Understanding and Generating Visual Worlds with Foundation Models

by

Weiming Ren

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2026

© Weiming Ren 2026

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Member:	Stella X. Yu Professor, Department of Electrical Engineering and Computer Science University of Michigan
Supervisor(s):	Wenhu Chen Assistant Professor, Cheriton School of Computer Science University of Waterloo
	Jimmy Lin Professor, Cheriton School of Computer Science University of Waterloo
Internal Member:	Yuntian Deng Assistant Professor, Cheriton School of Computer Science University of Waterloo
Internal Member:	Sihang Liu Assistant Professor, Cheriton School of Computer Science University of Waterloo
Internal-External Member:	Stephen L. Smith Professor, Department of Electrical and Computer Engineering University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis is based on seven published papers, for all of which I served as either the first author or a co-first author during my doctoral studies. The publication corresponding to each chapter is listed below, with my name underlined.

- **Chapter 4:** ConsistI2V: Enhancing Visual Consistency for Image-to-Video Generation. Weiming Ren, Huan Yang, Ge Zhang, Cong Wei, Xinrun Du, Wenhao Huang, Wenhu Chen. *Transactions on Machine Learning Research (TMLR)*, 2024.

- **Chapter 5:** AnyV2V: A Tuning-Free Framework for Any Video-to-Video Editing Tasks. Max Ku*, Cong Wei*, Weiming Ren*, Huan Yang, Wenhu Chen. *Transactions on Machine Learning Research (TMLR)*, 2024.

- **Chapter 7:** Vamba: Understanding Hour-Long Videos with Hybrid Mamba-Transformers. Weiming Ren, Wentao Ma, Huan Yang, Cong Wei, Ge Zhang, Wenhu Chen. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

- **Chapter 8:** VISTA: Enhancing Long-Duration and High-Resolution Video Understanding by Video Spatiotemporal Augmentation. Weiming Ren, Huan Yang, Jie Min, Cong Wei, Wenhu Chen. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

- **Chapter 9:** VideoEval-Pro: Robust and Realistic Long Video Understanding Evaluation. Wentao Ma*, Weiming Ren*, Yiming Jia, Zhuofeng Li, Ping Nie, Ge Zhang, Wenhu Chen. *Transactions on Machine Learning Research (TMLR)*, 2026.

- **Chapter 11:** Tuna: Taming Unified Visual Representations for Native Unified Multimodal Models. Zhiheng Liu*, Weiming Ren*, Haozhe Liu, Zijian Zhou, Shoufa Chen, Haonan Qiu, Xiaoke Huang, Zhaochong An, Fanny Yang, Aditya Patel, Viktar Atliha, Tony Ng, Xiao Han, Chuyan Zhu, Chenyang Zhang, Ding Liu, Juan-Manuel Perez-Rua, Sen He, Jürgen Schmidhuber, Wenhu Chen, Ping Luo, Wei Liu, Tao Xiang, Jonas Schult, Yuren Cong. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

- **Chapter 12:** Tuna-2: Pixel Embeddings Beat Vision Encoders for Multimodal Understanding and Generation. Zhiheng Liu*, Weiming Ren*, Xiaoke Huang, Shoufa Chen,

Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Belinda Zeng, Tao Xiang, Wenhu Chen, Ping Luo, Luke Zettlemoyer, Yuren Cong. *Advances in Neural Information Processing Systems (NeurIPS)*, 2026.

* denotes equal contribution.

I am the sole first author of the works presented in Chapters 4, 7, and 8. For each of these works, I proposed the research direction, designed and implemented the methods, conducted the experiments and analyses, and wrote the manuscript. My co-authors contributed guidance on the research direction, support for data collection and evaluation, and feedback on the manuscript.

For the works presented in Chapters 5, 9, 11, and 12, I am a joint first author and contributed equally with the other joint first authors to the design of the methods, the implementation and execution of the experiments, and the writing of the manuscripts.

Abstract

Foundation models have transformed how visual information is understood and generated, but they still typically operate on isolated *fragments* of visual experience, such as a short caption, a single image, or a few seconds of video. In contrast, people perceive and create within coherent *visual worlds* that comprise multiple modalities, evolve over time, and extend across long horizons. This thesis argues that closing this gap for foundation models requires not only larger parameter sizes, but also broader *interfaces*: models must be able to accept richer inputs and produce a wider range of outputs. It studies how these interfaces can be expanded along three axes: *control*, the conditioning signals that a generative model can faithfully follow; *horizon*, the spatial and temporal extent of the visual content that a model can process; and *modality*, the range of inputs and outputs that a single model can support.

The first part expands the control axis by allowing visual generation to be guided by multiple modalities rather than by text alone. I present ConsistI2V, an image-to-video diffusion model that adapts a pretrained text-to-image model into a video generator conditioned on a single keyframe. It introduces fine-grained spatiotemporal conditioning on the first frame, preserving its appearance, identity, and layout while allowing the generated content to move over time. It also introduces FrameInit, an inference-time noise initialization method that uses the low-frequency information of the first frame to guide the layout of the generated video. Building on this model, I present AnyV2V, a tuning-free framework that formulates a wide range of video editing tasks as first-frame-conditioned generation. This formulation enables diverse forms of video editing without task-specific training.

The second part expands the horizon axis by addressing architecture, training data, and evaluation as closely connected challenges. I present Vamba, a hybrid Mamba-Transformer model that avoids the quadratic cost of causal self-attention. In Vamba, text tokens access visual information through cross-attention, while video tokens are updated using linear-time Mamba-2 blocks. Without discarding or compressing video tokens, Vamba processes more than 1,024 frames on a single GPU and outperforms previous efficient video models on hour-long video benchmarks. To address the limited availability of long/high-resolution video training data, I present VISTA, a spatiotemporal augmentation framework that synthesizes long-duration and high-resolution video instruction data from existing video-caption pairs. Training with this synthetic dataset consistently improves multiple video models. To evaluate these advances more reliably, I present VideoEval-Pro, which demonstrates that widely used multiple-choice benchmarks for long video understanding are often inflated by answer-choice shortcuts and prior knowledge. It instead uses carefully

filtered open-ended questions, on which the strongest proprietary models achieve only 40.8% accuracy.

The third part expands the modality axis by combining visual understanding and generation within a single model. I present TUNA, a native unified multimodal model based on a shared continuous visual representation. This representation is produced by cascading a VAE encoder with a representation encoder, creating a feature space that preserves both the low-level visual structure needed for generation and the semantic information needed for understanding. A single LLM decoder performs autoregressive text generation for understanding and flow matching denoising for generation. The experiments demonstrate that joint training on multimodal understanding and generation data improves both capabilities for TUNA, compared with training on either type of data alone. To further simplify the model architecture, I present TUNA-2, which removes pretrained vision encoders entirely and operates directly on pixel embeddings, produced by simple patch embedding layers. With a minimalist model design, TUNA-2 matches its encoder-based counterpart in generation while achieving better performance on fine-grained visual understanding, demonstrating the effectiveness and scalability of pixel-space unified models.

Together, these contributions broaden the interface between foundation models and the visual world along all three axes. They trace a progression from generating and editing visual content under increasingly rich forms of control, to understanding visual information over extended periods, and finally to supporting understanding and generation within a single model. These advances represent concrete steps toward foundation models that process visual experience as a coherent, temporally extended, and multimodal world rather than as a collection of isolated fragments.

Acknowledgements

First and foremost, I am deeply grateful to my supervisors, Wenhu Chen and Jimmy Lin, for their guidance and support throughout my PhD journey. Wenhu helped shape my research interests and provided invaluable advice and feedback on every project I pursued. Jimmy guided me through many administrative matters and was consistently supportive of my research. I feel extremely fortunate to have had them as my supervisors.

I would also like to thank the members of my examining committee, Yuntian Deng, Sihang Liu, Stella Yu, and Stephen Smith, for taking the time to read my thesis and participate in my defence.

During my doctoral studies, I was fortunate to spend time in industry. I would like to thank Jonas Schult and Yuren Cong at Meta; Iqbal Mohomed, Alex Levinshtein, Raghav Goyal, Zhiming Hu, and Tristan Aumentado-Armstrong at Samsung; and Huan Yang and Wenhao Huang at 01.AI for mentoring me during these internships. Working with them exposed me to new research problems, engineering practices, and a scale of computation that I would not otherwise have encountered. Several of the ideas in this thesis grew directly out of these collaborations.

I have also been fortunate to share a lab with an extraordinary group of people. I thank Cong Wei, Dongfu Jiang, Xueguang Ma, Yubo Wang, Max Ku, Yuansheng Ni, Zhiheng Lyu, Tianle Li, and Ge Zhang for the discussions that clarified my thinking, the feedback on half-formed ideas, the willingness to read drafts on short notice, the companionship through many late nights before deadlines, and all the fun moments we shared along the way. You made my PhD journey a truly enjoyable experience.

To my parents, thank you for your unconditional support and for encouraging me to pursue this path, even though it has kept me far from home for so many years. Whatever I have managed to accomplish here rests on what you gave me long before this journey began.

Finally, to my girlfriend, Yujing Liu, thank you for your patience through the deadlines, the rejections, and the long periods when my attention was somewhere else entirely. You celebrated the good news with me and made every setback feel smaller. I am deeply grateful for every bit of it.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract vi

Acknowledgements viii

List of Figures xvi

List of Tables xxi

1 Introduction 1

1.1 Challenges of Current Foundation Models 2

1.2 Thesis Contributions 4

1.2.1 Part I: Controllable Visual World Generation from Diverse Conditioning Signals 4

1.2.2 Part II: Scalable Visual World Understanding over Long Horizons . 5

1.2.3 Part III: Unified Multimodal Models for Joint Visual Understanding and Generation 6

1.2.4 Summary of Contributions 6

2 Background **8**
2.1 Sequence Models . . . 8
2.1.1 Transformers . . . 9
2.1.2 Linear-Time Sequence Models . . . 12
2.2 Large Language Models and Large Multimodal Models . . . 13
2.2.1 Large Language Models . . . 13
2.2.2 Large Multimodal Models . . . 14
2.3 Diffusion Models . . . 16
2.3.1 Denoising Diffusion Probabilistic Models . . . 16
2.3.2 Latent Diffusion Models . . . 17
2.3.3 Flow Matching . . . 18

I Controllable Visual World Generation from Diverse Conditioning Signals **19**

3 Background and Roadmap **20**
3.1 Text-to-Video and Image-to-Video Generation . . . 21
3.2 Video Editing . . . 22

4 ConsistI2V: Enhancing Visual Consistency in Image-to-Video Generation **23**
4.1 Introduction . . . 23
4.2 Methodology . . . 25
4.2.1 Model Architecture . . . 25
4.2.2 Fine-Grained Spatial Feature Conditioning . . . 26
4.2.3 Window-based Temporal Feature Conditioning . . . 28
4.2.4 Inference-time Layout-Guided Noise Initialization . . . 29
4.3 I2V-Bench . . . 30

4.4 Experiments . . . 31
4.4.1 Implementation Details . . . 31
4.4.2 Quantitative Evaluation . . . 31
4.4.3 Qualitative Evaluation . . . 34
4.4.4 Ablation Studies . . . 34
4.4.5 More Applications . . . 37
4.5 Conclusion . . . 39

5 AnyV2V: A Tuning-Free Framework For Any Video-to-Video Editing Tasks 40
5.1 Introduction . . . 40
5.2 Preliminary . . . 43
5.2.1 DDIM Inversion . . . 43
5.2.2 Plug-and-Play (PnP) Diffusion Features . . . 43
5.3 AnyV2V . . . 44
5.3.1 Flexible First Frame Editing . . . 45
5.3.2 Structural Guidance using DDIM Inverison . . . 45
5.3.3 Appearance Guidance via Spatial Feature Injection . . . 45
5.3.4 Motion Guidance through Temporal Feature Injection . . . 46
5.3.5 Putting it Together . . . 47
5.4 Experiments . . . 47
5.4.1 Implementation Details . . . 47
5.4.2 Tasks Definition . . . 48
5.4.3 Evaluation Results . . . 49
5.4.4 Ablation Study . . . 53
5.5 Conclusion . . . 55

II Scalable Visual World Understanding over Long Horizons 56

6 Background and Roadmap 57

6.1 Large Multimodal Models for Video Understanding . . . 58
6.2 Efficient Models for Long Video Understanding . . . 58
6.3 Video Instruction-Following Data . . . 59
6.4 Video Understanding Benchmarks . . . 59

7 Vamba: Understanding Hour-Long Videos with Hybrid Mamba-Transformers 61

7.1 Introduction . . . 61
7.2 Preliminaries . . . 64
7.2.1 State Space Models and Mamba . . . 64
7.2.2 Transformer-based LMMs . . . 64
7.3 Our Method: VAMBA . . . 65
7.3.1 Updating Text Tokens via Cross-Attentions . . . 65
7.3.2 Updating Video Tokens with Mamba Blocks . . . 67
7.3.3 Training Paradigm . . . 68
7.4 Experiments . . . 68
7.4.1 Ablation Study . . . 68
7.4.2 Main Evaluation Results . . . 71
7.4.3 Runtime Efficiency Analysis . . . 72
7.4.4 Case Study . . . 74
7.5 Conclusion . . . 75

8 VISTA: Enhancing Long-Duration and High-Resolution Video Understanding by Video Spatiotemporal Augmentation 76

8.1 Introduction . . . 76
8.2 VISTA-400K Dataset . . . 78
8.2.1 Long Video Captioning & Event QA . . . 80

8.2.2 Video Needle-in-a-haystack (NIAH) QA . . . 81
8.2.3 High-Resolution Video Grid QA . . . 82
8.2.4 Summary of Video Instruction-Following Data . . . 82
8.3 Evaluation: HRVideoBench . . . 83
8.4 Experimental Results . . . 84
8.4.1 Evaluation Setup . . . 84
8.4.2 Quantitative Results . . . 85
8.4.3 Ablation Study . . . 86
8.4.4 Case Study . . . 87
8.5 Conclusion . . . 88

9 VideoEval-Pro: Robust and Realistic Long Video Understanding Evaluation 90
9.1 Introduction . . . 90
9.2 VideoEval-Pro . . . 92
9.2.1 Data Curation Pipeline . . . 92
9.2.2 Dataset Statistics . . . 94
9.2.3 Evaluation Pipeline . . . 96
9.3 Experiments . . . 97
9.3.1 Experimental Setup . . . 97
9.3.2 Main Results and Discussions . . . 98
9.3.3 Frame Scaling Properties of VideoEval-Pro . . . 100
9.3.4 Qualitative Analysis . . . 101
9.4 Conclusion . . . 103

III Unified Multimodal Models for Joint Visual Understanding and Generation 104

10 Background and Roadmap 105
10.1 Large Multimodal Models . . . 106

10.2 Diffusion Generative Models . . . 106
10.3 Unified Multimodal Models . . . 107
10.4 Visual Representations and Encoder-Free Models . . . 107

11 Tuna: Taming Unified Visual Representations for Native Unified Multimodal Models 109

11.1 Introduction . . . 109
11.2 Our Method: TUNA . . . 112
11.2.1 Motivation and Design Principles . . . 113
11.2.2 Model Architecture . . . 114
11.2.3 Training Pipeline . . . 115
11.3 Experiments . . . 116
11.3.1 Experiment Setup . . . 116
11.3.2 Main Results . . . 117
11.3.3 Ablation: Visual Representation Design . . . 121
11.3.4 Discussion: Unified Representation Analysis . . . 123
11.3.5 Qualitative Results . . . 125
11.4 Conclusion . . . 129

12 Tuna-2: Pixel Embeddings Beat Vision Encoders for Multimodal Understanding and Generation 130

12.1 Introduction . . . 130
12.2 Method . . . 134
12.2.1 Towards Encoder-Free Unified Models . . . 134
12.2.2 Learning Robust Visual Representations via Masking . . . 135
12.2.3 Training Pipeline . . . 137
12.3 Experiments . . . 137
12.3.1 Experiment Setup . . . 137
12.3.2 Main Results . . . 138

12.3.3 Ablation: Model Training Dynamics . . . 143
12.3.4 Ablation: Masking-based Feature Learning . . . 143
12.3.5 Analysis: TUNA-R vs. TUNA-2 . . . 144
12.3.6 Analysis: Attention Map Visualizations . . . 146
12.4 Conclusion . . . 147

13 Conclusion and Future Work **148**
13.1 Summary of Contributions . . . 148
13.2 Future Directions . . . 150
13.3 Closing Remarks . . . 152

References **153**

List of Figures

2.1 The standard encoder-decoder transformer architecture originally proposed for machine translation. Figure taken from [259]. 11

2.2 The dominant large multimodal model architecture. A pretrained vision encoder maps an input image to visual features, a lightweight projector turns these features into visual tokens in the language model's embedding space, and the visual tokens are prepended to the text tokens and fed to a shared transformer decoder that generates a text response autoregressively. Figure taken from [166]. 15

2.3 The two directions of a diffusion model. The forward process gradually adds Gaussian noise to a clean data sample $\mathbf{x}_0$ over T steps until it becomes an isotropic Gaussian $\mathbf{x}_T$. The reverse process, learned by a neural network, starts from pure noise and iteratively removes it to recover a data sample. Figure taken from [102]. 16

4.1 Comparison of image-to-video generation results obtained from SEINE [49] and our ConsistI2V. SEINE shows degenerated appearance and motion as the video progresses, while our result maintains visual consistency. 24

4.2 Our ConsistI2V framework. In our model, we concatenate the first frame latent z^1 to the input noise and perform first frame conditioning by augmenting the spatial and temporal self-attention operations in the model with the intermediate hidden states z_h^1. During inference, we incorporate the low-frequency component from z^1 to initialize the inference noise and guide the video generation process. 26

4.3 Visualization of our proposed spatial and temporal first frame conditioning schemes. For spatial self-attention layers, we employ cross-frame attention mechanisms and expand the keys and values with the features from all spatial positions in the first frame. For temporal self-attention layers, we augment the key and value vectors with a local feature window from the first frame. 27

4.4 Different frequency bands from the original latent $\mathbf{z}_0$ after spatiotemporal frequency decomposition. 29

4.5 Qualitative comparisons between DynamiCrafter, SEINE, AnimateAnything and our ConsistI2V. Input first frames are generated by PixArt-α [38] and SDXL [207]. 33

4.6 Visual comparisons of our method after disabling FrameInit (F.I.), temporal conditioning (T.C.) and spatial conditioning (S.C.). We use the same seed to generate all videos. 35

4.7 Visual comparisons of employing different FrameInit hyperparameters during I2V generation. All videos are generated using the same seed. 36

4.8 Applications of ConsistI2V. **Upper panel:** FrameInit enhances object consistency in long video generation. **Lower panel:** FrameInit enables training-free camera motion control. 38

5.1 AnyV2V is an evolving framework to handle all types of video-to-video editing tasks without any parameter tuning. AnyV2V disentangles video editing into two simpler problems: (1) Single image editing and (2) Image-to-video generation with video reference. 41

5.2 AnyV2V takes a source video V^S as input. In the first stage, we apply a block-box image editing method on the first frame I_1 according to the editing task. In the second stage, the source video is inverted to initial noise z_T^S, which is then denoised using DDIM sampling. During the sampling process, we extract spatial convolution, spatial attention, and temporal attention features from the I2V models' decoder layers. To generate the edited video, we perform a DDIM sampling by fixing z_T^* as z_T^T and use the edited first frame as the conditional signal. During sampling, we inject the features and attention into corresponding layers of the model. 44

5.3 AnyV2V is robust in a wide range of prompt-based editing tasks while preserving the background. The results align the most with the text prompt and maintain high motion consistency. 48

5.4 With different image editing models, AnyV2V can achieve a wide range of editing tasks, including reference-based style transfer, subject-driven editing, and identity manipulation. 50

5.5 AnyV2V can edit video length beyond the training frame while maintaining motion consistency. The first row is the source video frames while the second rows are the edited. The editing prompt of the image was "turn woman into a robot" using image model InstructPix2Pix [23]. 53

5.6 Visual comparisons of AnyV2V's editing results after disabling temporal feature injection (T.I.), spatial feature injection (S.I.) and DDIM inverted initial noise (D.I.). 54

7.1 VAMBA achieves strong long video understanding performance (42.1% on LVBench [268]) while being more computationally efficient compared to transformer-based LMMs. 62

7.2 Overview of our VAMBA model architecture. Compared to transformer-based LMMs (left), we replace the costly causal self-attention operations with the more efficient cross-attention layers and Mamba blocks to achieve better efficiency. 66

7.3 Comparison of training GPU memory usage and runtime per training step between Qwen2-VL-7B and VAMBA. 73

7.4 Comparison of GPU memory usage and FLOPs between Qwen2-VL-7B and VAMBA during inference. 73

7.5 Qualitative comparison between VAMBA and efficient LMM baselines. Red text denotes incorrect responses, while green text highlights the correct responses by our VAMBA. 74

8.1 VISTA is a simple but effective framework that generates high-quality video instruction data from existing video-caption pairs. Our VISTA-400K dataset enhances model performances on various long and high-resolution video benchmarks. 77

8.2 Our proposed video augmentation and instruction-following data synthesis schemes for VISTA-400K. Given input videos, We perform spatiotemporal video combinations to produce augmented video samples with longer duration and higher resolution. 79

8.3 Qualitative comparisons between the baseline models and our VISTA-finetuned models. Red text indicates hallucinations or incorrect responses, while green text highlights the correct responses that correspond accurately to the video content. 88

9.1 Comparison between VideoEval-Pro and MCQ benchmarks. **Left:** MCQ benchmarks yield inflated scores on identical questions (MCQ vs. Open) and can misrepresent model performance (LVBench [268]). **Right**: VideoEval-Pro cannot be effectively solved with a single input frame, and performance scales consistently with more frames. Video-MME [74] exhibits contradictory trends. 91

9.2 Summary of VideoEval-Pro data composition and task type distribution. . 95

9.3 Comparison between VideoEval-Pro and Video-MME accuracy across five LMMs. 101

9.4 Qualitative comparisons between VideoEval-Pro and the corresponding MCQ problems. 102

11.1 We present TUNA, a native unified multimodal model built on a unified visual representation, enabling diverse multimodal understanding and generation capabilities such as image and video understanding, image and video generation, and image editing. 110

11.2 Overview of the TUNA architecture. Our model employs a VAE encoder and a representation encoder to construct unified visual representations, which are then combined with text tokens and processed by an LLM decoder. The decoder performs autoregressive text generation for understanding tasks and flow-matching-based visual generation for generation tasks. *During visual generation, noise is added to the visual tokens to enable diffusion-based generation. 112

11.3 Attention masks in the LLM decoder for understanding and generation tasks. * indicates that the visual tokens are noised. 115

11.4 Comparison between TUNA and Show-o2 on how unified visual representations are produced. 124

11.5 Representation alignment analysis with SigLIP 2 and SD3-Medium. For both TUNA and Show-o2, we extract visual representations at the input layer of the LLM decoder. 125

11.6 Qualitative comparison between TUNA and baseline models on image generation tasks. The instructions that are correctly reflected in our results but failed in some of the baseline models are **bolded**. 126

11.7 Qualitative comparison between TUNA and baseline models on image editing tasks. 127

11.8 Qualitative results for TUNA on the task of text-to-video generation. . . . 128

12.1 Evolution of TUNA-2 architecture and multimodal performance comparison. We simplify TUNA [180] by progressively stripping away its visual encoding components. By removing the VAE, we first derive TUNA-R, a pixel-space UMM that relies solely on a representation encoder. TUNA-2 further streamlines the design by bypassing the representation encoder entirely, utilizing direct patch embedding layers for raw image inputs. TUNA-2 using pixel embeddings outperforms both TUNA-R and TUNA across a diverse suite of multimodal benchmarks. 131

12.2 While being completely encoder-free, TUNA-2 is capable of performing high-fidelity text-to-image generation and image editing. 133

12.3 Illustration of our proposed masking-based feature learning scheme. During training, we use the learnable mask token to regularize multimodal understanding and perform masked prediction for visual generation. 136

12.4 Qualitative results for different visual tokenizers. 141

12.5 Comparison of the image generation (MSE) and language modelling (CE) loss curves based on different understanding-generation data ratios during model training. 142

12.6 Model accuracy curves with respect to training dataset size (measured by the number of tokens consumed by the model) for TUNA-R and TUNA-2. . 144

12.7 Attention map visualization for TUNA-R, TUNA-2 and other baseline models. Red area denotes high attention scores and Blue area denotes low attention scores. 145

List of Tables

4.1 Quantitative evaluation results for ConsistI2V. †: the statistics also include the data for training the base video generation model. **Bold:** best results. Underline: second best. 32

4.2 Automatic evaluation results on I2V-Bench. Consist. denotes the consistency metrics. **Bold:** best results. Underline: second best. 32

4.3 UCF-101 ablation study results and runtime statistics for spatiotemporal first frame conditioning and FrameInit. T.Cond. and S.Cond. correspond to temporal and spatial first frame conditioning, respectively. 36

5.1 Quantitative comparisons for our AnyV2V with baselines on prompt-based video editing. Alignment: prompt alignment; Overall: overall preference. **Bold**: best results; Underline: top-2. 49

5.2 Comparisons for three I2V models under AnyV2V framework on novel video editing tasks. Align: reference alignment; Overall: overall preference. **Bold**: best results; Underline: top-2. 52

5.3 Ablation study results for AnyV2V (I2VGen-XL). T. Injection and S. Injection correspond to temporal and spatial feature injection mechanisms, respectively. 55

7.1 Theoretical time and memory complexity for transformer-based LMMs and Vamba in pre-filling. M and N represent the number of vision and text tokens. d denotes the hidden dimension. 63

7.2 Ablation study results for the Vamba design space explorations. "CA from SA?" denotes whether to initialize cross-attention weights from self-attention layers. "VidMME" represents the Video-MME benchmark. . . . 69

7.3 Ablation study results on pretraining VAMBA models with a transformer-based LMM teacher and a distillation loss. 69

7.4 Comparisons between baseline models and VAMBA on hour-long video understanding benchmarks. **Bold**: best results among efficient video LMMs. Underline: second-best. 70

7.5 Comparisons between baseline models and VAMBA on medium-length or short video understanding benchmarks. **Bold**: best results among each section. Underline: second-best. * indicates the results based on our evaluation scripts. 71

8.1 Statistics of our synthetic video instruction-following dataset. "(N)" and "(H)" corresponds to the "needle" (short or low-res videos) and the "haystack" (long or high-res videos) in NIAH subsets. 80

8.2 Comparisons between baseline models and VISTA-finetuned models on long/short video understanding benchmarks. The best results among open-source models are bolded. Δ denotes the performance differences before and after finetuning on VISTA-400K. 83

8.3 Quantitative results on HRVideoBench and open-ended video QA benchmarks. "acc." represents accuracy. 84

8.4 Ablation study results for VISTA-Mantis. Each "w/o [Subset]" denotes a Mantis-Idefics2 model finetuned on a modified VISTA-400K by replacing the corresponding subset with the same amount of training examples from VideoChat2-IT [147]. 87

9.1 Video and QA statistics for VideoEval-Pro. 94

9.2 Main results of our VideoEval-Pro benchmark. Δ indicates the gap between MCQ and open-ended questions. LP, LR, HP and HR correspond to Local Perception, Local Reasoning, Holistic Perception and Holistic Reasoning tasks. 98

9.3 Comparison between proprietary and open-source models on VideoEval-Pro and other standard medium and long video benchmarks. 100

11.1 Comparisons between TUNA and baseline models on multimodal understanding benchmarks. Results with model size greater than 13B are grayed. **Bold**: best results among each section. Underline: second-best. * indicates the results based on our evaluation scripts. 117

11.2 Image generation results on GenEval. † refers to methods using LLM rewriters. **Bold**: best results among each section. Underline: second-best. 118

11.3 Image generation results on DPG-Bench. **Bold**: best results among each section. Underline: second-best. 119

11.4 Image editing results on ImgEdit-Bench and GEdit-Bench. For ImgEdit-Bench, we test editing performance across various dimensions, including 'Add', 'Adjust', 'Extract', 'Replace', 'Remove', 'Background', 'Style', 'Hybrid', and 'Action'. For GEdit-Bench, "G-SC" and "G-PQ" denote "G-Semantic Consistency" and "G-Perceptual Quality", respectively. **Bold**: best results among each section. Underline: second-best. 119

11.5 Experimental results on video understanding benchmarks. #Frames denotes the number of frames used during inference. **Bold**: best results. Underline: second-best. 120

11.6 Video generation results on VBench. Full column names: QS: Quality Score, SS: Semantic Score, SC: Subject Consistency, BC: Background Consistency, TF: Temporal Flickering, MS: Motion Smoothness, DD: Dynamic Degree, AQ: Aesthetic Quality, IQ: Imaging Quality, OC: Object Class, MO: Multiple Objects, HA: Human Action, C: Color, SR: Spatial Relationship, S: Scene, AS: Appearance style, TS: Temporal Style, OC': Overall Consistency. **Bold**: best results among each section. Underline: second-best. 121

11.7 Ablation study results. "Und. Only", "Gen. Only" and "Und. & Gen." refer to models trained with understanding data only, generation data only, and both data, respectively. 122

12.1 Comparisons between TUNA-2 and baseline models on multimodal understanding benchmarks. Results with model size greater than 13B are grayed. **Bold**: best results among all UMMs. Underline: second-best among all UMMs. 138

12.2 Image generation results on GenEval and DPG-Bench. "Col. Attr." means "Color Attribute". † refers to methods using LLM rewriters in GenEval. **Bold**: best results among native UMMs. Underline: second-best. 139

12.3 Evaluation results under GPT-5.4 and Claude Opus 4.7. Quality measures image realism and fine-grained detail richness, while Diversity measures variation among images generated from the same prompt. **Bold** denotes the best result and underlining denotes the second-best. 140

12.4 Image editing results on ImgEdit. **Bold** denotes the best result among unified models and underlining denotes the second-best. 140

12.5 Image reconstruction performance for different visual tokenizers. 141

12.6 Ablation study results for TUNA-R and TUNA-2, with or without our proposed masking-based feature learning scheme during pretraining. 143

Chapter 1

Introduction

Over the past few years, foundation models have transformed the way visual information is processed and generated. Traditionally, visual understanding and generation were solved by small-scale, task-specific models, each trained from scratch on a narrow dataset for a single objective such as classification [130, 98], captioning [306] and text-to-image generation [86, 307]. Today, by scaling up model parameters and training data, large-scale multimodal foundation models handle many tasks at once within a single system. For multimodal understanding, large multimodal models (LMMs) [166, 265, 15, 14] can recognize objects [186, 51], describe scenes [264, 321], and answer questions about images and videos [52, 277]. For visual generation, diffusion-based models [102, 222, 66, 260] synthesize photorealistic images and videos from free-form text prompts [222, 66, 20, 260], perform instruction-guided image and video editing [281, 122], and generate visual content conditioned on diverse structural signals, such as depth maps, edge maps, and sketches [333, 314]. These capabilities, once unimaginable, are now the starting point rather than the goal.

Despite this rapid progress, current foundation models still largely operate on *fragments* of experience in both their inputs and outputs, such as a simple textual description, a single image, or a short video clip. Such fragment-level interfaces remain far removed from the *visual world* that people perceive and act within. First, humans do not experience the world as a disconnected sequence of snapshots. Instead, they perceive a continuous stream of visual information in which objects persist, scenes evolve coherently, and events unfold over minutes, hours, or even longer periods. Second, real-world perception and creation rarely involve a single modality in isolation. A complete visual experience, such as visiting a museum or an art gallery, may interleave written descriptions, artworks, spatial environments, and dynamic human interactions. Similarly, the creation of visual content, such as films, paintings, or calligraphy, often involves multiple forms of information, including

language, reference images, sketches, motion, and physical actions. If foundation models are to become general-purpose tools for perceiving, simulating, and interacting with the world, they must move beyond processing and producing isolated fragments toward modelling persistent, coherent, and multimodal visual experiences over extended spatial and temporal contexts.

This thesis investigates how foundation models can move beyond processing isolated fragments toward understanding and generating coherent visual worlds. I use the term *visual world* to describe visual content that remains coherent over time, extends across long horizons, and can be both read through understanding and written through generation. From this perspective, the fragmentary behavior of current foundation models is fundamentally a limitation of their *interfaces* to visual content: what they can take in as input and what they can produce as output.

The goal of this thesis is to broaden these interfaces along three complementary axes: *control*, *horizon*, and *modality*. The control axis concerns how precisely and flexibly a generative model can be steered. I enrich the forms of conditioning available to visual generators, enabling them to produce content that remains faithful not only to text but also to input images and videos. The horizon axis concerns the spatial and temporal extent of the visual input. I expand the amount of continuous visual history and the spatial resolution that a model can process, enabling it to understand high-resolution and hour-long videos rather than only low-resolution, sparse collections of frames. The modality axis concerns the type of multimodal input and output foundation models can process. I unify visual understanding and generation within a single model and a shared visual representation, allowing one system to both understand and generate across modalities rather than operate within a single modality or in only one direction.

1.1 Challenges of Current Foundation Models

Enhancing the three axis of foundation models raises concrete technical challenges, and this thesis addresses these challenges in turn.

Challenge 1: Faithful generation under richer conditioning. The control axis concerns a generative model's ability to faithfully follow conditioning signals beyond text. In many practical applications, including story visualization, video editing, and personalized video generation, the model is provided with only sparse observations, such as a

single keyframe and a short textual description, and must synthesize a temporally coherent video that preserves the semantic intent, visual appearance, and spatial layout of the initial scene. This setting presents two main challenges: on one hand, the generated video must contain sufficient motion and temporal variation to appear dynamic and engaging. On the other hand, it must preserve object identities, spatial relationships, and global visual style consistently across frames. Naively adapting an image diffusion model to video generation often fails to balance these objectives, yielding either nearly static clips with limited temporal evolution or dynamic sequences that rapidly drift from the source image in identity, structure, and appearance.

Challenge 2: Scalability to long horizons. The horizon axis addresses the scale of the visual content that the model can perceive. Real world visual contents, including movies, instructional videos, and long-form recordings, often span tens of minutes or hours and unfold through complex, multi-stage narratives. Scaling LMMs to such inputs is difficult because the quadratic cost of self-attention makes processing dense video tokens across thousands of frames computationally infeasible on standard hardware. The common workaround of compressing a video into a small number of tokens reduces computation but discards the fine-grained details required for long-horizon reasoning. A different approach is therefore needed if models are to interpret an entire visual world rather than a heavily summarized representation of it. Scaling along this axis also introduces two further challenges: high-quality instruction data for long videos remains scarce, and existing long-video benchmarks often overstate progress. An efficient architecture must therefore be paired with suitable training data and reliable evaluation.

Challenge 3: Diversity of input/output modality. The modality axis concerns the range of input and output modalities that a model can process. Current foundation models are typically designed for either visual understanding or visual generation, which differs substantially from how humans perceive and interact with the world. Moreover, these models often rely on distinct visual representations for each capability. This separation introduces format mismatches and conflicting inductive biases, making it difficult to integrate different tasks within a single unified model. A model that shares one visual representation across understanding and generation could allow the two capabilities to reinforce each other, but designing such a system remains challenging. Its visual representation must support both discriminative tasks, such as question answering, and generative tasks, such as image synthesis, while remaining flexible enough to process multiple modalities, including text, images, and videos.

1.2 Thesis Contributions

This thesis investigates how to widen the interface between foundation models and the visual world by addressing the challenges associated with each axis. Along the control axis, I expand the conditioning signals available to generative models, enabling them to produce visual content guided by images, videos and text. Along the horizon axis, I extend the temporal range that understanding models can process through efficient architectures, specialized training data, and reliable benchmarks for hour-long video. Along the modality axis, I unify visual understanding and generation through a shared visual representation, allowing a single model to both perceive and synthesis visual content. I present the discussions of each axis in a dedicated chapter and grounded in published works.

1.2.1 Part I: Controllable Visual World Generation from Diverse Conditioning Signals

In the first part of the thesis, I discuss methods to widen the control axis, enriching the conditioning mechanisms that a generative model accepts. I propose ConsistI2V [220], an image-to-video diffusion model that converts a pretrained text-to-image backbone (Stable Diffusion 2.1 [222]) into a generator capable of extending a single keyframe into a coherent video. ConsistI2V introduces two complementary mechanisms. First, it augments both the spatial and temporal self-attention layers with first-frame conditioning signals: in the spatial layers, each location attends back to the full feature map of the first frame to enforce fine-grained appearance alignment, while in the temporal layers, each query attends to a local spatial window of first-frame features so that the same entity stays in view as it moves. This anchors appearance to the input frame while still permitting natural motion. Second, it introduces FrameInit, an inference-time noise initialization scheme that injects the low-frequency component of the first frame into the initial noise, providing coarse layout guidance, narrowing the gap between training and inference, and stabilizing long-range generation. These mechanisms also enable downstream capabilities such as autoregressive long-video generation and camera motion control. Trained only on the public WebVid-10M [16] dataset and initialized from a text-to-image model, ConsistI2V outperforms strong image-to-video baselines across most automatic metrics and ranks first in human evaluations of both appearance and motion consistency.

Building on this generative engine, AnyV2V [132] further widens the conditioning interface from an input image to an input video, extending the framework from generation to editing. AnyV2V is a tuning-free approach that reformulates a broad range of video-to-

video editing tasks, including prompt-based editing, style transfer, and subject-driven editing, as first-frame-conditioned generation. An off-the-shelf image editor first modifies the initial frame, after which an image-to-video model propagates the edit across the remaining frames while preserving the original motion. By separating spatial editing from temporal propagation, AnyV2V supports diverse editing tasks without task-specific training, demonstrating how a consistent image-to-video generator can function as a general-purpose tool for manipulating visual worlds.

1.2.2 Part II: Scalable Visual World Understanding over Long Horizons

The second part of this thesis widens the horizon axis by extending the temporal span that a model can receive and process. Specifically, I introduce VAMBA [215], a hybrid Mamba-Transformer model for efficient hour-long video understanding. Rather than compressing or discarding video tokens, VAMBA redesigns the model architecture so that it can process longer sequences of video frames directly. Starting from a strong Transformer-based video LMM, VAMBA decomposes the original causal self-attention over input text tokens into two less expensive operations: self-attention over the relatively short text sequence and cross-attention in which the text tokens attend to the full set of video tokens. The video tokens are updated by Mamba-2 [57] blocks, which operate as linear-time recurrent sequence models and avoid the quadratic cost of self-attention. Without reducing the number of tokens, VAMBA can encode more than 1,024 video frames at 360p on a single GPU, providing roughly four times the context length of comparable Transformer-based LMMs while reducing training and inference memory by more than half and nearly doubling per-step training speed. On challenging hour-long benchmarks such as LVBench [267], VAMBA outperforms prior efficient video LMMs while remaining competitive on medium-length and short-video tasks, providing the long-horizon backbone needed for visual world modelling.

Efficient architectures alone are insufficient, as models also require suitable training data and trustworthy evaluation. To address the data challenge, I introduce VISTA [219], a video spatiotemporal augmentation framework that synthesizes long-duration, high-resolution video instruction-following data from existing video-caption datasets. By combining videos spatially and temporally, VISTA creates new, longer synthetic videos together with corresponding questions and answers, directly addressing the scarcity of high-quality supervision for long-video understanding. To evaluate progress reliably, I introduce VIDEOEVAL-PRO [188], a robust benchmark for long-video understanding. My analysis shows that many existing benchmarks rely heavily on multiple-choice questions, whose

scores can be inflated by guessing, and include questions that strong models can answer from prior knowledge without watching the video. In contrast, VideoEval-Pro uses open-ended questions that require genuine comprehension of the full video, providing a more realistic assessment of a model's long-horizon understanding.

1.2.3 Part III: Unified Multimodal Models for Joint Visual Understanding and Generation

The third part of this thesis focuses on the modality axis, which addresses the challenge of building unified multimodal models that can read and write across modalities. I introduce Tuna [179], a native unified multimodal model that constructs a single continuous visual representation shared by understanding and generation. Rather than aligning separate vision encoders for different tasks, Tuna cascades a VAE encoder with a representation encoder. The VAE produces high-fidelity latent features that preserve low-level visual structure for generation, while the representation encoder enriches these features with the semantic information needed for understanding. The resulting unified visual representations are combined with text tokens and processed by a language model decoder, which performs autoregressive text prediction for understanding tasks and flow-matching-based denoising for generation tasks. This design allows Tuna to use a single backbone for captioning, visual question answering, text-to-image and text-to-video generation, and image editing. Tuna consistently outperforms prior unified models based on decoupled representations, approaches or matches strong understanding-only and generation-only baselines, and surpasses contemporary unified models in image generation. Importantly, its unified design allows the two capabilities to reinforce each other, as joint training on understanding and generation data outperforms training either capability in isolation.

Finally, Tuna-2 [178] pushes the unification further by operating directly on pixel embeddings. It replaces the modular vision encoders, including the VAE and the representation encoder, with simple patch embedding layers, dramatically simplifying the architecture while achieving state-of-the-art performance across multimodal understanding and generation benchmarks. Tuna-2 demonstrates that a fully end-to-end, pixel-native design can match or exceed models built from separately pretrained encoders, pointing toward a simpler and more scalable recipe for unified multimodal models.

1.2.4 Summary of Contributions

In summary, this thesis makes the following contributions:

- ConsistI2V and AnyV2V: methods for controllable image-to-video generation and tuning-free video editing that turn rich conditioning signals, such as text, image, and video, into temporally coherent visual worlds while preserving identity, layout, and style.

- Vamba, VISTA, and VideoEval-Pro: a linear-time hybrid Mamba-Transformer architecture that scales video understanding to hour-long inputs, a spatiotemporal data-augmentation framework that supplies long video instruction data, and a robust open-ended benchmark that measures long video understanding without the inflation of multiple-choice evaluation.

- Tuna and Tuna-2: native unified multimodal models that share a single continuous visual representation across understanding and generation, demonstrating that unified representations let the two capabilities enhance each other, and that a pixel-native design can further simplify the architecture without sacrificing performance.

Taken together, these contributions widen the interface between foundation models and the visual world along all three axes: richer control for generation, longer horizons for understanding, and a shared representation that unifies multimodal understanding and generation. They trace a progression from generating and editing visual content under increasingly rich conditioning, to interpreting extended visual histories, and ultimately to unifying the understanding and generation of visual content within a single model. Collectively, they represent concrete steps toward foundation models that treat the visual world as a coherent, temporally extended, and multimodal signal that can be both understood and generated.

Chapter 2

Background

This chapter introduces the technical background needed to follow the remainder of the thesis. Section 2.1 covers sequence models, the architectural backbone shared by essentially every model in this thesis, including the transformer and the linear-time state space models that make long-horizon processing tractable. Section 2.2 describes large language models and their multimodal extensions, which underpin the visual understanding studied in Parts II and III. Section 2.3 presents diffusion models and flow matching, the generative machinery behind the visual-world generation of Part I and the generation branch of the unified models in Part III.

2.1 Sequence Models

The data that this thesis works with, such as natural language, images, and video, can all be represented as a sequence of tokens, where each token is a vector in $\mathbb{R}^d$ and d is the hidden dimension. Text is naturally a sequence of word or subword tokens. An image can be split into a grid of patches, each flattened and projected into a token, so that a picture becomes a short sequence [64]. A video is a sequence of frames, each of which yields many patch tokens, so even a short clip produces a long sequence. A *sequence model* maps an input sequence $x_1, \ldots, x_n$ to a sequence of output representations in which each position is contextualized by the others, that is, the representation at position i depends not only on x_i but on the surrounding tokens.

Historically, sequence data was processed by recurrent networks such as the long short-term memory (LSTM) network [105] and by encoder-decoder sequence-to-sequence models

built from them [240]. These architectures consume tokens one at a time, folding the history into a recurrent hidden state that is carried forward step by step. Attention was first introduced as an add-on to this recurrent pipeline, letting the decoder look back over all encoder states rather than a single fixed summary vector [11]. However, because computation flows through the hidden state in order, recurrence is inherently sequential and hard to parallelize, and information from distant tokens tends to be forgotten as it is repeatedly overwritten, making long-range dependencies difficult to capture.

Transformer models [259] replaced recurrence entirely with attention. By letting every token interact directly with every other token in a single step, it achieves full parallelism over the sequence and models long-range interactions without passing information through a bottlenecked hidden state. Transformer is now the backbone of essentially all the foundation models studied in this thesis. In the remainder of this section, I first present the transformer model in detail, and then present linear-time sequence models, specifically Mamba [89], that address the transformer's main scalability limitation.

2.1.1 Transformers

Token embeddings and positional encoding. In a transformer model, each input token is first mapped to a d-dimensional embedding vector. Because self-attention is permutation-equivariant, positional information must be introduced explicitly to distinguish tokens at different locations in the sequence. The original transformer model added fixed sinusoidal or learned positional encodings directly to the token embeddings, whereas most modern models use rotary position embeddings, which encode relative positional information by rotating the query and key vectors according to their token indices [238].

Scaled dot-product self-attention. After obtaining the token embeddings, a transformer layer collects the token embeddings into an input matrix $X \in \mathbb{R}^{n \times d}$. It then uses three learned projections to produce queries, keys, and values, such that:

$$Q = XW_Q, \qquad K = XW_K, \qquad V = XW_V, \tag{2.1}$$

where $Q, K \in \mathbb{R}^{n \times d_k}$ and $V \in \mathbb{R}^{n \times d_v}$. Self-attention is then formulated as a weighted sum of the value vectors, where the weights are determined by the similarity between queries and keys. The scaled dot-product attention is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V. \tag{2.2}$$

Intuitively, each token emits a query that is compared, by dot product, against the key of every token in the sequence. The softmax turns these n scores into a set of non-negative attention weights that sum to one, and the output for that token is the corresponding weighted average of the value vectors. In this way, each token gathers information from the whole sequence, weighted by relevance. The factor $\sqrt{d_k}$ rescales the dot products: without it, the scores grow in magnitude with d_k, pushing the softmax into saturated regions where gradients vanish, so the scaling stabilizes training.

Multi-head attention. Rather than computing a single attention operation, transformers usually run h attention heads in parallel, each with its own projections W_Q, W_K, W_V. Different heads can specialize on different subspaces and relations between tokens. Their outputs are concatenated and passed through a final linear layer back to dimension d.

Causal (masked) self-attention. For autoregressive generation, the sequence must be processed so that each position sees only the past. A causal mask sets the attention scores for all future positions to $-\infty$ before the softmax, restricting position i to attend only to positions $\leq i$. The model can then predict the next token from past context alone. This masked form of self-attention is the mechanism used by the language-model decoders discussed in the next section.

Feed-forward, residuals, and normalization. Each transformer layer follows the attention sublayer with a position-wise feed-forward network, a small MLP applied independently to every token. Both sublayers are wrapped in a residual connection and layer normalization [10], which together ease optimization and keep activations well scaled. A transformer is a stack of many such identical layers.

Architectural variants. As shown in Figure 2.1, the original transformer followed an encoder-decoder architecture designed for machine translation, in which the encoder processes the source sentence and the decoder generates the target sentence. In this thesis, the transformer-based language models instead use decoder-only architectures with masked self-attention throughout. By contrast, vision transformers [64] and diffusion transformers typically adopt encoder-only architectures, in which visual inputs are mapped to hidden representations for downstream multimodal modelling in LMMs or for noise prediction in diffusion models.

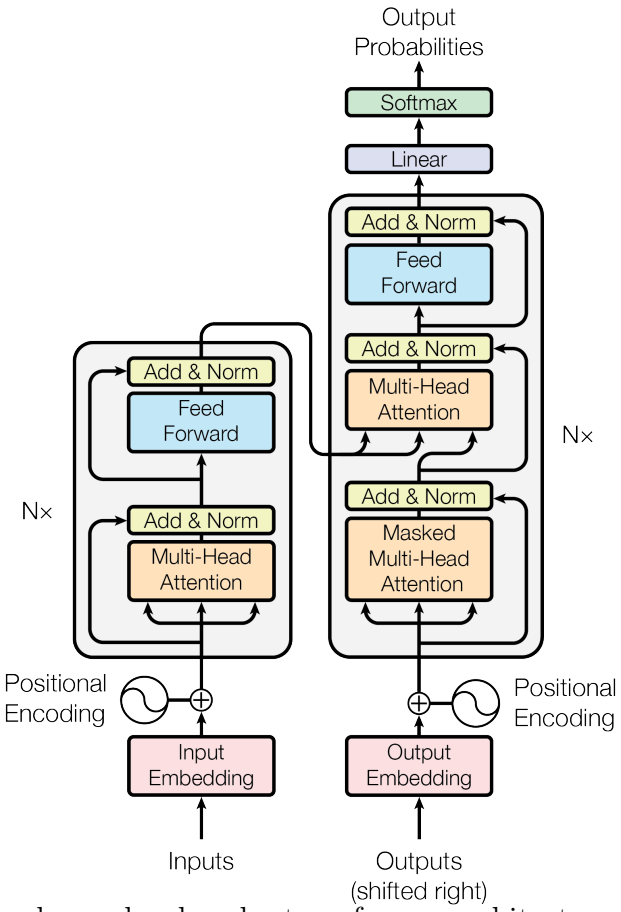


Figure 2.1: The standard encoder-decoder transformer architecture originally proposed for machine translation. Figure taken from [259].

Complexity. The attention matrix $QK^\top$ in Equation 2.2 compares every pair of tokens, so it has n^2 entries and both the time and the memory of self-attention scale as $O(n^2 d)$, that is, quadratically in the sequence length n. For short sequences this cost is negligible, but it becomes prohibitive as n grows into the tens of thousands, which is exactly the regime of hour-long video, where each frame already contributes many patch tokens. This quadratic cost is the central obstacle to processing very long visual sequences, and it directly motivates the linear-time sequence models covered next. I return to this scalability limitation, and to architectures that overcome it, in Part II.

2.1.2 Linear-Time Sequence Models

Recall from Section 2.1.1 that self-attention costs $O(n^2 d)$ in time and $O(n^2)$ in memory, quadratic in the sequence length n. This quadratic scaling becomes prohibitive when the sequence is very long: a single minute of video already yields many thousands of visual tokens, and the cost of dense attention grows with the square of that count. Classical recurrent networks avoid the quadratic term and run in linear time, but they are hard to parallelize across the time axis and tend to forget information over long horizons. State space models (SSMs) occupy a useful middle ground: they provide linear-time sequence modeling with strong long-range memory, and they can be computed as a parallel convolution at training time and as a lightweight recurrence at inference time.

An SSM is a linear time-invariant system that maps a continuous scalar input signal $x(t) \in \mathbb{R}$ to an output $y(t) \in \mathbb{R}$ through a hidden state $h(t) \in \mathbb{R}^N$, described by the ordinary differential equation:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \qquad y(t) = \mathbf{C}h(t), \tag{2.3}$$

with $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times 1}$, and $\mathbf{C} \in \mathbb{R}^{1\times N}$. To apply this system to a discrete sequence, the models S4 [91] and its successor Mamba [89] discretize it using a zero-order hold (ZOH) with step size Δ:

$$\overline{\mathbf{A}} = \exp(\Delta\mathbf{A}), \qquad \overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}\big(\exp(\Delta\mathbf{A}) - \mathbf{I}\big) \cdot \Delta\mathbf{B}. \tag{2.4}$$

The resulting discrete recurrence is:

$$h_t = \overline{\mathbf{A}}h_{t-1} + \overline{\mathbf{B}}x_t, \qquad y_t = \mathbf{C}h_t. \tag{2.5}$$

Here, the hidden state h_t acts as a compressed running summary of all past inputs: at each step, it decays the previous summary through $\overline{\mathbf{A}}$ and folds in the new token through $\overline{\mathbf{B}}$. This recurrence can be evaluated one step at a time, like an RNN. When the parameters are held fixed across time (the time-invariant case), the same computation also unrolls into a single global convolution of the input with a fixed kernel, which can be applied in parallel over the whole sequence and makes training efficient.

S4 [91] follows this recipe and initializes $\mathbf{A}$ as the diagonalized HiPPO matrix [90], a structured form that endows the state with long-range memory and lets the model remain effective and efficient on extremely long sequences. Mamba [89] adds the key idea of a *selective scan*: it makes $\mathbf{B}$, $\mathbf{C}$, and Δ functions of the current token, so the model can selectively propagate or forget information based on content. Making the parameters

input-dependent turns the system time-varying, so it is no longer a plain convolution, but a hardware-aware parallel scan recovers efficient parallel training. Mamba-2 [57] develops the state space duality (SSD) framework, which connects SSMs to attention, and simplifies $\mathbf{A}$ to a scalar times the identity. This structure supports a multi-head SSM formulation, admits a much larger state dimension N than Mamba, and is faster to train.

The payoff is favorable scaling. Processing n tokens with a Mamba block costs $O(d^2 n)$, which is linear in the sequence length, in contrast to the $O(n^2 d)$ cost of causal self-attention. For sequences that reach into the thousands or tens of thousands of tokens, this difference is decisive. In Part II, I adopt Mamba-2 blocks to process long video-token sequences, using their linear-time recurrence to scale visual-world understanding to hour-long horizons that dense attention cannot reach.

2.2 Large Language Models and Large Multimodal Models

The sequence models of the previous section become foundation models once they are trained at scale on natural language. This section describes large language models (LLMs) as autoregressive decoders, and then large multimodal models (LMMs), which extend an LLM so that it can perceive images and video. Both build directly on the causal-attention transformer decoder introduced earlier, and both are central to the visual understanding side of this thesis.

2.2.1 Large Language Models

A large language model is a decoder-only transformer [259] trained for autoregressive next-token prediction. The input text is first tokenized: a string is split into subword units drawn from a fixed vocabulary, so that a piece of text becomes a sequence of discrete tokens $x_1, \ldots, x_n$. The model treats language modelling as density estimation and factorizes the joint probability of a sequence by the chain rule, predicting each token from its predecessors:

$$p_\theta(x_1, \ldots, x_n) = \prod_{t=1}^{n} p_\theta(x_t \mid x_{<t}). \tag{2.6}$$

At each position t, the final decoder hidden state is passed through a linear language-modelling head followed by a softmax, yielding a distribution over the vocabulary. Training

minimizes the negative log-likelihood, i.e., the cross-entropy language-modelling loss:

$$\mathcal{L}_{\mathrm{LM}} = -\frac{1}{n}\sum_{t=1}^{n} \log p_\theta(x_t \mid x_{<t}). \tag{2.7}$$

This uses teacher forcing together with the causal attention mask, so that position t attends only to $x_{<t}$. Because the mask enforces this ordering, every position in a training sequence can be supervised in parallel within a single forward pass.

Generation. A trained LLM produces new text by running the decoder autoregressively. Given the current context, the model computes the next-token distribution, and one token is selected from it, either greedily (using arg-max) or by sampling under a temperature with top-k or top-p truncation. The chosen token is appended to the context, and the procedure repeats: each newly generated token becomes part of the input for the next step, so generation proceeds one token at a time until a stopping condition is reached. This autoregressive loop is the same generative mechanism that later carries over to large multimodal models.

Scaling model size and training data along this simple objective yields broad, general capabilities [213, 25], and open foundation models such as the LLaMA series [251, 252, 88] and the Qwen series [13, 211, 310] have made these capabilities widely available. A subsequent stage of instruction tuning and alignment turns the raw next-token predictor into a model that follows instructions and answers questions rather than merely continuing text.

2.2.2 Large Multimodal Models

An LLM processes only text, but this thesis concerns visual worlds, so the model must also perceive images and video. Large multimodal models (LMMs) extend an LLM so that it accepts visual inputs while still producing text. The dominant recipe, popularized by LLaVA [166], is illustrated in Figure 2.2 and has three parts. First, a pretrained vision encoder such as CLIP [212] or SigLIP [327] encodes an image into a set of visual features. Second, a lightweight projector, typically a small multilayer perceptron, maps these features into the LLM's token-embedding space, producing a sequence of visual tokens in $\mathbb{R}^d$. Third, the visual tokens are concatenated with the text tokens, and the combined sequence is processed by the same transformer decoder, which generates a natural-language response autoregressively exactly as in the LLM case above. The decoder still generates only text

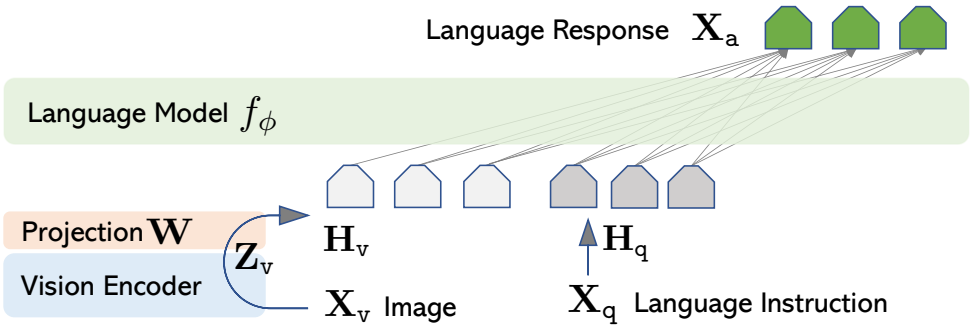


Figure 2.2: The dominant large multimodal model architecture. A pretrained vision encoder maps an input image to visual features, a lightweight projector turns these features into visual tokens in the language model's embedding space, and the visual tokens are prepended to the text tokens and fed to a shared transformer decoder that generates a text response autoregressively. Figure taken from [166].

tokens; the visual tokens never appear at the output and instead serve as additional context in the prefix.

The attention layout follows a common convention: the visual tokens are placed before the text tokens, and the decoder applies causal self-attention over the whole sequence, so each text token can attend to all visual tokens and to the preceding text tokens. Scaling this recipe with stronger vision encoders, larger decoders, and more instruction data results in strong models for image and video understanding, as in Qwen2.5-VL [15], Qwen3-VL [14], and InternVL3 [346].

The cost of long visual context. For video, frames are sampled over time and each frame is encoded into many visual tokens, so the sequence fed to the decoder becomes very long. Since the decoder relies on the quadratic self-attention of the transformers, processing long, high-resolution video is expensive in both computation and memory. This bottleneck is exactly what motivates the efficient long-video architecture I develop in Part II. It is also worth stressing a limitation of scope: current LMMs understand visual input but cannot generate images or videos. Visual generation is instead handled by the diffusion models described in the next section, and unifying the two directions, perception and synthesis, within a single model is the subject of Part III.

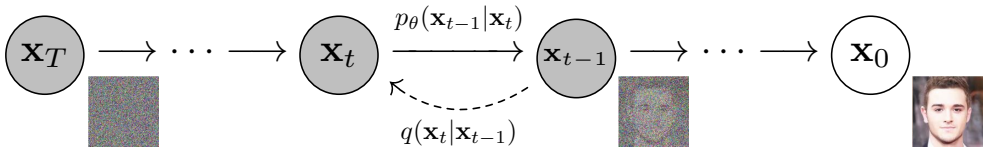


Figure 2.3: The two directions of a diffusion model. The forward process gradually adds Gaussian noise to a clean data sample $\mathbf{x}_0$ over T steps until it becomes an isotropic Gaussian $\mathbf{x}_T$. The reverse process, learned by a neural network, starts from pure noise and iteratively removes it to recover a data sample. Figure taken from [102].

2.3 Diffusion Models

Generative models learn a data distribution so that new samples, such as images or videos, can be drawn from it. Diffusion models [233, 102], together with their flow-matching generalization, are the dominant paradigm for high-fidelity visual generation. Their core idea is to gradually corrupt data into noise and to train a neural network to reverse this corruption, so that sampling begins from pure noise and iteratively denoises into a data sample. Figure 2.3 illustrates this forward and reverse process.

2.3.1 Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPMs) [102, 233] define generation through a pair of forward and reverse diffusion processes. The *forward* process is a fixed Markov chain that gradually adds Gaussian noise to a clean sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ over T steps according to a variance schedule $0 < \beta_1 < \beta_2 < \cdots < \beta_T < 1$. A convenient property of this chain is that the noisy sample at any timestep $t \in \{1, \ldots, T\}$ can be directly constructed from the clean sample $\mathbf{x}_0$ in closed-form. Writing $\bar{\alpha}_t = \prod_{i=1}^{t}(1 - \beta_i)$ and drawing $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the reparameterization is:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\,\boldsymbol{\epsilon}, \qquad \bar{\alpha}_t = \prod_{i=1}^{t}(1 - \beta_i). \tag{2.8}$$

As $t \to T$, the schedule drives $\bar{\alpha}_t \to 0$, so $\mathbf{x}_T$ contains almost no input signal and approaches an isotropic Gaussian.

The *reverse* process learns to invert this corruption. It is parameterized as a Markov chain $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ of Gaussian transitions whose mean is expressed through a neural

network that predicts the noise $\boldsymbol{\epsilon}$ added to reach $\mathbf{x}_t$. Denoting this noise-prediction network by $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, DDPM training reduces to the simplified mean-squared-error objective:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\big[\, \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2 \,\big], \tag{2.9}$$

where $\mathbf{x}_t$ is obtained from $\mathbf{x}_0$ and $\boldsymbol{\epsilon}$ based on Equation 2.8, and t is sampled from a predefined distribution (uniform in DDPM [102] and logit-norm in LDM [222]).

To generate a sample, one starts from a pure noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and applies the learned reverse transition iteratively from $t = T$ down to $t = 1$, arriving at a clean sample $\mathbf{x}_0$. This many-step procedure is accurate but costly. To reduce inference cost, DDIM [234] recasts sampling as a non-Markovian deterministic process that produces comparable samples in far fewer steps. An equivalent continuous-time formulation views the forward and reverse dynamics as a stochastic differential equation whose reverse direction depends on the score of the data distribution [235].

For conditional generation, such as text-to-image synthesis, the network additionally takes a condition $\mathbf{c}$ and becomes $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}, t)$. Classifier-free guidance [103] strengthens adherence to $\mathbf{c}$ at sampling time by extrapolating between the conditional prediction and an unconditional one produced by the same network with the condition dropped. This avoids the separate classifier required by classifier guidance [61] and has become the standard tool for trading sample diversity against faithfulness to the condition.

2.3.2 Latent Diffusion Models

Running diffusion directly in pixel space is expensive at high resolution, because every denoising step operates on a tensor as large as the image itself. Latent diffusion models [222] address this by moving the diffusion process into a compact latent space. A pretrained autoencoder, typically a KL-regularized variational autoencoder [128], provides an encoder $\mathcal{E}$ that maps an image to a lower-dimensional latent $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$ and a decoder $\mathcal{D}$ that inverts it. Both the forward and reverse diffusion processes are run entirely on the latent, and the denoised latent $\hat{\mathbf{z}}$ is decoded back to pixels as $\hat{\mathbf{x}} = \mathcal{D}(\hat{\mathbf{z}})$. Because the latent is much smaller than the image, this greatly reduces computation while preserving perceptual quality, and it forms the basis of Stable Diffusion.

The denoising network in latent space is commonly a U-Net [223] augmented with cross-attention layers that inject the conditioning signal, for example text embeddings. More recent systems, such as the Diffusion Transformer (DiT) [206], replace the U-Net with a transformer that treats latent patches as tokens, which scales favorably and underlies large

text-to-image models such as Stable Diffusion 3 [66]. The same latent-diffusion recipe extends naturally to video generation by diffusing spatiotemporal latents [104, 20, 260]. The controllable image-to-video generation models I develop in Part I build directly on latent diffusion.

2.3.3 Flow Matching

Flow matching [164, 173] is a continuous-time generative framework that learns a *velocity field* $\mathbf{v}_\theta(\mathbf{x}_t, t)$ defining an ordinary differential equation:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \mathbf{v}_\theta(\mathbf{x}_t, t), \tag{2.10}$$

which transports a simple prior, Gaussian noise, to the data distribution. Rather than simulating this ODE during training, conditional flow matching supplies a closed-form regression target along prescribed interpolation paths, which makes optimization simple and stable. The concrete instance used throughout this thesis is rectified flow [173], which connects noise and data by a straight line. Let $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be noise and $\mathbf{x}_1$ be a data sample, rectified flow obtains the noisy sample at timestep t by linearly interpolating between them:

$$\mathbf{x}_t = t\,\mathbf{x}_1 + (1-t)\,\mathbf{x}_0, \qquad t \in [0, 1]. \tag{2.11}$$

Differentiating the interpolation with respect to t gives a constant target velocity $\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_0$, independent of t. The network is trained by regressing to this velocity, such that the loss is the mean squared error between the predicted velocity and the true velocity:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, t}\big[\, \|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1 - \mathbf{x}_0)\|^2 \,\big]. \tag{2.12}$$

During inference, sampling a new data sample is done by integrating the learned ODE from $t = 0$ (noise) to $t = 1$ (data) with a numerical solver such as the Euler solver:

$$\mathbf{x}_{t'} = \mathbf{x}_t + (t' - t)\,\mathbf{v}_\theta(\mathbf{x}_t, t). \tag{2.13}$$

Because the interpolation paths are close to straight lines, the resulting ODE trajectories are nearly linear, allowing high-quality samples to be generated in relatively few steps. Flow matching is closely related to diffusion, as both transform noise into data through a continuous or iterative process, but it provides a simpler deterministic training target and straighter sampling trajectories. Recent state-of-the-art image and video generators, including Stable Diffusion 3 [66] and Wan 2.2 [260], have adopted flow matching. The unified models presented in Part III likewise use flow matching in their generation branches.

Part I

Controllable Visual World Generation from Diverse Conditioning Signals

Chapter 3

Background and Roadmap

The first part of this thesis studies the *control* axis of visual world generation: broadening the conditioning signals that a generative model can faithfully follow beyond a text prompt alone. A central difficulty in this setting is that the model is often given only sparse observations, such as a single keyframe together with a short textual description, from which it must synthesize a temporally coherent video that preserves the identity, spatial layout, and visual style of the input while still exhibiting natural motion. Naively adapting an image generator to video tends to fail at one of these two goals, producing either nearly static clips or dynamic sequences that quickly drift away from the input. In this part, we develop methods that turn diverse conditioning signals, such as images, videos and text, into coherent video clips for controllable video generation and editing.

The roadmap for this part is as follows. In Chapter 4, we present ConsistI2V, an image-to-video (I2V) diffusion model that converts a pretrained text-to-image diffusion model into a video generator capable of extending a single keyframe plus a text prompt into a coherent video. ConsistI2V augments the spatial and temporal attention layers of the backbone with fine-grained first-frame conditioning and introduces an inference-time, layout-guided noise initialization scheme. These two mechanisms together produces videos that remain visually consistent with the input frame while supporting downstream applications such as autoregressive long-video generation and camera motion control. In Chapter 5, we build on this I2V model to present AnyV2V, a tuning-free framework that recasts a broad range of video-to-video editing tasks as first-frame-conditioned video generation: an off-the-shelf image editor modifies the first frame, and an image-to-video model propagates the edit across the remaining frames while preserving the original motion. Together, these chapters widen the conditioning interface of video generation model from text to images and videos.

The remainder of this chapter reviews the related work that situates these contributions. We first survey text-to-video and image-to-video generation, and then discuss video editing.

3.1 Text-to-Video and Image-to-Video Generation

Text-to-video (T2V) generation has progressed from early approaches based on generative adversarial networks [72, 22, 248] and autoregressive transformers [78, 106] toward diffusion models, which now dominate the field. Most diffusion-based methods extend a text-to-image (T2I) architecture to model the temporal dimension. Video Diffusion Models [104] introduce a space-time factorized U-Net [223] with interleaved temporal attention layers, while Imagen-Video [101] and Make-A-Video [231] employ pixel-space diffusion for high-definition synthesis. To reduce computation, another line of work runs diffusion in the latent space of a pretrained autoencoder [99, 35, 127, 95], following the latent diffusion paradigm [222]. Within this family, MagicVideo [343] inserts lightweight adaptor layers, Latent-Shift [5] adopts temporal shift modules [162], and VideoLDM [20] inserts temporal convolution and attention layers into the LDM U-Net, a recipe that many later systems follow.

A complementary line of research studies noise initialization for video diffusion. PYoCo [79] proposes a video-specific noise prior in which each frame's initial noise is correlated with the others, FreeNoise [209] reschedules noise for training-free long-video generation, and FreeInit [294] injects the low-frequency component of a noisy video to close the gap between training and inference. The layout-guided noise initialization of ConsistI2V builds directly on this observation.

Steering generation with text alone offers only coarse control over the identity and layout of the output. Image-to-video (I2V) generation addresses this by conditioning on an additional initial frame in addition to text. Several methods [84, 49, 55, 326] concatenate the latent of the first frame with the input noise to enable first-frame conditioning, while others [337, 303, 329] add image cross-attention layers that inject stronger conditional signals into the generation process. ConsistI2V differs from these works in two aspects: first it introduces fine-grained spatiotemporal first-frame conditioning that yields better visual consistency and enables efficient training on public video-text data. Second, it develops inference-time noise initialization strategies, an aspect that prior I2V methods rarely address.

3.2 Video Editing

Because paired data of videos before and after an edit is difficult to obtain, most video editing methods adapt a pretrained T2I model in a zero-shot or few-shot manner [29, 291, 81, 284, 338, 54]. To maintain temporal coherence across independently edited frames, these methods introduce cross-frame attention. Tune-A-Video [291] and Pix2Video [29] let each frame attend to its immediate predecessor and to the first frame, whereas TokenFlow [81] and Fairy [284] propagate edits through a set of mutually attending anchor frames. Other methods add structural guidance: Control-A-Video [45] and ControlVideo [338] use ControlNet [333] for spatial control, CoDeF [203] propagates a single edited frame through a canonical representation, and UniEdit [12] injects inverted latents and feature maps into a pretrained T2V model [275].

Despite this progress, most methods offer limited controllability and are typically tied to a single editing paradigm. Approaches that rely on cross-attention, for example, are largely restricted to word-swapping prompts [170], since the ambiguity of natural language and the constraints of the underlying model make it difficult to express a user's exact intent. AnyV2V instead decouples spatial editing from temporal propagation: any off-the-shelf image editor performs the edit on the first frame, and an image-to-video model propagates it across the video while preserving motion. This design makes AnyV2V the first framework to support a diverse array of editing tasks, including prompt-based editing, reference-based style transfer, subject-driven editing, and identity manipulation, within a single unified approach.

Chapter 4

ConsistI2V: Enhancing Visual Consistency in Image-to-Video Generation

4.1 Introduction

Recent advancements in video diffusion models [104] have led to an unprecedented development in text-to-video (T2V) generation [101, 20]. However, such conditional generation techniques fall short of achieving precise control over the generated video content. For instance, given an input text prompt "a dog running in the backyard", the generated videos may vary from outputting different dog breeds, different camera viewing angles, as well as different background objects. As a result, users may need to carefully modify the text prompt to add more descriptive adjectives, or repetitively generate several videos to achieve the desired outcome.

To mitigate this issue, prior efforts have been focused on encoding customized subjects into video generation models with few-shot finetuning [193] or replacing video generation backbones with personalized image generation model [95]. Recently, incorporating additional first frame images into the video generation process has become a new solution to controllable video generation. This method, often known as (text-guided) image-to-video generation (I2V) or image animation, enables the foreground/background contents in the generated video to be conditioned on the objects as reflected in the first frame.

Nevertheless, training such conditional video generation models is a non-trivial task and existing methods often encounter appearance and motion inconsistency in the generated

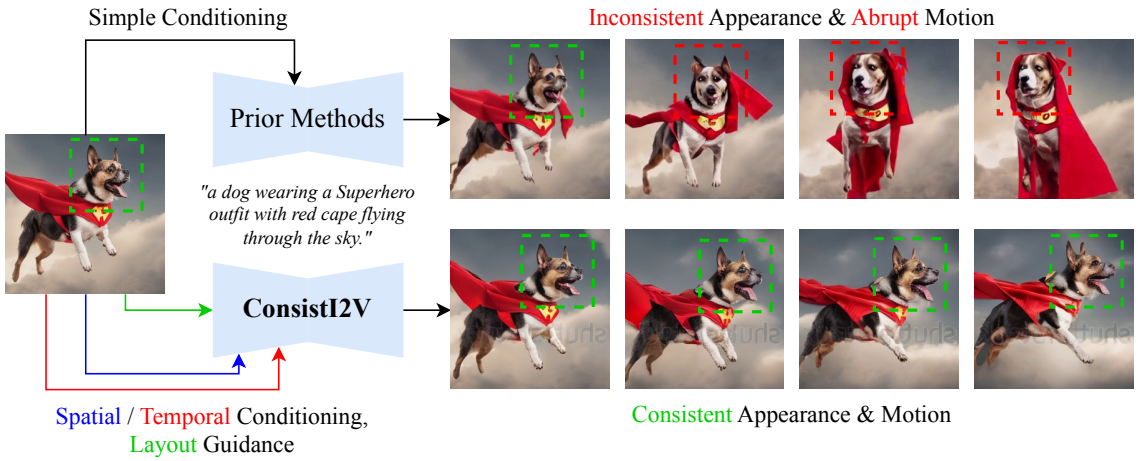


Figure 4.1: Comparison of image-to-video generation results obtained from SEINE [49] and our ConsistI2V. SEINE shows degenerated appearance and motion as the video progresses, while our result maintains visual consistency.

videos (shown in Figure 4.1). Initial efforts such as VideoComposer [271] and VideoCrafter1 [35] condition the video generation model with the semantic embedding (e.g. CLIP embedding) from the first frame but cannot fully preserve the local details in the generated video. Subsequent works either employ a simple conditioning scheme by directly concatenating the first frame latent features with the input noise [84, 49, 326, 55] or combine the two aforementioned design choices together [303, 337] to enhance the first frame conditioning. Despite improving visual appearance alignment with the first frame, these methods still suffer from generating videos with incorrect and jittery motion, which severely restricts their applications in practice.

To address the aforementioned challenges, we propose ConsistI2V, a simple yet effective framework capable of enhancing the visual consistency for I2V generation. Our method focuses on improving the first frame conditioning mechanisms in the I2V model and optimizing the inference noise initialization during sampling. To produce videos that closely resemble the first frame, we apply cross-frame attention mechanisms in the model's spatial layers to achieve fine-grained *spatial* first frame conditioning. To ensure the temporal smoothness and coherency of the generated video, we include a local window of the first frame features in the *temporal* layers to augment their attention operations. During inference, we propose FrameInit, which leverages the low-frequency component of the first frame image and combines it with the initial noise to act as a *layout* guidance and eliminate the noise discrepancy between training and inference. By integrating these design

optimizations, our model generates highly consistent videos and can be easily extended to other applications such as autoregressive long video generation and camera motion control. Our model achieves state-of-the-art results on public I2V generation benchmarks. We further conduct extensive automatic and human evaluations on a self-collected dataset I2V-Bench to verify the effectiveness of our method for I2V generation. Our contributions are summarized below:

1. We introduce ConsistI2V, a diffusion-based model that performs spatiotemporal conditioning over the first frame to enhance the visual consistency in video generation.
2. We devise FrameInit, an inference-time noise initialization strategy that uses the low-frequency band from the first frame to stabilize video generation. FrameInit can also support applications such as autoregressive long video generation and camera motion control.
3. We propose I2V-Bench, a comprehensive quantitative evaluation benchmark dedicated to evaluating I2V generation models. We will release our evaluation dataset to foster future I2V generation research.

4.2 Methodology

Given an image $x^1 \in \mathbb{R}^{C\times H\times W}$ and a text prompt $\mathbf{s}$, the goal of our model is to generate an N frame video clip $\hat{\mathbf{x}} = \{x^1, \hat{x}^2, \hat{x}^3, ... \hat{x}^N\} \in \mathbb{R}^{N\times C\times H\times W}$ such that x^1 is the first frame of the video, and enforce the appearance of the rest of the video to be closely aligned with x^1 and the content of the video to follow the textual description in $\mathbf{s}$. We approach this task by employing first-frame conditioning mechanisms in spatial and temporal layers of our model and applying layout-guided noise initialization during inference. The overall model architecture and inference pipeline of ConsistI2V are shown in Figure 4.2.

4.2.1 Model Architecture

U-Net Inflation for Video Generation Our model is developed based on text-to-image (T2I) LDMs [222] that employ the U-Net [223] model for image generation. This U-Net model contains a series of spatial downsampling and upsampling blocks with skip connections. Each down/upsampling block is constructed with two types of basic blocks: spatial convolution and spatial attention layers. We insert a 1D temporal convolution block

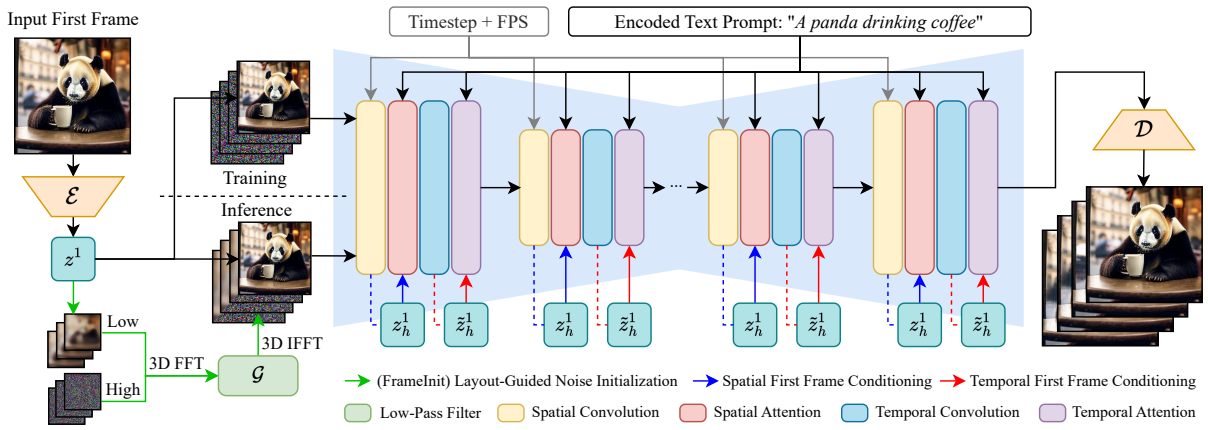


Figure 4.2: Our ConsistI2V framework. In our model, we concatenate the first frame latent z^1 to the input noise and perform first frame conditioning by augmenting the spatial and temporal self-attention operations in the model with the intermediate hidden states z_h^1. During inference, we incorporate the low-frequency component from z^1 to initialize the inference noise and guide the video generation process.

after every spatial convolution block and temporal attention blocks at certain attention resolutions to make it compatible with video generation tasks. Our temporal convolution and attention blocks share the exact same architecture as their spatial counterparts, apart from the convolution and attention operations are operated along the temporal dimension. We incorporate RoPE [238] embeddings to represent positional information in the temporal layers and employ the PYoCo [79] *mixed* noise prior for noise initialization.

First Frame Condition Injection We leverage the variational autoencoder (VAE) [128] of the T2I LDM to encode the input first frame into latent representation $z^1 = \mathcal{E}(x^1) \in \mathbb{R}^{C' \times H' \times W'}$ and use z^1 as the conditional signal. To inject this signal into our model, we directly replace the first frame noise ϵ^1 with z^1 and construct the model input as $\hat{\epsilon} = \{z^1, \epsilon^2, \epsilon^3, ..., \epsilon^N\} \in \mathbb{R}^{N \times C' \times H' \times W'}$.

4.2.2 Fine-Grained Spatial Feature Conditioning

The spatial attention layer in the LDM U-Net contains a self-attention layer that operates on each frame independently and a cross-attention layer that operates between frames and the encoded text prompt. Given an intermediate hidden state z^i of the i^{th} frame, the

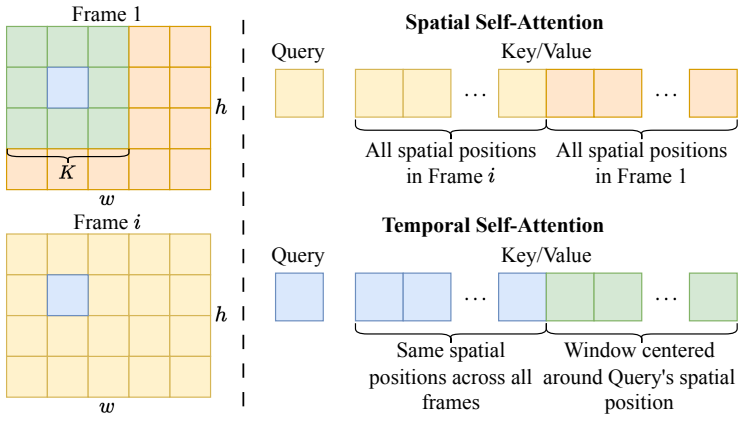


Figure 4.3: Visualization of our proposed spatial and temporal first frame conditioning schemes. For spatial self-attention layers, we employ cross-frame attention mechanisms and expand the keys and values with the features from all spatial positions in the first frame. For temporal self-attention layers, we augment the key and value vectors with a local feature window from the first frame.

self-attention operation is formulated as the attention between different spatial positions of z^i:

$$Q_s = W_s^Q z^i, K_s = W_s^K z^i, V_s = W_s^V z^i, \tag{4.1}$$

$$\text{Attention}(Q_s, K_s, V_s) = \text{Softmax}(\frac{Q_s K_s^\top}{\sqrt{d}})V_s, \tag{4.2}$$

where W_s^Q, W_s^K and W_s^V are learnable projection matrices for creating query, key and value vectors from the input. d is the dimension of the query and key vectors. To achieve better visual coherency in the video, we modify the key and value vectors in the self-attention layers to also include features from the first frame z^1 (*cf.* Figure 4.3):

$$Q_s = W_s^Q z^i, K_s' = W_s^K [z^i, z^1], V_s' = W_s^V [z^i, z^1], \tag{4.3}$$

where $[\cdot]$ represents the concatenation operation such that the token sequence length in K_s' and V_s' are doubled compared to the original K_s and V_s. In this way, each spatial position in all frames gets access to the complete information from the first frame, allowing fine-grained feature conditioning in the spatial attention layers.

4.2.3 Window-based Temporal Feature Conditioning

The temporal self-attention operations in video diffusion models [104, 20, 275, 35] share a similar formulation to spatial self-attention (*cf.* Equation 4.1), with the exception that the intermediate hidden state being formulated as $\bar{\mathbf{z}} \in \mathbb{R}^{(H\times W)\times N\times C}$, where N is the number of frames and C, H, W correspond to the channel, height and width dimension of the hidden state tensor. Formally, given an input hidden state $\mathbf{z} \in \mathbb{R}^{N\times C\times H\times W}$, $\bar{\mathbf{z}}$ is obtained by reshaping the height and width dimension of $\mathbf{z}$ to the batch dimension such that $\bar{\mathbf{z}} \leftarrow \texttt{rearrange}(\mathbf{z}, \texttt{"N C H W -> (H W) N C"})$ (using `einops` [221] notation). Intuitively, every $1 \times N \times C$ matrices in $\bar{\mathbf{z}}$ represents features at the same 1×1 spatial location across all N frames. This limits the temporal attention layers to always look at a small spatial window when attending to features between different frames.

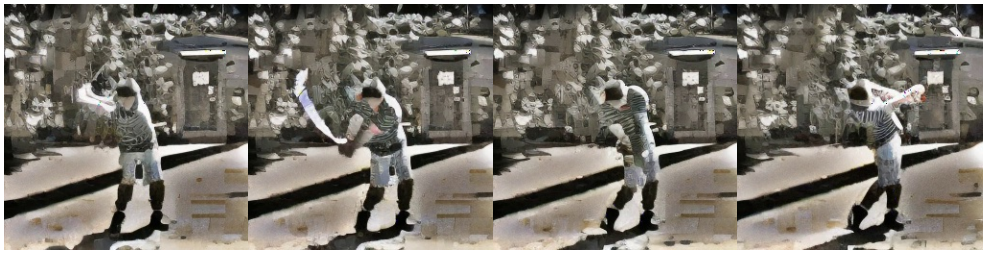
To effectively leverage the first frame features, we also augment the temporal self-attention layers to include a broader window of features from the first frame. We propose to compute the query, key and value from $\bar{\mathbf{z}}$ as:

$$Q_t = W_t^Q \bar{\mathbf{z}}, K_t' = W_t^K [\bar{\mathbf{z}}, \tilde{z}^1], V_t' = W_t^V [\bar{\mathbf{z}}, \tilde{z}^1], \tag{4.4}$$

where $\tilde{z}^1 \in \mathbb{R}^{(H\times W)\times(K\times K-1)\times C}$ is a tensor constructed in a way that its $h \times w$ position in the batch dimension corresponds to an $K \times K$ window of the first frame features, centred at the spatial position of (h, w). The first frame feature vector at (h, w) is not included in $\tilde{z}^1$ as it is already presented in $\bar{\mathbf{z}}$, making the second dimension of $\tilde{z}^1$ as $K \times K - 1$. We pad $\mathbf{z}$ in its spatial dimensions by replicating the boundary values to ensure that all spatial positions in $\mathbf{z}$ will have a complete window around them. We then concatenate $\bar{\mathbf{z}}$ with $\tilde{z}^1$ to enlarge the sequence length of the key and value matrices. See Figure 4.3 for a visualization of the augmented temporal self-attention. Our main rationale is that the visual objects in a video may move to different spatial locations as the video progresses. The vanilla or common formulation of temporal self-attention in video diffusion models essentially assumes $K = 1$, as shown by the blue tokens in Figure 4.3. Having an extra window of keys and values around the query location with the window size $K > 1$ (green tokens in Figure 4.3) increases the probability of attending to the same entity in the first frame when performing temporal self-attention. In practice, we set $K = 3$ to control the time complexity of the attention operations. As the input latent is downsampled $8\times$ by the VAE, and the denoising U-Net further downsamples the feature map in its intermediate layers, our selected window size ($K = 3$) can still create a large receptive field in deeper U-Net layers. For example, in the second last level of the U-Net ($32\times$ total downsampling), a 256×256 video will be compressed into a 8×8 feature map. Consequently, a 3×3 window on this feature map covers a region of 96×96 in the input first frame. This is a significant



Reconstructed sample from the original latent $\mathbf{z}_0$



Reconstructed sample from the high-frequency component of $\mathbf{z}_0$



Reconstructed sample from the low-frequency component of $\mathbf{z}_0$

Figure 4.4: Different frequency bands from the original latent $\mathbf{z}_0$ after spatiotemporal frequency decomposition.

increase from previous methods where a 1×1 window on the feature map corresponded to only a 24×24 region in the original frame.

4.2.4 Inference-time Layout-Guided Noise Initialization

Existing literature [163] in image diffusion models has identified that there exists a noise initialization gap between training and inference, due to the fact that common diffusion noise schedules create an information leak to the diffusion noise during training, causing it to be inconsistent with the random Gaussian noise sampled during inference. In the domain of video generation, this initialization gap has been further explored by FreeInit

[294], showing that the information leak mainly comes from the low-frequency component of a video after spatiotemporal frequency decomposition and adding this low-frequency component to the initial inference noise greatly enhances the quality of the generated videos. Inspired by the significance of frequency bands in images and videos for human perception, prior work like Blurring Diffusion Models [108] employs diffusion processes in the frequency domain to enhance generation results. Here, we also aim to study how to effectively leverage the video's different frequency components to enhance I2V generation.

To better understand how spatiotemporal frequency bands and visual features are correlated in videos, we visualize the videos decoded from the VAE latents after spatiotemporal frequency decomposition, as shown in Figure 4.4. We observe that the video's high-frequency component captures the fast-moving objects and the fine details in the video, whereas the low-frequency component corresponds to those slowly moving parts and represents an overall layout. Based on this observation, we propose FrameInit, which duplicates the input first frame into a static video and uses its low-frequency component as a coarse layout guidance during inference. Formally, given the latent representation $\mathbf{z}_0$ of a static video and an inference noise ϵ, we first add τ step inference noise to the static video to obtain $\mathbf{z}_\tau = \texttt{add_noise}(\mathbf{z}_0, \epsilon, \tau)$. We then extract the low-frequency component of $\mathbf{z}_\tau$ and mix it with ϵ:

$$\mathcal{F}^{low}_{\mathbf{z}_\tau} = \texttt{FFT_3D}(\mathbf{z}_\tau) \odot \mathcal{G}(D_0), \tag{4.5}$$

$$\mathcal{F}^{high}_{\epsilon} = \texttt{FFT_3D}(\epsilon) \odot (1 - \mathcal{G}(D_0)), \tag{4.6}$$

$$\epsilon' = \texttt{IFFT_3D}(\mathcal{F}^{low}_{\mathbf{z}_\tau} + \mathcal{F}^{high}_{\epsilon}), \tag{4.7}$$

where `FFT_3D` is the 3D discrete fast Fourier transformation operating on spatiotemporal dimensions and `IFFT_3D` is the inverse FFT operation. $\mathcal{G}$ is the Gaussian low-pass filter parameterized by the normalized space-time stop frequency D_0. The modified noise ϵ' containing the low-frequency information from the static video is then used for denoising.

By implementing FrameInit, our empirical analysis reveals a significant enhancement in the stabilization of generated videos, demonstrating improved video quality and consistency. FrameInit also enables our model with two additional applications: (1) autoregressive long video generation and (2) camera motion control. We showcase more results for each application in Section 4.4.5.

4.3 I2V-Bench

Existing video generation benchmarks such as UCF-101 [236] and MSR-VTT [305] fall short in video resolution, diversity, and aesthetic appeal. To bridge this gap, we introduce

the I2V-Bench evaluation dataset, featuring 2,950 high-quality YouTube videos curated based on strict resolution and aesthetic standards. We organized these videos into 16 distinct categories, such as Scenery, Sports, Animals, and Portraits.

Evaluation Metrics Following VBench [111], our evaluation framework encompasses two key dimensions, each addressing distinct aspects of I2V performances: (1) **Visual Quality** assesses the perceptual quality of the video output regardless of the input prompts. We measure the subject and background consistency, temporal flickering, motion smoothness and dynamic degree. (2) **Visual Consistency** evaluates the video's adherence to the text prompt given by the user. We measure object consistency, scene consistency and overall video-text consistency.

4.4 Experiments

4.4.1 Implementation Details

We use Stable Diffusion 2.1-base [222] as the base T2I model to initialize ConsistI2V and train the model on the WebVid-10M [16] dataset, which contains ∼10M video-text pairs. For each video, we sample 16 frames with a spatial resolution of 256×256 and a frame interval between $1 \leq v \leq 5$, which is used as a conditional input to the model to enable FPS control. We use the first frame as the image input and learn to denoise the subsequent 15 frames during training. Our model is trained with the ϵ objective over all U-Net parameters using a batch size of 192 and a learning rate of 5e-5 for 170k steps. During training, we randomly drop input text prompts with a probability of 0.1 to enable classifier-free guidance [103]. During inference, we employ the DDIM sampler [234] with 50 steps and classifier-free guidance with a guidance scale of $w = 7.5$ to sample videos. We apply FrameInit with $\tau = 850$ and $D_0 = 0.25$ for inference noise initialization.

4.4.2 Quantitative Evaluation

UCF-101 & MSR-VTT We evaluate ConsistI2V on two public datasets UCF-101 [236] and MSR-VTT [305]. We report Fréchet Video Distance (FVD) [257] and Inception Score (IS) [226] for video quality assessment, Fréchet Inception Distance (FID) [100] for frame quality assessment and CLIP similarity (CLIPSIM) [285] for video-text alignment evaluation. We evaluate FVD, FID and IS on UCF-101 over 2048 videos and FVD and

Table 4.1: Quantitative evaluation results for ConsistI2V. †: the statistics also include the data for training the base video generation model. **Bold:** best results. Underline: second best.

Method	#Data	UCF-101			MSR-VTT		Human Eval: Consistency	
		FVD ↓	IS ↑	FID ↓	FVD ↓	CLIPSIM ↑	Appearance ↑	Motion ↑
AnimateAnything	10M+20K†	642.64	**63.87**	**10.00**	218.10	0.2661	43.07%	20.26%
I2VGen-XL	35M	597.42	18.20	42.39	270.78	0.2541	1.79%	9.43%
DynamiCrafter	10M+10M†	404.50	41.97	32.35	219.31	0.2659	44.49%	31.10%
SEINE	25M+10M†	306.49	54.02	26.00	152.63	**0.2774**	48.16%	36.76%
ConsistI2V	10M	**177.66**	56.22	15.74	**104.58**	0.2674	**53.62%**	**37.04%**

Table 4.2: Automatic evaluation results on I2V-Bench. Consist. denotes the consistency metrics. **Bold:** best results. Underline: second best.

Method	Temporal Flickering ↑	Motion Smoothness ↑	Dynamic Degree ↑	Background Consist. ↑	Subject Consist. ↑	Object Consist. ↑	Scene Consist. ↑	Overall Consist. ↑
AnimateAnything	**99.08**	**99.23**	3.69	**98.50**	**97.90**	32.56	27.18	18.74
I2VGen-XL	94.03	96.03	57.88	94.52	89.36	29.25	23.50	16.89
DynamiCrafter	93.81	95.89	**64.11**	93.47	88.01	32.18	23.93	18.68
SEINE	96.08	97.92	42.12	93.98	88.21	**35.43**	**27.68**	**20.21**
ConsistI2V	96.65	97.77	37.48	94.69	90.85	32.06	24.81	19.50

CLIPSIM on MSR-VTT's test split (2990 samples). We focus on evaluating the I2V generation capability of our model: given a video clip from the evaluation dataset, we randomly sample a frame and use it along with the text prompt as the input to our model. All evaluations are performed in a zero-shot manner.

We compare ConsistI2V against four open-sourced I2V generation models: I2VGen-XL [337], AnimateAnything [55], DynamiCrafter [303] and SEINE [49]. Quantitative evaluation results are shown in Table 4.1. We observe that while AnimateAnything achieves better IS and FID, its generated videos are mostly near static (see Figure 4.5 for visualizations), which severely limits the video quality (highest FVD of 642.64). Our model significantly outperforms the rest of the baseline models in all metrics, except for CLIPSIM on MSR-VTT, where the result is slightly lower than SEINE. We note that SEINE, initialized from LaVie [275], benefited from a larger and superior-quality training dataset, including Vimeo25M, WebVid-10M, and additional private datasets. In contrast, ConsistI2V is directly initialized from T2I models and only trained on WebVid-10M, showcasing our method's effectiveness.

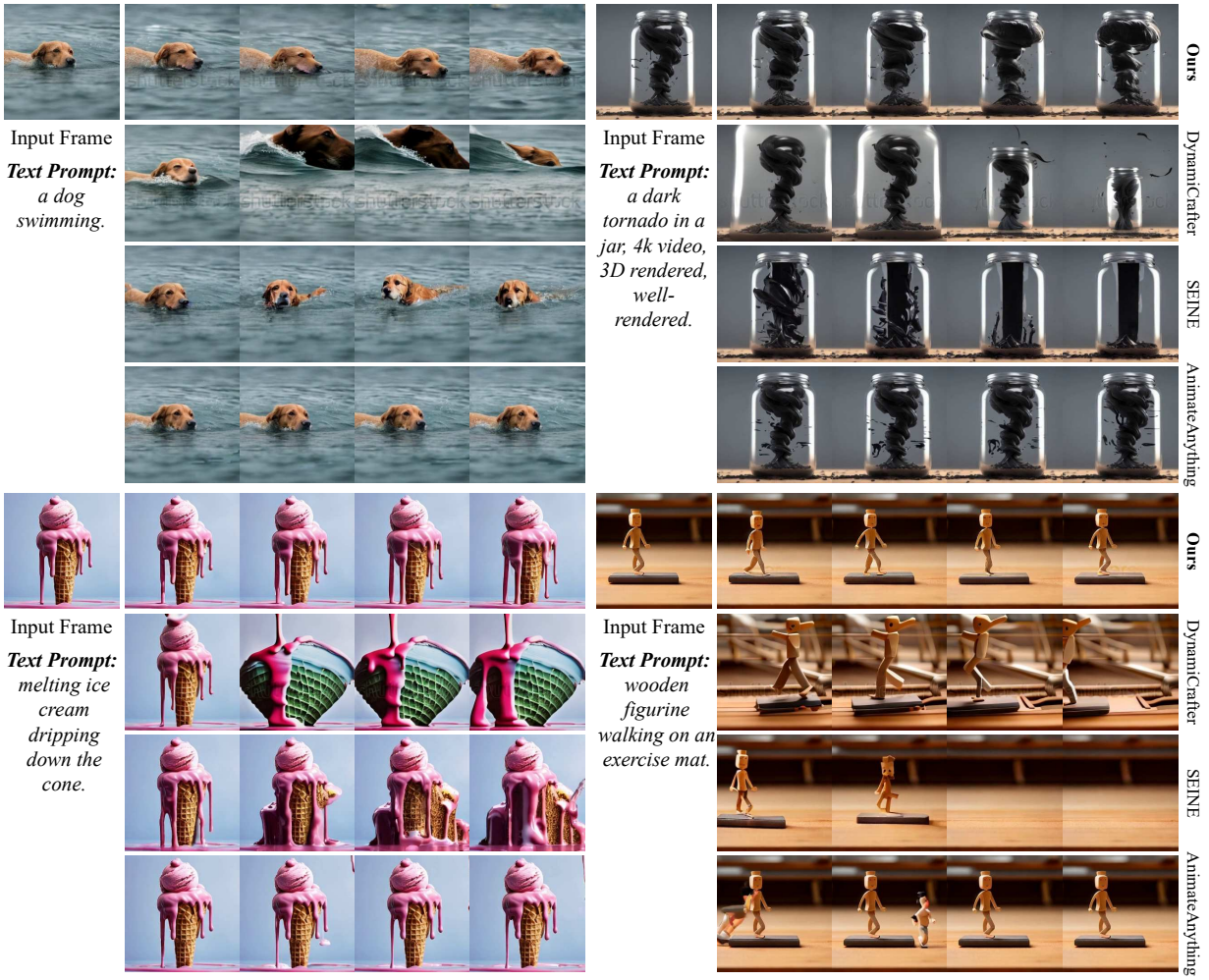


Figure 4.5: Qualitative comparisons between DynamiCrafter, SEINE, AnimateAnything and our ConsistI2V. Input first frames are generated by PixArt-α [38] and SDXL [207].

I2V-Bench We present the automatic evaluation results for ConsistI2V and the baseline models in Table 4.2. Similar to previous findings, we observe that AnimateAnything achieves the best motion smoothness and appearance consistency among all models. However, it significantly falls short in generating videos with higher motion magnitude, registering a modest dynamic degree value of only 3.69 (visual results as shown in Figure 4.5). On the other hand, our model achieves a better balance between motion magnitude and video quality, outperforms all other baseline models excluding AnimateAnything in terms of motion quality (less flickering and better smoothness) and visual consistency (higher

background/subject consistency) and achieves a competitive overall video-text consistency.

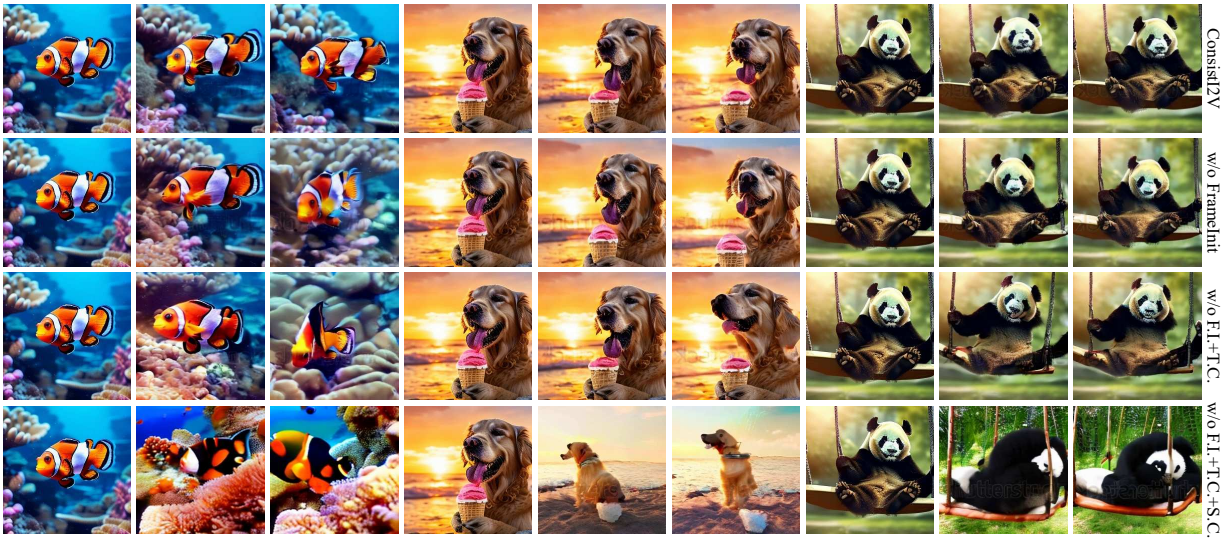
Human Evaluation To further validate the generation quality of our model, we conduct a human evaluation based on 548 samples from our ConsistI2V and the baseline models. We randomly distribute a subset of samples to each participant, presenting them with the input image, text prompt, and all generated videos. Participants are then asked to answer two questions: to identify the videos with the best overall appearance and motion consistency. Each question allows for one or more selections. We collect a total of 1061 responses from 13 participants and show the results in the right part of Table 4.1. As demonstrated by the results, our model ranked top in both metrics, achieving a comparable motion consistency with SEINE and a significantly higher appearance consistency than all other baseline models.

4.4.3 Qualitative Evaluation

We present a visual comparison of our model with DynamiCrafter, SEINE and AnimateAnything in Figure 4.5. We exclude I2VGen-XL in this section as its generated video cannot fully adhere to the visual details from the input first frame. As shown in the figure, current methods often struggle with maintaining appearance and motion consistency in video sequences. This can include (1) sudden changes in subject appearance mid-video, as demonstrated in the "ice cream" case for DynamiCrafter and SEINE; (2) background inconsistency, as observed in DynamiCrafter's "wooden figurine walking" case; (3) unnatural object movements, evident in the "dog swimming" case (DynamiCrafter) and "tornado in a jar" case (SEINE) and (4) minimal or absent movement, as displayed in most of the generated videos by AnimateAnything. On the other hand, ConsistI2V produces videos with subjects that consistently align with the input first frame. Additionally, our generated videos exhibit more natural and logical motion, avoiding abrupt changes and thereby ensuring improved appearance and motion consistency.

4.4.4 Ablation Studies

To verify the effectiveness of our design choices, we conduct an ablation study on UCF-101 by iteratively disabling FrameInit, temporal first frame conditioning and spatial first frame conditioning. We follow the same experiment setups in Section 4.4.2 and show the results in Table 4.3.



Text Prompt: *clown fish swimming through the coral reef.*

Text Prompt: *a golden retriever eating ice cream on a beautiful tropical beach at sunset.*

Text Prompt: *a panda playing on a swing set.*

Figure 4.6: Visual comparisons of our method after disabling FrameInit (F.I.), temporal conditioning (T.C.) and spatial conditioning (S.C.). We use the same seed to generate all videos.

Effectiveness of FrameInit According to the results, we find that applying FrameInit greatly enhances the model performance for all metrics. Our empirical observation suggests that FrameInit can stabilize the output video and reduce the abrupt appearance and motion changes. As shown in Figure 4.6, while our model can still generate reasonable results without enabling FrameInit, the output videos also suffer from a higher chance of rendering sudden object movements and blurry frames (last frame in the second row). This highlights the effectiveness of FrameInit in producing more natural motion and higher-quality frames.

Hyperparameters of FrameInit To further investigate how the FrameInit hyperparameters τ (number of steps of inference noise added to the first frame) and D_0 (spatiotemporal stop frequency) affect the generated videos, we conduct an additional qualitative experiment by generating the same video using different FrameInit hyperparameters and show the results in Figure 4.7. Comparing (1) and (4) in Figure 4.7, we find that while using our default setting of FrameInit achieves more stable outputs, it also slightly restricts the motion magnitude of the generated video compared to not using FrameInit at all. From comparisons between (1), (5), and (1), (6), we observe that either increasing D_0 alone or reducing τ alone does not significantly impact the motion magnitude of the videos. However, comparisons between (1), (2), and (3) reveal that when both D_0 is increased and τ is decreased, there is a further restriction in video dynamics. This suggests that

Table 4.3: UCF-101 ablation study results and runtime statistics for spatiotemporal first frame conditioning and FrameInit. T.Cond. and S.Cond. correspond to temporal and spatial first frame conditioning, respectively.

	UCF-101 (Zero-Shot)			Inference Cost Statistics		
	FVD ↓	IS ↑	FID ↓	GPU Mem.	Runtime	TFLOPs (UNet)
ConsistI2V	177.66	56.22	15.74	9249 Mb	18.48s	5.168
w/o FrameInit	245.79	42.21	24.08	9245 Mb	18.37s	5.168
w/o FrameInit & T.Cond.	224.16	42.98	24.04	9233 Mb	17.93s	5.117
w/o FrameInit & T.Cond. & S.Cond.	704.48	21.97	68.39	9017 Mb	17.14s	5.015

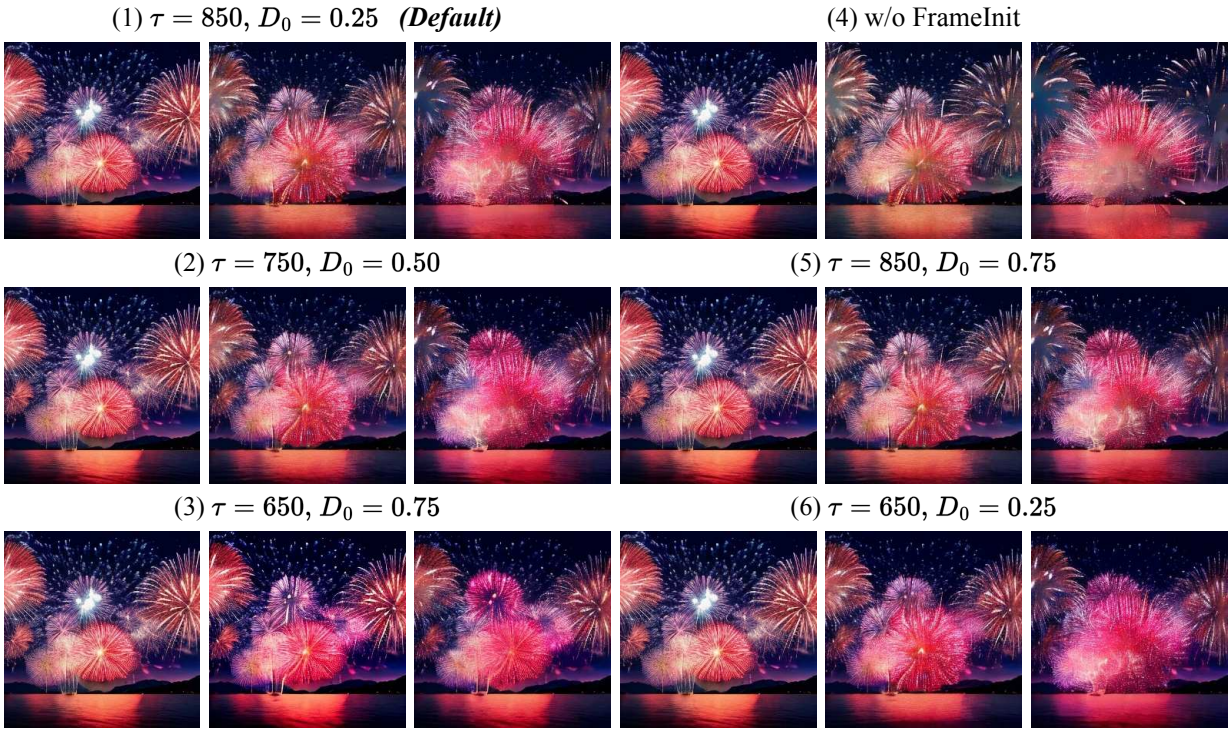


Figure 4.7: Visual comparisons of employing different FrameInit hyperparameters during I2V generation. All videos are generated using the same seed.

the static first frame video exerts a stronger layout guidance, intensifying its influence on video dynamics.

Spatiotemporal First Frame Conditioning Our ablation results on UCF-101 (*c.f.*

Table 4.3) reflects a significant performance boost after applying the proposed spatial and temporal first frame conditioning mechanisms to our model. Although removing temporal first frame conditioning leads to an overall better performance for the three quantitative metrics, in practice we find that only using spatial conditioning often results in jittering motion and larger object distortions, evident in the third row of Figure 4.6. When both spatial and temporal first frame conditioning is removed, our model loses the capability of maintaining the appearance of the input first frame.

Runtime Efficiency To better understand how the proposed model components affect the inference cost, we also benchmark the runtime statistics of our ConsistI2V and its variants after disabling different components. We measure all models on a single Nvidia RTX 4090 with float32 and an inference batch size of 1. For TFLOPs computation, we only consider the denoising U-Net and exclude components such as the VAE and text encoder as these modules share the same computation across all model variants. As shown in the right part of Table 4.3, we find that the proposed spatiotemporal feature conditioning and FrameInit incur limited computational overhead during inference, introducing approximately 200Mb extra GPU memory and ∼1.3s additional inference time. These results demonstrate that our method is highly efficient during inference.

4.4.5 More Applications

Autoregressive Long Video Generation While our I2V model provides native support for long video generation by reusing the last frame of the previous video to generate the next video, we observe that directly using the model to generate long videos may lead to suboptimal results, as the artifacts in the previous video clip will often accumulate throughout the autoregressive generation process. We find that using FrameInit to guide the generation of each video chunk helps stabilize the autoregressive video generation process and results in a more consistent visual appearance throughout the video, as shown in Figure 4.8.

Camera Motion Control When adopting FrameInit for inference, instead of using the static first frame video as the input, we can alternatively create synthetic camera motions from the first frame and use it as the layout condition. For instance, camera panning can be simulated by creating spatial crops in the first frame starting from one side and gradually moving to the other side. As shown in the first two examples under camera motion control in Figure 4.8, by simply tweaking the FrameInit parameters to $\tau = 750$ and $D_0 = 0.5$ and using the synthetic camera motion as the layout guidance, we are able

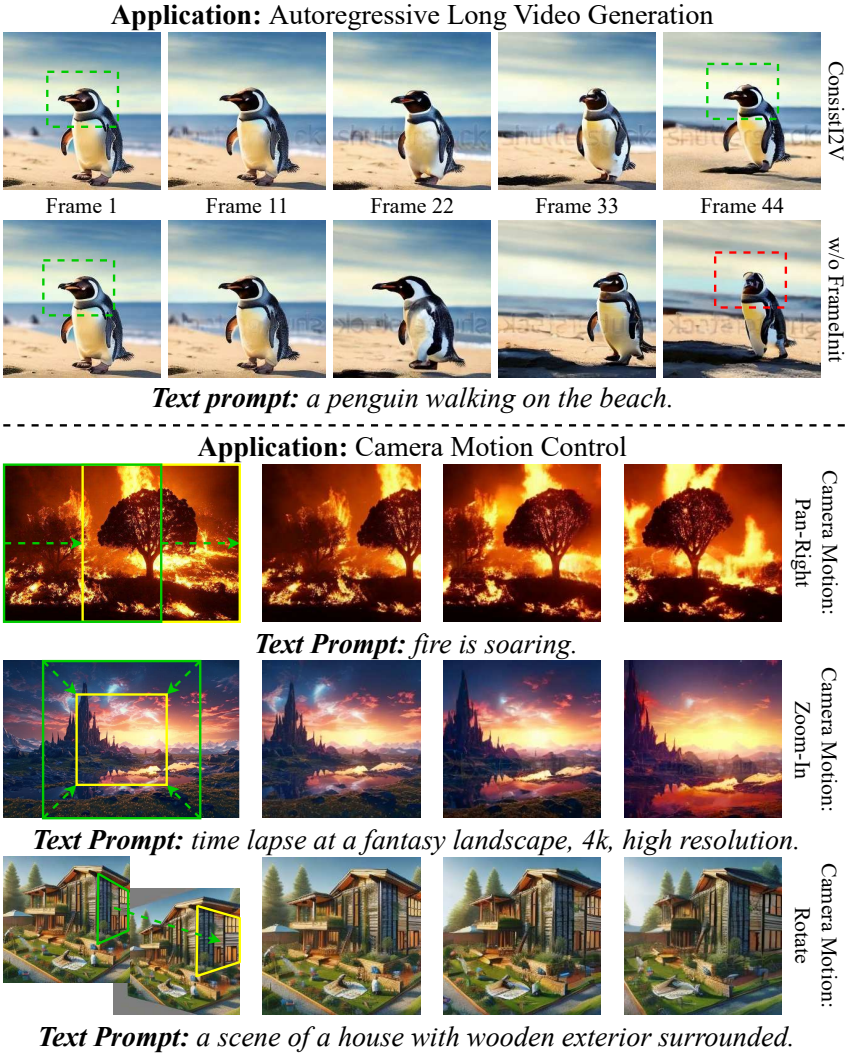


Figure 4.8: Applications of ConsistI2V. **Upper panel:** FrameInit enhances object consistency in long video generation. **Lower panel:** FrameInit enables training-free camera motion control.

to achieve camera panning and zoom-in/zoom-out effects in the generated videos without any additional training. For camera motions that involve perspective view change (e.g. rotating), we propose to apply image warping to the input first frame to create a sequence of perspective transformations. We then use this synthetic perspective view change as the layout condition to generate new videos. As shown in the last example of Figure 4.8, we

can synthesize camera rotation by creating synthetic perspective view changes based on the coordinates of four selected key points in the first frame. As perspective view changes are often more complex than 2D panning/zooming, the FrameInit hyperparameters for this example are selected as $\tau = 700$ and $D_0 = 0.5$ to enforce a stronger layout guidance.

4.5 Conclusion

We presented ConsistI2V, an I2V generation framework designed to improve the visual consistency of generated videos by integrating our novel spatiotemporal first frame conditioning and FrameInit layout guidance mechanisms. Our approach enables the generation of highly consistent videos and supports applications including autoregressive long video generation and camera motion control. We conducted extensive automatic and human evaluations on various benchmarks, including our proposed I2V-Bench and demonstrated exceptional I2V generation results. For future work, we plan to refine our training paradigm and incorporate higher-quality training data to further scale up the capacity of our ConsistI2V.

Chapter 5

AnyV2V: A Tuning-Free Framework For Any Video-to-Video Editing Tasks

5.1 Introduction

The development of deep generative models [102] has led to significant advancements in content creation and manipulation, especially in digital images [222, 195, 23, 131, 46, 150]. However, video generation and editing have not reached the same level of advancement as images [275, 35, 36, 104]. In the context of video editing, training a large-scale video editing model presents considerable challenges due to the scarcity of paired data and the substantial computational resources required.

To overcome these challenges, researchers proposed various approaches [81, 54, 284, 291, 171, 156, 94, 54, 338, 208, 29, 312, 118, 95], which can be categorized into two types: (1) zero-shot adaptation from pre-trained text-to-image (T2I) models or (2) fine-tuned motion module from a pre-trained T2I or text-to-video (T2V) models. The zero-shot methods [81, 54, 118, 338] usually suffer from flickering issues due to a lack of temporal understanding. On the other hand, fine-tuning methods [284, 45, 291, 94, 95] require more time and computational overhead to edit videos. Moreover, all the methods can only adhere to certain types of edits. For example, a user might want to perform edits on visuals that are out-of-distribution from the learned text encoder (e.g. change a person to a character from their artwork). Thus, a highly customizable solution is more desired in video editing applications. It would be ideal if there were methods that allowed for a seamless

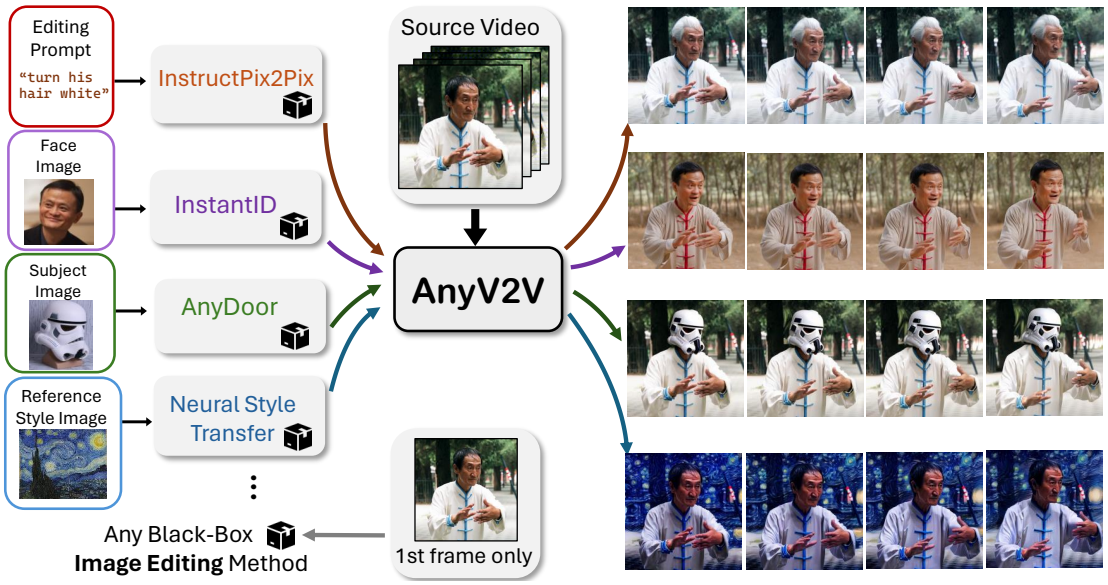


Figure 5.1: AnyV2V is an evolving framework to handle all types of video-to-video editing tasks without any parameter tuning. AnyV2V disentangles video editing into two simpler problems: (1) Single image editing and (2) Image-to-video generation with video reference.

combination of human artistic input and the assistance provided by AI, allowing for a synergy between human effort and AI, maintaining creators' creativity while producing high-quality outputs.

With the above motivations, we aim to develop a video editing framework that requires no fine-tuning and caters to user demand. In this chapter, we present AnyV2V, designed to enhance the controllability of zero-shot video editing by decomposing the task into two pivotal stages:

1. Apply image edit on the first frame with any off-the-shelf image editing model or human effort.
2. Leverage the innate knowledge of the image-to-video (I2V) model to generate the edited video with the edited first frame, source video latent, and the intermediate temporal features.

Our objective is to propagate the edited first frame across the entire video while ensuring alignment with the source video. To achieve this, we employ I2V models [337, 49, 217] for DDIM inversion to enable first-frame conditioning. With the inverted latents as initial noise and the modified first frame as the conditional signal, the I2V model can generate

videos that are not only faithful to the edited first frame but also follow the appearance and motion of the source video. To further enforce the consistency of the appearance and motion with the source video, we perform feature injection in the I2V model. The two-stage editing process effectively offloads the editing operation to existing image editing tools. This design (detail in Section 5.3) helps AnyV2V excel in:

- **Compatibility**: It provides a highly customized interface for a user to perform video edits on any modality. It can seamlessly integrate any image editing methods to perform diverse editing.
- **Simplicity**: It does not require any fine-tuning nor additional video features like previous works [94, 291, 203] to achieve high appearance and temporal consistency for video editing tasks.

From our findings, without any fine-tuning, AnyV2V can perform video editing tasks beyond the scope of current publicly available methods, such as reference-based style transfer, subject-driven editing, and identity manipulation. AnyV2V also can perform prompt-based editing and achieve superior results on common video editing evaluation metrics compared to the baseline models [81, 54, 291]. We show both quantitatively and qualitatively that our method outperforms existing SOTA baselines in Section 5.4.3. AnyV2V is favoured in 69.7% of samples for prompt alignment and 46.2% overall preference in human evaluation, while the best baseline only achieves 31.7% and 20.7%, respectively. Our method also reaches the highest CLIP-Text score of 0.2932 in text alignment and a competitive CLIP-Image score of 0.9652 in temporal consistency.

All these achievements are thanks to the AnyV2V's design to harness the power of off-the-shelf image editing models from advanced image editing research. Through a comprehensive study and evaluation of the effectiveness of our design, our key observation is that the inverted noise latent and feature injection serve as critical components to guide the video motion, and the I2V model itself has good capabilities in generating motions. We also found that by inverting long videos exceeding the I2V models' training frames, the inverted latents enable the I2V model to produce longer videos, making long video editing possible. To summarize, The main contributions of our work are three-fold:

- We proposed AnyV2V as a first fundamentally different solution for video editing, treating video editing as a simpler image editing problem.
- We showed that AnyV2V can support long video editing by inverting videos that extend beyond the training frame lengths of I2V models.
- Our extensive experiments showcased the superior performance of AnyV2V when compared to the existing SOTA methods, highlighting the potential of leveraging I2V models for video editing.

5.2 Preliminary

5.2.1 DDIM Inversion

The denoising process for DDPM-based image-to-video generation models from $\mathbf{z}_t$ to $\mathbf{z}_{t-1}$ can be achieved using the DDIM [234] sampling algorithm. The reverse process of DDIM sampling, known as DDIM inversion [192, 61], allows obtaining $\mathbf{z}_{t+1}$ from $\mathbf{z}_t$ such that $\mathbf{z}_{t+1} = \sqrt{\frac{\alpha_{t+1}}{\alpha_t}}\mathbf{z}_t + (\sqrt{\frac{1}{\alpha_{t+1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1}) \cdot \epsilon_\theta(\mathbf{z}_t, x_0, \mathbf{s}, t)$, where α_t is derived from the variance schedule of the diffusion process.

5.2.2 Plug-and-Play (PnP) Diffusion Features

Tumanyan et al. [255] proposed PnP diffusion features for image editing, based on the observation that intermediate convolution features f and self-attention scores $A = \text{Softmax}(\frac{QK^\top}{\sqrt{d}})$ in a text-to-image (T2I) denoising U-Net capture the semantic regions (e.g. legs or torso of a human body) during the image generation process.

Given an input source image I^S and a target prompt P, PnP first performs DDIM inversion to obtain the image's corresponding noise $\{\mathbf{z}_t^S\}_{t=1}^T$ at each time step t. It then collects the convolution features $\{f_t^l\}$ and attention scores $\{A_t^l\}$ from some predefined layers l at each time step t of the backward diffusion process $\mathbf{z}_{t-1}^S = \epsilon_\theta(\mathbf{z}_t^S, \varnothing, t)$, where $\varnothing$ denotes the null text prompt during denoising.

To generate the edited image I^*, PnP starts from the initial noise of the source image (i.e. $\mathbf{z}_T^* = \mathbf{z}_T^S$) and performs feature injection during denoising: $\mathbf{z}_{t-1}^* = \epsilon_\theta(\mathbf{z}_t^*, P, t, \{f_t^l, A_t^l\})$, where $\epsilon_\theta(\cdot, \cdot, \cdot, \{f_t^l, A_t^l\})$ represents the operation of replacing the intermediate feature and attention scores $\{f_t^{l*}, A_t^{l*}\}$ with $\{f_t^l, A_t^l\}$. This feature injection mechanism ensures I^* to preserve the layout and structure from I^S while reflecting the description in P. To control the feature injection strength, PnP also employs two thresholds τ_f and τ_A such that the feature and attention scores are only injected in the first τ_f and τ_A denoising steps. Our method extends this feature injection mechanism to I2V generation models, where we inject features in convolution, spatial, and temporal attention layers. We show the detailed design of AnyV2V in Section 5.3.

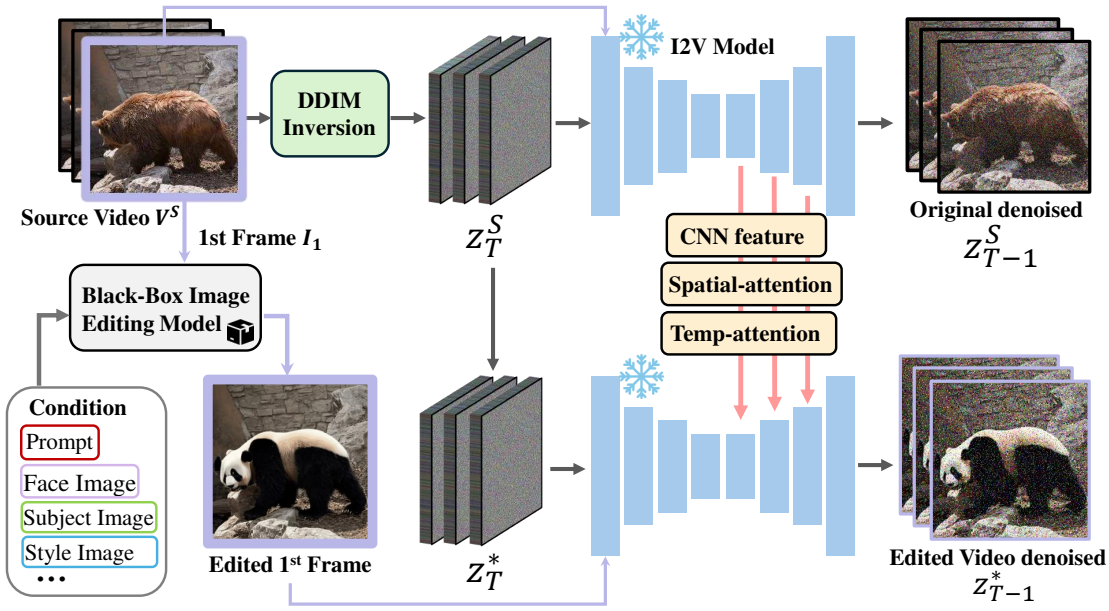


Figure 5.2: AnyV2V takes a source video V^S as input. In the first stage, we apply a black-box image editing method on the first frame I_1 according to the editing task. In the second stage, the source video is inverted to initial noise z_T^S, which is then denoised using DDIM sampling. During the sampling process, we extract spatial convolution, spatial attention, and temporal attention features from the I2V models' decoder layers. To generate the edited video, we perform a DDIM sampling by fixing z_T^* as z_T^T and use the edited first frame as the conditional signal. During sampling, we inject the features and attention into corresponding layers of the model.

5.3 AnyV2V

Our method presents a two-stage approach to video editing. Given a source video $V^S = \{I_1, I_2, I_3, ..., I_n\}$, where I_i is the frame at time i and n denotes the video length, we first extract the initial frame I_1 and pass it into an image editing model ϕ_{img} to obtain an edited first frame $I_1^* = \phi_{\text{img}}(I_1, C)$. C is the auxiliary conditions for image editing models such as text prompt, mask, style, etc. In the second stage, we feed the edited first frame I_1^* and a target prompt $\mathbf{s}^*$ into an I2V generation model ϵ_θ and employ the inverted latent from the source video V^S to guide the generation process such that the edited video V^* follows the motion of the source video V^S, the semantic information represented in the edited first frame I_1^*, and the target prompt $\mathbf{s}^*$. An overall illustration of our video editing pipeline is shown in Figure 5.2. In this section, we explain each core component of our method.

5.3.1 Flexible First Frame Editing

In visual manipulation, controllability is a key element in performing precise editing. AnyV2V enables more controllable video editing by utilizing image editing models to modify the video's first frame. This strategic approach enables highly accurate modifications in the video and is compatible with a broad spectrum of image editing models, including other deep learning models that can perform image style transfer [77, 82, 185, 262], mask-based image editing [195, 8], image inpainting [241, 133], identity-preserving image editing [266], and subject-driven image editing [46, 150, 93].

5.3.2 Structural Guidance using DDIM Inverison

To ensure the generated videos from the I2V generation model follow the general structure as presented in the source video, we employ DDIM inversion to obtain the latent noise of the source video at each time step t. Specifically, we perform the inversion *without* text prompt condition but *with* the first frame condition. Formally, given a source video $V^S = \{I_1, I_2, I_3, ..., I_n\}$, we obtain the inverted latent noise for time step t as:

$$\mathbf{z}_t^S = \text{DDIM_Inv}(\epsilon_\theta(\mathbf{z}_{t+1}, I_1, \varnothing, t)), \tag{5.1}$$

where $\text{DDIM_Inv}(\cdot)$ denotes the DDIM inversion operation as described in Section 5.2. In ideal cases, the latent noise $\mathbf{z}_T^S$ at the final time step T (initial noise of the source video) should be used as the initial noise for sampling the edited videos. In practice, we find that due to the limited capability of certain I2V models, the edited videos denoised from the last time step are sometimes distorted. Following [150], we observe that starting the sampling from a previous time step $T' < T$ can be used as a simple workaround to fix this issue.

5.3.3 Appearance Guidance via Spatial Feature Injection

Our empirical observation (Section 5.4.4) suggests that I2V generation models already have some editing capabilities by only using the edited first frame and DDIM inverted noise as the model input. However, we find that this simple approach is often unable to correctly preserve the background in the edited first frame and the motion in the source video, as the conditional signal from the source video encoded in the inverted noise is limited.

To enforce consistency with the source video, we perform feature injection in both convolution layers and spatial attention layers in the denoising U-Net. During the video

sampling process, we simultaneously denoise the source video using the previously collected DDIM inverted latents $\mathbf{z}_t^S$ at each time step t such that $\mathbf{z}_{t-1}^S = \epsilon_\theta(\mathbf{z}_t^S, I_1, \varnothing, t)$. We preserve two types of features during source video denoising: convolution features f^{l_1} before skip connection from the l_1^{th} residual block in the U-Net decoder, and the spatial self-attention scores $\{A_s^{l_2}\}$ from $l_2 = \{l_{low}, l_{low+1}, ..., l_{high}\}$ layers. We collect the queries $\{Q_s^{l_2}\}$ and keys $\{K_s^{l_2}\}$ instead of directly collecting $A_s^{l_2}$ as the attention score matrices are parameterized by the query and key vectors. We then replace the corresponding features during denoising the edited video in both the normal denoising branch and the negative prompt branch for classifier-free guidance [103]. We use two thresholds τ_{conv} and τ_{sa} to control the convolution and spatial attention injection to only happen in the first τ_{conv} and τ_{sa} steps during video sampling.

5.3.4 Motion Guidance through Temporal Feature Injection

The spatial feature injection mechanism described in Section 5.3.3 significantly enhances the background and overall structure consistency of the edited video. While it also helps maintain the source video motion to some degree, we observe that the edited videos will still have a high chance of containing incorrect motion compared to the source video. On the other hand, we notice that I2V generation models, or video diffusion models in general, are often initialized from pre-trained T2I models and continue to be trained on video data. During the training process, parameters in the spatial layers are often frozen or set to a lower learning rate such that the pre-trained weights from the T2I model are less affected, and the parameters in the temporal layers are more extensively updated during training. Therefore, it is likely that a large portion of the motion information is encoded in the temporal layers of the I2V generation models. Concurrent work [12] also observes that features in the temporal layers show similar characteristics with optical flow [109], a pattern that is often used to describe the motion of the video.

To better reconstruct the source video motion in the edited video, we propose to also inject the temporal attention features in the video generation process. Similar to spatial attention injection, we collect the source video temporal self-attention queries $Q_t^{l_3}$ and keys $K_t^{l_3}$ from some U-Net decoder layers represented by l_3 and inject them into the edited video denoising branches. We also only apply temporal attention injection in the first τ_{ta} steps during sampling.

5.3.5 Putting it Together

Overall, combining the spatial and temporal feature injection mechanisms, we replace the editing branch features $\{f^{*l_1}, Q_s^{*l_2}, K_s^{*l_2}, Q_t^{*l_3}, K_t^{*l_3}\}$ with the features from the source denoising branch:

$$\mathbf{z}_{t-1}^{*} = \epsilon_\theta(\mathbf{z}_t^{*}, I^{*}, \mathbf{s}^{*}, t\ ; \{f^{l_1}, Q_s^{l_2}, K_s^{l_2}, Q_t^{l_3}, K_t^{l_3}\}), \tag{5.2}$$

where $\epsilon_\theta(\cdot\ ; \{f^{l_1}, Q_s^{l_2}, K_s^{l_2}, Q_t^{l_3}, K_t^{l_3}\})$ denotes the feature replacement operation across different layers l_1, l_2, l_3. Our proposed spatial and temporal feature injection scheme enables tuning-free adaptation of I2V generation models for video editing. Our experimental results demonstrate that each component in our design is crucial to the accurate editing of source videos. We showcase more qualitative results for the effectiveness of our model components in Section 5.4.

5.4 Experiments

5.4.1 Implementation Details

We employ AnyV2V on three off-the-shelf I2V generation models: I2VGen-XL[1] [337], ConsistI2V [217] and SEINE [49]. For all I2V models, we use $\tau_{conv} = 0.2T$, $\tau_{sa} = 0.2T$ and $\tau_{ta} = 0.5T$, where T is the total number of sampling steps. We use the DDIM [234] sampler and set T to the default values of the selected I2V models. Following PnP [256], we set $l_1 = 4$ for convolution feature injection and $l_2 = l_3 = \{4, 5, 6, ..., 11\}$ for spatial and temporal attention injections. During sampling, we apply text classifier-free guidance (CFG) [103] for all models with the same negative prompt *"Distorted, discontinuous, Ugly, blurry, low resolution, motionless, static, disfigured, disconnected limbs, Ugly faces, incomplete arms"* across all edits. To obtain the initial edited frames in our implementation, we use a set of image editing model candidates including prompt-based image editing model InstructPix2Pix [23], style transfer model Neural Style Transfer (NST) [77], subject-driven image editing model AnyDoor [46], and identity-driven image editing model InstantID [266]. We experiment with only the successfully edited frames, which is crucial for our method. We conducted all the experiments on a single Nvidia A6000 GPU. To edit a 16-frame video, it requires around 15G GPU memory and around 100 seconds for the whole inference process.

[1]We use the version provided in https://huggingface.co/ali-vilab/i2vgen-xl.

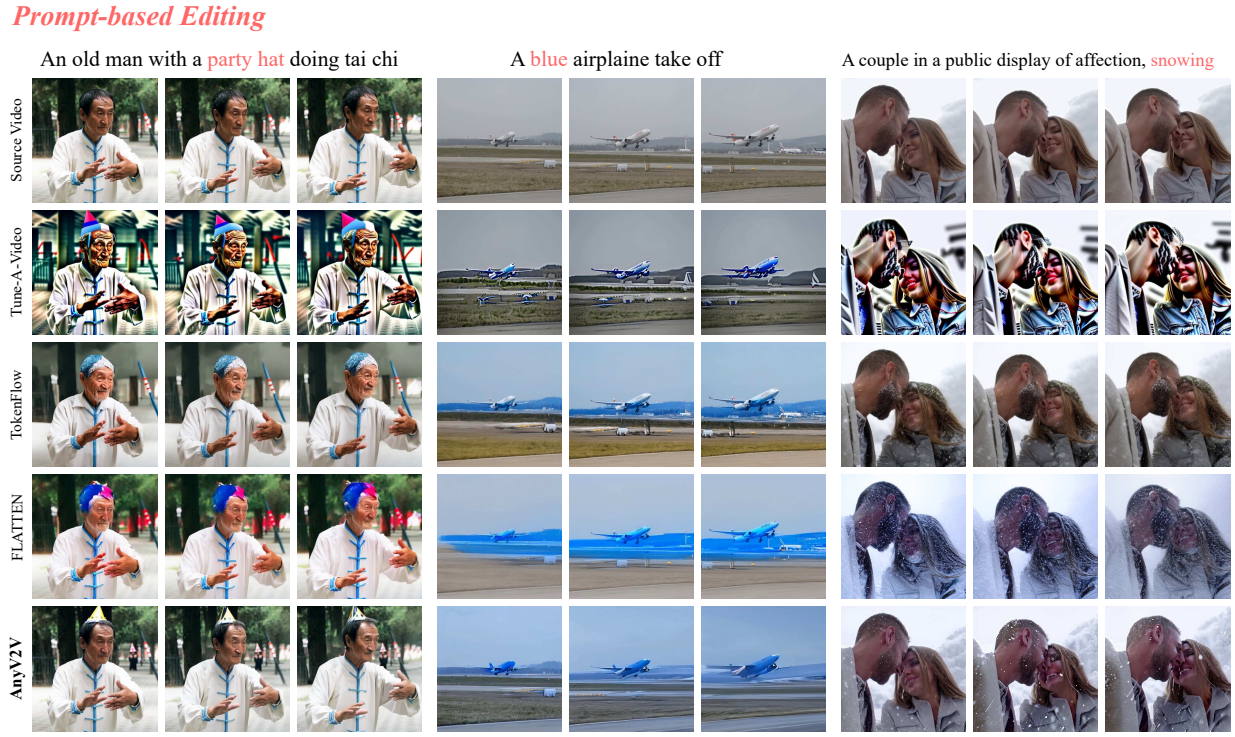


Figure 5.3: AnyV2V is robust in a wide range of prompt-based editing tasks while preserving the background. The results align the most with the text prompt and maintain high motion consistency.

5.4.2 Tasks Definition

1. **Prompt-based Editing:** allows users to manipulate video content using only natural language. This can include descriptive prompts or instructions. With the prompt, Users can perform a wide range of edits, such as incorporating accessories, spawning or swapping objects, adding effects, or altering the background.

2. **Reference-based Style Transfer:** In the realm of style transfer tasks, the artistic styles of Monet and Van Gogh are frequently explored, but in real-life examples, users might want to use a distinct style based on one particular artwork. In reference-based style transfer, we focus on using a style image as a reference to perform video editing. The edited video should capture the distinct style of the referenced artwork.

3. **Subject-driven Editing:** In subject-driven video editing, we aim at replacing an

Table 5.1: Quantitative comparisons for our AnyV2V with baselines on prompt-based video editing. Alignment: prompt alignment; Overall: overall preference. **Bold**: best results; Underline: top-2.

Method	Human Evaluation ↑		CLIP Scores ↑	
	Alignment	Overall	CLIP-Text	CLIP-Image
Tune-A-Video	15.2%	2.1%	0.2902	0.9704
TokenFlow	31.7%	20.7%	0.2858	**0.9783**
FLATTEN	25.5%	16.6%	0.2742	0.9739
AnyV2V (SEINE)	28.9%	8.3%	0.2910	0.9631
AnyV2V (ConsistI2V)	33.8%	11.7%	0.2896	0.9556
AnyV2V (I2VGen-XL)	**69.7%**	**46.2%**	**0.2932**	0.9652

object in the video with a target subject based on a given subject image while maintaining the video motion and persevering the background.

4. **Identity Manipulation:** Identity manipulation allows the user to manipulate video content by replacing a person with another person's identity in the video based on an input image of the target person.

5.4.3 Evaluation Results

As shown in Figure 5.3 and 5.4, AnyV2V can perform the following video editing tasks: (1) prompt-based editing, (2) reference-based style transfer, (3) subject-driven editing, and (4) identity manipulation. we compare AnyV2V against three baseline models Tune-A-Video [291], TokenFlow [81] and FLATTEN [54] for (1) prompt-based editing. Since there exists no publicly available baseline method for task (2) (3) (4), we evaluate the performance of three I2V generation models under AnyV2V.

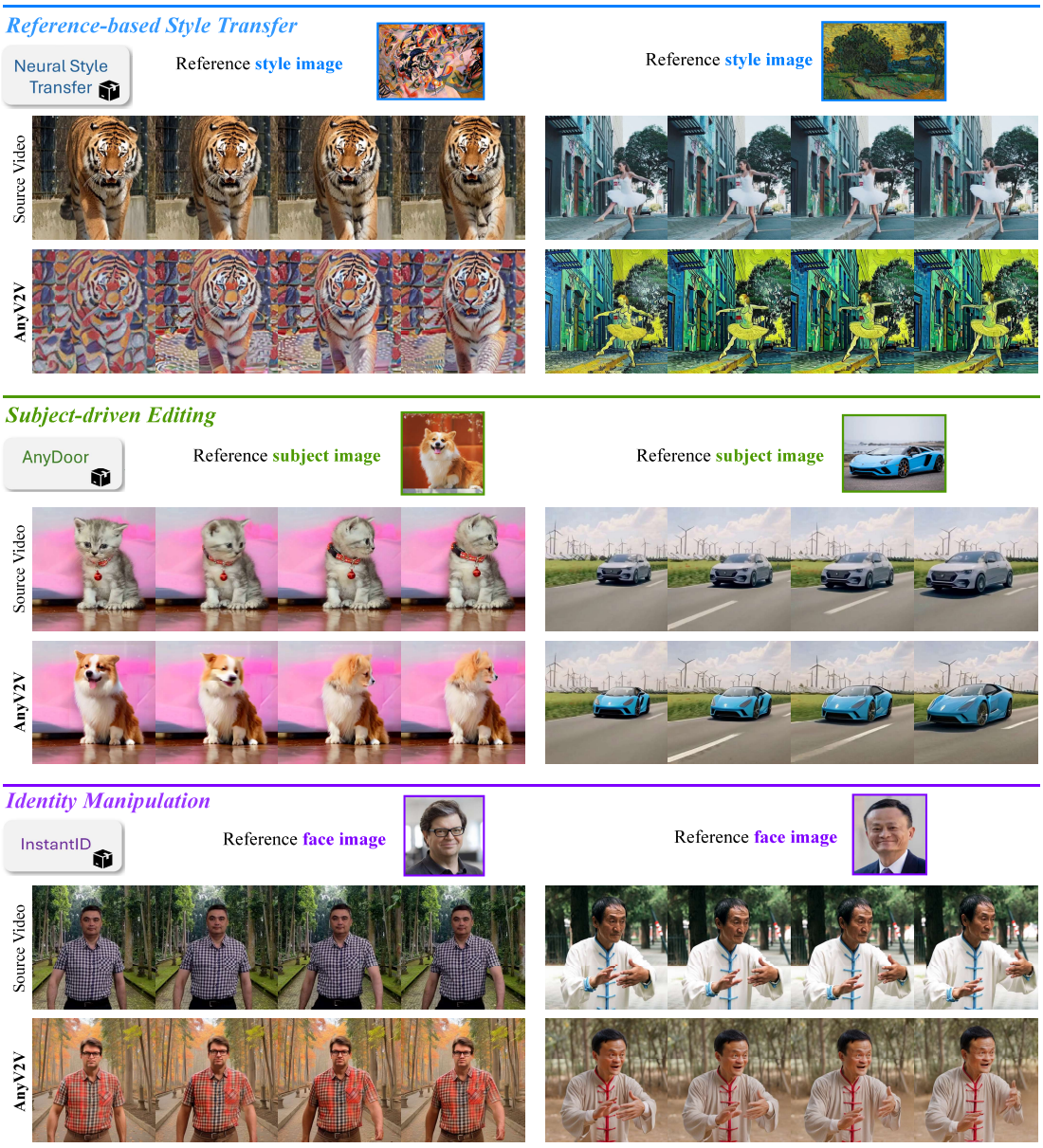


Figure 5.4: With different image editing models, AnyV2V can achieve a wide range of editing tasks, including reference-based style transfer, subject-driven editing, and identity manipulation.

Prompt-based Editing Unlike the baseline methods [291, 81, 54] that often introduce unwarranted changes not specified in the text commands, AnyV2V utilizes the precision of image editing models to ensure only the targeted areas of the scene are altered, leaving the rest unchanged. Combining AnyV2V with the instruction-guided image editing model InstructPix2Pix [23], AnyV2V accurately places a party hat on an elderly man's head and correctly paints the plane in blue. Additionally, it maintains the original video's background and fidelity, whereas, in comparison, baseline methods often alter the color tone and shape of objects, as illustrated in Figure 5.3. Also, for motion tasks such as adding snowing weather, I2V models from AnyV2V provide inherent support for animating the snowing while the baseline methods would result in flickering. For quantitative evaluations, we conduct a human evaluation to examine the degree of prompt alignment and overall preference of the edited videos based on user voting, and also compute the metrics CLIP-Text for text alignment and CLIP-Image for temporal consistency. In detail, CLIP-Text is computed by the average cosine similarity between text embeddings from CLIP model [212] and CLIP-Image is computed in the same way but between image embeddings for every pair of consecutive frames. Table 5.1 shows that AnyV2V generally achieves high text-alignment and temporal consistency, while AnyV2V with I2VGen-XL backbone is the most preferred method because it does not edit the video precisely.

Style Transfer, Subject-Driven Editing and Identity Manipulation For these novel tasks, we stress the alignment with the reference image instead of the text prompt. As shown in Figure 5.4, we can observe that for task (2), AnyV2V can capture one particular style that is tailor-made, and even if the style is not learned by the text encoder. In the examples, AnyV2V captures the style of Vassily Kandinsky's artwork "Composition VII" and Vincent Van Gogh's artwork "Chateau in Auvers at Sunset" accurately. For task (3), AnyV2V can replace the subject in the video with other subjects even if the new subject differs from the original subject. In the examples, a cat is replaced with a dog according to the reference image and maintains highly aligned motion and background as reflected in the source video. A car is replaced by our desired car while the wheel is still spinning in the edited video. For task (4), AnyV2V can swap a person's identity to anyone. We report both human evaluation results and find that AnyV2V with I2VGen-XL backbone is the most preferred method in terms of reference alignment and overall performance, plotted in Table 5.2.

I2V Backbones We find that AnyV2V (I2VGen-XL) tends to be the most robust both qualitatively and quantitatively. It has a good generalization ability to produce consistent

Table 5.2: Comparisons for three I2V models under AnyV2V framework on novel video editing tasks. Align: reference alignment; Overall: overall preference. **Bold**: best results; Underline: top-2.

Task	Reference-based Style Transfer		Subject-driven Editing		Identity Manipulation	
Image Editing Method	NST		AnyDoor		InstantID	
Image Editing Success Rate	≈90%		≈10%		≈80%	
Human Evaluation	Align ↑	Overall ↑	Align ↑	Overall ↑	Align ↑	Overall ↑
AnyV2V (SEINE)	92.3%	30.8%	48.4%	15.2%	72.7%	18.2%
AnyV2V (ConsistI2V)	38.4%	10.3%	63.6%	42.4%	72.7%	27.3%
AnyV2V (I2VGen-XL)	**100.0%**	**76.9%**	**93.9%**	**84.8%**	**90.1%**	**45.4%**

motions in the video with high visual quality. AnyV2V (ConsistI2V) can generate consistent motion, but sometimes the watermark would appear due to its training data, thus harming the visual quality. AnyV2V (SEINE)'s generalization ability is relatively weaker but still produces consistent and high-quality video if the motion is simple enough, such as a person walking.

Editing Video beyond Training Frames of I2V model Current state-of-the-art I2V models [49, 217, 337] are mostly trained on video data that contains only 16 frames. To edit videos that have length beyond the training frames of the I2V model, an intuitive approach would be generating videos in an auto-regressive manner as used in ConsistI2V [217] and SEINE [49]. However, we find that such an experiment setup cannot maintain semantic consistency in our case. As many works in extending video generation exploit the initial latent to generate longer video [209, 294], we leverage the longer inverted latent as the initial latent and force an I2V model to generate longer frames of output. Our experiments found that the inverted latent contains enough temporal and semantic information to allow the generated video to maintain temporal and semantic consistency, as shown in Figure 5.5.

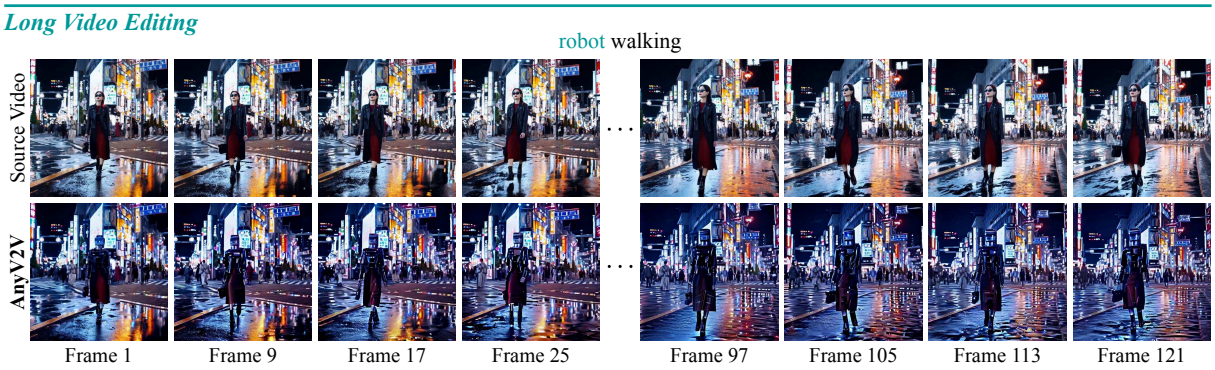


Figure 5.5: AnyV2V can edit video length beyond the training frame while maintaining motion consistency. The first row is the source video frames while the second rows are the edited. The editing prompt of the image was "turn woman into a robot" using image model InstructPix2Pix [23].

5.4.4 Ablation Study

To verify the effectiveness of our design choices, we conduct an ablation study by iteratively disabling the three core components in our model: temporal feature injection, spatial feature injection, and DDIM inverted latent as initial noise. We use AnyV2V (I2VGen-XL) and a subset of 20 samples in this ablation study and report both the frame-wise consistency results using CLIP-Image score in Table 5.3 and qualitative comparisons in Figure 5.6.

Effectiveness of Temporal Feature Injection According to the results, after disabling temporal feature injection in AnyV2V (I2VGen-XL), while we observe a slight increase in the CLIP-Image score value, the edited videos often demonstrate less adherence to the motion presented in the source video. For example, in the second frame of the "couple sitting" case (3^{rd} row, 2^{nd} column in the right panel in Figure 5.6), the motion of the woman raising her leg in the source video is not reflected in the edited video without applying temporal injection. On the other hand, even when the style of the video is completely changed, AnyV2V (I2VGen-XL) with temporal injection is still able to capture this nuance motion in the edited video.

Effectiveness of Spatial Feature Injection As shown in Table 5.3, we observe a drop in the CLIP-Image score after removing the spatial feature injection mechanisms from our model, indicating that the edited videos are not smoothly progressed across

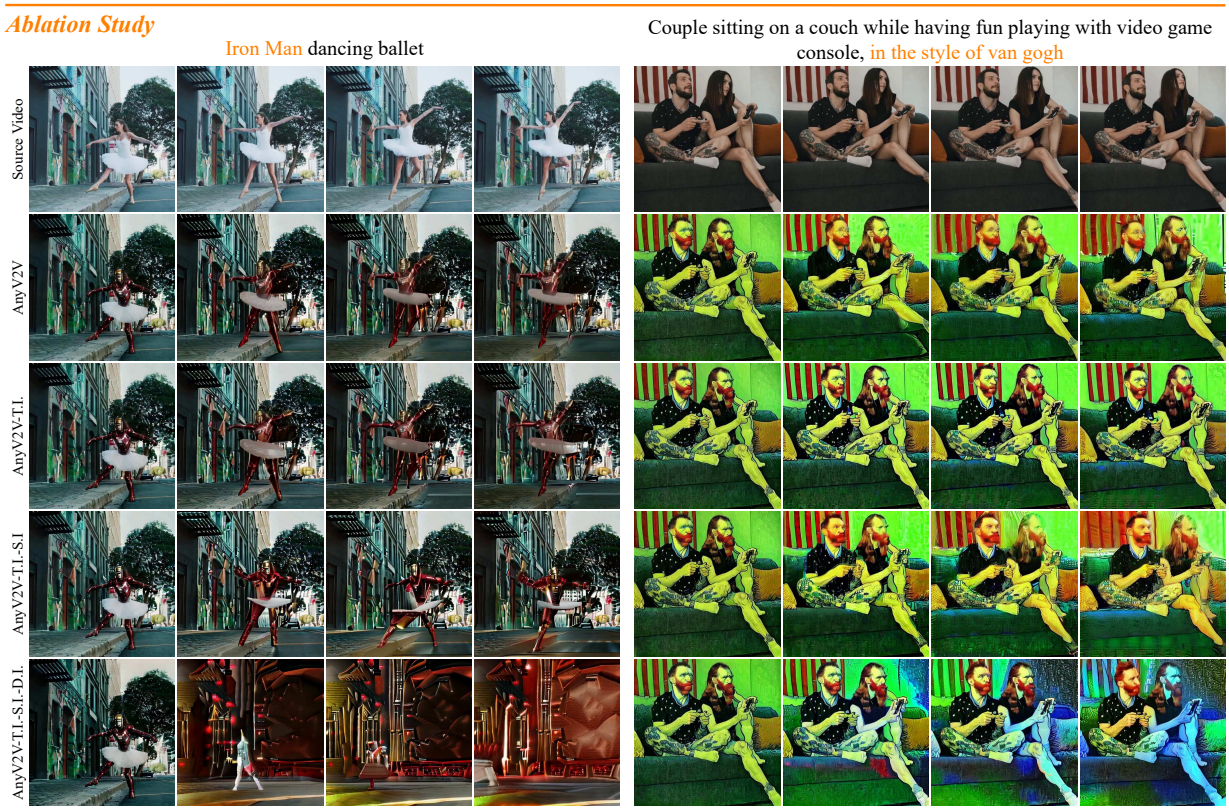


Figure 5.6: Visual comparisons of AnyV2V's editing results after disabling temporal feature injection (T.I.), spatial feature injection (S.I.) and DDIM inverted initial noise (D.I.).

consecutive frames and contain more appearance and motion inconsistencies. Further illustrated in the third row of Figure 5.6, removing spatial feature injection will often result in incorrect subject appearance and pose (as shown in the "ballet dancing" case) and degenerated background appearance (evident in the "couple sitting" case). These observations demonstrate that directly generating edited videos from the DDIM inverted noise is often not enough to fully preserve the source video structures, and the spatial feature injection mechanisms are crucial for achieving better editing results.

DDIM Inverted Noise as Structural Guidance Finally, we observe a further decrease in CLIP-Image scores and a significantly degraded visual appearance in both examples in Figure 5.6 after replacing the initial DDIM inverted noise with random noise during sampling. This indicates that the I2V generation models become less capable of animating

Table 5.3: Ablation study results for AnyV2V (I2VGen-XL). T. Injection and S. Injection correspond to temporal and spatial feature injection mechanisms, respectively.

Model	CLIP-Image ↑
AnyV2V (I2VGen-XL)	0.9648
AnyV2V (I2VGen-XL) w/o T. Injection	0.9652
AnyV2V (I2VGen-XL) w/o T. Injection & S. Injection	0.9637
AnyV2V (I2VGen-XL) w/o T. Injection & S. Injection & DDIM Inversion	0.9607

the input image when the editing prompt is completely out-of-domain and highlights the importance of the DDIM inverted noise as the structural guidance of the edited videos.

5.5 Conclusion

In this chapter, we presented AnyV2V, a novel unified framework for video editing. Our framework is training-free, highly cost-effective, and can be applied to any image editing model and I2V generation model. To perform video editing with high precision, we propose a two-stage approach to first edit the initial frame of the source video and then condition an I2V model with the edited first frame and the source video features and inverted latents to produce the edited video at any length. Comprehensive experiments have shown that our method achieves outstanding outcomes across a broad spectrum of applications that are beyond the scope of existing SOTA methods while achieving superior results on both common video metrics and human evaluation. For future work, we aim to find a tuning-free method to bridge the I2V properties into T2V models so that we can leverage existing strong T2V models for video editing.

Part II

Scalable Visual World Understanding over Long Horizons

Chapter 6

Background and Roadmap

The second part of this thesis studies the *horizon* axis: extending the spatial and temporal extent of the visual input that a model can perceive and reason over. Real-world visual content, such as movies, instructional videos, and long recordings, often spans tens of minutes or hours and unfolds through complex, multi-stage narratives. Scaling large multimodal models (LMMs) to such inputs is difficult because a single minute of video can produce thousands of visual tokens, and the quadratic cost of causal self-attention makes processing dense tokens across thousands of frames computationally infeasible on standard hardware. The common workaround of compressing a video into a handful of tokens reduces computation but discards the fine-grained detail required for long-horizon reasoning. Scaling along this axis also raises two further challenges: high-quality instruction data for long, high-resolution videos is scarce, and existing benchmarks often overstate progress. An efficient architecture must therefore be paired with suitable training data and reliable evaluation.

This part comprises three chapters that address these challenges in turn. In Chapter 7, we present VAMBA, a hybrid Mamba-Transformer model for efficient hour-long video understanding that replaces the costly causal self-attention over video tokens with cross-attention for the text tokens and linear-time Mamba-2 blocks for the video tokens, encoding far more frames on a single GPU without discarding any tokens. In Chapter 8, we present VISTA, a video spatiotemporal augmentation framework that synthesizes long-duration, high-resolution video instruction-following data from existing video-caption datasets, directly addressing the scarcity of long-video supervision. In Chapter 9, we present VideoEval-Pro, a robust benchmark for long video understanding that replaces multiple-choice questions, whose scores can be inflated by guessing and prior knowledge, with open-ended questions that require genuine comprehension of the full video. Together,

these chapters extend the horizon that an understanding model can process, and provide the data and evaluation needed to measure progress reliably.

The remainder of this chapter reviews the related work that situates these contributions, covering large multimodal models for video understanding, efficient architectures for long video, video instruction-following data, and video understanding benchmarks.

6.1 Large Multimodal Models for Video Understanding

Large multimodal models (LMMs) generally follow the LLaVA paradigm [166], in which a pretrained vision encoder maps an image or video into a sequence of visual tokens that are concatenated with text tokens and jointly processed by a language model decoder. Early video LMMs such as Video-ChatGPT [190] and Video-LLaVA [160] were restricted to a small number of low-resolution frames. Subsequent research improved these models along several directions: through better instruction-following data [165, 147, 40, 339], through stronger vision encoders capable of processing higher resolutions, and by extending LMMs to interleaved image [135, 141, 120] and video inputs. To optimize visual feature extraction, Idefics2 [135] combines the NaViT [58] strategy with a SigLIP [327] encoder to support native-resolution inputs, and Qwen2-VL [265] introduces a multimodal rotary position embedding to handle varying input sizes. Despite these advances, state-of-the-art LMMs remain inefficient at long multimodal inputs because of the quadratic complexity of causal self-attention, which motivates the efficient architecture we develop in Chapter 7.

6.2 Efficient Models for Long Video Understanding

One prominent strategy for handling long videos is to reduce the length of the visual token sequence. Video-LLaMA [330] and the VideoChat series [145, 147] use a Q-Former [144] to compress dense visual tokens into a compact set of learnable queries, and Video-CCAM [70] adds causal cross-attention masks to better capture temporal order. VideoLLaMA 2 [53] designs a 3D convolution-based connector for token compression, LLaVA-Mini [336] reduces each image to a single token, Video-XL [229] summarizes visual information into special tokens, and LongVU [228] applies spatiotemporal adaptive compression. An orthogonal approach increases the context length of the language backbone itself, as in LongVA

[334]. These methods reduce computation but risk discarding information essential for fine-grained, long-horizon reasoning.

A second strategy modifies the model architecture to avoid the quadratic cost directly. Flamingo [4], OpenFlamingo [9], Idefics [134], and LLaMA 3.2 [65] interleave gated cross-attention layers with self-attention layers for vision-language modelling, and mPlug-Owl3 [315] integrates self-attention with cross-attention in a hyper-attention layer, although such cross-attention-based models generally underperform LLaVA-style transformers. Linear-time sequence models offer another route: the Mamba family [89, 57] and Mamba-Transformer hybrids such as Jamba [158] and Zamba [85] have succeeded in language modelling, and LongLLaVA [273] took an initial step by adopting a hybrid decoder for multimodal understanding, though its long-video performance remained limited. Concurrent hybrid video models include STORM [121] and BIMBA [116]. Our VAMBA (Chapter 7) differs from these efforts by employing Mamba-2 blocks directly in the decoder to update the video tokens with linear complexity, achieving efficient long-video processing without any token reduction.

6.3 Video Instruction-Following Data

Early video-text datasets typically targeted a specialized domain, such as video retrieval [305], video question answering [297, 320, 304], or action recognition [236, 28, 87]. To build instruction-following data, VideoChat2 [147] combines multiple specialized datasets and rewrites their instructions with a language model, though the underlying videos are often of limited quality. More recent efforts such as LLaVA-Hound [335] and ShareGPT4Video [40] generate higher-quality captions and open-ended questions using GPT-4V [3], but their source videos tend to have limited motion and variety. LLaVA-Video-178K [339] provides high-quality instruction data at scale, yet relies on a costly annotation process that feeds densely sampled frames to a proprietary model. In contrast, our VISTA (Chapter 8) synthesizes long-duration, high-resolution video instruction data from existing short or low-resolution videos, offering a more scalable and cost-effective source of supervision.

6.4 Video Understanding Benchmarks

The progress of video LMMs has driven the creation of many evaluation benchmarks. MVBench [147] assembles a comprehensive benchmark by regenerating question-answer

pairs from existing datasets, while TempCompass [174] and VideoVista [154] target temporal reasoning. To assess understanding of extremely long videos, benchmarks such as Video-MME [74], MLVU [344], LVBench [268], and LongVideoBench [289] have emerged. Two limitations of these benchmarks motivate the contributions in this part. First, none specifically evaluates the understanding of high-resolution videos, which is important for perceiving small objects and localized actions; VISTA (Chapter 8) introduces a high-resolution benchmark to fill this gap. Second, these benchmarks predominantly rely on multiple-choice questions, whose scores can be inflated by answer hints and prior knowledge, and this effect has not been systematically studied. VideoEval-Pro (Chapter 9) addresses this concern with concise, free-form answers that better reflect a model's true long-video understanding.

Chapter 7

Vamba: Understanding Hour-Long Videos with Hybrid Mamba-Transformers

7.1 Introduction

The field of large multimodal models (LMMs) has seen tremendous progress in recent years. The seminal work of LLaVA [167] successfully transferred the power of autoregressive next-token prediction from large language models (LLMs) [197, 3, 251, 13, 119] to the multimodal domain, establishing a new standard for visual understanding by autoregressively modelling visual and language tokens through causal transformer layers. Since then, transformer-based LMMs have been widely studied and enhanced to tackle a diverse range of tasks, such as high-resolution image understanding [165, 135, 265] and interleaved image-text reasoning [120, 141, 137]. Nevertheless, the quadratic complexity inherent in causal self-attention operations presents significant challenges for long-context inputs, making it difficult for transformer-based LMMs to effectively handle the task of understanding extremely long videos.

A key challenge for advanced LMMs when processing long video inputs is that they tend to encode each frame into a large number of vision tokens, which leads to substantial computational and memory costs during both training and inference. For example, Qwen2-VL [265] can only process 256 frames (360p) on a single GPU, which is far from sufficient for hour-long video understanding. To reduce the computational/memory cost, previous efforts have primarily focused on reducing the vision tokens in the input sequence.

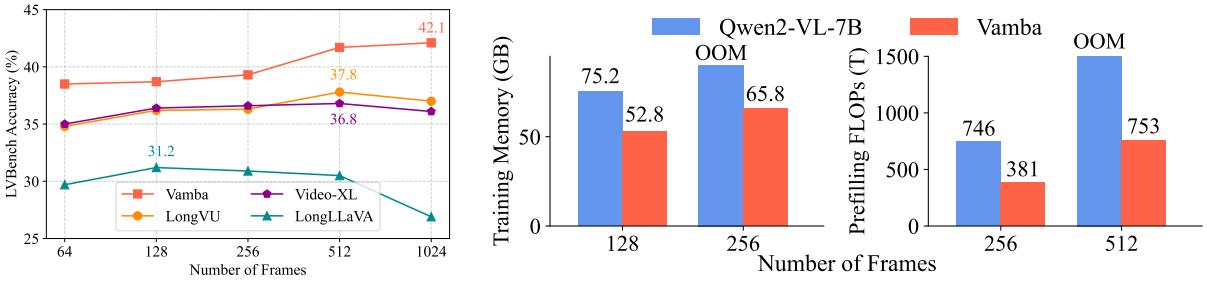


Figure 7.1: VAMBA achieves strong long video understanding performance (42.1% on LVBench [268]) while being more computationally efficient compared to transformer-based LMMs.

Several approaches [331, 145, 147, 70] utilize a Q-Former [144] to compress vision tokens. More recent methods, such as LongVU [228] and Video-XL [229], partition vision token sequences into chunks and employ adaptive token compression mechanisms to decrease the token count. Another line of work, such as InternVideo2.5 [278] and Orxy-1.5 [181], employs various mechanisms to evaluate the vision tokens' importance, and thus dropping or merging those less significant tokens [21] during training and inference. Despite these advancements, key challenges in LMMs for long video understanding still persist. First, aggressive token reduction can lead to critical information loss, particularly when processing extremely long videos. Second, these methods still suffer from quadratic computational complexity as the number of input frames increases. Third, token reduction-based methods require additional overhead, which might increase the actual wall-clock time.

In this chapter, we investigate an orthogonal direction to previous approaches: instead of compressing the video tokens, we seek to develop an alternative model architecture that improves the efficiency of processing video tokens during training and pre-filling stage of inference. We propose VAMBA, a hybrid Mamba-Transformer model for efficient hour-long video understanding. The key insight of our method is that we can design efficient modules to approximate the causal self-attention operation for both text and video tokens in transformer-based LMMs. In particular, we propose to (1) utilize cross-attentions to update text tokens based on video tokens, which is affordable due to the short length of text tokens, and (2) adopt Mamba-2 [57] to process the massive video tokens with linear complexity.

Assuming a combined input sequence of $M+N$ tokens, where M is the number of video tokens and N is the number of text tokens, we find that M could be at least 100 times larger than N on many long video tasks ($M \gg N$). Our model can reduce the training/pre-filling computational complexity from $O(d(M+N)^2)$ to $O(dMN + d^2M)$, where d is the

Table 7.1: Theoretical time and memory complexity for transformer-based LMMs and VAMBA in pre-filling. M and N represent the number of vision and text tokens. d denotes the hidden dimension.

Model	Time Complexity	Memory Complexity
Transformer	$O(d(M+N)^2)$	$O((M+N)^2)$
VAMBA	$O(dMN+d^2M)$	$O(MN)$

hidden dimension, according to Table 7.1. In practice, this theoretical improvement may not be fully realized due to the hardware under-optimization for Mamba [89]. Nevertheless, we still observe a more than 50% reduction in GPU memory usage and FLOPs/runtime during training and inference for long video inputs, as shown in Figure 7.1. VAMBA can be efficiently trained using 8×A800 GPUs, whereas other efficient video LMMs such as LongVU [228] and LongLLaVA [273] require 64 and 24 GPUs for training, respectively. By performing a two-stage training, our VAMBA achieves a 4.3% improvement over the best efficient video LMMs on the challenging hour-scale video understanding benchmark LVBench [268]. On other long video understanding datasets like Video-MME [74], MLVU [344] and LongVideoBench [290], VAMBA also achieves top-notch performance.

Our contributions can be summarized as follows:

1. We propose VAMBA, a hybrid Mamba-Transformer model for hour-long video understanding. VAMBA's design features efficient modules such as Mamba-2 blocks and cross-attention layers, effectively reducing the computational overhead of transformer-based LMMs.

2. We conduct a comprehensive ablation study and show that employing Mamba-2 blocks and initializing cross-attention weights from pretrained self-attention layers are crucial for achieving high performance in VAMBA.

3. Our extensive evaluations demonstrate that VAMBA achieves strong video understanding capabilities. Specifically, VAMBA achieves 4.3% improvement over state-of-the-art efficient video LMMs on the challenging hour-long benchmark LVBench.

7.2 Preliminaries

7.2.1 State Space Models and Mamba

State space models (SSMs) are linear time-invariant systems for modelling continuous signals. A continuous SSM can be expressed by ordinary differential equations (ODEs):

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \quad y(t) = \mathbf{C}h(t),$$

here, $x(t) \in \mathbb{R}$ is the input signal, $h(t) \in \mathbb{R}^N$ is the N-dimensional hidden state and $y(t) \in \mathbb{R}$ is the output signal. $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the state transition matrix and $\mathbf{B} \in \mathbb{R}^{N\times 1}, \mathbf{C} \in \mathbb{R}^{1\times N}$ are projection matrices.

Mamba [89], and its predecessor S4 [91], are discretized SSMs for discrete sequence modelling. The discretization of the SSM is done by the zero-order hold (ZOH) technique:

$$\overline{\mathbf{A}} = \exp(\Delta\mathbf{A}), \quad \overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B},$$

where $\overline{\mathbf{A}}, \overline{\mathbf{B}}$ are discretized state variables and Δ is the step size. The discretized SSM can thus be rewritten as:

$$h_t = \overline{\mathbf{A}}h_{t-1} + \overline{\mathbf{B}}x_t, \quad y_t = \mathbf{C}h_t.$$

Mamba proposes the selective scan algorithm that dynamically determines $\mathbf{B}, \mathbf{C}, \Delta$ based on the sequence inputs. Following S4, the $\mathbf{A}$ matrix is initialized as the diagonalized HiPPO matrix [90]. These design choices are proven to be efficient and effective for modelling extremely long sequences. Mamba-2 [57] further simplifies the formulation of $\mathbf{A}$ to a scalar times identity structure. Mamba-2 supports multi-head SSM and allows a much larger state dimension N than Mamba, while being faster during training.

7.2.2 Transformer-based LMMs

We study how text and video tokens are being updated in transformers. As shown in Figure 7.2, we assume a standard case where video tokens are placed before text tokens and ignore the chat template tokens for simplicity. In each LMM decoder layer, the video and text tokens are updated as:

- All video tokens get updated by an input layer normalization (LN) layer, a self-attention layer, a post LN layer and a final MLP layer. There are also residual connections for self-attention and MLP layers. The i^{th} video token v_i computes self-attention based on the first i video tokens:

$$o_{v_i} = (\sigma(\frac{q_{v_i}\mathbf{K}_{[v_1:v_i]}^{\top}}{\sqrt{d}})\mathbf{V}_{[v_1:v_i]})\mathbf{W}_o, \tag{7.1}$$

  where $\sigma(\cdot)$ denotes the softmax operation, q_{v_i} is the query vector for v_i and $\mathbf{K}_{[v_1:v_i]}$, $\mathbf{V}_{[v_1:v_i]}$ are key and value matrices of the first i video tokens. d denotes the dimension of the query and key vectors and $\mathbf{W}_o$ is the out projection matrix. o_{v_i} represents the final output for token v_i.

- The text tokens share a similar update route to the video tokens. The self-attention updates are slightly different: for the j^{th} text token t_j, it computes self-attention based on all video tokens and the first j text tokens:

$$o_{t_j} = (\sigma(\frac{q_{t_j}[\mathbf{K}_v, \mathbf{K}_{[t_1:t_j]}]^{\top}}{\sqrt{d}})[\mathbf{V}_v, \mathbf{V}_{[t_1:t_j]}])\mathbf{W}_o, \tag{7.2}$$

  where $[\mathbf{K}_v, \mathbf{K}_{[t_1:t_j]}]$ and $[\mathbf{V}_v, \mathbf{V}_{[t_1:t_j]}]$ are the key and value matrices of the combination of all video tokens and text tokens up to t_j.

The main computation overhead in the transformer-based LMMs comes from the **quadratic complexity of the self-attention in the video tokens**. To overcome this issue, we design a hybrid Mamba Transformer architecture to process text and video tokens differently. Our detailed model architecture designs are listed in the sections below.

7.3 Our Method: Vamba

Our goal is to devise a model Θ' that preserves the performance of transformer Θ, while being more efficient. We approach this problem by approximating the causal transformer layers in the pretrained video LMMs.

7.3.1 Updating Text Tokens via Cross-Attentions

The key idea of our method is to split the expensive self-attention operation over the entire video and text token sequence into two more efficient components. Since video

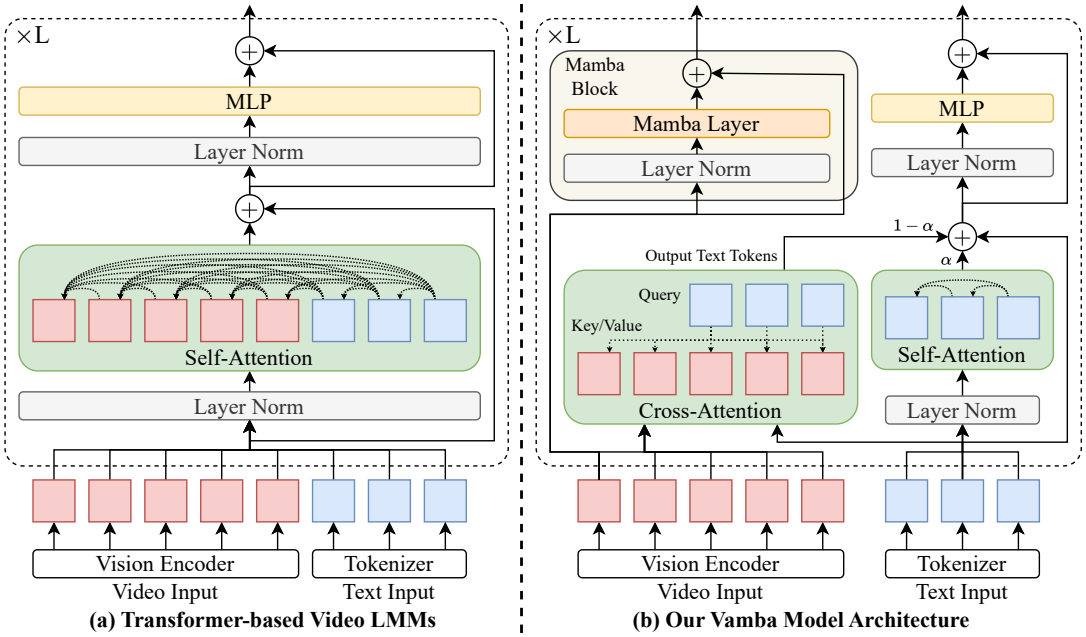


Figure 7.2: Overview of our VAMBA model architecture. Compared to transformer-based LMMs (left), we replace the costly causal self-attention operations with the more efficient cross-attention layers and Mamba blocks to achieve better efficiency.

tokens typically dominate the sequence while text tokens remain few, we maintain the self-attention mechanism exclusively for the text tokens and eliminate it for the video tokens. Instead, we add cross-attention layers that use text tokens as queries and video tokens as keys and values. As shown in Figure 7.2, this design updates the text tokens during the attention layers in our model as follows:

$$\begin{aligned} o_{t_j} =& (1-\alpha)(\sigma(\frac{q_{t_j}\mathbf{K}_v^\top}{\sqrt{d}})\mathbf{V}_v)\mathbf{W}_o^c \qquad \text{(Cross-Attention)} \\ &+ \alpha(\sigma(\frac{q_{t_j}\mathbf{K}_{[t_1:t_j]}^\top}{\sqrt{d}})\mathbf{V}_{[t_1:t_j]})\mathbf{W}_o^s, \qquad \text{(Self-Attention)} \end{aligned} \tag{7.3}$$

where $\mathbf{W}_o^c$ and $\mathbf{W}_o^s$ are the output projection matrices for the cross- and self-attention layers. $\alpha \in [0, 1]$ is a learnable weighting parameter that balances the cross- and self-attention outputs. Assuming a total of M video tokens and N text tokens, the self-attention operations in the transformer-based video LMMs have a (pre-filling) computational complexity of $O(d(M+N)^2)$. Our self+cross-attention design has the complexity of $O(dN^2)$ (self-attention) and $O(dMN)$ (cross-attention), effectively reducing the pre-filling complexity

to $O(dN^2 + dMN) \sim O(dMN)$, given that $M \gg N \approx d$.

Our attention formulation can be viewed as an approximation of the causal self-attention operation (Equation 7.2), as each text token q_{t_j} can still attend to all video tokens and the first j text tokens. As the video and text tokens share the same channel dimension, the projection matrices in the cross-attention layer will have identical dimensions as in the self-attention layers. This allows us to consider two model design choices: we can either randomly initialize the weights in the cross-attention layers, or we can initialize the cross-attention layers using the self-attention layer weights (i.e. $\mathbf{W}_o^c = \mathbf{W}_o^s$, same applies to the query, key and value projection matrices $\mathbf{W}_q^c, \mathbf{W}_k^c, \mathbf{W}_v^c$). *We add these two design choices to the* VAMBA *design space.*

7.3.2 Updating Video Tokens with Mamba Blocks

While the self+cross-attention design significantly reduces the model's complexity, relying solely on cross-attention layers can compromise the model's representational power. Specifically, rather than updating video tokens through self-attention and MLP layers (i.e., through the causal transformer block, *c.f.* Section 7.2.2), the video tokens remain unchanged after the cross-attention layers. To address this limitation, we seek an efficient architecture to approximate the effects of the transformer blocks. Motivated by the success of the Mamba [89, 57] architectures in image and video modelling [347, 175, 146], we propose to employ Mamba blocks to effectively process the video tokens. As shown in Figure 7.2, the video token updates can be formulated as:

$$o_{v_i} = \text{Mamba}(\text{LN}(v_i), \mathbf{h}_{v_{i-1}}, \overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{C}), \tag{7.4}$$

where $\text{LN}(\cdot)$ is the layer normalization operator and $\mathbf{h}_{v_{i-1}}$ is the context (hidden states) at token v_{i-1}. As a recurrent model, Mamba operates similarly to causal self-attention layers, enabling each video token to update itself based on all preceding tokens. Crucially, Mamba reduces the complexity of updating video tokens from $O(dM^2)$ in causal self-attentions to $O(d^2M)$. Combining the Mamba blocks and the self+cross-attention design, our model achieves an overall complexity of $O(dN^2 + dMN + d^2M) \sim O(dMN + d^2M)$, which is substantially lower than the quadratic complexity $O(d(M+N)^2)$ in the transformer-based video LMMs. In Section 7.3.3, we further detail how the Mamba layers are trained to approximate Equation 7.1.

We consider employing Mamba or Mamba-2 blocks in Equation 7.4 in the VAMBA *design space.*

7.3.3 Training Paradigm

We utilize a two-stage training strategy to optimize VAMBA, including a pretraining stage and an instruction-tuning stage. In the pretraining stage, the model is initialized with pretrained transformer-based LMM weights in all modules except for the newly introduced cross-attention and Mamba layers. We freeze the pretrained components and train only the new layers using image captioning data to restore the model's visual understanding capabilities. We explore two types of training objectives during pretraining: (1) the standard language modelling loss $\mathcal{L}_{\text{LM}} = -\frac{1}{T}\sum_{t=1}^{T}\log p(x_t|x_{<t})$, which is the cross-entropy loss for next-token (x_t) prediction; (2) the distillation loss $\mathcal{L}_{\text{Distill}}$, where we extract the top 100 logits $\mathcal{P}_\Theta$ with the highest values from the transformer-based model Θ and compute a KL-divergence loss with the logit outputs $\mathcal{P}_{\Theta'}$ of our model Θ' at the same indices, such that $\mathcal{L}_{\text{Distill}} = D_{KL}(\mathcal{P}_\Theta||\mathcal{P}_{\Theta'})$ ($D_{KL}(\cdot||\cdot)$ denotes KL-divergence). The final loss is formulated as $\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda\mathcal{L}_{\text{Distill}}$, where we use λ to balance the weights of the two losses. *We include* $\lambda = 0, 0.001, 0.1, 0.5, 1, 2$ *to the* VAMBA *design space.*

In the instruction-tuning stage, we leverage both image and video instruction-following data to fully finetune VAMBA, thereby enhancing its instruction-following capability. When GPU memory limitations prevent full finetuning, we freeze the vision encoder and finetune only the LMM decoder. We only employ the language modelling loss during the instruct-tuning stage to ensure that the teacher model does not restrict the student's performance.

The full design space for VAMBA *includes whether to initialize cross-attention weights from self-attention; whether to use Mamba or Mamba-2 blocks to update video tokens, and the choice of* λ *in the pretraining stage.*

7.4 Experiments

In this section, we first ablate the VAMBA design choices and then adopt the best setup for full-scale training.

7.4.1 Ablation Study

We explore the VAMBA design space to determine the effectiveness of our model components. We initialize all our models based on a strong pretrained transformer-based LMM Qwen2-VL-7B [265]. For all ablation experiments, we randomly select one million images from CC12M [34] and employ the PixelProse [232] captions for pretraining. We

Table 7.2: Ablation study results for the VAMBA design space explorations. "CA from SA?" denotes whether to initialize cross-attention weights from self-attention layers. "VidMME" represents the Video-MME benchmark.

Model ID	CA from SA?	Mamba Block Type	Stage 1	Stage 2		
			G-VEval	LVBench	VidMME	MVBench
			ref. free	test	w/o sub.	test
A	✗	N/A	75.7	23.7	47.6	40.9
B	✓	N/A	81.0	34.2	51.7	51.8
C	✓	Mamba	81.8	34.2	53.4	53.5
D	✓	Mamba-2	82.2	35.3	54.1	53.5

Table 7.3: Ablation study results on pretraining VAMBA models with a transformer-based LMM teacher and a distillation loss.

Model D + Distill	λ value in Stage 1 Training					
	0	0.001	0.01	0.5	1	2
G-VEval	82.19	81.05	80.68	73.69	63.65	47.61

use the LLaVA-Video-178K [339] dataset with a total of 1.3M video-instruction pairs for instruction-tuning.

Evaluation Metrics For pretraining (stage 1) evaluation, we sample 500 images from the COCO-2017 [48] dataset and have the models generate captions for each image. We then apply the reference-free version of G-VEval [250], a metric based on GPT-4o [198] to evaluate the quality of the captions. For instruction-tuning (stage 2), we evaluate model performances across three benchmarks: LVBench [268] for hour-long videos, Video-MME [74] for medium-to-long videos, and MVBench [147] for short videos. Ablation study results are shown in Table 7.2 and Table 7.3.

Cross-Attention Layer Initialization Strategy According to Table 7.2, we find that it is crucial to initialize the cross-attention layer weights using the corresponding self-attention layer from the same decoder level. Compared to Model A, the dramatic performance boost observed in Model B across all metrics underscores the critical impact of this weight initialization strategy. We believe this observation stems from the fact that initializing the cross-attention layer weights from the self-attention layers allows our self+cross-attention operation (Equation 7.3) to more closely approximate the pretrained

Table 7.4: Comparisons between baseline models and VAMBA on hour-long video understanding benchmarks. **Bold**: best results among efficient video LMMs. Underline: second-best.

Models	Size	Hour-Long Video Understanding		
		LVBench	HourVideo	HourEval
		test	dev	test
Transformer-based LMMs				
Gemini-1.5-Pro [245]	-	33.1	37.3	-
GPT-4o [198]	-	34.7	19.6	-
Qwen2-VL [265]	7B	42.0	33.8	53.0
Efficient LMMs				
LLaVA-Mini [336]	7B	17.6	20.6	24.2
LongLLaVA [273]	9B	31.2	27.7	39.1
LongVU [228]	7B	37.8	30.8	46.8
Video-XL [229]	7B	36.8	33.0	47.1
VAMBA	10B	**42.1**	**33.6**	**50.7**

causal self-attention operation (Equation 7.2): once $\mathbf{W}_q^c, \mathbf{W}_k^c, \mathbf{W}_v^c$ and $\mathbf{W}_o^c$ are set equal to $\mathbf{W}_q^s, \mathbf{W}_k^s, \mathbf{W}_v^s$ and $\mathbf{W}_o^s$, the only discrepancy between the two equations becomes their attention score matrices. This difference can be mitigated by our two-stage training process, making it easier for Model B to recover its multimodal understanding capabilities.

Mamba Block Design Choices Comparing Model B with Model C and D in Table 7.2, we observe that both Mamba and Mamba-2 blocks improve the model performances across all metrics, indicating that Mamba blocks bring extra representation power to the model's visual understanding capabilities. Furthermore, when comparing Model C and Model D, we find that Mamba-2 demonstrates superior image and video modelling capabilities. Despite having a simplified $\mathbf{A}$ matrix structure, Mamba-2 accommodates a larger state dimension than Mamba (64 versus 16 in our case), which likely contributes to its improved performance.

Pretraining Distillation Loss We employ the pretrained Qwen2-VL-7B model as the teacher model and perform a series of pretraining experiments with the additional distillation loss based on our best model setting (Model D in Table 7.2). The results are shown in Table 7.3. In contrast to previous findings from cross-attention-based LLMs (e.g. CEPE [317]), where distillation loss is reported to be beneficial, we observe that incorporating

Table 7.5: Comparisons between baseline models and VAMBA on medium-length or short video understanding benchmarks. **Bold**: best results among each section. Underline: second-best. * indicates the results based on our evaluation scripts.

Models	Size	Medium-Length Understanding			Short Video Understanding		Video Captioning
		Video-MME	MLVU	LongVideoBench	MVBench	NExT-QA	DREAM-1K
		w/o subtitles	m-avg	val	test	mc	F1
Proprietary Models							
GPT-4V [3]	-	59.9	49.2	59.1	43.5	70.4	34.4
GPT-4o [198]	-	71.9	64.6	66.7	-	76.0	39.2
Gemini-1.5-Pro [245]	-	75.0	62.9	64.0	54.2	76.4	36.2
Open-source Transformer-based LMMs							
VideoChat2 [147]	7B	39.5	47.9	39.3	51.9	78.6	26.6
ShareGPT4Video [41]	7B	39.9	46.4	39.7	51.2	-	19.5
LongVA [334]	7B	52.4	56.3	51.8	49.2	68.3	-
Video-CCAM [70]	9B	50.3	58.5	-	64.6	-	-
Kangaroo [168]	8B	56.0	61.0	54.8	61.1	-	-
InternVL2 [52]	8B	54.0	48.1*	51.8*	66.4	-	26.9
LLaVA-OneVision [137]	7B	58.2	62.6*	**56.4**	56.7	**79.4**	**31.7**
Qwen2-VL [265]	7B	**63.3**	**64.2***	52.4*	**67.0**	-	29.6
Phi-4-Mini [2]	5.6B	55.0	60.1	46.7	60.4	-	-
Open-source Efficient LMMs							
LLaVA-Mini [336]	7B	40.3	44.3	19.3*	44.5	47.6*	22.9
LongLLaVA [273]	9B	51.6	53.3*	42.1*	54.6	72.2*	24.6
LongVU [228]	7B	55.3*	65.4	53.5*	**66.9**	78.0*	**28.1**
Video-XL [229]	7B	55.5	64.9	50.7	55.3	77.5*	23.5
VAMBA	10B	**57.8**	**65.9**	**55.9**	60.4	**78.1**	**28.1**

an additional distillation loss does not further improve our model's performance, as the G-VEval score decreases for all $\lambda > 0$. Therefore, we exclude the distillation loss and only employ the language modelling loss in both stages in our final training setting.

7.4.2 Main Evaluation Results

Implementation Details Based on our ablation study (Table 7.2), we adopt Model D as our final design and initialize VAMBA from Qwen2-VL-7B. We source ∼3M image caption data for pretraining and ∼7M image and video QA data for instruction-tuning.

Hour-Long Video Understanding We evaluate our model's ability to handle extremely long videos using two public benchmarks: LVBench [268] and HourVideo [31] (development set). We further compose a new benchmark called HourEval by selecting all questions related to videos longer than 30 minutes from Video-MME [74], MLVU [344]

development set, and LongVideoBench [290] validation set. The average video lengths for LVBench, HourVideo, and HourEval are 68.4, 47.2, and 54.7 minutes, respectively. We compare our model against four efficient video LMMs: LLaVA-Mini [336], LongLLaVA [273], Video-XL [229] and LongVU [228]. We also include the results from Qwen2-VL-7B (our baseline transformer-based LMM) as a reference.

Experimental results are shown in Table 7.4. Our VAMBA consistently outperforms all efficient video LMMs across the three hour-long video benchmarks, highlighting its exceptional ability to understand and reason over hour-scale videos. Notably, our model surpasses the baseline Qwen2-VL-7B on the LVBench benchmark, and its performance on HourVideo is also very close to Qwen2-VL-7B. These results underscore that our VAMBA is competitive with the best open-sourced transformer-based LMMs, while being significantly more efficient during training and inference.

Medium-Length or Short Video Understanding To further demonstrate our model's generalizability across various video durations, we test VAMBA on several medium-length or short video understanding benchmarks. Specifically, we report multiple-choice question performance on Video-MME, MLVU, LongVideoBench, NExT-QA [297] and MVBench [147]. We also evaluate VAMBA on DREAM-1K [264] for video captioning assessment.

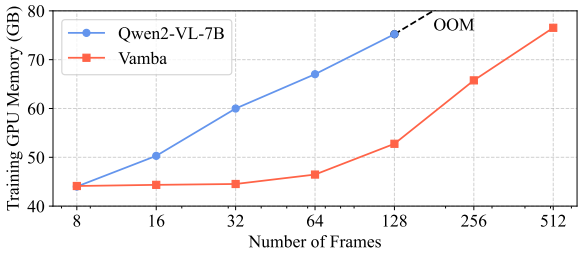
According to Table 7.5, our VAMBA demonstrates superior performance across three medium-length video understanding benchmarks (with average video durations between 10–20 minutes), ranking first among efficient video LMMs on all metrics. Despite the newly integrated cross-attention and Mamba layers in VAMBA have only been trained on relatively smaller datasets, VAMBA's performance remains competitive with large-scale pretrained transformer-based LMMs. Furthermore, all our evaluations can be conducted on a single 80G GPU, while some transformer-based LMMs, such as Qwen2-VL-7B, require additional inference strategies like sliding window attention or ring attention over multiple GPUs to achieve optimal results.

For short video understanding and video captioning benchmarks, VAMBA also achieves competitive performances, ranking first on NExT-QA and DREAM-1K and second on MVBench among efficient LMMs. Overall, our model delivers the best results on medium-length and long video benchmarks, demonstrating its strong ability to handle long-context video-language inputs.

7.4.3 Runtime Efficiency Analysis

To understand our model's runtime efficiency gains over the baseline transformer-based LMM (Qwen2-VL-7B), we conduct an efficiency analysis for both training and inference.

(a) Training GPU Memory Usage Comparison



(b) Training Runtime Comparison (time for 1 training step)

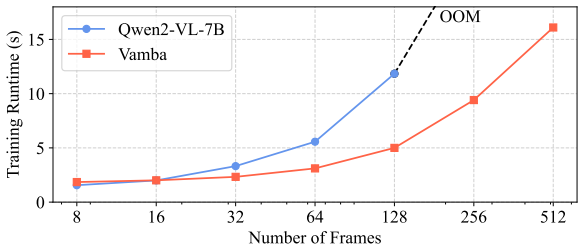
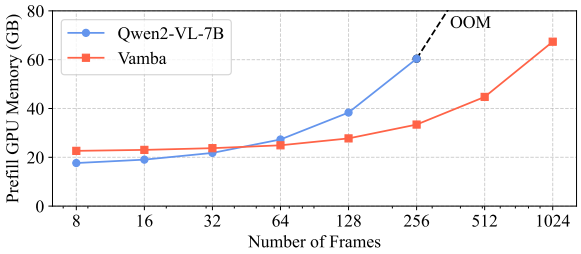


Figure 7.3: Comparison of training GPU memory usage and runtime per training step between Qwen2-VL-7B and VAMBA.

(a) Pre-filling GPU Memory Usage Comparison



(b) Pre-filling FLOPs Comparison (measured using `calflops`)

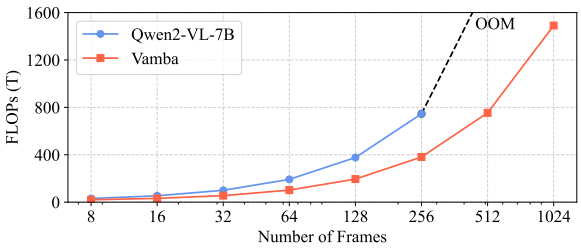


Figure 7.4: Comparison of GPU memory usage and FLOPs between Qwen2-VL-7B and VAMBA during inference.

For model training, we ensure a consistent training environment by using 8 NVIDIA A800 80G GPUs for both models. All input videos are resized to a resolution of 640×360, and a batch size of 1 is maintained on each GPU. We apply standard optimization methods such as Flash-Attention 2 [56], DeepSpeed ZeRO-3 [214] and gradient checkpointing [43] for both models. Based on different numbers of input frames, we measure the average GPU memory across 8 GPUs and the runtime per training step during training. As shown in Figure 7.3a, although VAMBA and Qwen2-VL-7B share the same vision encoder design and generate vision tokens of identical sequence lengths, our model requires over 50% less training memory when processing videos with more than 16 frames. This efficiency gain allows us to handle a larger number of video frames during training (512 vs. 128). Furthermore, as depicted in Figure 7.3b, our efficient design also accelerates model training, achieving nearly twice the speedup per training step when working with more than 64 frames.

For model inference, we focus on the pre-fill stage and analyze GPU memory usage and FLOPs for both models. All measurements are conducted on an input video with a resolution of 640×360. As shown in Figure 7.4a, VAMBA requires slightly more GPU



Figure 7.5: Qualitative comparison between VAMBA and efficient LMM baselines. Red text denotes incorrect responses, while green text highlights the correct responses by our VAMBA.

memory than Qwen2-VL-7B when the number of input frames is low (fewer than 32 frames), due to the additional ∼3B parameters in its cross-attention and Mamba layers. However, VAMBA's memory usage increases more slowly as the frame increases, allowing it to handle four times as many frames on a single NVIDIA A800 80G GPU compared to Qwen2-VL-7B (256 vs. 1024). Regarding computational cost, VAMBA reduces FLOPs by 30% to 50% during inference (Figure 7.4b), demonstrating a significantly lower complexity than its transformer-based LMM counterparts.

7.4.4 Case Study

In this section, we perform a case study for Video-XL and VAMBA using different video inputs and show the results in Figure 7.5. The "cookie" example focuses on object property identification in a long video (36 minutes). Video-XL incorrectly identifies the cookie colors as green and white, possibly due to the video also containing scenes for making other

desserts (e.g. white ice cream in the frame highlighted in red). On the other hand, VAMBA more accurately describes them as green and brown. In the "magic trick" example, Video-XL misinterprets the rubber band as a green ring, whereas VAMBA correctly describes the performer manipulating the rubber band around their fingers, making it appear to pass through. These comparisons highlight how VAMBA is better at retrieving relevant events in long videos and capturing fine-grained visual details and actions.

7.5 Conclusion

We presented VAMBA, a hybrid Mamba-Transformer model for efficient hour-long video understanding. By integrating cross-attention for text tokens and Mamba-2 blocks for video token updates, our approach reduces computational complexity and GPU memory usage while achieving competitive performance across long, medium, and short video benchmarks. Extensive evaluations on datasets such as LVBench demonstrate VAMBA's superiority over existing efficient video LMMs. For future work, since our model design is orthogonal to token reduction-based methods, we will focus on combining the strengths of both approaches to develop more efficient VAMBA variants.

Chapter 8

VISTA: Enhancing Long-Duration and High-Resolution Video Understanding by Video Spatiotemporal Augmentation

8.1 Introduction

Recent advancements in large language models (LLMs) and large multimodal models (LMMs) have brought transformative changes to video understanding tasks. Traditionally, video understanding relied on training task-specific models using domain-specific datasets (e.g. action recognition on Kinetics [28], video retrieval on MSR-VTT [305], and video captioning on YouCook2 [345]). In contrast, it is now possible to process video inputs and address diverse tasks using a single video LMM [160, 53, 190] through instruction following. However, most current (open-sourced) video LMMs are optimized for understanding and reasoning over short, low-resolution videos. **Processing long-sequence video input, such as long or high-resolution videos, remains a significant challenge for video LMMs.**

Most efforts to enhance long-sequence video understanding have centered around designing better model architectures. Conversely, the creation of higher-quality (long/high-resolution) video instruction-following datasets remains highly under-explored. The primary challenges stem from the scarcity of publicly available (license-friendly) high-quality

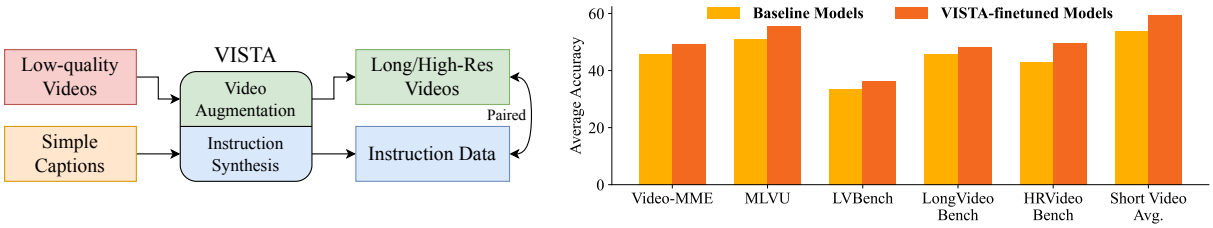


Figure 8.1: VISTA is a simple but effective framework that generates high-quality video instruction data from existing video-caption pairs. Our VISTA-400K dataset enhances model performances on various long and high-resolution video benchmarks.

video-instruction data. Consequently, open-source video instruction-following datasets often face limitations such as low resolution or short duration. For example, VideoChat2 [147] collects a video instruction-following dataset by combining videos from multiple domain-specific datasets and rewriting the instructions using ChatGPT [197]. However, the video sources used in this dataset primarily contain short videos. ShareGPT4Video [40] curates a video-instruction dataset containing 40K videos with detailed video captions. Nevertheless, the source videos are collected at a sparse sampling rate of 0.15 fps, and the video contents often lack motion and can appear nearly static. FineVideo [69] is a long video dataset featuring diverse video contents and high-quality metadata such as detailed captions and narrative progressions. While it offers satisfactory video durations, the dataset is limited by its resolution, as it predominantly comprises 360p videos.

Though some concurrent studies have improved LMMs' long-sequence video understanding capabilities, they often release only the model weights, hiding their video training data. For instance, Kangaroo [168]'s training data contains 700K long video data, but specifics about the curation of the videos and the construction of the instructions are not provided. Similarly, Qwen2-VL [265], MiniCPM-V-2.6 [313] and Aria [140] all claim that their instruction-following dataset contains long video training data, yet without providing the sources and statistics of the video data. This lack of transparency hinders a clear understanding of what types of video instruction data truly benefit long-sequence video understanding tasks and impedes further advancements in improving existing models.

In this chapter, we propose VISTA, a simple yet effective video augmentation pipeline to synthesize **long and high-resolution video instruction data from existing video datasets**. VISTA leverages insights from image and video classification data augmentation techniques such as CutMix [324], MixUp [332] and VideoMix [325], which demonstrate that training on synthetic data created by overlaying or mixing multiple images or videos results in more robust classifiers. Similarly, our method spatially and temporally combines videos

to create (artificial) augmented video samples with longer durations and higher resolutions, followed by synthesizing instruction data based on these new videos. Our data synthesis pipeline utilizes existing public video-caption datasets, making it fully open-sourced and scalable. This allows us to construct VISTA-400K, a high-quality video instruction-following dataset aimed at improving the long and high-resolution video understanding capabilities of video LMMs. By finetuning various video LMMs on our dataset, we observe an average of 3.3% improvement across multiple long video understanding benchmarks. Additionally, we compile a new benchmark HRVideoBench that focuses on high-resolution videos. Results on HRVideoBench (+6.5% after finetuning) demonstrate that our method produces models well-suited for high-resolution video understanding. Our contributions are summarized below:

1. We present VISTA-400K, a high-quality synthetic video instruction-following dataset. VISTA-400K includes challenging QA pairs designed to enhance video LMMs' ability to understand long and high-res video inputs.
2. We collect HRVideoBench, a comprehensive benchmark dedicated to evaluating video LMMs' capability in understanding fine object details and subtle, localized actions within high-resolution videos.
3. Our VISTA-finetuned models achieve an average gain of 3.3% on four challenging long-video benchmarks and 6.5% on HRVideoBench compared to the vanilla models. Our ablation study indicates that disabling our proposed video augmentations significantly reduces performance.

8.2 VISTA-400K Dataset

Our goal is to create high-quality video instruction-following data from existing video-caption datasets. Specifically, we aim to (1) generate extended or higher-resolution videos by spatially or temporally combining existing videos, and (2) produce high-quality question-answer pairs by leveraging existing video captions. Formally, given a set of candidate videos $\mathbf{V} = \{V_1, V_2, ..., V_N\}$ with captions $\mathbf{C} = \{C_1, C_2, ..., C_N\}$, we seek to create an augmented video V^* and synthesize a QA pair (q, a):

$$V^* = \Phi(\mathbf{V}), \qquad (q, a) = \Theta(\mathbf{C}). \tag{8.1}$$

For the video augmentation operator Φ, we draw inspiration from data augmentation techniques used in image and video classification, such as CutMix [324] and VideoMix [325],

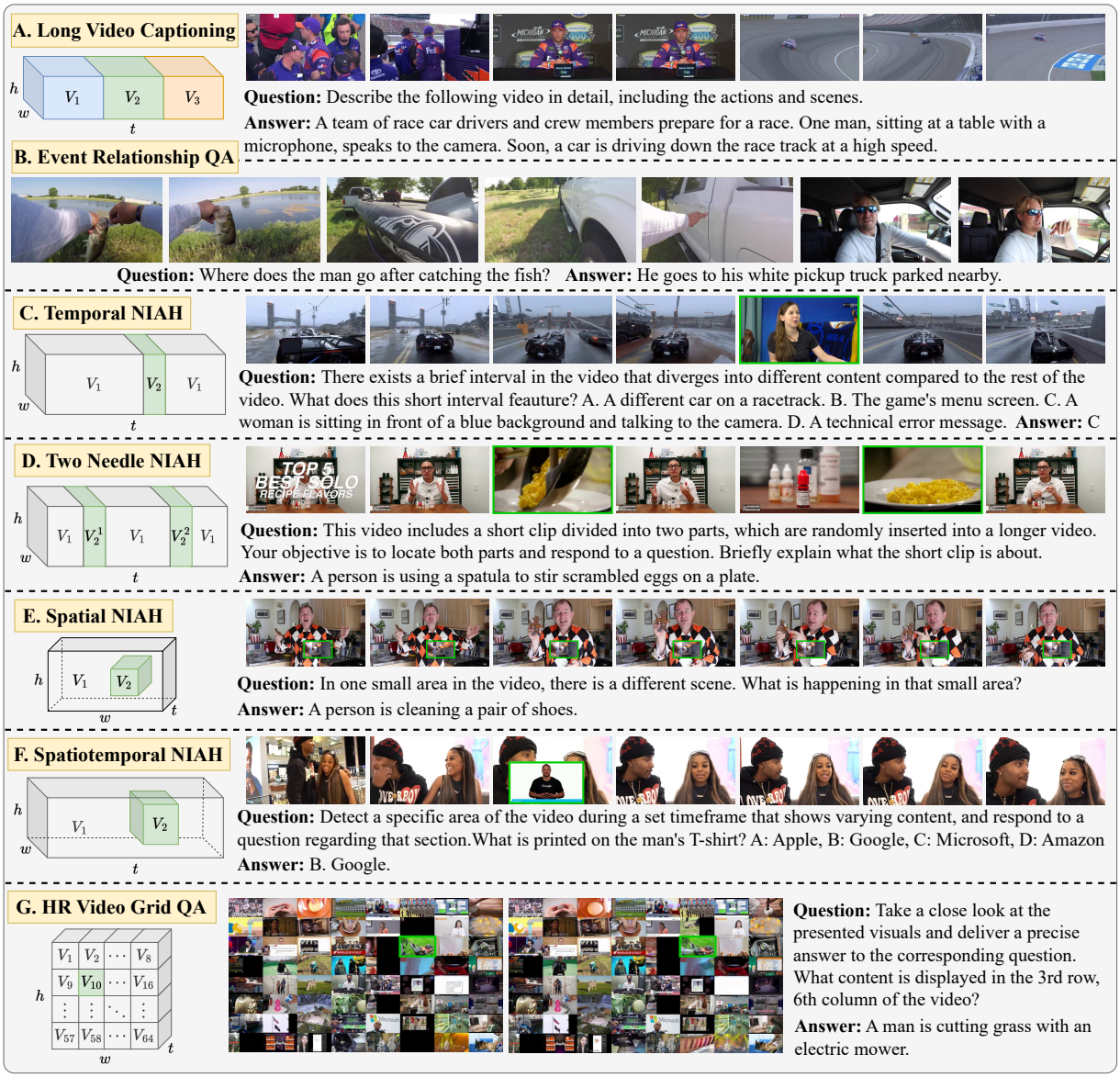


Figure 8.2: Our proposed video augmentation and instruction-following data synthesis schemes for VISTA-400K. Given input videos, We perform spatiotemporal video combinations to produce augmented video samples with longer duration and higher resolution.

Table 8.1: Statistics of our synthetic video instruction-following dataset. "(N)" and "(H)" corresponds to the "needle" (short or low-res videos) and the "haystack" (long or high-res videos) in NIAH subsets.

Subset	Instruction Type	Video Source	#Videos	Avg. Duration	Avg. Resolution
Long Video Captioning	Video Captioning	Panda-70M [44]	58,617	33.2s	1277×720
Event Relationship QA	Freeform QA/MCQ	Panda-70M [44]	56,854	33.4s	1278×720
Temporal NIAH	Freeform QA/MCQ	Panda-70M [44] (N), MiraData [123] (H)	59,751	67.6s	640×358
Two Needle NIAH	Freeform QA	Panda-70M [44] (N), FineVideo [69] (H)	52,349	112.4s	591×382
Spatial NIAH	Freeform QA/MCQ	InternVid [276] (N), OpenVid-1M [194] (H)	59,978	9.9s	1726×971
Spatiotemporal NIAH	Freeform QA/MCQ	OpenVid-1M [194] (N), FineVideo [69] (H)	56,494	89.9s	591×383
HR Video Grid QA	Freeform QA/MCQ	InternVid [276]	59,901	3s	1920×1080
VISTA-400K	-	-	403,944	48.6s	1160×666

to perform video mixing and combination. We use a highly capable language model Gemini-1.5-Pro [244] as Θ to generate synthetic QA pairs. As shown in Figure 8.2, our dataset contains a total of seven subsets, each featuring different video augmentation methods. In this section, we detail the dataset construction process for each subset.

8.2.1 Long Video Captioning & Event QA

Our first approach to synthesizing long video instruction data is to generate longer videos by temporally concatenating multiple short clips, as illustrated by Figure 8.2-A. We observe that public video-text datasets like InternVid [276] and Panda-70M [44] often contain short clips that are drawn from longer videos, which can be combined to form extended sequences. To harvest long video data, we combine multiple short clips from the same source video, ensuring that the interval between them does not exceed five seconds. This preserves natural scenes and content transitions in the extended videos. Given sampled videos containing multiple short clips and their accompanying captions, we generate two types of instruction-following data using Gemini:

1. **Long Video Captioning**: given the captions for each short clip, we prompt Gemini to generate a longer caption describing the entire video. This task is designed to enhance the summarization ability of video LMMs for input videos with longer durations.

2. **Event Relationship QA**: we let Gemini generate freeform or multiple choice questions related to the order of the events based on the short clip captions. Figure 8.2-B shows an example of such instruction data. These QA pairs focus on improving video LMM's ability to recognize action and event sequences.

8.2.2 Video Needle-in-a-haystack (NIAH) QA

To effectively understand and reason over long or high-resolution videos, video LMMs must learn to accurately retrieve relevant information from this long sequence of video tokens, i.e. finding "needles in a haystack". NIAH experiments have been widely adopted to evaluate LLMs [124] and have also been adapted to LMM evaluations [270, 263]. In this approach, we propose to spatially or temporally combine videos to form various NIAH QA data:

1. **Temporal NIAH**: as shown in Figure 8.2-C, this approach randomly inserts a short clip within a longer video, creating a "temporal needle" in the sequence. Based on this combined video, our instruction data contain questions that ask video LMMs to describe the content of the needle video. This task challenges video LMMs to locate the relevant needle tokens within a long video and accurately summarize their content.

2. **Two Needle NIAH**: we consider a variant of Temporal NIAH, where a short video clip is split into two parts and inserted at different timestamps within a longer clip. Video LMMs must summarize the short clip's content by locating both "needles", challenging them to retrieve relevant information from multiple temporal locations within the video sequence.

3. **Spatial NIAH**: recognizing local and small objects in high-resolution videos is essential for video LMMs, yet obtaining suitable training data for high-resolution videos with detailed captions or QA pairs is challenging. We propose to generate spatial NIAH data to simulate such training data. As depicted in Figure 8.2-E, we overlay a small and low-resolution video onto a high-resolution video at a random position, and then ask a question related to this small video. This approach focuses our QA data on a specific "needle" region, forcing video LMMs to extract local information from high-resolution videos.

4. **Spatiotemporal NIAH**: finally, we generate synthetic training data by integrating spatial and temporal NIAH. As shown in Figure 8.2-F, we embed a low-resolution, short-duration "needle" video within a longer, high-resolution video at a random spatial position and timestamp. This combined setup encourages video LMMs to understand video contents across both spatial and temporal dimensions, fostering a more comprehensive understanding of complex video inputs.

For all four NIAH variants, we obtain freeform QA pairs by prompting Gemini using the captions of the "needle" videos. We further convert half of these QA pairs into multiple-choice questions by prompting Gemini to generate incorrect distractors derived from the captions of the "haystack" videos. This method ensures the distractor options are contextually related to the haystack content. Consequently, if the model fails to identify the needle video correctly, it is more likely to choose these distractor options, increasing the challenge and rigour of the NIAH tasks.

8.2.3 High-Resolution Video Grid QA

In this task, we explore the idea of generating high-resolution video instruction data by combining multiple low-resolution videos. From a large collection of low-resolution video clips, we randomly sample 64 videos and arrange them in a 8×8 grid (*c.f.* Figure 8.2-G). Each video is resized to 240×135, resulting in a combined video resolution of 1920×1080. We then randomly select a cell at row i and column j in the video grid and synthesize a question about the content in this specific cell. This design enhances video LMM's ability to understand high-resolution videos by requiring it to locate the correct cell based on indices and accurately interpret the content within that small area. For MCQ data, we randomly select other cells within the grid and use their captions to create distractor options.

8.2.4 Summary of Video Instruction-Following Data

Table 8.1 provides a summary of our video instruction-following dataset. Our dataset comprises ∼400K entries, with each long video over 30 seconds and each high-resolution video at least 960p. A key advantage of our data synthesis pipeline is that our QA synthesis process (*c.f.* Equation 8.1) requires only text processing via the Gemini API, without needing Gemini's multimodal functionality. This makes our approach significantly more cost-efficient compared to existing methods [40, 339]. We leverage five datasets: Panda-70M [44], MiraData [123], FineVideo [69], InternVid [276] and OpenVid-1M [194] to generate synthetic video instruction data. However, our method also applies to any raw video source by first using an off-the-shelf video captioning model to generate simple captions, followed by our pipeline to produce high-quality video instruction data.

Table 8.2: Comparisons between baseline models and VISTA-finetuned models on long/short video understanding benchmarks. The best results among open-source models are bolded. Δ denotes the performance differences before and after finetuning on VISTA-400K.

Models	Size	Long Video Understanding							Short Video Understanding	
		Video-MME w/o subtitles				MLVU	LVBench	LongVideoBench	MVBench	NExT-QA
		avg	short	medium	long	m-avg	test	val	test	mc
Proprietary Models										
GPT-4V [3]	-	59.9	70.5	55.8	53.5	49.2	-	59.1	43.5	-
GPT-4o [198]	-	71.9	80.0	70.3	65.3	64.6	34.7	66.7	-	76.0
Gemini-1.5-Pro [244]	-	75.0	81.7	74.3	67.4	-	33.1	64.0	-	-
Open-source Models										
VideoChat2 [147]	7B	39.5	48.3	37.0	33.2	47.9	-	39.3	51.9	**78.6**
LLaMA-VID [153]	7B	-	-	-	-	33.2	23.9	-	41.3	-
ST-LLM [169]	7B	37.9	45.7	36.8	31.3	-	-	-	54.9	-
ShareGPT4Video [40]	7B	39.9	48.3	36.3	35.0	46.4	-	39.7	51.2	-
LongVILA [309]	7B	50.5	61.8	50.4	46.2	-	-	-	-	-
LongLLaVA [273]	7B	52.9	61.9	51.4	45.4	-	-	-	54.6	-
Video-XL [229]	7B	**55.5**	64.0	**53.2**	**49.2**	**64.9**	-	49.5	**55.3**	77.2
VideoLLaVA [160]	7B	39.9	45.3	38.0	36.2	47.3	29.3	39.9	43.8	61.8
VISTA-VideoLLaVA	7B	43.7	48.2	43.9	38.9	49.5	33.8	42.3	47.2	63.0
Δ - VideoLLaVA		**+3.8**	**+2.9**	**+5.9**	**+2.7**	**+2.2**	**+4.5**	**+2.4**	**+3.4**	**+1.2**
Mantis-Idefics2 [120]	8B	45.4	55.9	43.0	37.2	49.4	35.0	45.8	51.4	75.8
VISTA-Mantis	8B	48.2	58.4	46.7	39.6	55.5	36.4	49.1	52.5	75.2
Δ - Mantis-Idefics2		**+2.8**	**+2.5**	**+3.7**	**+2.4**	**+6.1**	**+1.4**	**+3.3**	**+1.1**	**-0.6**
LongVA [334]	7B	52.4	61.4	50.9	45.0	56.3	35.9	51.8	49.2	68.3
VISTA-LongVA	7B	**55.5**	**66.0**	53.1	47.4	62.1	**39.0**	**53.1**	51.1	69.3
Δ - LongVA		**+3.1**	**+4.6**	**+2.2**	**+2.4**	**+5.8**	**+3.1**	**+1.3**	**+1.9**	**+1.0**

8.3 Evaluation: HRVideoBench

We observe that existing video understanding benchmarks are inadequate for accurately assessing the ability of video LMMs to understand high-resolution videos, especially the details inside the videos. Prior benchmarks mainly consist of low-resolution videos. More recent benchmarks focus on evaluating the long video understanding capability of video LMMs, which contain questions that typically pertain to a short segment in the long video. As a result, a model's high-resolution video understanding performance can be undermined if it struggles to sample or retrieve the relevant frames from a lengthy video sequence.

To address this gap, we introduce HRVideoBench, a comprehensive benchmark with 200 multiple-choice questions designed to assess video LMMs for high-resolution video understanding. HRVideoBench focuses on the perception and understanding of small regions

Table 8.3: Quantitative results on HRVideoBench and open-ended video QA benchmarks. "acc." represents accuracy.

Models	High-Res Video Understanding			Open-Ended Video QA							
	HRVideoBench			MSVD-QA		MSRVTT-QA		TGIF-QA		ActivityNet-QA	
	avg	object	action	acc.	score	acc.	score	acc.	score	acc.	score
VideoLLaVA [160]	32.5	36.0	27.9	60.3	3.7	42.1	3.0	63.5	3.8	48.6	3.3
VISTA-VideoLLaVA	47.5	50.0	44.2	71.5	4.0	58.5	3.5	78.0	4.3	49.1	3.4
Δ - VideoLLaVA	**+15.0**	**+14.0**	**+16.3**	**+11.2**	**+0.3**	**+16.4**	**+0.5**	**+14.5**	**+0.5**	**+0.5**	**+0.1**
Mantis-Idefics2 [120]	48.5	50.9	45.4	57.4	3.5	34.9	2.7	65.7	3.8	46.5	3.1
VISTA-Mantis	51.0	53.5	47.7	65.2	3.8	46.4	3.1	71.4	4.0	48.8	3.3
Δ - Mantis	**+2.5**	**+2.6**	**+2.3**	**+7.8**	**+0.3**	**+11.5**	**+0.4**	**+5.7**	**+0.2**	**+2.3**	**+0.2**
LongVA [334]	48.0	52.6	41.9	56.3	3.5	37.7	2.8	55.4	3.4	48.0	3.2
VISTA-LongVA	50.0	56.1	41.9	61.0	3.7	42.5	3.0	67.5	3.9	51.8	3.4
Δ - LongVA	**+2.0**	**+3.5**	**+0.0**	**+4.7**	**+0.2**	**+4.8**	**+0.2**	**+12.1**	**+0.5**	**+3.8**	**+0.2**

and subtle actions in the video. Our test videos are at least 1080p and contain 10 different video types collected with real-world applications in mind. For example, key applications of high-resolution video understanding include autonomous driving and video surveillance. We correspondingly collect POV driving videos and CCTV footage for the benchmark. Our benchmark consists of 10 types of questions, all of which are manually annotated and can be broadly categorized into object and action-related tasks. The question types are:

- Object-related Tasks: Object Counting, OCR problem, Object Recognition, Entity Recognition, Object Property Recognition, Object Status Change Recognition.
- Action-related Tasks: Action Recognition, Moving Direction Identification, Interaction Detection, Temporal Sequence Recognition.

8.4 Experimental Results

8.4.1 Evaluation Setup

To validate the effectiveness of VISTA-400K, we finetune a diverse set of LMMs on our dataset. Specifically, we choose VideoLLaVA [160], Mantis-Idefics2 [120] and LongVA [334] as the base models because these models disclose details about their training dataset. We directly finetune LongVA and Mantis-Idefics2 on our dataset as they are mainly pretrained on image data. For VideoLLaVA, we additionally include 300K short video from

VideoChat2-IT [147] to preserve the model's short video understanding capability. We use 8 frames for VideoLLaVA, 24 frames for Mantis-Idefics2, and 64 frames for LongVA for finetuning and evaluation. The input resolution is fixed at 224×224 for VideoLLaVA and 336×336 for LongVA, while Mantis-Idefics2 supports dynamic resolutions up to 980×980. The finetuned model is denoted as VISTA-[base model].

Our evaluation assesses video LMMs' capabilities in long and high-resolution video understanding. We test the finetuned models on four comprehensive long video understanding benchmarks: Video-MME [74], MLVU [344], LVBench [268] and LongVideoBench [289]. For high-resolution video understanding, we utilize our HRVideoBench. Additionally, we report results on prior short video understanding benchmarks including MVBench [147] and NExT-QA [297], as well as open-ended video QA benchmarks such as MSVD-QA [304], MSRVTT-QA [304], TGIF-QA [117] and ActivityNet-QA [320].

8.4.2 Quantitative Results

Long Video Understanding Quantitative results for long video understanding are shown in Table 8.2. All three models demonstrate a consistent performance boost across the evaluation benchmarks, with average improvements of 3.3% for Video-MME, 4.7% for MLVU, 3.0% for LVBench, and 2.3% for LongVideoBench. For Video-MME, our method yields the largest average improvement for medium-length questions, which correspond to videos ranging from 4 to 15 minutes. Given that most of the augmented videos in our dataset are under 3 minutes, this result suggests that our approach generalizes effectively to longer videos. One possible explanation for this observation is that longer videos often contain more scene cuts and transitions, which is aligned with our synthetic data where multiple clips with different scenes are concatenated. Our VISTA-LongVA achieves state-of-the-art performance on Video-MME, LVBench and LongVideoBench among open-source models, demonstrating the effectiveness of our approach.

High-Resolution Video Understanding We present the HRVideoBench evaluation results for VISTA-finetuned models and the baseline models in Table 8.3. Among the baseline models, we observe that Mantis-Idefics2, with its higher input video resolution, outperforms LongVA on both object and action-related tasks, despite being considered weaker for long video understanding. This suggests that our HRVideoBench requires detailed comprehension of high-resolution videos, and simply adding more input frames during testing does not necessarily lead to enhanced model performance. Overall, we find that finetuning on our dataset enhances the high-resolution video understanding capabilities for all three models. Additionally, we observe that recognizing subtle actions remains more

challenging than identifying objects and their properties, suggesting that there remains significant potential for improvement in high-resolution video understanding for video LMMs.

Short Video Understanding We further validate our proposed method on several prior short video understanding benchmarks. According to Table 8.2, all three VISTA-finetuned models show improvement on both MVBench and NExT-QA benchmarks except Mantis. This result is likely due to NExT-QA's training split being included as part of the in-domain training data for Mantis-Idefics2. Consequently, finetuning on our out-of-domain data causes a slight decrease in performance. While our results are slightly worse than Video-XL [229], we note that their pretraining and instruction-tuning datasets are larger (>2.7M training samples). Here, we focus more on the relative improvement of our method over the baseline models.

Since the benchmarks mentioned above consist solely of multiple-choice questions, we also include several open-ended QA benchmarks to verify the text generation capabilities of the finetuned models. Following Video-ChatGPT [190], we use GPT-3.5-Turbo [197] to evaluate the accuracy of the responses and rate their quality on a scale of 1 to 5. As shown in Table 8.3, finetuning on our dataset improves both accuracy and generation quality across all four benchmarks for the VISTA-finetuned models. This demonstrates that our dataset can enhance the generation quality of LMMs.

8.4.3 Ablation Study

To verify that each subset in VISTA-400K contributes to the performance gain of long or high-resolution video understanding, we conduct an ablation study and train several Mantis-Idefics2 models by excluding each subset from the training data. In addition, we add the same amount of video training data from VideoChat2-IT [147] to each of the ablation experiments to ensure the total number of training examples does not change. The results on Video-MME and HRVideoBench are shown in Table 8.4. We observe that excluding the long video subsets decreases the model performances on Video-MME. Similarly, excluding high-resolution video subsets leads to a drop in scores on HRVideoBench. These observations indicate that all seven video augmentation methods are effective at their corresponding tasks. We also report results for a Mantis-Idefics2 model without our proposed video augmentations by using the original videos directly for training before combining them with others (e.g., using the needle video in NIAH tasks to train the model directly). According to the last row in Table 8.4, this yields very low scores on Video-MME and HRVideoBench, highlighting the necessity of our augmentations to enhance video instruction data quality.

Table 8.4: Ablation study results for VISTA-Mantis. Each "w/o [Subset]" denotes a Mantis-Idefics2 model finetuned on a modified VISTA-400K by replacing the corresponding subset with the same amount of training examples from VideoChat2-IT [147].

Models	Video-MME		HRVideoBench
	w/o sub.	avg	avg
VISTA-Mantis	**48.2**		**51.0**
w/o Long Video Captioning	47.9		48.0
w/o Event Relationship QA	47.7		49.5
w/o Temporal NIAH	47.5		48.0
w/o Two Needle NIAH	48.1		50.5
w/o Spatial NIAH	47.2		47.5
w/o Spatiotemporal NIAH	47.7		50.0
w/o HR Video Grid QA	47.8		48.0
w/o Video Augmentation	45.7		44.5

Furthermore, previous studies [141, 334] have shown that training on high-resolution image data yields strong long video understanding models. Our ablation study supports this conclusion and demonstrates that the proposed high-resolution video augmentation methods enhance the performances of long video understanding tasks. In addition, our results indicate that the reverse is also true: our long video augmentation methods also improve the performance of high-resolution video understanding. We believe this transferability between long and high-resolution video understanding capabilities is because both types of inputs convert to a long sequence of video tokens in video LMM's embedding space. As a result, identifying short moments from a long video and recognizing small regions in a high-resolution video both correspond to the task of finding key information needles in a haystack of video tokens.

8.4.4 Case Study

To qualitatively evaluate our method, we compare the generated responses of VISTA-finetuned models with the baseline models, as illustrated in Figure 8.3. The "helicopter" example involves recognizing event orders within a long video sequence. The baseline models LongVA and VideoLLaVA exhibit hallucination issues, while our VISTA-LongVA describes the action accurately. Due to the sparse sampling rate of VideoLLaVA (8 frames per video), VISTA-VideoLLaVA misses frames related to the action immediately following



Figure 8.3: Qualitative comparisons between the baseline models and our VISTA-finetuned models. Red text indicates hallucinations or incorrect responses, while green text highlights the correct responses that correspond accurately to the video content.

"the helicopter takes off". Nevertheless, it successfully describes the later event, "the helicopter crashes onto a rooftop", which the vanilla VideoLLaVA fails to recognize. The "table tennis" example challenges video LMMs to identify a subtle and unusual action within a high-resolution video. As demonstrated in the second part of Figure 8.3, Mantis-Idefics2 fails to recognize this action, instead generating a response based on common knowledge about table tennis. Conversely, VISTA-Mantis correctly identifies the unusual action that the player hits the table tennis ball with his leg.

8.5 Conclusion

We presented VISTA, a video augmentation framework to enhance long-duration and high-resolution video understanding by generating high-quality synthetic video instruction-following data from existing video-caption datasets. VISTA performs spatiotemporal video

combination over input videos to create new video samples and utilizes video captions for instruction synthesis. Through extensive experiments on various long video understanding benchmarks and our proposed HRVideoBench, we demonstrated that VISTA-400K consistently improves the performance of three different video LMMs. For future work, we aim to design and explore additional video augmentation methods to further strengthen our approach's robustness.

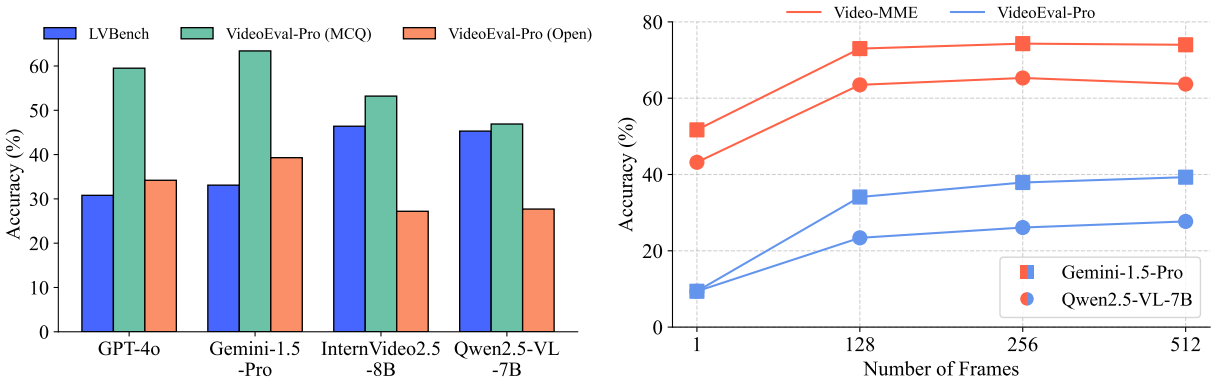
Chapter 9

VideoEval-Pro: Robust and Realistic Long Video Understanding Evaluation

9.1 Introduction

Long video understanding (LVU) refers to the task of using AI systems to process, interpret, and reason over long-duration video content. Key applications of long video understanding include event and anomaly detection in video surveillance [184], temporal reasoning and behaviour prediction in autonomous driving [68], as well as content summarization or key information retrieval in instructional/lecture videos [345]. Designing AI systems capable of understanding and reasoning over long videos is therefore a fundamental challenge in artificial intelligence.

Recently, large multimodal models (LMMs) have emerged as a promising solution for understanding long videos. Ongoing research enhances the capability of LMMs to process longer videos by extending their context length [334, 50], dropping or merging video tokens [151, 279], and leveraging efficient, linear-complexity models [216, 121, 116]. Besides model architecture improvements, recent studies also investigate curating better training data [339, 218] and applying reinforcement learning approaches [152, 71] for LMMs tailored to LVU tasks. As a result, LMMs have been rapidly advanced to achieve stronger LVU capabilities: the preliminary attempt of Video-LLaVA [160] (November 2023) can only process short videos with eight frames. Today, LMMs such as Vamba [216], Video-XL-Pro



(a) MCQ Benchmarks vs. VideoEval-Pro (b) VideoEval-Pro vs. Video-MME

Figure 9.1: Comparison between VideoEval-Pro and MCQ benchmarks. **Left:** MCQ benchmarks yield inflated scores on identical questions (MCQ vs. Open) and can misrepresent model performance (LVBench [268]). **Right**: VideoEval-Pro cannot be effectively solved with a single input frame, and performance scales consistently with more frames. Video-MME [74] exhibits contradictory trends.

[172], and InternVideo2.5 [279] (early 2025) can encode thousands of frames and reason over hour-long videos.

To rigorously evaluate ongoing advances in video LMMs, researchers have introduced dedicated long video understanding benchmarks [74, 344, 268, 289, 30], which provide standardized scores to quantitatively measure and compare different models' ability to reason over long videos. Nevertheless, by taking a deeper look at current LVU benchmarks, our findings are sobering. First, most current LVU benchmarks rely almost exclusively on multiple-choice questions (MCQs), a format that can inadvertently provide hints to the model, enabling it to answer correctly through guesswork. As shown in Figure 9.1a, for the same set of questions, we observe over 20% accuracy drop when switching from MCQ to open-ended question answering. This significant gap suggests that MCQ-based accuracies may be substantially inflated and do not reliably reflect the model's true understanding of the video content. Second, many questions in existing LVU benchmarks exhibit strong priors that allow models to answer correctly without meaningfully processing the input video. As illustrated in Figure 9.1b, both proprietary (Gemini-1.5-Pro [245]) and open-source (Qwen2.5-VL-7B [15]) models achieve around 50% accuracy on Video-MME [74] with only one input frame. These issues lead to performance plateauing or even declining as input frames increase—an outcome that contradicts the expectation that more frames should offer richer context and improve long video understanding. Our investigations of existing LVU benchmarks prompt us to think about two central questions:

Do existing long video benchmarks faithfully reflect models' real capacity to understand long video content? (2) **Do the gains reported by newer models genuinely translate into stronger long video comprehension capability, or are they illusional?**

To probe these questions, we present VideoEval-Pro, a more robust and realistic long video understanding benchmark containing open-ended, short-answer QA problems. To construct VideoEval-Pro, we source the questions from four existing long video understanding MCQ benchmarks: Video-MME [74], MLVU [344], LVBench [268] and LongVideoBench [289], and reformat these questions into free-form questions. We apply a series of filtering methods based on video duration, question and answer type, answerability and QA difficulty to ensure the quality of our benchmark. Our final benchmark contains a total of 1,289 short-answer questions based on 465 videos, with an average duration of 38 minutes. By evaluating a wide array of proprietary and open-source models on VideoEval-Pro, our main findings can be summarized as below:

1. VideoEval-Pro and its open-ended QA format introduce substantial challenges for video LMMs, as evidenced by performance drops exceeding 25% compared to the MCQ format.
2. VideoEval-Pro's results show that LMMs perform better at questions about local video segments and struggle more with those requiring holistic video understanding. They also perform better on perception tasks than on reasoning tasks.
3. Unlike existing MCQ benchmarks, VideoEval-Pro reveals that proprietary models still hold a significant lead, indicating the brittleness of open-source models on challenging LVU tasks.
4. Current LMMs achieve only ∼10% accuracy on VideoEval-Pro with a single input frame, but performance steadily improves with more frames. These results highlight our benchmark's requirement for rich temporal information and show that VideoEval-Pro is a more suitable benchmark for long video understanding.

9.2 VideoEval-Pro

9.2.1 Data Curation Pipeline

VideoEval-Pro's data curation pipeline comprises two main steps: Data Collection and Data Filtering. In this section, we describe each step in detail.

Data Collection To construct VideoEval-Pro, we first collect source QA pairs from four publicly available long video understanding benchmarks: Video-MME [74], MLVU

[344], LVBench [268] and LongVideoBench [289]. These benchmarks span diverse video content and question types, providing a rich source for long video understanding tasks. Our initial seed question set contains a total of 5,562 questions, which are all in MCQ format with 4-6 options. To create an open-ended evaluation benchmark, we transform each multiple-choice question into a free-form question: the correct MCQ option becomes the reference answer, while the distractors are discarded. During evaluation, the models receive only the question itself, forcing them to generate an answer based on the input video rather than exploiting hints from different options.

Video Duration Filtering Once the initial pool of questions is collected, we apply a multi-stage filtering process to ensure that the resulting dataset emphasizes long-term video comprehension and presents a meaningful challenge for current models. In the first stage, we filter out all samples associated with videos shorter than 10 minutes. Shorter clips often contain less complicated long-term temporal dependencies, which may lower the difficulty of video perception and reasoning tasks. In order to maintain the difficulty and reliability of VideoEval-Pro, we only selected questions with medium-to-long videos (¿10 minutes).

Question and Answer Type Filtering In the second stage, we remove questions for which the average word count of answer options in the original MCQ format exceeds five words. For example, questions such as "What is this video about?" often yield overly detailed responses, which complicates answer evaluation. This word-count constraint reduces uncertainty from overly verbose options and ensures that the converted open-ended questions have concise yet meaningful answers, making it easier for LLM judges (*c.f.* Section 9.2.3) to evaluate model responses, thereby enhancing the stability and precision of our benchmark and improving its overall validity.

Answerability Filtering In the third stage, we assess whether each multiple-choice question can be reasonably reformulated into a free-form question without losing clarity or answerability. From the question pools we collected, we notice three types of questions with low answerability: (1) Option-evaluating or comparing questions, which require the model to compare different options and pick the most reasonable option; (2) Timestamp-dependent questions, which requires the model to answer questions for a given numerical timestamp; (3) Subtitle-dependent questions, which queries information that only appeared in the subtitles. We prompt Gemini-2.0-Flash with the question (excluding the answer choices) and ask it to determine whether the question can be answered solely based on

the video content. This step helps identify and discard questions that rely heavily on inspecting MCQ options, which are unsuitable for open-ended evaluation.

Difficulty Filtering Finally, we filter out questions that are too easy to answer. To identify such cases, we randomly sample a single frame from each input video and prompt Gemini-2.0-Flash to generate an answer to the corresponding MCQ and open-ended question using only that frame. We then use Gemini-2.0-Flash to judge the open-ended answers. Questions for which Gemini-2.0-Flash produces a correct response on both MCQ and open-ended formats are excluded from the benchmark. This filtering step ensures that the remaining questions require broader temporal understanding and cannot be resolved using minimal visual context.

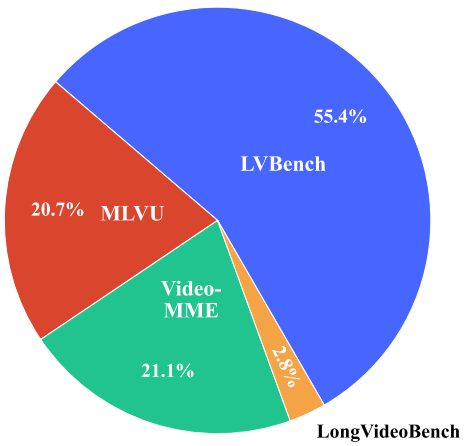
9.2.2 Dataset Statistics

Our rigorous data collection and filtering pipeline ensures that the final benchmark questions demand deeper temporal comprehension and reasoning beyond surface-level cues. Our final dataset comprises 1,289 question-answer pairs in free-form QA, each grounded in a long video with a duration greater than 10 minutes. As shown in Table 9.1, VideoEval-Pro includes a total of 465 videos, with an average length of 38.25 minutes. Among them, 204 videos are between 10 and 30 minutes and 261 videos exceed 30 minutes. For the 1,289 questions used in our benchmark, 371 are associated with videos in the 10–30 minute range, while 918 are based on videos longer than 30 minutes. The average length of an answer is 2.1 words. These design choices ensure the evaluation focuses on the model's ability to retrieve concise and accurate information from long video content.

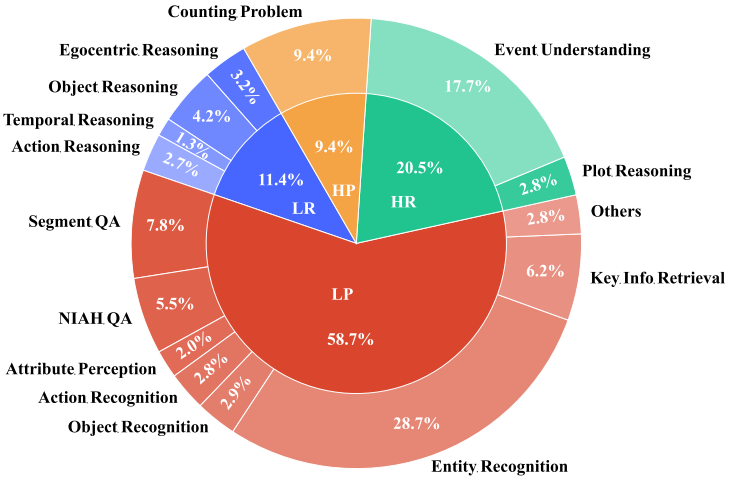
Table 9.1: Video and QA statistics for VideoEval-Pro.

VideoEval-Pro	Total	10–30 min	>30 min	Note
Videos	465	204	261	Average video duration: 38.25 minutes
QA Pairs	1,289	371	918	Average answer length: 2.1 words

QA Source Distribution As mentioned in Section 9.2.1, the questions in VideoEval-Pro are drawn from four existing benchmarks: Video-MME [74], MLVU [344], LVBench [268], and LongVideoBench [289]. Given the variability in video duration and question quality across these sources, their contributions to the final dataset differ. As illustrated in



(a) Video and QA source distribution of VideoEval-Pro



(b) Task type distribution of VideoEval-Pro

Figure 9.2: Summary of VideoEval-Pro data composition and task type distribution.

Figure 9.2a, LVBench accounts for the largest portion, contributing 714 questions (55%) due to its long and information-rich video sources. MLVU contributes 267 questions (21%), with many QAs excluded because of the relatively short video lengths. Video-MME adds 272 questions (21%), although a significant number were filtered out due to limited answerability. LongVideoBench, which is smaller in scale and subject to strict selection criteria, contributes 36 questions (3%). This diverse composition ensures that VideoEval-Pro spans a wide range of content domains and video types, ensuring a comprehensive model evaluation.

Task Definition and Distribution Given the questions we collected, we propose a unified and generalizable task taxonomy to categorize our benchmark questions into four main types and 15 subtypes. These task types capture both perception and reasoning demands for both local video segments and holistic long video understanding tasks. The four main task types are:

- **Local Perception (LP)**: LP focuses on identifying and retrieving visual elements or actions from a short video clip in a long video. Subtypes in this category include *Segment QA*, *Needle-In-A-Haystack (NIAH) QA*, *Attribute Perception*, *Action Recognition*, *Object Recognition*, *Entity Recognition*, *Key Information Retrieval* and a combined *Other* subtype.

- **Local Reasoning (LR)**: LR focuses on reasoning within short temporal windows, such as inferring causality, temporal order, or changes that happen over a local sequence of events. The four subtypes in this category are *Egocentric Video Reasoning*, *Object Reasoning*, *Temporal Reasoning* and *Action Reasoning*.
- **Holistic Perception (HP)**: HP involves a global and holistic understanding of statistical, structural, or spatial information, typically requiring visual aggregation. In VideoEval-Pro, HP is comprised of *Visual Counting* problems.
- **Holistic Reasoning (HR)**: HR requires abstract or high-level understanding of long videos across events or scenes, often involving narrative or intent understanding. The two subtypes for HR are *Event Understanding* and *Plot Reasoning*.

This taxonomy enables fine-grained evaluation of model capabilities across different cognitive demands required by long video understanding. Based on this taxonomy, the distribution of questions in our dataset is summarized in Figure 9.2b. The majority of questions (59%) fall under *Local Perception*, reflecting the VideoEval-Pro's emphasis on fine-grained tracking and understanding of visual dynamics. *Holistic Reasoning* accounts for 21% of the questions, while *Local Reasoning* and *Holistic Perception* represent 11% and 10% of the questions in the dataset, respectively.

9.2.3 Evaluation Pipeline

For each question in the benchmark, we uniformly sample a fixed number of frames from the corresponding video. We use all frames if the total number of available frames is fewer than the required frame count. The sampled frames, along with the open-ended question, are passed to the evaluated model to generate an answer. To evaluate the correctness of each model's responses, we adopt the evaluation criteria introduced in SimpleQA [282] and Video-SimpleQA [27]. Specifically, each model response is classified into one of the following categories:

- **Correct**: The predicted answer comprehensively includes all essential information present in the reference answer and contains no contradictory content.
- **Incorrect**: The predicted answer includes statements that contradict the reference answer, or provides uncertain responses such as "possibly" or "I think".
- **Not Attempted**: The predicted answer omits critical elements of the reference answer but does not contradict it, or the model refuses to answer the question.

We follow the *LLM-as-a-Judge* [92, 341] paradigm and employ `GPT-4o-0806` as our evaluation model to assess the accuracy of generated short answers. Finally, we report the overall correct rate as the proportion of responses labelled "Correct" across the entire dataset. This metric reflects the model's ability to provide accurate, faithful answers grounded in the visual content. Note that we do not report the F-score (harmonic mean of overall correct and correct given attempted) adopted in SimpleQA and Video-SimpleQA, as we want the results from our open-ended questions and the corresponding MCQ scores to be comparable.

9.3 Experiments

9.3.1 Experimental Setup

We consider a total of 21 proprietary and open-source LMMs and conduct evaluations on our VideoEval-Pro. For proprietary models, we consider the GPT series [198, 199, 200] (GPT-4o, GPT-4.1, GPT-4.1-mini) and the Gemini series [245, 125] (Gemini-2.5-Flash, Gemini-1.5-Flash, Gemini-1.5-Pro). For open-source models, we include prior methods such as Video-LLaVA [160], Mantis-Idefics2 [120], and LongVA [334]; recent large-scale pretrained LMMs such as the Qwen-VL model family (Qwen2-VL [265] and Qwen2.5-VL [15]), InternVL model family (InternVL2.5 [51], InternVL3 [346], InternVideo2.5 [279]), LLaVA-Video [339] and Phi-4 [1]; finally, we also consider extra-long video understanding LMMs such as LongLLaVA [273], Video-XL [229], LongVU [228], Vamba [216] and VideoChat-Flash [151].

As different candidate models are trained using different numbers of frames, we evaluate each one with inputs of 32, 64, 128, 256, and 512 frames and report its highest score. If a model cannot handle larger inputs (e.g. due to API restrictions or model context length limits), we instead report its best score among the frame counts that fit within its allowable context window. We employ the LLM-based scoring method described in Section 9.2.3 to compute each model's final accuracy on the short-answer tasks. For all models and the LLM judge, we use greedy decoding (set temperature to 0) to ensure deterministic outputs. For comparison, we also measure multiple-choice performance: we rerun the evaluation and provide the original answer options to each model. All models are prompted to output only the selected choice, and accuracy is computed via exact string matching.

Table 9.2: Main results of our VideoEval-Pro benchmark. Δ indicates the gap between MCQ and open-ended questions. LP, LR, HP and HR correspond to Local Perception, Local Reasoning, Holistic Perception and Holistic Reasoning tasks.

Models	Size	Frames	LP		LR		HP		HR		Overall		
			Open	MCQ	Open	MCQ	Open	MCQ	Open	MCQ	Open	MCQ	Δ
Proprietary Models													
GPT-4o	-	256	39.4	64.8	23.1	62.6	26.4	42.1	29.2	50.4	34.2	59.5	25.3
Gemini-1.5-Flash	-	512	41.5	65.5	25.9	63.9	27.3	36.4	25.8	55.7	35.1	60.6	25.5
Gemini-2.5-Flash	-	256	42.4	64.1	30.6	65.3	25.6	33.9	26.9	54.2	36.3	59.3	23.0
Gemini-1.5-Pro	-	512	43.7	66.7	32.7	69.4	35.5	40.5	31.8	61.0	39.3	63.4	24.1
GPT-4.1-mini	-	256	46.0	68.6	32.0	68.7	27.3	38.8	32.6	57.6	39.9	63.5	23.6
GPT-4.1	-	256	47.2	68.8	29.9	68.7	28.1	38.0	34.5	59.5	40.8	64.0	23.2
Open-source Models													
Video-LLaVA	8B	8	13.2	27.5	6.1	33.3	14.0	24.8	6.1	26.5	11.0	27.7	16.7
Mantis-Idefics2	8B	24	17.8	33.2	9.5	29.9	16.5	16.5	8.3	29.9	14.8	30.6	15.8
LongVA	7B	64	20.5	43.3	6.8	33.3	19.0	24.0	9.5	31.8	16.5	38.0	21.5
Phi-4-Mini	5.6B	128	19.2	46.4	12.9	47.6	18.2	30.6	10.2	31.4	16.5	42.0	25.5
LongLLaVA	9B	512	21.7	41.2	15.0	34.0	14.0	29.8	10.2	29.2	17.8	36.9	19.1
Video-XL	7B	512	22.3	41.9	15.0	34.0	18.2	28.1	10.2	29.2	18.6	38.2	19.6
LongVU	7B	512	25.9	45.6	12.9	38.8	19.8	24.0	17.4	37.1	22.1	41.0	18.9
Vamba	10B	512	28.1	52.4	10.9	40.8	21.5	26.4	12.5	37.9	22.3	45.7	23.4
LLaVA-Video	7B	64	28.5	53.5	13.6	47.6	20.7	28.9	19.3	40.2	24.2	47.8	23.6
InternVL2.5	8B	64	28.8	54.3	19.7	46.3	21.5	35.5	16.7	39.0	24.6	48.5	23.9
InternVL3	8B	64	30.3	54.6	17.0	49.0	24.0	34.7	13.3	36.7	24.7	48.4	23.7
Qwen2-VL	7B	512	31.7	53.9	14.3	51.7	21.5	28.1	20.5	39.0	26.5	48.2	21.7
InternVideo2.5	8B	512	33.6	59.8	17.0	47.6	19.8	34.7	18.2	45.8	27.2	53.2	26.0
VideoChat-Flash	7B	512	33.3	57.7	16.3	43.5	21.5	33.9	17.4	44.7	27.0	51.2	24.2
Qwen2.5-VL	7B	512	33.9	51.7	15.6	48.3	24.8	31.4	17.8	39.8	27.7	46.9	19.2

9.3.2 Main Results and Discussions

Evaluation results are shown in Table 9.2. Our VideoEval-Pro open-ended QA accuracy is denoted as "Open" while "MCQ" represents the corresponding multiple choice accuracy.

Overall, we observe that GPT-4.1 [200] delivers the strongest performance among proprietary models, while Qwen2.5-VL [15] leads the open-source models. We summarize our key findings as follows:

MCQ vs. VideoEval-Pro As shown in Table 9.2, compared to MCQ accuracy, all models demonstrate a substantial drop in performance on open-ended questions. Moreover, the scores obtained from MCQ and open-ended questions are not necessarily correlated. For example, although InternVL2.5 and InternVL3 outperform Qwen2.5-VL on MCQ accuracy, their open-ended QA scores are lower than those of Qwen2.5-VL. These findings suggest that MCQ-based accuracy may overestimate model performance and fail to capture the true capacity of models to understand long videos. Consequently, MCQ results may not serve as a reliable indicator for ranking video LMMs.

Local vs. Holistic Tasks When comparing performance on local versus holistic understanding tasks, we observe that most models perform better on local tasks, suggesting that holistic tasks are generally more challenging. This disparity is expected, as holistic tasks require models to process the entire video and reason over complex temporal dynamics that span long durations. In contrast, local tasks are confined to short video segments, where the actions or events are typically simpler and more temporally localized, making them easier to identify and interpret.

Perception vs. Reasoning Tasks Comparing results between perception and reasoning tasks, we find that although models often achieve similar MCQ accuracy across both task types, their performance on open-ended questions diverges significantly. Specifically, models tend to perform considerably better on perception tasks than on reasoning tasks in the open-ended setting. For instance, Gemini-2.5-Flash achieves comparable MCQ accuracies of 64.1% on local perception tasks and 65.3% on local reasoning tasks. However, its open-ended QA accuracy drops to 30.6% on local reasoning tasks, whereas it maintains a much higher accuracy of 42.4% on local perception tasks. This discrepancy highlights the increased difficulty of long video reasoning tasks, which can be correctly reflected by our VideoEval-Pro.

Proprietary vs. Open-source Models We compare proprietary and open-source models across several benchmarks and observe an interesting phenomenon. As shown in Table 9.3, though the best open-source video LMMs like InternVideo2.5 or InternVL3

Table 9.3: Comparison between proprietary and open-source models on VideoEval-Pro and other standard medium and long video benchmarks.

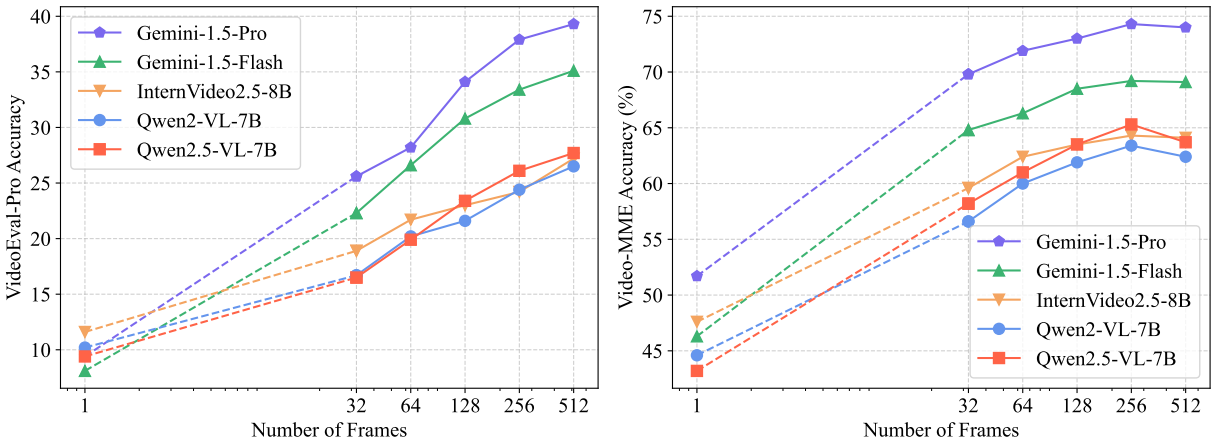
Model	VideoEval-Pro	MVBench	LVBench	LongVideoBench	MLVU
Proprietary Models					
GPT-4o	34.2	**64.6**	30.8	**66.7**	**64.6**
Gemini-1.5-Pro	**39.3**	60.5	**33.1**	64.0	61.2
Open-source Models					
InternVideo2.5-8B	**27.2**	**75.5**	46.4	60.6	72.8
InternVL3-8B	24.7	75.4	44.2	58.8	70.8
VideoChat-Flash-7B	27.0	73.2	**47.2**	**64.2**	**74.5**
Δ (Open - Proprietary)	-13.1	+9.9	+14.1	-2.5	+9.9

have surpassed GPT-4o/Gemini-1.5-Pro by as much as 14% across existing long video understanding benchmarks, their performances on VideoEval-Pro are lagging behind GPT-4o/Gemini-1.5-Pro by 13%. This prominent contrast reveals the brittleness of open-source models on more challenging long video understanding tasks.

9.3.3 Frame Scaling Properties of VideoEval-Pro

In this section, we examine how performance on VideoEval-Pro scales with varying numbers of input frames. We evaluate two proprietary models: Gemini-1.5-Flash and Gemini-1.5-Pro, alongside three open-source models: Qwen2-VL, Qwen2.5-VL, and InternVideo2.5. For each model, we plot the VideoEval-Pro accuracy across different frame counts (1, 32, 64, 128, 256 and 512) in Figure 9.3a. For comparison, we also present the corresponding results from the Video-MME benchmark using the same models and frame settings in Figure 9.3b.

Our first observation is that existing benchmarks such as Video-MME yield relatively high accuracy even when only one frame is provided to the model. As shown in Figure 9.3b, both proprietary and open-source models achieve around 45% accuracy under this setting, with Gemini-1.5-Pro surpassing 50% accuracy. These results suggest that current long video benchmarks may include insufficiently challenging questions, allowing models to answer correctly even when most of the video information is missing. In contrast, all models achieve only around 10% accuracy on our VideoEval-Pro when provided with a single input frame, as shown in Figure 9.3a. This performance drop highlights that



(a) VideoEval-Pro Accuracy Curve (b) Video-MME Accuracy Curve

Figure 9.3: Comparison between VideoEval-Pro and Video-MME accuracy across five LMMs.

VideoEval-Pro cannot be easily solved without incorporating richer visual cues from the input video, demonstrating that VideoEval-Pro poses a substantially more challenging and discriminative benchmark for long video understanding evaluation.

We also find that performance on existing long video benchmarks tends to saturate or even decline as the number of input frames increases. As illustrated in Figure 9.3b, all models achieve their highest accuracy on Video-MME with 256 input frames, but performance begins to plateau or drop when the input is extended to 512 frames. This is a counterintuitive finding, as one would expect that providing more input frames would supply additional contextual information that models could leverage to improve performance. On the other hand, the five tested models exhibit a consistent improvement in accuracy on VideoEval-Pro as the number of input frames increases. This divergence suggests that VideoEval-Pro is a more robust benchmark in assessing long video tasks, and offers a more faithful evaluation of a model's ability to integrate and reason over longer video contexts.

9.3.4 Qualitative Analysis

We conduct a qualitative analysis using results from Gemini-2.0-Flash to better understand the challenges posed by our VideoEval-Pro. We identify several interesting cases where the model selects the correct answer in the MCQ setting but fails to produce accurate factual details in the free-form response. The results are shown in Figure 9.4.

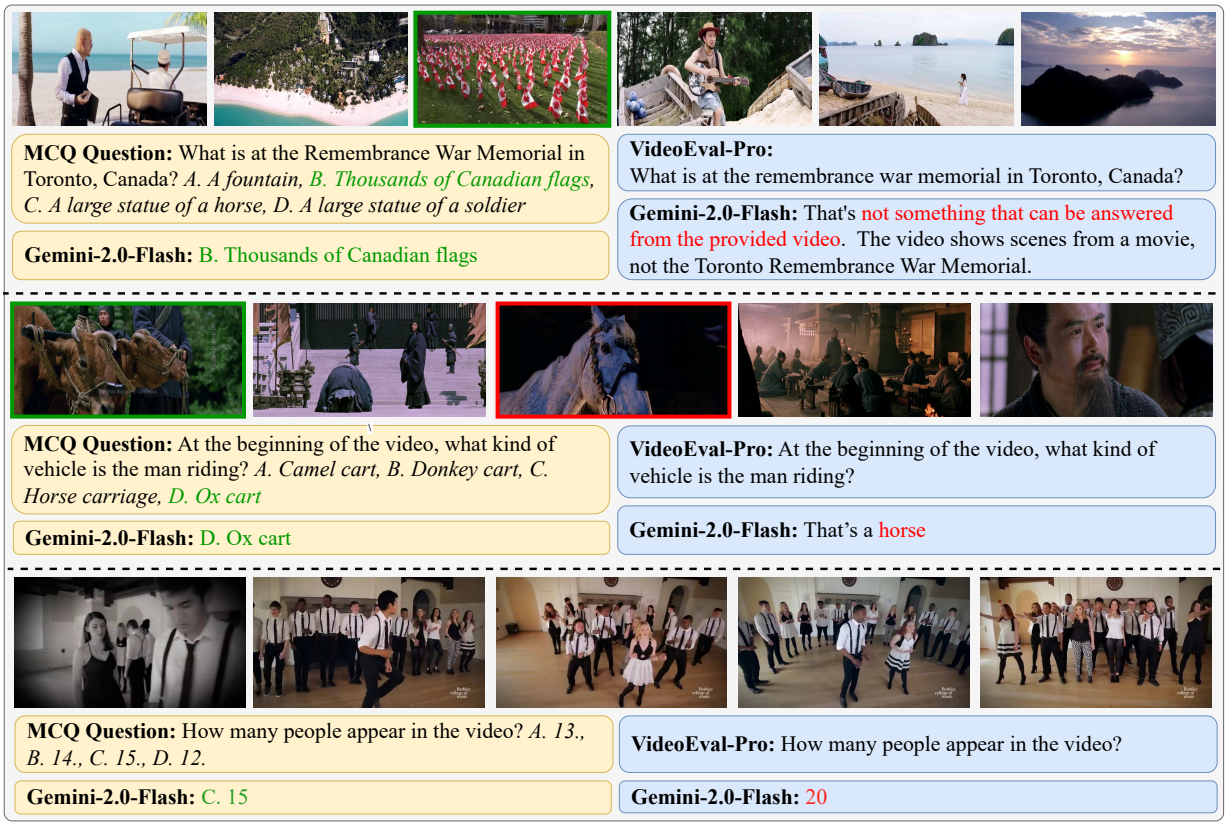


Figure 9.4: Qualitative comparisons between VideoEval-Pro and the corresponding MCQ problems.

In the first example, the question asks about the appearance of the Toronto Remembrance War Memorial. While Gemini correctly selects the answer "Thousands of Canadian flags" in the multiple-choice (MCQ) format, it fails to produce the correct response in the open-ended setting. This suggests that when MCQ options are available, the model may rely on common knowledge (Toronto and Canada are associated), rather than engaging in detailed video analysis. In the second example, although the model correctly identifies the option "Ox cart" in the MCQ format, it incorrectly describes the content as "That's a horse" in its open-ended response. This indicates that fine-grained visual recognition in long videos remains a significant challenge for LMMs, and MCQ options may provide cues that help the model circumvent this difficulty. Similarly, in the third example where the question asks how many people appear in the video, the model correctly selects "15" in the MCQ format but responds with "20" in the open-ended version. This discrepancy suggests that the correct MCQ answer may have been chosen through guesswork or elimination

strategies rather than precise analysis of the video content.

9.4 Conclusion

In this chapter, we introduced VideoEval-Pro, a robust and realistic LVU benchmark designed to faithfully evaluate LMM's understanding and reasoning capabilities over long videos. Compared to existing LVU benchmarks, VideoEval-Pro reformulates MCQ problems into open-ended questions, preventing models from exploiting shortcuts inherent in the options and reducing performance variations caused by the MCQ format. VideoEval-Pro also employs a rigorous data filtering pipeline to eliminate questions with strong priors that allow LMMs to answer based on common knowledge or stereotypical associations, without truly reading the video. By evaluating 21 proprietary and open-source models, we find that VideoEval-Pro poses significant challenges to current video LMMs, with the best-performing model GPT-4.1 achieving 40.8% accuracy. We also observe that, in contrast to other LVU benchmarks where model performance tends to saturate with an increasing number of input frames, performance on VideoEval-Pro consistently improves as more frames are provided. These observations demonstrate that our VideoEval-Pro is a more reliable benchmark to track the progress of long video understanding.

Part III

Unified Multimodal Models for Joint Visual Understanding and Generation

Chapter 10

Background and Roadmap

The third part of this thesis studies the *modality* axis: unifying visual understanding and generation within a single model. Historically, visual understanding and generation have been pursued by separate research communities based on distinct model architectures: causal large multimodal models for perception and reasoning, and diffusion or flow matching models for visual synthesis. This separation prevents the two capabilities from reinforcing each other, even though a model that deeply understands a scene should be better equipped to generate coherent renditions of it, while a model trained to synthesize complex visual entities may learn representations that benefit understanding. Even recent unified models often rely on separate visual representations for the two tasks, introducing format mismatches and conflicting inductive biases that limit effective unification. The chapters in this part address this challenge by building unified multimodal models around a single shared visual representation.

In Chapter 11, we present TUNA, a native unified multimodal model that constructs a single continuous visual representation by cascading a VAE encoder with a representation encoder: the VAE preserves the low-level structure needed for generation, while the representation encoder supplies the semantics needed for both tasks. The resulting unified visual representations are processed by a language model decoder that performs autoregressive text prediction for understanding and flow-matching-based denoising for generation. In Chapter 12, we present TUNA-2, which pushes the unification further by operating directly on pixel embeddings. TUNA-2 replaces the vision encoder modules such as the VAE and the representation encoder with simple patch embedding layers, and demonstrates that a fully pixel-native design can match or exceed models built from separately pretrained encoders. Together, these chapters present practical methods to unify visual understanding

and generation using a shared visual representation and a simplified model architecture, without sacrificing model performance.

The remainder of this chapter reviews the related work that situates these contributions, including large multimodal models, diffusion and flow matching models, unified multimodal models, and the visual representations that underpin them.

10.1 Large Multimodal Models

Large multimodal models (LMMs) generate text responses from multimodal inputs spanning images, videos, and text. Early LMMs such as Flamingo [4] and Idefics [134] introduced cross-attention layers to let visual and linguistic features interact. Modern LMMs generally follow the LLaVA paradigm [166], in which visual inputs are encoded by a vision encoder such as CLIP [212] and then concatenated with text tokens for joint processing by a language model decoder. Subsequent work scaled this recipe to larger models and datasets, developed stronger vision encoders capable of handling higher resolutions [135, 265, 15], and extended LMMs to interleaved image and video understanding [120, 190, 160, 339]. These models excel at visual understanding but do not generate visual content by themselves, which motivates further investigation of unified multimodal models.

10.2 Diffusion Generative Models

Diffusion generative models have become the de facto backbone of high-fidelity image [66, 17] and video [129, 260] synthesis. Large-scale visual generators typically apply diffusion in a continuous latent space defined by a learned VAE, following the latent diffusion paradigm [222], which offers better perceptual quality and sampling efficiency than autoregressive decoding of long sequences of discrete VQ-VAE tokens [258, 67]. Within diffusion, latent-space models [222, 207, 66] are generally preferred over pixel-space approaches [61, 225] because they reduce computational cost and let the denoiser focus on semantically meaningful structure rather than low-level pixel noise. Architecturally, diffusion backbones have evolved from convolutional U-Nets [223, 102] to diffusion transformers [206], and the learning objective has been generalized from Gaussian noise prediction and score matching [102, 235] to the more expressive rectified flow [173] and flow matching [164] formulations that our models adopt.

10.3 Unified Multimodal Models

Unified multimodal models (UMMs) aim to perform multimodal understanding and generation within a single framework, typically pairing autoregressive text generation with diffusion or flow-matching visual generation. One family of approaches connects understanding-only and generation-only models through learnable adapters, as in MetaQuery [205], BLIP-3o [37], and UniWorld-V1 [159]. While effective, these composite models inherit their capabilities from frozen task-specific components, limiting the synergy between understanding and generation. Native UMMs instead train both capabilities from scratch in a single architecture. Among them, the Janus series [287, 189, 47] adopts decoupled visual representations for the two tasks, and models such as BAGEL [59] and Mogao [157] further route the two representations through mixture-of-experts architectures to mitigate task conflict between understanding and generation. Alternatively, models like Chameleon [243], Transfusion [342], and the Show-o series [299, 300] employ a unified visual representation for both tasks. These models are more efficient but often exhibit imbalanced performance, excelling at one task while underperforming at the other. Our TUNA (Chapter 11) targets this imbalance by learning a single, balanced, continuous representation that performs strongly at both understanding and generation.

10.4 Visual Representations and Encoder-Free Models

A closely related line of work studies the visual representation that a unified model should operate on. To support both tasks with one tokenizer, TokenFlow [210] merges features from separate understanding and generation encoders, while UniTok [187] and TokLIP [161] train a single encoder to produce vector-quantized representations. However, discretized visual tokens limit visual generation fidelity. UniFlow [323] and UniLIP [242] instead adapt representation encoders into continuous unified tokenizers, but rely on relatively complex alignment schemes or a composite design. Separately, recent studies such as REPA [318] show that aligning generative models with semantic features improves generation quality, suggesting that semantics and generation are complementary. TUNA (Chapter 11) draws these threads together by learning a continuous unified representation end-to-end under joint understanding and generation objectives.

A further simplification removes the pretrained vision encoder entirely. While most LMMs pair a representation encoder with a language decoder, encoder-free models such

as Fuyu [18], EVE [62], Mono-InternVL [183], and NEO [63] tokenize raw pixels with simple patch embedding layers and process them jointly with text in a single transformer. Such designs have so far been applied mostly to visual understanding, whereas pixel-space visual generation has been considered difficult solve. Nevertheless, recent work on pixel-space flow matching [42, 148] suggests it can match the performance of latent diffusion, although these works usually only validate their methods at a relatively small scale, such as class-conditioned image generation on ImageNet [224]. Our TUNA-2 (Chapter 12) unifies these ideas, showing that an encoder-free, pixel-space model can be scaled to large-scale unified multimodal pretraining and support free-form text-to-image generation and image editing.

Chapter 11

Tuna: Taming Unified Visual Representations for Native Unified Multimodal Models

11.1 Introduction

A long-term aspiration of multimodal AI is *natively*[1] unified multimodal generation, in which a single model can seamlessly understand and generate diverse modalities such as text, images, and videos. Recent advances in unified multimodal models [243, 59, 300] have shown promising results towards this vision, suggesting that truly integrated multimodal intelligence is increasingly within reach.

A central challenge in developing native UMMs lies in how visual inputs are encoded into representations. Current UMMs adopt one of two approaches: (1) decoupled visual representations for understanding and generation tasks, or (2) a unified visual representation shared across both tasks. Intuitively, learning a unified visual representation for both tasks offers compelling advantages for UMMs.

First, UMMs with decoupled representations, such as BAGEL [59] and Mogao [157], often adopt MoE-style architectures to handle different visual encoders, introducing additional parameters that increase training and inference costs. In contrast, a unified representation allows the model to operate within a single representation space, simplifying

[1]We define *native* UMMs as models that are pretrained jointly on both understanding and generation objectives, instead of assembling understanding-only and generation-only models with learnable connectors. We refer to the latter as *composite* UMMs.

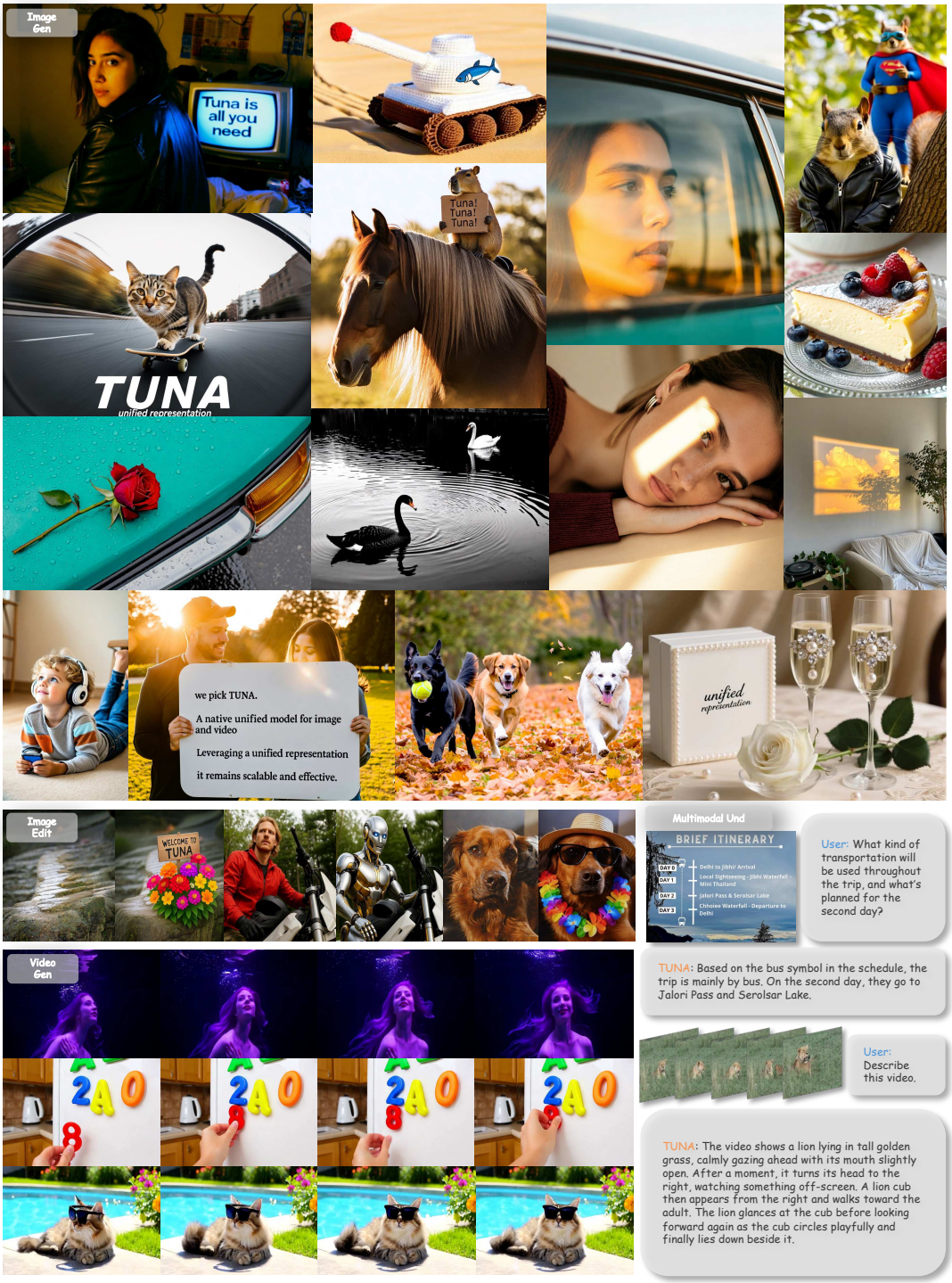


Figure 11.1: We present TUNA, a native unified multimodal model built on a unified visual representation, enabling diverse multimodal understanding and generation capabilities such as image and video understanding, image and video generation, and image editing.

training and improving efficiency. Second, different vision encoders typically produce representations with incompatible formats. For the same input, features from a representation encoder (*e.g.* SigLIP [327]) and a causal VAE encoder (*e.g.* Wan 2.1 VAE [260]) differ in (1) spatial compression (16× vs. 8×), (2) temporal compression (none vs. 4×), and (3) channel dimension (1152 vs. 16). These discrepancies can cause representation conflicts in decoupled models, whereas unified representations inherently avoid such inconsistencies. Finally, unified visual representations provide a clear pathway for achieving mutual enhancement between understanding and generation. While recent studies like Ross [261] and REPA [318] show task-specific improvements in understanding-only and generation-only models, this synergy remains underexplored in existing UMMs.

Nevertheless, current UMMs with unified visual representations often underperform their decoupled counterparts. Most existing approaches adopt a single type of vision encoder for both understanding and generation. For example, Chameleon [243] and Transfusion [342] use VQ-VAE [67], while Harmon [293] utilizes the MAR encoder [149]. This unified design tends to favor one task at the expense of the other. Show-o2 [300] attempts to mitigate this issue by fusing SigLIP [327] and VAE [260] features through a late-fusion strategy. However, our analysis in Section 11.3.4 reveals that its learned representation remains biased toward semantic features, resulting in limited generation quality.

To systematically address these limitations, we propose Tuna, a native UMM that employs unified visual representations across understanding and generation. Our design is simple yet highly effective: by directly connecting a VAE encoder to a representation encoder, we obtain representations that are sufficiently expressive for diverse multimodal tasks. These unified visual features are fused with text tokens and processed by an LLM decoder, which subsequently generates new text tokens and denoised images through autoregressive next-token prediction and flow matching. As illustrated in Figure 11.1, our unified visual representation enables Tuna to handle image and video understanding, image and video generation, and image editing within a single framework. By conducting a three-stage training, our model achieves state-of-the-art performance on multimodal understanding and generation benchmarks (*e.g.* 61.2% on MMStar [39] and 0.90 on GenEval [83]).

Our main contributions can be summarized as follows:

1. We propose Tuna, a native unified multimodal model with unified visual representations, enabling image/video understanding, image/video generation, and image editing within a single framework.

2. Our extensive experiments show that Tuna's unified visual representation is highly effective, achieving state-of-the-art performance across multiple multimodal understanding and generation tasks.

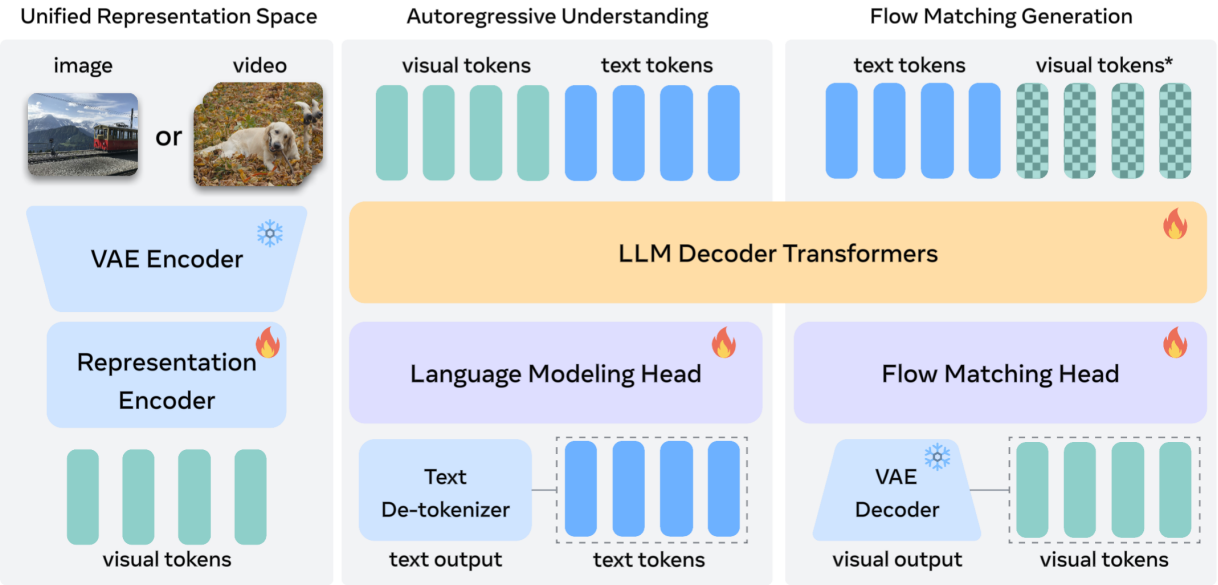


Figure 11.2: Overview of the TUNA architecture. Our model employs a VAE encoder and a representation encoder to construct unified visual representations, which are then combined with text tokens and processed by an LLM decoder. The decoder performs autoregressive text generation for understanding tasks and flow-matching-based visual generation for generation tasks. *During visual generation, noise is added to the visual tokens to enable diffusion-based generation.

3. We further perform a comprehensive ablation study, demonstrating the superiority of our unified visual representation design over existing methods such as Show-o2 and other models employing decoupled representations.

11.2 Our Method: Tuna

In this section, we introduce TUNA, a native unified multimodal model that employs unified visual representations across all multimodal understanding and generation tasks. We begin by outlining the key motivations behind our model design in Section 11.2.1, followed by a detailed description of TUNA's architecture and training pipeline in Sections 11.2.2 and 11.2.3, respectively. An overview of our overall framework is shown in Figure 11.2.

11.2.1 Motivation and Design Principles

We discuss the following observations, which motivate the design of TUNA and its unified visual representations:

Autoregressive vs. diffusion. Both text and image/video generation can be achieved using autoregressive [88, 310, 239] or diffusion models [196, 17, 260]. In practice, leading understanding-only models [15, 269] adopt autoregressive models for text generation. On the other hand, state-of-the-art image and video generators [66, 260] employ (latent[2]) diffusion models with flow matching.

Continuous vs. discrete visual representations. We observe that image and video generation models operating in a continuous (*e.g.*, KL-regularized) VAE latent space [66, 260] outperform those using discrete representations [239], as discretization causes information loss and reduces fidelity. Similarly, multimodal understanding models [15, 269] typically rely on continuous semantic features (*e.g.*, CLIP [212] features), suggesting that continuous visual representations are inherently more effective for both understanding and generation tasks.

Semantic representations benefit visual generation. Recent studies suggest that semantic features enhance visual generation. For instance, REPA [318] demonstrates that diffusion transformers benefit from aligning intermediate features with pretrained representation encoders like DINOv2 [202]. Concurrent to our work, RAE [340] employs a frozen representation encoder to encode images into latent representations, showing that pretrained semantic features alone can reconstruct input images effectively.

VAE latents can also support understanding tasks. We observe that both discrete and continuous VAE latents, originally designed for visual reconstruction, can also support semantic understanding tasks. Recent approaches such as UniTok [187] and TokLIP [161] enhance VQ-VAE latents with semantic understanding capability through contrastive learning. Other works explore diffusion models with continuous VAE latents for semantic understanding and dense prediction tasks, including semantic segmentation [348], object recognition [136], and image retrieval [349].

Building on these observations, we design TUNA with the following key characteristics:

- TUNA integrates **autoregressive** text generation with **flow matching** for image and video generation.

[2]Due to higher generation fidelity and lower training costs, latent diffusion models are generally favoured over pixel diffusion [340].

- TUNA builds its unified visual representation on continuous VAE latents, as these latents effectively support both understanding and generation tasks.
- To further enhance performance, TUNA employs **representation encoders** to extract higher-level features from the VAE latents, improving the quality of both understanding and generation.

11.2.2 Model Architecture

Unified visual representations. As illustrated in Figure 11.2, TUNA constructs its unified visual representation using a VAE encoder and a representation encoder. Given an input image or video $\mathbf{X}$, we apply the 3D causal VAE encoder from Wan 2.2 [247], which downsamples the input by 16× spatially and 4× temporally, producing the latent $\mathbf{x}_1$. We then generate a noisy latent $\mathbf{x}_t = t\mathbf{x}_1 + (1-t)\mathbf{x}_0$, where $t \in [0, 1]$ is a sampled timestep and $\mathbf{x}_0 \sim \mathcal{N}(0, 1)$. Next, we use the SigLIP 2 vision encoder Φ (patch size 16, pretrained resolution 512) to extract semantic features from the VAE latents. Since the VAE encoder has 16× downsampling, we replace SigLIP 2's original 16×16 patch embedding layer with a randomly initialized 1×1 patch embedding layer, forming a modified encoder Φ'. This ensures that the token sequence lengths of $\Phi(\mathbf{X})$ and $\Phi'(\mathbf{x}_t)$ are consistent. Finally, we apply a two-layer MLP connector to obtain the unified visual representations $\mathbf{z} = \texttt{MLP}(\Phi'(\mathbf{x}_t))$. During training, we randomly sample t between $[0, 1]$ for visual generation and fix $t = 1$ for multimodal understanding such that $\mathbf{x}_t$ always corresponds to the clean latent.

For video inputs, where $\mathbf{x}_t \in \mathbb{R}^{b\times c\times f\times h\times w}$ (with b as batch size, f as the number of latent frames, and c, h, w as channel, height, and width), we aim to prevent the representation encoder Φ' from processing excessively long sequences. Instead of flattening all latent frames into a single sequence, we apply a window-based attention mechanism by reshaping the frame dimension into the batch dimension in Φ'. In `einops` notation, the unified visual representation $\mathbf{z}_v$ can be expressed as:

$$\bar{\mathbf{x}}_t = \texttt{rearrange}(\mathbf{x}_t, \texttt{b c f h w} \to (\texttt{b f})\ \texttt{c h w}), \tag{11.1}$$

$$\bar{\mathbf{z}}_v = \texttt{MLP}(\Phi'(\bar{\mathbf{x}}_t)) \in \mathbb{R}^{(b\times f)\times d}, \tag{11.2}$$

$$\mathbf{z}_v = \texttt{rearrange}(\bar{\mathbf{z}}_v, (\texttt{b f})\ \texttt{d} \to \texttt{b}\ (\texttt{f d})), \tag{11.3}$$

where d is the hidden dimension of the video tokens. This operation effectively allows Φ' to operate independently on each 4-frame window, significantly improving efficiency when processing video tokens.

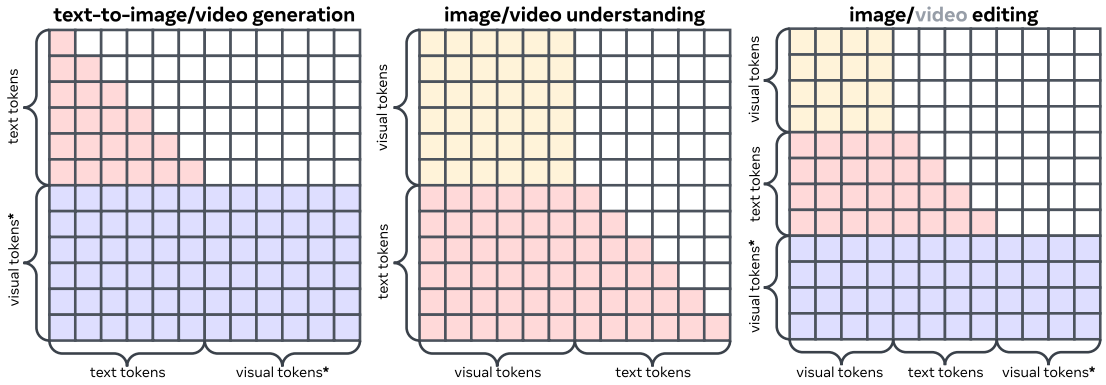


Figure 11.3: Attention masks in the LLM decoder for understanding and generation tasks. * indicates that the visual tokens are noised.

LLM decoder and flow matching head. After obtaining the unified visual representation $\mathbf{z}$, we prepend a timestep token representing the sampled timestep t to $\mathbf{z}$, concatenate this visual token sequence with language tokens and feed the combined sequence into an LLM decoder (Qwen-2.5 [15]) for joint multimodal processing. Following standard UMM practices [299, 59], we apply a causal attention mask on language tokens and a bidirectional attention mask on visual tokens within the LLM decoder layers, as illustrated in Figure 11.3. For multimodal understanding tasks, the LLM decoder output is passed through a language modeling head to generate text token predictions. For visual generation and image editing, we feed the full token sequence to a randomly initialized flow matching head to predict the velocity for flow matching. This head shares the LLM decoder architecture and adds timestep conditioning via AdaLN-Zero, following Show-o2 [300] and DiT [206]. For generation and editing tasks, we adopt multimodal 3D-RoPE [227, 238] over the concatenated text-visual sequence to handle interleaved instructions and visual content.

11.2.3 Training Pipeline

To effectively train our unified model, we adopt a three-stage training strategy that progressively adapts each model component to both understanding and generation tasks.

Stage 1: unified representation and flow matching head pretraining. In the first training stage, our objective is to adapt the semantic representation encoder to generate unified visual representations and to establish a robust initialization for the flow matching head. To this end, we train the representation encoder and flow matching head while

freezing the LLM decoder, using two objectives: image captioning and text-to-image generation.

Our image captioning objective aligns with the pretraining objectives of strong semantic encoders, such as SigLIP 2 [253] and the Qwen2.5-VL [15] vision encoder. Image captioning has also been shown to provide semantic richness comparable to contrastive learning [254], thereby enhancing our unified representation's visual understanding capability. Meanwhile, the text-to-image generation objective trains the flow matching head to generate images from text conditions, laying the groundwork for later image editing and text-to-video generation tasks. Additionally, this objective allows generation gradients to flow back into the representation encoder, further aligning our unified visual representation with both understanding and generation tasks.

Stage 2: full model continue pretraining. In the second training stage, we unfreeze the LLM decoder and pretrain the entire model using the same image captioning and text-to-image generation objectives from Stage 1. During later training steps of Stage 2, we further introduce image instruction-following, image editing, and video-captioning datasets to extend the model's capabilities. This stage enables TUNA to perform more complex multimodal reasoning and generation tasks, bridging the gap between basic visual-text alignment and higher-level instruction-driven multimodal understanding and generation.

Stage 3: supervised finetuning (SFT). Finally, in the third stage, we conduct supervised fine-tuning (SFT) using a combination of image editing, image/video instruction-following, and high-quality image/video generation datasets, trained with a reduced learning rate. This stage further refines TUNA 's capabilities, improving its performance and generalization across diverse multimodal understanding and generation tasks.

11.3 Experiments

11.3.1 Experiment Setup

Implementation details. We verify TUNA with two LLM models at different scales, *i.e.*, Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct [15]. In the pretraining stage, we optimize the representation encoder, projection layers, and diffusion head with AdamW [182] using a learning rate of 1×10^{-4}. We train on images with a base resolution of 512×512, as well as alternative aspect ratios that yield a similar number of visual tokens. In the second stage, we enable end-to-end training after a linear warm-up of 2K steps and continue optimization

Table 11.1: Comparisons between TUNA and baseline models on multimodal understanding benchmarks. Results with model size greater than 13B are grayed. **Bold**: best results among each section. Underline: second-best. * indicates the results based on our evaluation scripts.

Models	Size	MME	GQA	RealWorldQA	SEED	MMMU	MMStar	AI2D	ChartQA	OCRBench
		perception	test-dev	test	image	val	avg	test	test	test
Understanding-only Models (LMMs)										
LLaVA-1.5 [166]	7B	1510.7	62.0	54.8	65.8	35.7	33.1	55.5	17.8	31.8
Qwen-VL-Chat [13]	7B	1487.6	57.5	49.3	64.8	37.0	34.5	57.7	49.8	48.8
LLaVA-OV [141]	7B	1580.0	-	69.9	76.7	48.8	57.5	81.4	80.9	62.2
Composite UMMs										
TokenFlow-XL [210]	14B	1551.1	62.5	56.6	72.6	43.2	-	-	-	-
BLIP3-o [37]	4B	1527.7	-	60.4	73.8	46.6	-	-	-	-
Tar [96]	7B	1571.0	61.3	-	73.0	39.0	-	-	-	-
X-Omni [80]	7B	-	62.8	62.6	74.3	47.2	-	76.8	81.5	70.4
1.5B-scale Native UMMs										
Show-o [299]	1.3B	1097.2	58.0	-	51.5	27.4	-	-	-	-
Harmon [293]	1.5B	1155.0	58.9	49.8*	67.1	38.9	35.3*	57.0*	29.8*	11.2*
JanusFlow [189]	1.3B	1333.1	60.3	41.2*	**70.5**	29.3	40.6*	54.2	42.4*	53.2*
SynerGen-VL [143]	2.4B	1381.0	-	-	-	34.2	-	-	-	-
Janus-Pro [47]	1.5B	1444.0	59.3	52.6*	68.3	36.3	43.1*	64.5*	23.4	48.7
Show-o2 [300]	1.5B	1450.9	60.0	56.5*	65.6	37.1	43.4	69.0	40.0*	24.5*
TUNA	1.5B	**1461.5**	**61.4**	**62.5**	69.3	**39.1**	**54.6**	**71.4**	**82.1**	**71.9**
7B-scale Native UMMs										
BAGEL [59]	14B	1687.0	-	72.8	78.5	55.3	-	89.2	78.5	73.3
Emu3 [274]	8B	-	60.3	57.4	68.2	31.6	-	70.0	-	68.7
VILA-U [295]	7B	1401.8	60.8	-	59.0	-	-	-	-	-
MUSE-VL [301]	7B	-	-	-	69.1	39.7	49.6	69.8	-	-
Janus-Pro [47]	7B	1567.1	62.0	58.0*	72.1	41.0	48.3*	71.3*	25.8	59.0
Mogao [157]	7B	1592.0	60.9	-	74.6	44.2	-	-	-	-
Show-o2 [300]	7B	1620.5	63.1	64.7*	69.8	48.9	56.6	78.6	52.3*	32.4*
TUNA	7B	**1641.5**	**63.9**	**66.1**	**74.7**	**49.8**	**61.2**	**79.3**	**85.8**	**74.3**

with the same learning rate. At this point, we extend the training data to include video-caption pairs and editing data. In the final stage, we perform supervised fine-tuning for instruction following on our curated SFT corpus with a smaller learning rate of 2×10^{-5}. Due to the substantial computational cost of video training, the 7B variant is trained without video data.

11.3.2 Main Results

Image understanding. We evaluate TUNA's multimodal understanding capabilities on nine benchmarks, including general VQA benchmarks such as MME [73], GQA [114], RealWorldQA [296] and SEED-Bench [139]; knowledge-intensive benchmarks such as MMMU [322], MMStar [39], and AI2D [126]; and text-centric benchmarks including ChartQA [191] and OCRBench [176]. As shown in Table 11.1, both 1.5B and 7B TUNA achieve state-of-

Table 11.2: Image generation results on GenEval. † refers to methods using LLM rewriters. **Bold**: best results among each section. Underline: second-best.

Models	Size	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attr.	Overall
Generation-only Models								
SD3-Medium [66]	2B	0.99	0.94	0.72	0.89	0.33	0.60	0.74
FLUX.1 [Dev]† [17]	12B	0.98	0.93	0.75	0.93	0.68	0.65	0.82
Composite UMMs								
MetaQuery-XL† [205]	7B	-	-	-	-	-	-	0.80
Tar [96]	7B	0.99	0.92	0.83	0.85	0.80	0.65	0.84
BLIP3-o [37]	8B	-	-	-	-	-	-	0.84
UniWorld-V1† [159]	12B	0.98	0.93	0.81	0.89	0.74	0.71	0.84
OmniGen2† [288]	7B	0.99	0.96	0.74	0.98	0.71	0.75	0.86
1.5B-scale Native UMMs								
D-DiT [155]	2B	0.97	0.80	0.54	0.76	0.32	0.50	0.65
Show-o [299]	1.5B	0.98	0.80	0.66	0.84	0.31	0.50	0.68
Janus-Pro [47]	1.5B	0.98	0.82	0.51	0.89	0.65	0.56	0.73
Show-o2 [300]	1.5B	0.99	0.86	0.55	0.86	0.46	0.63	0.73
Harmon [293]	1.5B	0.99	0.86	0.66	0.85	0.74	0.48	0.76
TUNA	1.5B	**1.00**	**0.94**	**0.83**	**0.91**	**0.81**	**0.79**	**0.88**
7B-scale Native UMMs								
MUSE-VL [302]	7B	-	-	-	-	-	-	0.57
Transfusion [342]	7B	-	-	-	-	-	-	0.63
Emu3 [274]	8B	-	-	-	-	-	-	0.66
Show-o2 [300]	7B	**1.00**	0.87	0.58	0.92	0.52	0.62	0.76
Janus-Pro [47]	7B	0.99	0.89	0.59	0.90	0.79	0.66	0.80
BAGEL† [59]	14B	0.98	0.95	**0.84**	**0.95**	0.78	0.77	0.88
Mogao [157]	7B	**1.00**	**0.97**	0.83	0.93	0.84	0.80	0.89
TUNA	7B	**1.00**	**0.97**	0.81	0.91	**0.88**	**0.83**	**0.90**

the-art results across nearly all benchmarks, demonstrating strong and consistent performance. Notably, TUNA delivers competitive image understanding results compared to understanding-only models and outperforms many composite UMMs and UMMs with larger model sizes, highlighting the effectiveness of its unified representations.

Image generation. We evaluate TUNA's image generation performance on three benchmarks: GenEval [83], DPG-Bench [110] and OneIG-Bench [33]. Results are presented in Table 11.2 and Table 11.3. Across all three benchmarks, TUNA consistently outperforms contemporary approaches such as Janus-Pro, BAGEL and Mogao, achieving state-of-the-art results for both the 1.5B and 7B variants. Notably, TUNA shows a substantial advantage in text rendering quality in OneIG-Bench, indicating its strong semantic understanding capability when generating images from complex instructions containing visual text-related

Table 11.3: Image generation results on DPG-Bench. **Bold**: best results among each section. Underline: second-best.

Models	Size	DPG-Bench						OneIG-Bench					
		Global	Entity	Attribute	Relation	Other	Overall	Alignment	Text	Reasoning	Style	Diversity	Average
Generation-only Models													
FLUX.1 [Dev] [17]	12B	82.10	89.50	88.70	91.10	89.40	84.00	0.79	0.52	0.25	0.37	0.24	0.43
Qwen-Image [286]	20B	91.32	91.56	92.02	94.31	92.73	88.32	0.88	0.89	0.31	0.42	0.20	0.54
1.5B-scale Native UMMs													
Show-o [299]	1.3B	-	-	-	-	-	-	0.70	0.00	0.21	**0.36**	**0.24**	0.25
Show-o2 [300]	1.5B	87.53	**90.38**	91.34	90.30	**91.21**	85.02	0.80	0.13	**0.27**	0.35	0.19	0.35
TUNA	1.5B	**88.87**	90.32	**91.71**	**91.79**	90.14	**86.03**	**0.82**	**0.77**	0.25	**0.36**	0.20	**0.48**
7B-scale Native UMMs													
Emu3-DPO [274]	8B	-	-	-	-	-	81.60	-	-	-	-	-	-
Janus-Pro [47]	7B	86.90	88.90	89.40	89.32	89.48	84.19	0.55	0.00	0.14	0.28	**0.37**	0.27
Mogao [157]	7B	82.37	90.03	88.26	**93.18**	85.40	84.33	-	-	-	-	-	-
BAGEL [59]	14B	88.94	90.37	**91.29**	90.82	88.67	85.07	0.77	0.24	0.17	0.37	0.25	0.36
Show-o2 [300]	7B	89.00	**91.78**	89.96	91.81	**91.64**	86.14	0.82	0.00	0.23	0.32	0.18	0.31
TUNA	7B	**90.42**	91.68	90.94	91.87	90.73	**86.76**	**0.84**	**0.82**	**0.27**	**0.40**	0.19	**0.50**

Table 11.4: Image editing results on ImgEdit-Bench and GEdit-Bench. For ImgEdit-Bench, we test editing performance across various dimensions, including 'Add', 'Adjust', 'Extract', 'Replace', 'Remove', 'Background', 'Style', 'Hybrid', and 'Action'. For GEdit-Bench, "G-SC" and "G-PQ" denote "G-Semantic Consistency" and "G-Perceptual Quality", respectively. **Bold**: best results among each section. Underline: second-best.

Models	Size	ImgEdit-Bench										GEdit-Bench		
		Add	Adj.	Ext.	Rep.	Rm.	Bg.	Sty.	Hyb.	Act.	Overall	G-SC	G-PQ	G-Overall
Generation-only Models														
FLUX.1 Kontext [Pro] [17]	12B	4.25	4.15	2.35	4.56	3.57	4.26	4.57	3.68	4.63	4.00	7.02	7.60	6.56
Qwen-Image [286]	20B	4.38	4.16	3.43	4.66	4.14	4.38	4.81	3.82	4.69	4.27	8.00	7.86	7.56
Native or Composite UMMs														
OmniGen [298]	3.8B	3.47	3.04	1.71	2.94	2.43	3.21	4.19	2.24	3.38	2.96	5.96	5.89	5.06
BAGEL [59]	14B	3.56	3.31	1.70	3.30	2.62	3.24	4.49	2.38	4.17	3.20	7.36	6.83	6.52
UniWorld-V1 [159]	12B	3.82	3.64	2.27	3.47	3.24	2.99	4.21	2.96	2.74	3.26	4.93	7.43	4.85
OmniGen2 [288]	4B	3.57	3.06	1.77	3.74	3.20	3.57	**4.81**	2.52	4.68	3.44	7.16	6.77	6.41
TUNA	7B	**4.46**	**4.52**	**2.47**	**4.68**	**4.58**	**4.56**	4.73	**4.07**	**4.69**	**4.31**	**7.79**	**7.48**	**7.29**

information. Our results show that TUNA consistently outperforms models with decoupled visual representations on image generation tasks, underscoring the strength and robustness of its unified representation design.

Image editing. We employ ImgEdit-Bench [316] and GEdit-Bench as our evaluation suite for image editing. As shown in Table 11.4, TUNA achieves an overall score of 4.31 on ImgEdit-Bench, ranking highest among all UMMs. TUNA's performance is also comparable to generation-only models such as FLUX.1 Kontext [17] and Qwen-Image [286]. For GEdit-Bench, although TUNA performs slightly below the best generation-only model (Qwen-

Table 11.5: Experimental results on video understanding benchmarks. #Frames denotes the number of frames used during inference. **Bold**: best results. Underline: second-best.

Models	Size	#Frames	MVBench	Video-MME	LongVideoBench	LVBench
			test	w/o sub	val	test
Understanding-only Models (LMMs)						
GPT-4o [198]	-	-	-	71.9	66.7	48.9
Gemini-1.5-Pro [245]	-	-	54.2	75.0	64.0	33.1
LongVA [334]	7B	64	49.2	52.6	51.8	-
VideoLLaMA2 [53]	7B	16	54.6	47.9	-	-
LLaVA-OV [141]	7B	32	56.7	58.2	56.5	26.9
1.5B-scale Native UMMs						
Show-o2 [300]	1.5B	32	49.8	48.0	49.2	-
TUNA	1.5B	49	**54.4**	**49.1**	**49.7**	**27.4**

Image [286]), it again achieves the highest overall score among all unified models. TUNA's consistently strong results on both ImgEdit-Bench and GEdit-Bench demonstrate its robust image editing capability and highlight the effectiveness of our unified visual representation when handling visual generation tasks that require precise semantic understanding and accurate prompt following.

Video understanding. We employ four video understanding benchmarks to evaluate TUNA: MVBench [147], Video-MME [75], LongVideoBench [289] and LVBench [267]. As shown in Table 11.5, TUNA outperforms Show-o2 on MVBench and Video-MME, while achieving competitive results on LongVideoBench and LVBench. Notably, despite being only a 1.5B-parameter model, TUNA performs on par with larger understanding-only models on MVBench and LVBench, demonstrating the efficiency and effectiveness of our unified representation for video understanding tasks.

Video generation. We evaluate TUNA on VBench [112] for text-to-video generation, comparing it against other UMMs and generation-only models. As shown in Table 11.6, TUNA achieves state-of-the-art performance, surpassing all existing UMMs capable of video generation, while using only a 1.5B-parameter LLM decoder. This demonstrates the efficiency and scalability of our unified architecture for high-quality video generation.

Table 11.6: Video generation results on VBench. Full column names: QS: Quality Score, SS: Semantic Score, SC: Subject Consistency, BC: Background Consistency, TF: Temporal Flickering, MS: Motion Smoothness, DD: Dynamic Degree, AQ: Aesthetic Quality, IQ: Imaging Quality, OC: Object Class, MO: Multiple Objects, HA: Human Action, C: Color, SR: Spatial Relationship, S: Scene, AS: Appearance style, TS: Temporal Style, OC': Overall Consistency. **Bold**: best results among each section. Underline: second-best.

Models	Size	QS	SS	SC	BC	TF	MS	DD	AQ	IQ	OC	MO	HA	C	SR	S	AS	TS	OC'	Total
Generation-only Models																				
CogVideoX	5B	82.75	77.04	96.23	96.52	98.66	96.92	70.97	61.98	62.90	85.23	62.11	99.40	82.81	66.35	53.20	24.91	25.38	27.59	81.61
Native or Composite UMMs																				
VILA-U	7B	76.26	65.04	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	74.01
HaploOmni	7B	-	-	96.40	97.60	-	96.80	65.30	-	-	-	-	-	-	-	34.60	-	-	-	78.10
Emu3	8B	-	-	95.32	**97.69**	-	**98.93**	**79.27**	59.64	-	86.17	44.64	77.71	-	68.73	37.11	20.92	-	-	80.96
Show-o2	1.5B	82.10	78.31	**97.28**	96.78	97.68	98.25	40.83	65.15	**67.06**	94.81	76.01	95.20	80.89	62.61	57.67	**23.29**	**25.27**	27.00	81.34
TUNA	1.5B	**84.32**	**83.04**	95.99	96.72	**98.02**	98.33	69.39	**65.88**	66.83	**95.41**	**92.31**	**97.50**	**87.67**	**78.12**	**58.59**	23.18	24.68	**27.71**	**84.06**

11.3.3 Ablation: Visual Representation Design

In this section, we conduct a series of ablation experiments to systematically assess the effectiveness of our model architecture and training pipeline. For all experiments, we use a lightweight variant of TUNA built on the Qwen2.5-1.5B LLM with a smaller flow matching head. Using this setup, we evaluate three visual representation designs:

1. Decoupled representations, using SigLIP 2 features for understanding and Wan 2.2 VAE latents for generation (denoted as "Decoupled").
2. Show-o2-style unified representations, using a dual-path late fusion strategy to obtain the final representations (denoted as "Show-o2"). A detailed explanation of this design can be found in Section 11.3.4.
3. TUNA's unified representation, initialized from three different pretrained representation encoders: SigLIP [327], SigLIP 2 [253], and DINOv3[3] [230].

All models are trained on a subset of our training data using a two-stage training pipeline (corresponding to Stage 1 and Stage 3 in Section 11.2.3), with an equal number of training steps per stage. Our ablation study results are shown in Table 11.7.

Unified vs. decoupled visual representation. Comparing Model 8 and Model 12 in Table 11.7, we observe that our unified representation consistently outperforms the decoupled setting across all understanding and generation benchmarks. Comparing Models 2 and

[3]Model code: `siglip-so400m-patch14-384`, `siglip2-so400m-patch16-512` and `dinov3-vith16plus-pretrain-lvd1689m`.

Table 11.7: Ablation study results. "Und. Only", "Gen. Only" and "Und. & Gen." refer to models trained with understanding data only, generation data only, and both data, respectively.

Models	ID	Data	Understanding				Generation	
			MME-p	MMMU	SEED	GQA	GenEval	DPG
Show-o2 (Wan 2.1 VAE + SigLIP)	1	Und. Only	1351	36.1	62.1	56.8	-	-
Decoupled (SigLIP 2 only)	2		1392	38.2	62.9	58.1	-	-
TUNA (Wan 2.2 VAE + SigLIP 2)	3		1386	37.6	62.9	57.4	-	-
Show-o2 (Wan 2.1 VAE + SigLIP)	4	Gen. Only	-	-	-	-	76.2	82.56
Decoupled (Wan 2.2 VAE only)	5		-	-	-	-	77.3	82.87
TUNA (Wan 2.2 VAE + SigLIP 2)	6		-	-	-	-	77.8	83.33
Show-o2 (Wan 2.1 VAE + SigLIP)	7	Und. & Gen.	1339	35.4	61.7	55.9	75.9	82.32
Decoupled (Wan 2.2 VAE + SigLIP 2)	8		1346	37.2	61.4	56.5	78.3	83.50
TUNA (Wan 2.1 VAE + SigLIP)	9		1358	35.9	64.2	57.2	77.2	83.29
TUNA (Wan 2.2 VAE + SigLIP)	10		1349	36.3	64.6	57.4	76.9	83.10
TUNA (Wan 2.1 VAE + SigLIP 2)	11		1379	37.7	65.9	58.4	79.1	83.98
TUNA (Wan 2.2 VAE + SigLIP 2)	12		1361	38.1	66.5	58.2	79.4	84.20
TUNA (Wan 2.2 VAE + DINOv3)	13		1396	37.3	65.6	58.6	78.9	84.08

5 with Model 8, we find that training a unified model using decoupled visual representations results in substantial degradation on understanding tasks, compared to only training the model on understanding data. In contrast, Model 12 surpasses Model 3 on most understanding benchmarks and outperforms Model 6 across all generation benchmarks. These results indicate that our unified representation suffers far less from representation conflicts than decoupled designs, enabling stronger performance in both understanding and generation.

Selection of representation encoders. We find that TUNA's unified representation generally benefits from stronger representation encoders. As shown in Table 11.7, comparing Models 10, 12, and 13, both SigLIP 2 (400M parameters) and DINOv3 (800M) outperform SigLIP (400M) on all benchmarks. Furthermore, comparing Model 9 to Model 11 and Model 10 to Model 12, we observe that regardless of which VAE encoder is used in the model, replacing SigLIP with SigLIP 2 in the representation encoder consistently improves performance across all understanding and generation benchmarks. We ultimately adopt SigLIP 2 for TUNA as it delivers comparable understanding performance, superior generation quality relative to DINOv3, and maintains a significantly smaller model size.

Understanding-generation synergy. Our experimental results on training models exclusively on either understanding (Models 1, 2 and 3) or generation data (Models 4, 5 and 6) demonstrate that TUNA benefits from joint training on both data types. Specifically,

we observe that Model 12 surpasses Model 3 on understanding benchmarks and Model 6 on generation benchmarks. Although the comparison between Model 2 and Model 3 shows that TUNA's VAE + representation encoder architecture incurs a slight performance drop relative to using only a representation encoder (the standard setup for understanding-only models), our joint understanding + generation training pipeline largely compensates for this degradation. Specifically, Model 12 recovers its understanding performance and becomes comparable to or even better than Model 2 on several understanding benchmarks. Moreover, Model 12 substantially outperforms both Model 5 and Model 6 on all generation benchmarks. These results demonstrate the mutual enhancement between understanding and generation made possible by our unified visual representation design.

Comparison with Show-o2. A closely related work to ours is Show-o2 [300], which uses a dual-path late-fusion mechanism, merging features from separate VAE and semantic encoders via a fusion layer to build unified representations. In contrast, TUNA extracts unified representations directly from VAE latents using a semantic encoder, achieving deep feature fusion across all layers in the semantic encoder. Comparing Model 7 with Models 9, 10, 11 and 12 in Table 11.7, our unified representation consistently outperforms Show-o2 on all benchmarks, regardless of the choice of the VAE encoder and the representation encoder. Model 1 vs. 3 and Model 4 vs. 6 further show that Show-o2 underperforms even when trained on a single task. We attribute this to its late-fusion strategy, which introduces representation conflicts and degrades overall performance.

11.3.4 Discussion: Unified Representation Analysis

As discussed in Section 11.3.3, both TUNA and Show-o2 [300] employ unified visual representations for understanding and generation, but they construct these representations in fundamentally different ways. In this section, we first describe Show-o2's unified visual representation design in detail, and then provide an in-depth analysis of why TUNA's unified representation achieves superior performance than Show-o2.

As illustrated in Figure 11.4, Show-o2 constructs unified visual representations using a dual-path feature fusion mechanism. The input image or video is first encoded by a VAE encoder, after which the latent is processed through two parallel branches. The semantic projection branch feeds the VAE latents into a set of semantic layers to extract features for understanding tasks. The VAE projection branch applies 2D patch embedding layers to produce features tailored for generation tasks. Importantly, the semantic layers are *pre-distilled* using a frozen representation encoder: given the same image, their outputs are first aligned with a pretrained SigLIP model before conducting end-to-end training of the

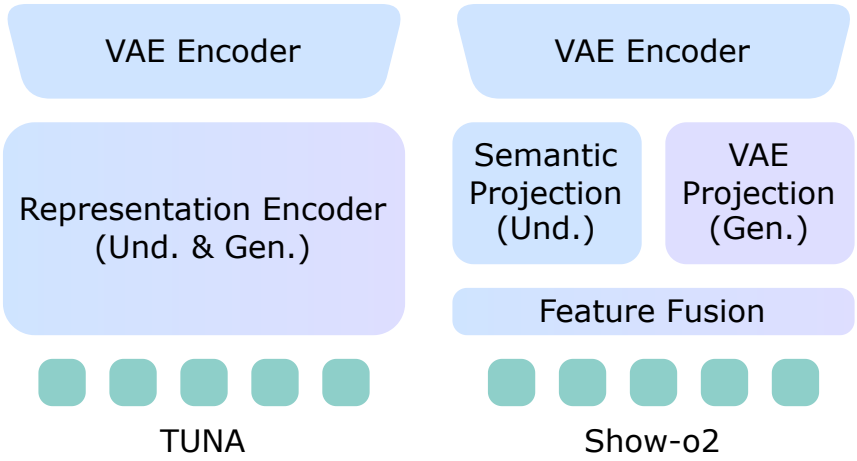


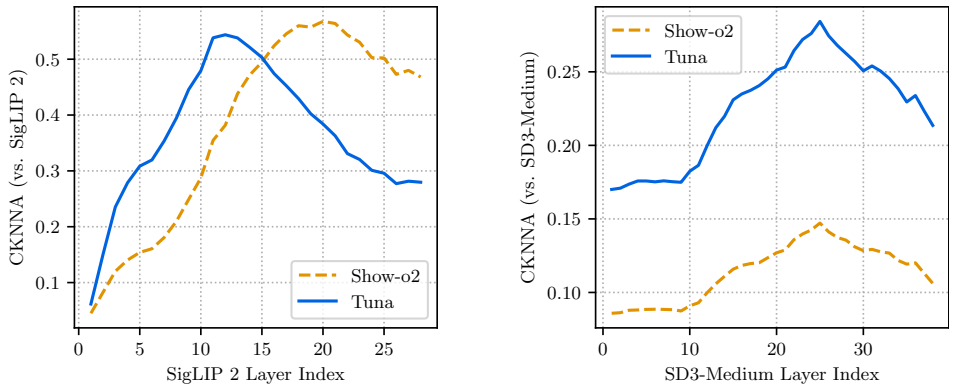
Figure 11.4: Comparison between TUNA and Show-o2 on how unified visual representations are produced.

Show-o2 model. This pre-distillation stage is proposed to preserve semantic understanding capability. Finally, Show-o2 merges the outputs of both paths using a feature fusion layer to obtain its unified visual representation.

To better understand why our unified representation yields superior performance, we perform a representation alignment analysis using CKNNA scores [115] with respect to two reference models: (1) a strong semantic encoder SigLIP 2 [253], and (2) a strong generation model SD3-Medium [66]. Concretely, we extract unified visual representations from TUNA and Show-o2 based on 1,024 images from the Wikipedia Captions dataset [237] and compute their CKNNA scores relative to features from all intermediate layers of the two reference models. The results are presented in Figure 11.5a and Figure 11.5b.

As shown in the figures, both TUNA and Show-o2 exhibit strong alignment with the SigLIP 2 intermediate features, with CKNNA scores exceeding 0.5. This high similarity reflects their strong semantic understanding capability, consistent with their solid performance on multimodal understanding tasks. On the other hand, TUNA's unified representation shows consistently higher alignment with the SD3-Medium intermediate features compared to Show-o2, indicating that TUNA learns a more balanced unified representation suited for both understanding and generation. In contrast, Show-o2 remains biased toward semantic features, which limits its generation quality.

The above findings prompt us to further investigate why Show-o2's dual-path fusion mechanism produces biased features toward semantic understanding. To analyze this, we compute CKNNA scores between Show-o2's final fused features and the intermediate



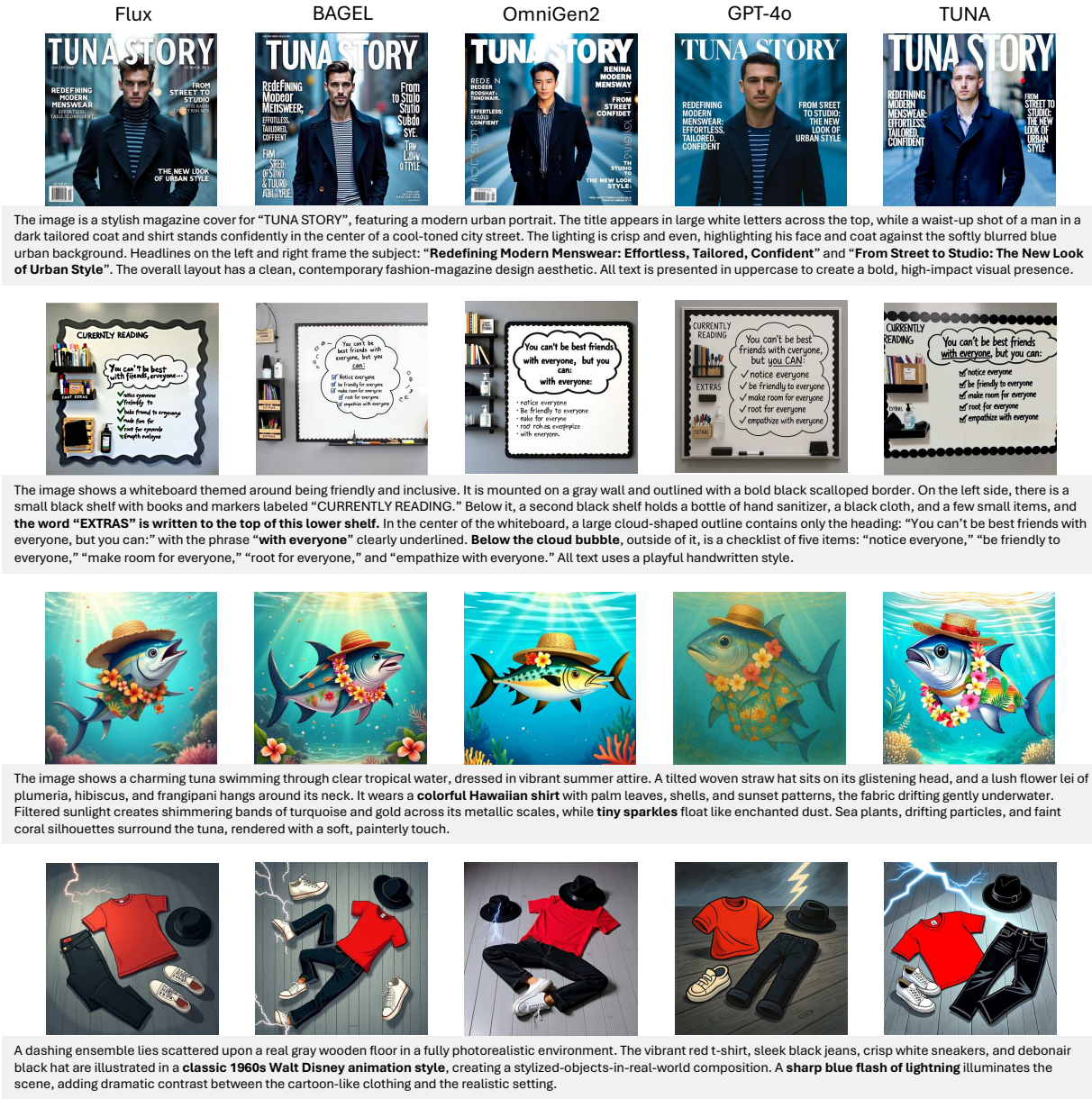
(a) Alignment to SigLIP 2 (b) Alignment to SD3-Medium

Figure 11.5: Representation alignment analysis with SigLIP 2 and SD3-Medium. For both TUNA and Show-o2, we extract visual representations at the input layer of the LLM decoder.

features from its understanding (semantic projection) and generation (VAE projection) branches before fusion. We find that Show-o2's unified representation exhibits a strong correlation with its understanding branch (CKNNA=0.45) but a very weak correlation with the generation branch (CKNNA=0.07). This demonstrates that the late-fusion strategy merges features in an imbalanced manner, causing the representation to remain dominated by semantic information. In contrast, TUNA's end-to-end training of the unified representation on both objectives enables early fusion of understanding and generation signals at every layer of the representation encoder. This layer-wise interaction captures richer cross-task dependencies and is inherently more robust than the late-fusion strategy adopted in Show-o2.

11.3.5 Qualitative Results

Image generation. We compare TUNA with state-of-the-art generation-only and unified models across diverse image generation instructions in Figure 11.6. In the first two examples, TUNA exhibits strong text rendering ability, accurately reproducing all visual text in the prompts without errors. In the whiteboard example, TUNA is the only model that correctly places an underline beneath "with everyone", demonstrating precise prompt-following capability. Moreover, TUNA accurately generates two black shelves, one containing books and markers on top and the other containing black cloth and hand sanitizer at the bottom, each in the correct position. Other models either fail to produce the correct number of shelves or place the wrong items on them. These results show that TUNA excels



The image is a stylish magazine cover for "TUNA STORY", featuring a modern urban portrait. The title appears in large white letters across the top, while a waist-up shot of a man in a dark tailored coat and shirt stands confidently in the center of a cool-toned city street. The lighting is crisp and even, highlighting his face and coat against the softly blurred blue urban background. Headlines on the left and right frame the subject: "**Redefining Modern Menswear: Effortless, Tailored, Confident**" and "**From Street to Studio: The New Look of Urban Style**". The overall layout has a clean, contemporary fashion-magazine design aesthetic. All text is presented in uppercase to create a bold, high-impact visual presence.

The image shows a whiteboard themed around being friendly and inclusive. It is mounted on a gray wall and outlined with a bold black scalloped border. On the left side, there is a small black shelf with books and markers labeled "CURRENTLY READING." Below it, a second black shelf holds a bottle of hand sanitizer, a black cloth, and a few small items, and **the word "EXTRAS" is written to the top of this lower shelf.** In the center of the whiteboard, a large cloud-shaped outline contains only the heading: "You can't be best friends with everyone, but you can:" with the phrase "**with everyone**" clearly underlined. **Below the cloud bubble**, outside of it, is a checklist of five items: "notice everyone," "be friendly to everyone," "make room for everyone," "root for everyone," and "empathize with everyone." All text uses a playful handwritten style.

The image shows a charming tuna swimming through clear tropical water, dressed in vibrant summer attire. A tilted woven straw hat sits on its glistening head, and a lush flower lei of plumeria, hibiscus, and frangipani hangs around its neck. It wears a **colorful Hawaiian shirt** with palm leaves, shells, and sunset patterns, the fabric drifting gently underwater. Filtered sunlight creates shimmering bands of turquoise and gold across its metallic scales, while **tiny sparkles** float like enchanted dust. Sea plants, drifting particles, and faint coral silhouettes surround the tuna, rendered with a soft, painterly touch.

A dashing ensemble lies scattered upon a real gray wooden floor in a fully photorealistic environment. The vibrant red t-shirt, sleek black jeans, crisp white sneakers, and debonair black hat are illustrated in a **classic 1960s Walt Disney animation style**, creating a stylized-objects-in-real-world composition. A **sharp blue flash of lightning** illuminates the scene, adding dramatic contrast between the cartoon-like clothing and the realistic setting.

Figure 11.6: Qualitative comparison between TUNA and baseline models on image generation tasks. The instructions that are correctly reflected in our results but failed in some of the baseline models are **bolded**.

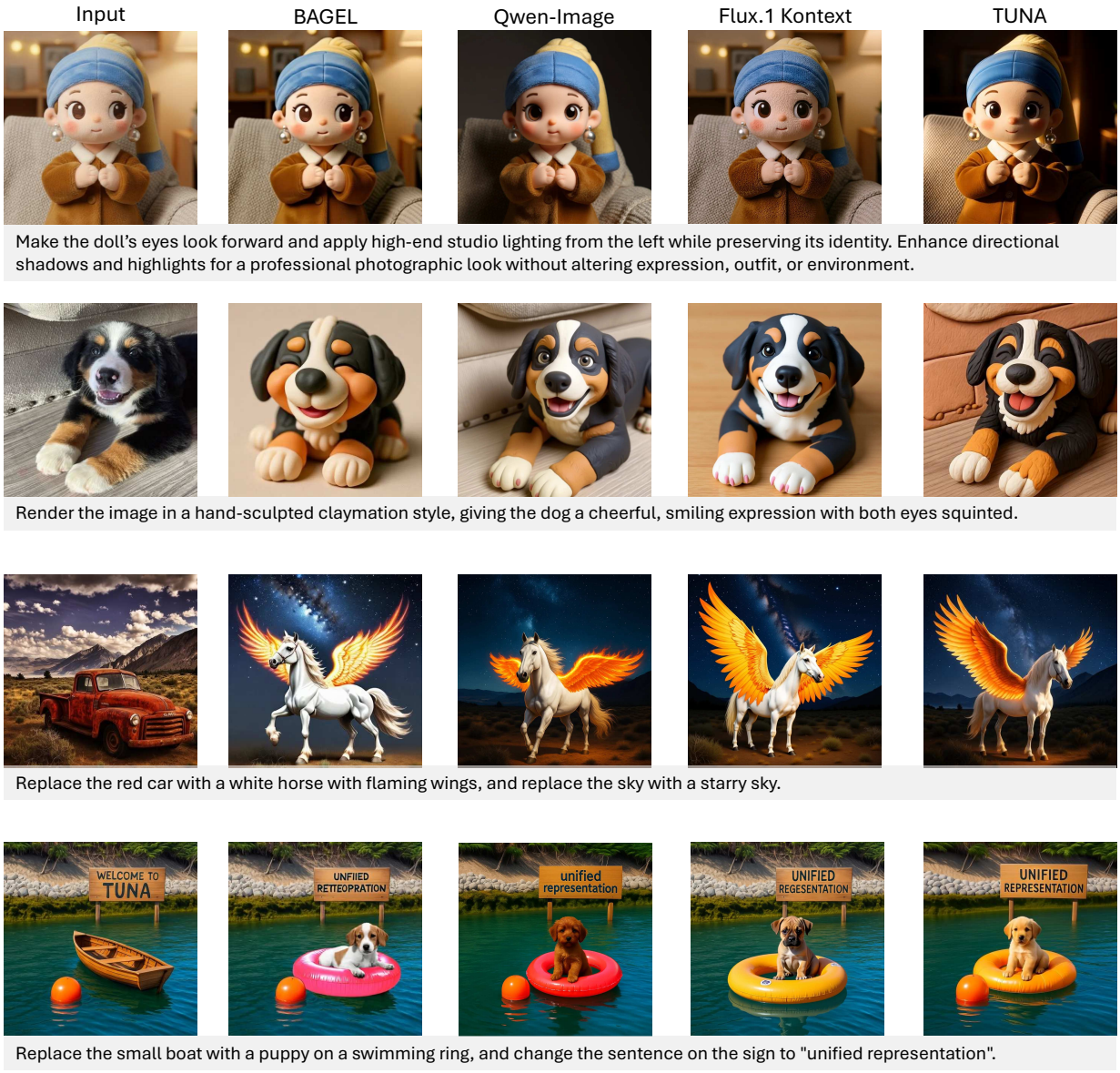


Figure 11.7: Qualitative comparison between TUNA and baseline models on image editing tasks.

at compositional image generation, enabled by its unified visual representation with strong semantic understanding capabilities. In the "tuna" example, both TUNA and Flux [17] successfully render the Hawaiian shirt, while other models either fail to depict the shirt or



A bird building a nest from twigs and leaves.



An elderly woman with short, curly white hair and a beige sweater sits on a gray couch, holding a light blue mug in her right hand. She rests her left arm on a green pillow, smiling as she looks off to the right. Behind her, a white shelf holds a glass vase with a gray candle, a stack of books, and a wicker basket. A large green plant stands to the left, and the background is a plain white wall. The woman's gaze shifts to the camera, and she raises the mug to her lips, taking a sip. She then sets the mug down on the couch beside her, her eyes never leaving the camera. The scene is well-lit, with soft, natural light illuminating the room.



A person stands at the edge of an infinity pool, arms outstretched, gazing out at a vibrant sunset over the ocean. The individual is silhouetted against the colorful sky, wearing a long-sleeved shirt and shorts. The pool's edge is barely visible, with ripples on the water's surface reflecting the sky's hues. In the background, a small island with palm trees and rocks meets the horizon, where the sun sets over the calm ocean waters. The sky transitions from blue to orange, yellow, and pink, with scattered clouds. The scene exudes a serene, peaceful atmosphere, capturing a moment of contemplation or joy at the natural beauty of the surroundings.



A majestic male lion with a thick, dark-brown mane and golden-brown fur lies on the ground, facing right. His front paws are stretched out in front of him, and his head is slightly raised, gazing into the distance. The lion's mane is long and shaggy, framing his face and neck. In the background, a tree trunk and some dry grass are visible, along with a large rock formation to the right. The ground is covered in light-brown dirt, and the overall atmosphere suggests a sunny day in a savannah or zoo enclosure. The lion's expression is calm and regal, exuding a sense of power and serenity.



The video shows a single red balloon drifting gently forward, its string hanging loosely as it floats at a steady, unhurried pace. The balloon rises and dips slightly with the movement of the air, shifting left or right in small, natural sways as it continues down the street. The scene unfolds along a old-town street treated as a full visual object rather than simple background: narrow cobblestone paths, tall stone buildings with ornate window frames, iron balconies, and warm-toned façades lining both sides. Soft afternoon light casts long shadows across café tables, bicycles, and shop signs, all subtly passing by as the balloon moves. The architecture's depth, the texture of the street, and the layered rows of buildings create a clear sense of space, framing the slow, floating journey of the red balloon through the historic European street.

Figure 11.8: Qualitative results for TUNA on the task of text-to-video generation.

generate an incorrect tuna body. Finally, in the "red t-shirt" example, TUNA accurately reflects the "classic 1960s Walt Disney animation style" and correctly includes all required elements from the prompt, maintaining a coherent and well-structured composition.

Image editing. We compare TUNA with BAGEL [59], Qwen-Image [286], and Flux.1 Kontext [17] on image editing tasks in Figure 11.7. As shown, TUNA not only correctly

performs explicit editing operations, such as style transfer (photorealistic → hand-sculpted claymation in the "dog" example), environment change (daylight → nighttime in the "red car" example), and object replacement (boat → puppy with a swim ring in the "boat" example), but also handles more implicit and nuanced instructions, such as applying lighting from the left side in the "doll" example. These results further highlight TUNA's strong semantic understanding and high-fidelity image generation capabilities.

Video generation. We present TUNA's video generation results in Figure 11.8. The model produces high-fidelity videos across a wide range of instructions, demonstrating the strength of its unified visual representation space for jointly modeling both images and videos.

11.4 Conclusion

We introduced TUNA, a native unified multimodal model that constructs a unified visual representation space by cascading a VAE encoder with a representation encoder. We train an LLM decoder and a flow matching head on this unified representation, achieving strong performance across image and video understanding, image and video generation, and image editing. TUNA not only surpasses prior UMM baselines but also performs competitively with leading understanding-only and generation-only models. Our ablation studies further show that (1) TUNA's unified representation space outperforms both Show-o2-style unified representations and decoupled representation designs, (2) stronger pretrained representation encoders consistently yield better performance within our framework, and (3) our unified visual representation design enables mutual enhancement between understanding and generation.

Chapter 12

Tuna-2: Pixel Embeddings Beat Vision Encoders for Multimodal Understanding and Generation

12.1 Introduction

Visual understanding and generation are two core capabilities in multimodal AI. Recent work has increasingly focused on native unified multimodal models (UMMs) [342, 59, 180], which aim to integrate both capabilities within a single framework. A central challenge in building such models is encoding input images into visual representations that effectively support both understanding and generation. Early approaches [59, 47] adopted decoupled representations, using representation encoders such as CLIP [212] for understanding and reconstruction-oriented encoders such as VQ-VAE [67] for generation. To address the representation mismatch introduced by this design, more recent UMMs [300, 180] have moved toward modelling both tasks using unified visual representations through a shared vision encoder.

Despite the significant progress, both decoupled and unified visual representation designs still rely heavily on pretrained vision encoders [260, 253] for visual feature extraction. In parallel, recent research on multimodal understanding and generation has begun to move away from encoder-based modular designs toward simpler monolithic, encoder-free architectures. In multimodal understanding, newer native vision-language models [63] remove the pretrained representation encoder and instead align images and natural language

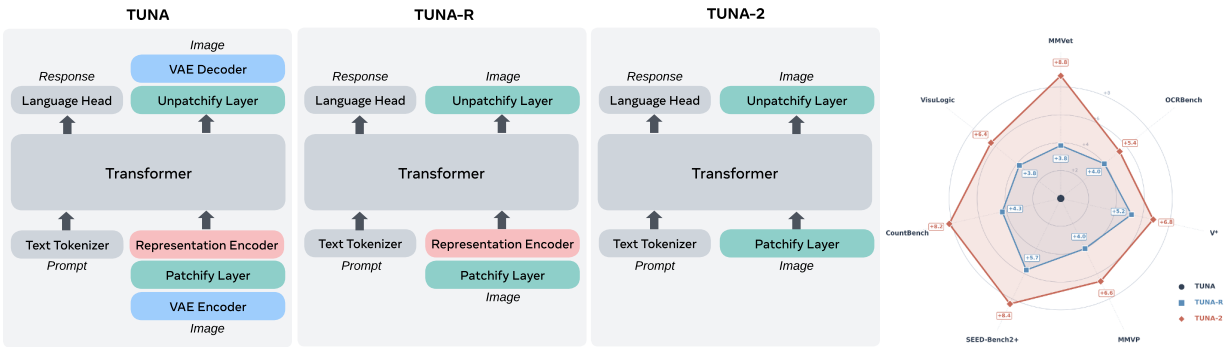


Figure 12.1: Evolution of TUNA-2 architecture and multimodal performance comparison. We simplify TUNA [180] by progressively stripping away its visual encoding components. By removing the VAE, we first derive TUNA-R, a pixel-space UMM that relies solely on a representation encoder. TUNA-2 further streamlines the design by bypassing the representation encoder entirely, utilizing direct patch embedding layers for raw image inputs. TUNA-2 using pixel embeddings outperforms both TUNA-R and TUNA across a diverse suite of multimodal benchmarks.

within a unified, end-to-end architecture. In visual generation, pixel-space diffusion models [107, 42, 148] have shown increasing flexibility, stronger scalability, and state-of-the-art performance on a wide range of tasks, suggesting that pretrained VAE encoders may no longer be essential even for high-fidelity image synthesis.

Motivated by these observations, we ask a natural but largely unexplored question: can we move beyond pretrained vision encoders altogether, and build unified multimodal models through end-to-end native modelling directly from raw pixels?

We answer this question affirmatively by introducing TUNA-2, a native unified multimodal model that attempts to progressively simplify the encoder modules, and ultimately remove vision encoders completely. We first introduce TUNA-R, which eliminates the VAE model while keeping a representation encoder in the model architecture. TUNA-R performs multimodal understanding similar to standard encoder-based LMMs, and supports visual generation through pixel-space flow matching with an x-prediction objective. We then propose TUNA-2, which further simplifies the architecture by removing the encoder entirely and using only a single transformer decoder to process image and video tokens. As a result, TUNA-2 enables end-to-end native unified modelling directly from raw pixels, without relying on any pretrained encoder modules.

Since learning unified representations directly in high-dimensional pixel space is sub-

stantially more challenging than learning them in a compact latent space, we further introduce a masking-based visual feature learning scheme to stabilize training and encourage the learning of more robust pixel-space representations. Together, these designs enable TUNA-2 to achieve state-of-the-art performance across a diverse set of multimodal understanding and generation benchmarks. More importantly, our controlled comparison reveals a clear design insight: after sufficient visual pretraining, the encoder-free TUNA-2 becomes competitive with the encoder-based TUNA-R on visual generation, while consistently outperforming it on multimodal understanding, especially on benchmarks that require fine-grained visual perception. These findings suggest that removing pretrained vision encoders can be advantageous for learning stronger fine-grained visual representations in end-to-end pretraining. As shown in Figures 12.1 and 12.2, this leads to highly competitive performances in both multimodal understanding and generation.

Our main contributions are summarized as follows:

1. We propose TUNA-2, a native unified multimodal model that supports multimodal understanding and generation with encoder-free designs, achieving state-of-the-art performance across a wide range of understanding and generation benchmarks.

2. We conduct controlled comparisons between TUNA-2 and an encoder-based pixel-space UMM variant TUNA-R, and show that after sufficient multimodal pretraining, TUNA-2 and its encoder-free design are competitive on generation and advantageous for understanding, especially on fine-grained, perception-intensive tasks.

3. We conduct comprehensive ablations and analyses on pixel-space UMMs to study their training dynamics and model behaviours, offering useful insights for the development of future native unified multimodal models.



Figure 12.2: While being completely encoder-free, TUNA-2 is capable of performing high-fidelity text-to-image generation and image editing.

12.2 Method

In this section, we present TUNA-2, a native unified multimodal model that performs visual understanding and generation both in pixel space. We start by detailing our approach to progressively remove vision encoder components to derive TUNA-2 in Section 12.2.1. We then describe our masked feature learning scheme in Section 12.2.2 and our model training pipeline in Section 12.2.3.

12.2.1 Towards Encoder-Free Unified Models

As shown in Figure 12.1, existing UMMs with unified visual representations, such as TUNA [180], typically consist of a vision encoder and an LLM decoder for joint vision-language modeling, followed by modality-specific heads, including a language modelling head for autoregressive text generation and a flow matching head for image generation. In this chapter, we propose TUNA-2 as an encoder-free UMM formulation by progressively simplifying the vision encoder components in existing architectures. Our design process for this architectural simplification is as follows:

Representation encoder-based architecture. First, we attempt to remove the VAE model and only employ a pretrained representation encoder in the vision encoder. As shown in Figure 12.1, this resonates a standard paradigm for vision-language modelling: the representation encoder first encodes input images into visual tokens, which are then combined with the text tokens in the LLM decoder for joint vision-language modelling. Originally proposed in LLaVA [166], this paradigm has been verified and scaled up by later works such as Qwen3-VL [14] and InternVL3.5 [269], and remains the most popular framework for multimodal understanding. We refer to this intermediate design as **Tuna-R**. Although our ultimate goal is to move beyond encoder-based architectures, we view TUNA-R as an important intermediate step that enables a controlled comparison with TUNA-2.

Encoder-free (non-encoder) architecture. Second, we consider a further simplified architecture that removes the representation encoder entirely, which becomes our main design for **Tuna-2**. As shown in Figure 12.1, this design replaces pretrained vision encoders with simple patch embedding layers that convert images into visual tokens, which are then processed jointly with text tokens by the LLM decoder. Similar encoder-free designs have recently been explored in models such as Mono-InternVL [183] and NEO [63]. By removing the pretrained representation encoder, this design avoids its built-in inductive biases, such as fixed input resolutions and limited access to fine-grained low-level visual details. It also

simplifies the model architecture into a single unified transformer. In Section 12.3, we present a series of in-depth analyses comparing TUNA-2 with TUNA-R, and demonstrate the effectiveness and scalability of TUNA-2.

Pixel-space image generation. Our VAE-free design allows us to directly perform multimodal understanding and text generation using the LLM decoder and the language modelling head. However, discarding the VAE also means that we can no longer adopt the designs from existing UMMs and generation-only models that follow the latent diffusion architecture. To effectively perform pixel-space image generation, we adopt the x-prediction and v-loss paradigm from JiT [148] for pixel-space flow matching. Specifically, given the source image x_1, the sampled noise $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the timestamp t, we employ rectified flow [173, 164] and its linear schedule to construct a noisy sample in pixel space:

$$x_t = tx_1 + (1-t)x_0, t \in [0, 1]. \tag{12.1}$$

TUNA-2 is then formulated to directly predict the clean image from the noisy image in pixel space:

$$x_\theta = \pi_\theta(x_t, c, t), \tag{12.2}$$

where π_θ is our unified model (vision-language backbone and flow matching head) and c is the conditioning signals (text for text-to-image generation and text+image for image editing). As suggested in JiT, while our model directly predicts x_θ, we still transform it into the velocity term v_θ and regress v_θ as our learning objective:

$$v_\theta = \frac{x_\theta - x_t}{1-t}, \tag{12.3}$$

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t,c,x_1,x_0}||v_\theta - v||_2^2, \tag{12.4}$$

where v is the ground truth velocity defined by $v = x_1 - x_0$. During inference, we employ the Euler solver and predict the denoised image at t' from the noisier image at $t < t'$ based on the velocity term v_θ, such that $x_{t'} = x_t + (t' - t)v_\theta$, where v_θ is transformed from our model prediction x_θ, based on Equation 12.3.

12.2.2 Learning Robust Visual Representations via Masking

While removing the VAE simplifies our model architecture and enables fully end-to-end unified multimodal training, it also shifts visual modelling from a compact latent space to the much higher-dimensional pixel space. As a result, learning a unified visual representation becomes more challenging: the increased redundancy in pixel-space inputs makes it

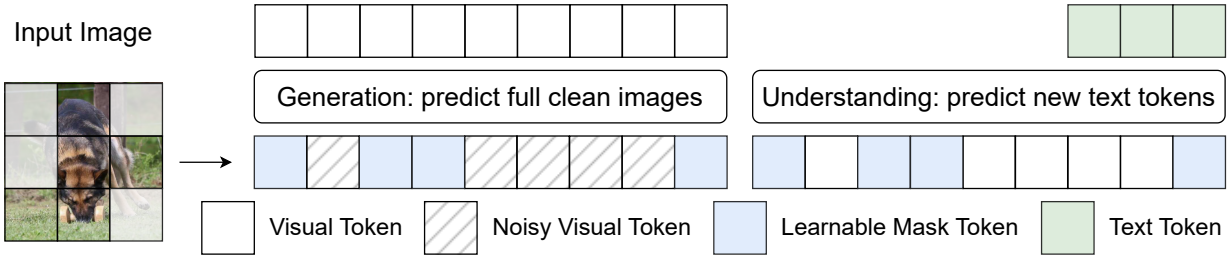


Figure 12.3: Illustration of our proposed masking-based feature learning scheme. During training, we use the learnable mask token to regularize multimodal understanding and perform masked prediction for visual generation.

easier for the model to rely on superficial shortcuts, rather than learning visual cues that are genuinely informative for both understanding and generation. To learn more robust visual representations in pixel space, we introduce a masking-based visual feature learning scheme. As shown in Figure 12.3, during training, we (optionally) randomly select a subset of image patches according to a masking ratio and replace the masked visual tokens with a learnable mask token before feeding them into the LLM decoder. The same masking operation is applied to both generation and understanding examples, but plays different roles in the two settings:

1. For generation examples, we let the model predict the clean image patches in *both the masked and the unmasked regions*, such that (1) we create a harder denoising problem for the model to predict clean images from partially observed noisy images; and (2) it encourages the learnable mask token to absorb useful information for image reconstruction conditioned on the visible context.

2. For understanding examples, our model predicts the ground truth text response based on the *masked visual input*. In this case, masking serves as a regularization mechanism that forces the model to perform multimodal reasoning under partial visual observation, leading to more robust visual representations.

Our masking-based feature learning scheme resembles masked modelling methods in visual understanding and generation, such as MAE [97] and SigLIP 2 [253] for semantic learning and MaskGIT [32] and DeTok [311] for visual generation. Empirically, we find that applying masking leads to enhanced model performance during pretraining stages.

12.2.3 Training Pipeline

Our encoder-free design enables fully end-to-end training of TUNA-2, without requiring separate stages to train connector layers, which is a common design in encoder-based modular approaches. As described below, our training pipeline consists of two stages, both of which are carried out in a fully end-to-end manner:

Stage 1: full model pretraining. In the first stage, we aim to establish a strong initialization for the flow matching head, and adapt pixel-space visual inputs for unified multimodal understanding and generation. To achieve this, we train the full model jointly on two tasks: image captioning and text-to-image generation.

Stage 2: supervised finetuning (SFT). Next, we perform supervised fine-tuning (SFT) of the full model with a lower learning rate. We use datasets for image editing, image instruction-following, and high-quality image generation. This step refines TUNA-2's abilities, boosting performance and generalization across various multimodal tasks.

For TUNA-R, which includes a connector layer between the representation encoder and the LLM decoder, we add an extra alignment stage before Stage 1. In this stage, we train only the connector layer for a short period using image captioning and text-to-image generation data. As noted above, TUNA-2 does not require this additional stage because of its encoder-free design.

12.3 Experiments

12.3.1 Experiment Setup

We employ Qwen2.5-7B-Instruct [211] as the LLM decoder and use a patch embedding size of 16 for TUNA-2. For Stage 1 pretraining, we use 550M in-house image-text pairs, consisting of 30% image captioning data for multimodal understanding and 70% text-to-image generation data. In addition, we include text-only data from Nemotron [19], which accounts for 20% of the total pretraining data. The full model is trained end-to-end for 300k steps on 64 nodes with the AdamW optimizer [182] and a learning rate of 1×10^{-4}. We employ the proposed masking-based feature learning strategy during the final 40% of pretraining, applying masking in 50% of training examples and randomly sampling the masking ratio from 0% to 50%. For Stage 2 supervised finetuning, we use a curated SFT corpus covering image instruction-following, image editing, and high-quality image generation. Specifically, for image instruction-following, we include 13M conversational examples

Table 12.1: Comparisons between TUNA-2 and baseline models on multimodal understanding benchmarks. Results with model size greater than 13B are grayed. **Bold**: best results among all UMMs. Underline: second-best among all UMMs.

Models	Size	General Benchmarks									Pixel-centric Benchmarks		
		GQA	RealWorldQA	MMVet	MMMU	MMVP	SEED-Bench2+	AI2D	ChartQA	OCRBench	V*	CountBench	VisuLogic
Understanding-only Models (LMMs)													
LLaVA-1.5 [166]	7B	62.0	54.8	32.9	35.7	-	-	55.5	17.8	31.8	-	-	-
Qwen-VL-Chat [13]	7B	57.5	49.3	47.3	37.0	-	-	57.7	49.8	48.8	-	-	-
LLaVA-OV [141]	7B	-	69.9	51.9	48.8	77.3	62.2	81.4	80.9	62.2	72.7	76.2	24.8
Qwen2.5-VL [15]	7B	60.7	69.9	61.7	58.6	78.0	70.5	82.7	83.0	83.7	71.2	74.1	20.0
Composite UMMs													
TokenFlow-XL [210]	14B	62.5	56.6	-	43.2	-	-	-	-	-	-	-	-
BLIP3-o [37]	4B	-	60.4	-	46.6	-	-	-	-	-	-	-	-
Tar [96]	7B	61.3	-	-	39.0	74.3	46.2	-	-	-	41.4	64.2	24.3
X-Omni [80]	7B	62.8	62.6	-	47.2	-	-	76.8	81.5	70.4	-	-	-
Native UMMs													
BAGEL [59]	14B	66.4	72.8	67.2	55.3	85.0	71.9	89.2	78.5	73.3	70.2	82.5	41.7
Ming-UniVision [113]	16B	59.4	59.1	64.2	40.3	71.0	56.8	82.8	76.7	72.4	48.2	76.8	26.7
Harmon [293]	1.5B	58.9	49.8	-	38.9	61.7	41.6	57.0	29.8	11.2	41.9	67.0	25.1
JanusFlow [189]	1.3B	60.3	41.2	36.2	29.3	67.7	39.8	54.2	42.4	53.2	42.9	78.6	22.0
Emu3 [274]	8B	60.3	57.4	23.5	31.6	71.0	44.6	70.0	69.4	68.7	53.4	65.2	24.7
VILA-U [295]	7B	60.8	46.8	26.3	31.2	62.7	31.9	49.0	11.4	23.3	38.7	55.2	25.4
Janus-Pro [47]	7B	62.0	58.0	41.1	41.0	73.3	56.3	71.3	25.8	59.0	47.6	53.2	23.8
Show-o2 [300]	7B	63.1	64.7	39.6	48.9	76.7	61.3	78.6	52.3	32.4	44.5	63.5	26.9
OneCat [142]	9B	63.1	65.2	**52.2**	41.9	71.3	**61.6**	77.8	81.2	79.0	**63.4**	34.2	24.9
TUNA [180]	7B	63.9	66.1	42.9	49.8	70.7	52.7	79.3	**85.8**	74.3	52.4	73.5	22.4
TUNA-R	7B	63.5	**67.9**	46.7	**51.1**	74.7	58.4	79.4	85.6	78.3	57.6	77.8	26.2
TUNA-2	7B	**65.0**	67.7	51.7	50.7	**77.3**	61.1	**79.6**	85.6	**79.7**	59.2	**81.7**	**28.8**

from the open-source FineVision [283] dataset. For image editing, we use approximately 2M examples from OmniEdit [280]. This stage is trained for 50k steps with AdamW and a learning rate of 2×10^{-5}. For all training stages, we pad the input sequence length to 16k tokens per GPU.

For TUNA-R, we use the same Qwen2.5-7B-Instruct as the LLM decoder. We follow TUNA and adopt SigLIP 2 So400M [253] as the representation encoder. For the connector-alignment stage in TUNA-R, we train the model for 3k steps with AdamW and a learning rate of 5×10^{-4}.

12.3.2 Main Results

Image understanding. We employ a comprehensive evaluation suite consisting of nine VQA benchmarks, including GQA [114], RealWorldQA [296], MMVet [319], MMMU [322], MMVP [249], SEED-Bench2+ [138], AI2D [126], ChartQA [191] and OCRBench [176], to evaluate the image understanding capabilities for TUNA-2. As shown in Table 12.1, after removing the VAE model, both TUNA-R and TUNA-2 outperform TUNA and achieve state-of-the-art results among all 7B-scale native UMMs, demonstrating the effectiveness

Table 12.2: Image generation results on GenEval and DPG-Bench. "Col. Attr." means "Color Attribute". † refers to methods using LLM rewriters in GenEval. **Bold**: best results among native UMMs. Underline: second-best.

Models	Size	GenEval							DPG-Bench					
		1-Obj.	2-Obj.	Count	Colors	Position	Col. Attr.	Overall	Global	Entity	Attribute	Relation	Other	Overall
Generation-only Models														
SD3-M [66]	2B	0.99	0.94	0.72	0.89	0.33	0.60	0.74	87.90	91.01	89.96	80.70	88.68	84.08
FLUX.1 [dev]† [17]	12B	0.98	0.93	0.75	0.93	0.68	0.65	0.82	82.10	89.50	88.70	91.10	89.40	84.00
LongCat-Image [246]	6B	0.99	0.98	0.86	0.86	0.75	0.73	0.87	89.10	92.54	92.00	93.28	87.50	86.80
Qwen-Image [286]	20B	0.99	0.92	0.89	0.88	0.76	0.77	0.87	91.32	91.56	92.02	94.31	92.73	88.32
Seedream 3.0 [76]	-	0.99	0.96	0.91	0.93	0.47	0.80	0.84	94.31	92.65	91.36	92.78	88.24	88.27
Z-Image-Turbo [26]	6B	1.00	0.95	0.77	0.89	0.65	0.68	0.82	91.29	89.59	90.14	92.16	88.68	84.86
Composite UMMs														
Tar [96]	7B	0.99	0.92	0.83	0.85	0.80	0.65	0.84	83.98	88.62	88.05	93.98	84.86	84.19
BLIP3-o [37]	8B	-	-	-	-	-	-	0.84	-	-	-	-	-	81.60
UniWorld-V1† [159]	12B	0.98	0.93	0.81	0.89	0.74	0.71	0.84	83.64	88.39	88.44	89.27	87.22	81.38
OmniGen2† [288]	7B	0.99	0.96	0.74	0.98	0.71	0.75	0.86	88.81	88.83	90.18	89.37	90.27	83.57
Native UMMs														
MUSE-VL [302]	7B	-	-	-	-	-	-	0.57	-	-	-	-	-	-
Transfusion [342]	7B	-	-	-	-	-	-	0.63	-	-	-	-	-	-
Emu3 [274]	8B	-	-	-	-	-	-	0.66	-	-	-	-	-	81.60
Show-o2 [300]	7B	**1.00**	0.87	0.58	0.92	0.52	0.62	0.76	89.00	91.78	89.96	91.81	**91.64**	86.14
Janus-Pro [47]	7B	0.99	0.89	0.59	0.90	0.79	0.66	0.80	86.90	88.90	89.40	89.32	89.48	84.19
HBridge† [272]	7B	**1.00**	0.96	0.80	0.94	0.77	0.78	0.87	**91.78**	**91.82**	90.23	90.06	88.42	85.23
Ming-UniVision [113]	16B	**1.00**	0.93	0.59	0.93	**0.92**	0.70	0.85	-	-	-	-	-	82.12
BAGEL† [59]	14B	0.98	0.95	**0.84**	**0.95**	0.78	0.77	0.88	88.94	90.37	91.29	90.82	88.67	85.07
Mogao [157]	7B	**1.00**	**0.97**	0.83	0.93	0.84	0.80	0.89	82.37	90.03	88.26	93.18	85.40	84.33
TUNA [180]	7B	**1.00**	**0.97**	0.81	0.91	0.88	**0.83**	**0.90**	90.42	91.68	90.94	91.87	90.73	**86.76**
TUNA-R	7B	**1.00**	0.95	0.82	0.89	0.86	0.79	0.88	86.00	91.80	91.03	**93.48**	84.89	86.35
TUNA-2	7B	0.99	0.96	0.80	0.91	0.84	0.76	0.87	89.50	91.40	**92.07**	91.91	88.81	86.54

of our pixel-space unified representations. Notably, TUNA-2 outperforms TUNA-R even after replacing the representation encoder with the simple patchify layer. This may indicate that large-scale joint training of a unified, monolithic architecture achieves better understanding performance than relying on the inductive bias in the pretrained representation encoders in the modular setting. We show more analysis comparing TUNA-R and TUNA-2 in Section 12.3.5.

To further understand the benefit of our pixel-space unified models, we include several pixel-centric benchmarks that focus heavily on visual reasoning over fine-grained visual details (e.g. recognizing very small objects in high-resolution images). These benchmarks include V* [292], CountBench [204] and VisuLogic [308]. As shown in Table 12.1, both TUNA-R and TUNA-2 outperform latent-space UMMs (e.g. Show-o2, TUNA, etc.) across all benchmarks, indicating the necessity of pixel-space visual representations when reason-

Table 12.3: Evaluation results under GPT-5.4 and Claude Opus 4.7. Quality measures image realism and fine-grained detail richness, while Diversity measures variation among images generated from the same prompt. **Bold** denotes the best result and underlining denotes the second-best.

	GPT-5.4		Claude Opus 4.7	
Models	Quality	Diversity	Quality	Diversity
TUNA	22.3%	20.6%	28.1%	28.2%
TUNA-R	**35.7%**	30.9%	**37.2%**	29.9%
TUNA-2	32.1%	**48.4%**	34.8%	**41.9%**

Table 12.4: Image editing results on ImgEdit. **Bold** denotes the best result among unified models and underlining denotes the second-best.

Models	Add	Adj.	Ext.	Rep.	Rm.	Bg.	Sty.	Hyb.	Act.	Total
Generation-only Models										
FLUX.1	4.25	4.15	2.35	4.56	3.57	4.26	4.57	3.68	4.63	4.00
Qwen-Image	4.38	4.16	3.43	4.66	4.14	4.38	4.81	3.82	4.69	4.27
Unified Models										
OmniGen	3.47	3.04	1.71	2.94	2.43	3.21	4.19	2.24	3.38	2.96
BAGEL	3.56	3.31	1.70	3.30	2.62	3.24	4.49	2.38	4.17	3.20
UniWorld	3.82	3.64	2.27	3.47	3.24	2.99	4.21	2.96	2.74	3.26
OmniGen2	3.57	3.06	1.77	3.74	3.20	3.57	4.81	2.52	4.68	3.44
GPT-Image	**4.61**	4.33	**2.90**	4.35	3.66	**4.57**	**4.93**	3.96	**4.89**	4.20
TUNA	4.43	**4.48**	2.46	**4.65**	**4.55**	4.52	4.69	**4.22**	4.76	**4.31**
TUNA-R	4.46	4.27	2.38	4.61	4.48	4.44	4.54	4.06	4.43	4.18
TUNA-2	4.34	4.13	2.22	4.53	4.42	4.36	4.58	3.91	4.28	4.09

ing over fine-grained visual details.

Image generation. We evaluate the image generation performance of TUNA-2 on GenEval [83] and DPG-Bench [110]. As shown in Table 12.2, both TUNA-R and TUNA-2 achieve state-of-the-art results on these benchmarks and perform competitively with contemporary approaches such as BAGEL, Mogao, and TUNA. Between the two variants, TUNA-R consistently performs slightly better than TUNA-2, suggesting that the semantic prior introduced by the representation encoder helps our method learn a stronger image generation model. Overall, these results show that even without a VAE and while performing image generation entirely in pixel space, TUNA-2 remains competitive with recent state-of-the-art unified models, highlighting the effectiveness and scalability of our pixel-space generation design.

Since existing image generation benchmarks such as GenEval mainly evaluate text-image alignment and the model's world knowledge, we further adopt an LLM-judge-based evaluation to assess the quality and diversity of images generated by TUNA-2, TUNA-R, and TUNA. Specifically, we sample 1.5K text prompts and ask each model to generate four images per prompt. We then use GPT-5.4 [201] and Claude Opus 4.7 [7] as judges to select the best model among the three for each prompt based on the quality and diversity of the generated images, where quality mainly refers to realism and fine-grained detail and texture fidelity, while diversity measures the extent to which the four generated images exhibit distinct visual variations under the same prompt. The results are reported in

Table 12.5: Image reconstruction performance for different visual tokenizers.

Tokenizer	Res.	rFID↓	PSNR↑	SSIM↑
Specialized tokenizers				
SD-VAE	256	1.06	28.62	0.86
GigaTok	256	0.51	21.32	0.69
VA-VAE	256	0.26	28.59	0.80
DC-AE	512	0.22	26.15	0.71
MAE-Tok	512	0.62	–	–
TexTok	512	0.73	24.45	0.66
FLUX.1[dev]-VAE†	512	0.06	33.65	0.93
Unified tokenizers				
UniTok	256	0.38	–	–
TokenFlow	384	0.63	22.77	0.73
X-Omni†	512	8.30	15.66	0.38
MingTok†	512	0.53	23.49	0.61
RAE	256	0.61	19.20	0.44
PS-VAE	256	0.20	28.79	0.82
Tuna-R	512	**0.12**	32.22	**0.93**
Tuna-2	512	0.15	**32.80**	**0.93**

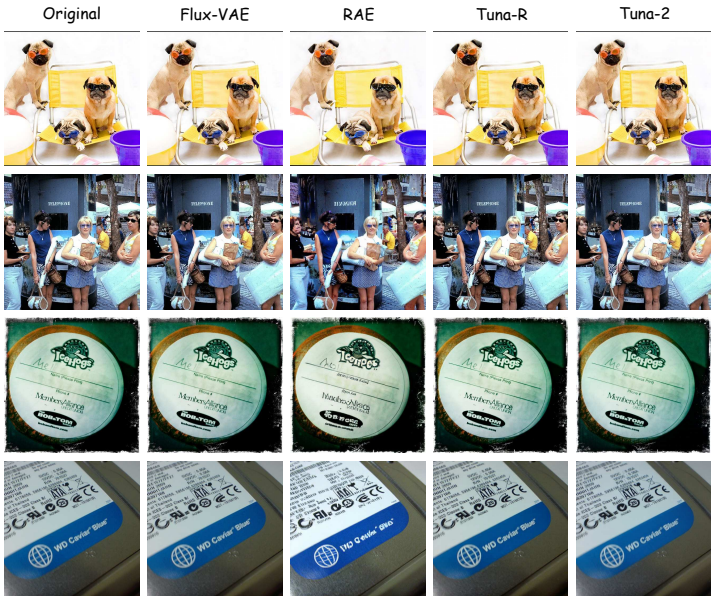


Figure 12.4: Qualitative results for different visual tokenizers.

Table 12.3. We find that under both LLM judges, TUNA-2 achieves competitive image generation quality, performing comparably to TUNA-R and better than TUNA, while being significantly preferred in terms of diversity. These results demonstrate that our encoder-free design is highly effective and enables TUNA-2 to generate both high quality and diverse images.

Image editing. We further evaluate the image editing capability of TUNA-2 on ImgEdit [316]. As shown in Table 12.4, TUNA-2 achieves strong editing performance among unified models, outperforming earlier baselines such as OmniGen [298], BAGEL [59], UniWorld [159], and OmniGen2 [288]. While being slightly behind TUNA and TUNA-R, TUNA-2 remains competitive with strong generation-only and unified editing systems, despite performing editing directly in pixel space without relying on vision encoders. These results suggest that our pixel-space unified modelling framework can effectively support instruction-guided image editing, while the small gap between TUNA-2 and encoder-based variants indicates that pretrained visual priors may still provide benefits for fine-grained editing fidelity.

Image reconstruction. To further investigate the image generation capability of TUNA-2, we examine whether the model can faithfully reconstruct an input image from its corresponding pixel-space visual representations. To this end, we perform lightweight finetuning on an image reconstruction task and evaluate reconstruction quality on the ImageNet

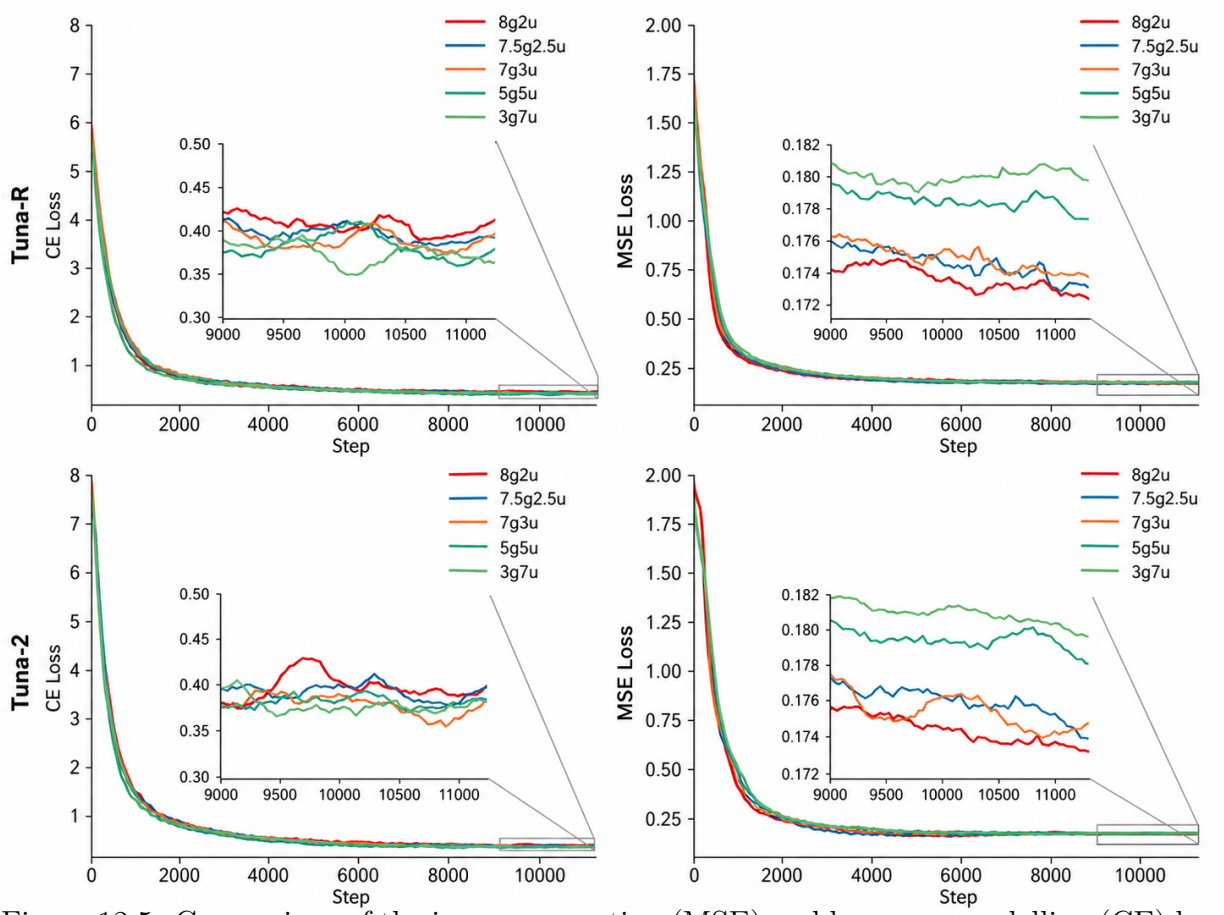


Figure 12.5: Comparison of the image generation (MSE) and language modelling (CE) loss curves based on different understanding-generation data ratios during model training.

validation set [60]. As shown in Table 12.5, both TUNA-R and TUNA-2 achieve strong reconstruction performance, ranking first among unified tokenizers and approaching specialized image tokenizers such as the VAE model in FLUX.1 [dev] [17]. Figure 12.4 further shows that our models produce substantially better reconstructions than other non-KL-regularized VAE approaches such as RAE [340]. These results indicate that our pixel-space unified visual representation can support strong image reconstruction and generation quality even without relying on VAE models.

Table 12.6: Ablation study results for TUNA-R and TUNA-2, with or without our proposed masking-based feature learning scheme during pretraining.

Models	OCRBench	MMVP	CountBench	GenEval
TUNA	56.9	54.0	55.6	57.2
TUNA-R w/o Masking	58.3	56.7	57.2	55.7
TUNA-R w/ Masking	59.2	58.0	58.2	56.0
TUNA-2 w/o Masking	55.4	52.3	53.4	47.6
TUNA-2 w/ Masking	56.8	55.7	57.6	48.2

12.3.3 Ablation: Model Training Dynamics

To better understand the learning dynamics of our joint multimodal training paradigm, we conduct a series of pretraining ablation studies on both TUNA-R and TUNA-2 by varying the sampling ratio between generation and understanding data. We use the notation `xgyu` (e.g., `8g2u`) to represent a generation-to-understanding sampling ratio of $x : y$. As shown in Figure 12.5, increasing the proportion of either generation or understanding data consistently reduces its corresponding training loss, namely MSE for flow matching and cross entropy loss (CE) for language modelling. Notably, the MSE loss is more sensitive to changes in the sampling ratio, while the CE loss varies within a relatively smaller range across different ratios. This suggests that both tasks benefit from scaling up the amount of their corresponding training data, with the generation objective being more affected by the data mixture. We also observe that a generation-to-understanding ratio of $7 : 3$ (`7g3u`) achieves the best trade-off between the two objectives, yielding a strong balance between generation and understanding performance. We therefore adopt this data sampling ratio in all experiments.

12.3.4 Ablation: Masking-based Feature Learning

To verify the effectiveness of our proposed masking-based feature learning strategy, we conduct controlled experiments on both TUNA-R and TUNA-2 using the smaller Qwen-2.5-Instruct-1.5B backbone. Since we expect this strategy to serve as a representation enhancement strategy after the model has acquired basic multimodal knowledge, we do not apply this objective from the beginning of pretraining. Instead, we first train all models for 50k steps; We then split each model variant into two groups: one continues standard pretraining, while the other continues pretraining with the masking-based feature learning

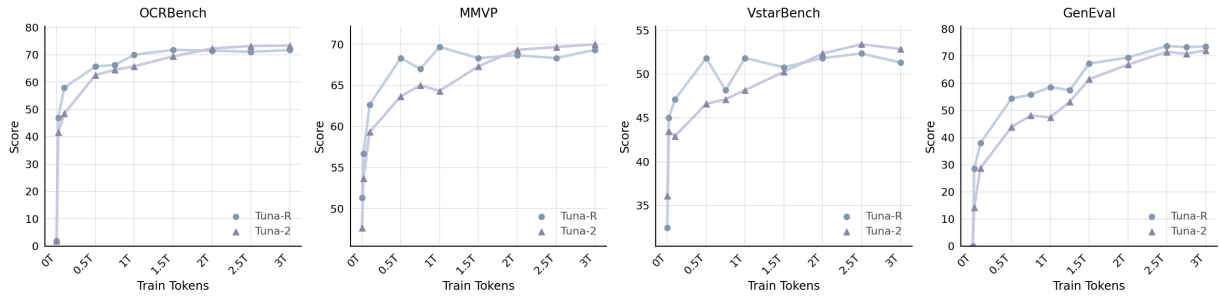


Figure 12.6: Model accuracy curves with respect to training dataset size (measured by the number of tokens consumed by the model) for Tuna-R and Tuna-2.

objective applied with a probability of 50%. Both groups are further trained for another 50k steps. As shown in Table 12.6, this strategy consistently improves performance on both understanding and generation benchmarks for both model variants. We further observe that Tuna-2 benefits more from masked training than Tuna-R. We hypothesize that this difference is related to the SigLIP 2 representation encoder used in Tuna-R, since SigLIP 2 itself is pretrained with a similar masked prediction objective. In addition, the results suggest that compressing images into the VAE latent space before visual encoding (Tuna) can introduce certain information loss for visual understanding, compared with directly encoding pixel-level inputs through the vision encoder (Tuna-R). Based on these findings, we apply the masking-based feature learning strategy during the final 40% of pretraining, encouraging more robust visual representation learning for both multimodal understanding and generation.

12.3.5 Analysis: Tuna-R vs. Tuna-2

Our experimental results in Section 12.3.2 reveal two interesting findings: First, without vision encoders, Tuna-2 outperforms Tuna-R on most image understanding benchmarks. Second, Tuna-R achieves similar or slightly stronger performance than Tuna-2 on generation tasks. To better understand these trends, we plot the performance curves of both models on understanding and generation benchmarks as the training data scale increases. As shown in Figure 12.6, on the three understanding benchmarks (OCRBench [176], MMVP [249] and V* [292]), Tuna-R actually outperforms Tuna-2 during the early stage of the training. We believe this advantage comes from the pretrained representation encoder in Tuna-R, whose rich semantic priors help the model acquire multimodal understanding capabilities more quickly at the beginning of training. Nevertheless, Tuna-2 later catches up and eventually surpasses Tuna-R. This suggests that the monolithic, encoder-free design of Tuna-2 may be better suited to benefit from large-scale unified multimodal

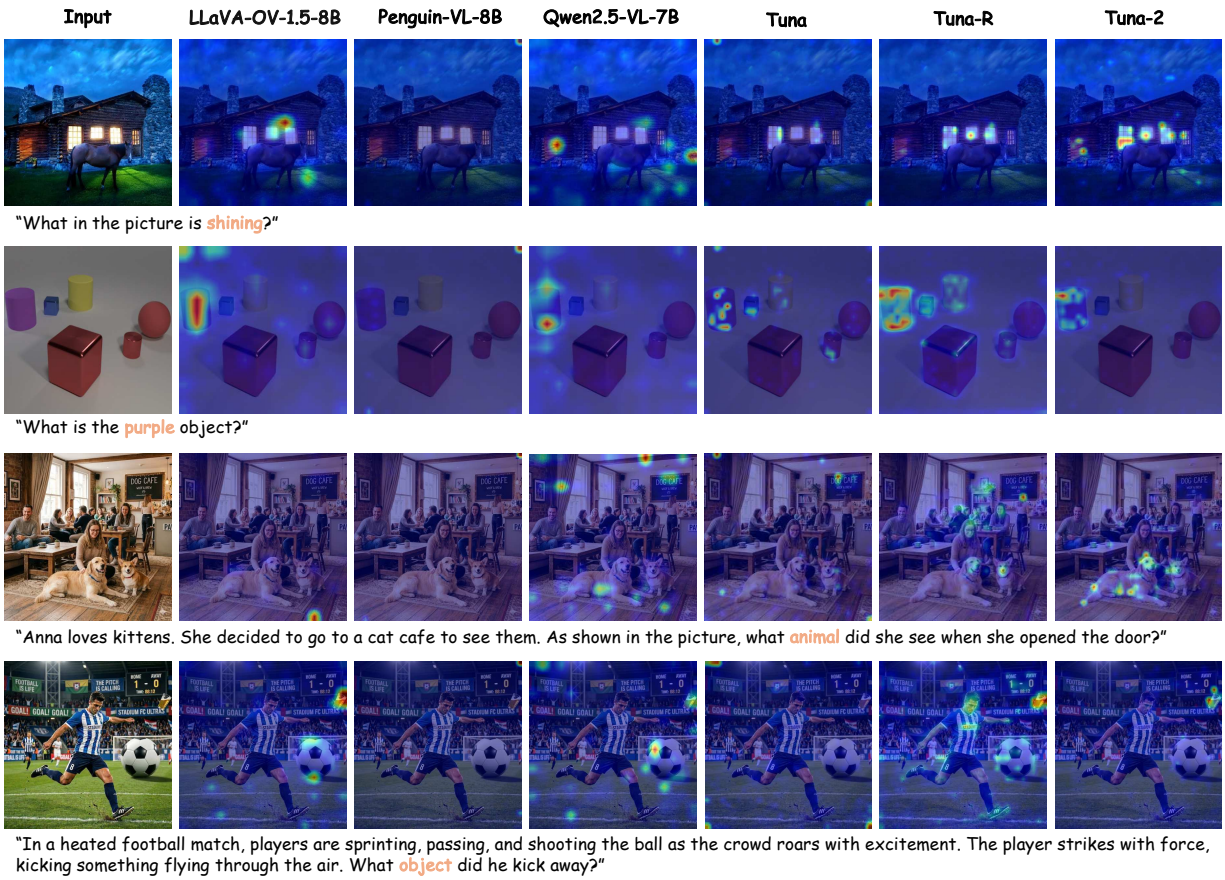


Figure 12.7: Attention map visualization for Tuna-R, Tuna-2 and other baseline models. Red area denotes high attention scores and Blue area denotes low attention scores.

pretraining, enabling it to develop stronger multimodal understanding capabilities.

For generation evaluation on GenEval [83], we observe that Tuna-R consistently outperforms Tuna-2 throughout the entire training process. This suggests that the semantic priors provided by the representation encoder play an important role in improving generation performance, which is consistent with prior findings from REPA [318] and Tuna [180]. However, this trend gradually weakens as the training data scale increases. As shown in the figure, the two model variants achieve nearly identical performance after SFT. Overall, compared with latent-space generation, both Tuna-R and Tuna-2's pixel-space generation paradigm achieves competitive performances.

12.3.6 Analysis: Attention Map Visualizations

To qualitatively analyze and compare the cross-modal alignment capabilities learned by different models, we visualize the attention maps of selected keywords in the input prompt with respect to the input image. We focus on the TUNA model family, including TUNA, TUNA-R, and TUNA-2, and compare them with representative encoder-based open-source LMMs, including models with ViT-based encoders, such as LLaVA-OneVision-1.5 [6] and Qwen2.5-VL [15], as well as models with LLM-based encoders, such as Penguin-VL [328]. The results are shown in Figure 12.7.

Our first observation is that TUNA-2 exhibits more accurate vision-language alignment in basic perception scenarios. For example, in the "shining window" case, TUNA-2 consistently highlights the regions that are semantically associated with "shining", while other models tend to provide only coarse or incomplete localization. In the "purple object" case, the attention activations of TUNA-2 are well aligned with the target object, whereas other methods often show dispersed attention or spurious activations in irrelevant regions.

We further construct examples with misleading linguistic contexts to evaluate whether models overly rely on language priors. For example, in the "dog cafe" case, the prompt suggests a cat cafe scenario, whereas the actual animals in the image are dogs. Some models tend to rely on textual cues within the image (e.g. "dog cafe" sign) instead of the actual visual subjects. In contrast, TUNA-2 places its attention on the truly relevant visual regions, demonstrating more robust cross-modal alignment under misleading linguistic contexts.

In the more challenging counterintuitive "football match" case, we introduce both strong linguistic priors and salient visual distractors. Specifically, the prompt contains strong football-related cues, such as "football match", "players", and "kicking". The image also includes a large and visually salient football as a distractor. However, the actual object being kicked by the player is a glass cup. Most models are easily misled by either the textual prior or the salient distractor and therefore attend to incorrect regions. In contrast, TUNA-2 accurately localizes the key object that is consistent with the question semantics, showing stronger robustness in such counterintuitive settings.

Overall, as an encoder-free native multimodal architecture, TUNA-2 produces more accurate and stable attention distributions across all cases. These results suggest that TUNA-2 learns more reliable visual representations, leading to more consistent cross-modal alignment and better robustness to misleading language priors and visual distractors.

12.4 Conclusion

We introduced TUNA-2, a family of native unified multimodal models that perform multimodal understanding and visual generation directly in pixel space, without relying on VAE encoders or latent diffusion. By combining a unified vision-language backbone with a pixel-space flow matching head, TUNA-2 supports image understanding, text-to-image generation, and image editing within a single framework. We further instantiated TUNA-2 with both a representation encoder-based variant and an encoder-free monolithic variant, and showed that both designs achieve strong performance across multimodal understanding and generation benchmarks. Our experiments demonstrate that TUNA-2 surpasses prior latent-space unified models such as TUNA and Show-o2 on fine-grained visual understanding benchmarks while remaining competitive with state-of-the-art unified models on image generation. Overall, these results highlight the effectiveness and scalability of pixel-space unified multimodal modelling and suggest a promising direction for future native UMMs.

Chapter 13

Conclusion and Future Work

This thesis started from the observation that today's visual foundation models, despite their remarkable capabilities, still operate on *fragments* of visual experience: a short caption, a single image, a few seconds of video. People, by contrast, perceive and create within *visual worlds* that persist over time, extend across long horizons, and are simultaneously read through perception and written through creation. I argued that this gap is not primarily a question of scale but of *interface*: what a model is allowed to take in, and what it is allowed to produce. Accordingly, the thesis set out to widen that interface along three axes, namely *control*, *horizon*, and *modality*, and developed a concrete system for each.

13.1 Summary of Contributions

Control: generating coherent visual worlds from sparse observations. Part I broadened the conditioning signals that a generative model can faithfully follow beyond simple text prompts. In Chapter 4, I presented ConsistI2V, an image-to-video diffusion model that converts a pretrained text-to-image backbone into a video generator, conditioned on both text prompts and input first frames. It introduces two mechanisms to enforce faithful image conditioning and prompt following. First, fine-grained spatiotemporal first-frame conditioning ties appearance, identity, and layout to the input frame while still allowing motion progresses in the video. Second, FrameInit introduces a layout-guided noise initialization scheme applied at inference time, injects the low-frequency component of the first frame into the initial noise to narrow the train-test gap and stabilize long-range generation. Trained only on public data and initialized from a text-to-image model, ConsistI2V outperformed strong image-to-video baselines on automatic metrics

and ranked first in human evaluation of appearance and motion consistency. Chapter 5 then showed that this image-to-video generator is more than a single-task model. AnyV2V recasts a broad family of video-to-video editing problems, including prompt-based editing, reference-based style transfer, subject-driven editing, and identity manipulation, as first-frame-conditioned generation: an off-the-shelf image editor first performs the edit spatially, the image-to-video model then propagates the edit temporally under structural and motion guidance from the source video. By decoupling *what* to change from *how it moves*, AnyV2V supports various video editing tasks without any task-specific training, and widens the conditioning interface from text and images to full videos.

Horizon: understanding visual worlds over long time spans. Part II extended the amount of continuous visual history a model can process, and treated architecture, data, and evaluation as three parts of one problem. Chapter 7 presented VAMBA, which attacks the quadratic cost of causal self-attention in video LMMs by changing how video tokens are processed within the decoder layers. VAMBA decomposes the expensive causal self-attention into cross-attention and linear-time Mamba-2 blocks, where the short text sequence attends to the full set of video tokens using cross-attention and the Mamba-2 blocks update the video tokens. This resulting in VAMBA encoding over 1,024 frames on a single GPU, roughly four times the context of comparable transformer-based models, while outperforming prior efficient video LMMs on various video understanding benchmarks. Chapter 8 argues that efficient architectures alone are not enough and addressed the scarcity of training data with VISTA, a spatiotemporal augmentation framework that synthesizes long-duration and high-resolution instruction data by combining existing short video-caption pairs along the temporal and spatial dimensions. The resulting VISTA-400K dataset improved three different video LMMs by an average of 3.3% on four long video benchmarks and 6.5% on HRVideoBench. Chapter 9 then questioned whether existing benchmarks measure what they claim to. My analysis showed that multiple-choice long video benchmarks are systematically inflated by option-based shortcuts and by questions answerable from prior knowledge alone. VideoEval-Pro replaces them with filtered, open-ended questions, and demonstrates that across 21 proprietary and open-source models, the strongest model only reached 40.8% accuracy, which is far beyond perfect. Critically, unlike saturating multiple-choice benchmarks, accuracy on VideoEval-Pro keeps rising as more frames are provided, which is exactly the behaviour a faithful measure of long-horizon understanding should exhibit.

Modality: unifying understanding and generation in one model. Part III focuses on closing the separation between models that understand visual content and models that

generate it. Chapter 11 presented TUNA, a native unified multimodal model built around a unified continuous visual representation obtained by cascading a VAE encoder with a representation encoder, so that one feature space carries both the low-level structure that generation needs and the semantics that understanding needs. A single language model decoder then performs autoregressive next-token prediction for multimodal understanding and flow-matching denoising for visual generation, covering tasks such as image and video understanding, image and video generation, and image editing in one model. TUNA surpassed prior unified models built on decoupled representations and remained competitive with specialized understanding-only and generation-only systems. Its most important result is not any single score but the ablation showing *mutual enhancement*: with a shared representation, joint training on understanding and generation outperforms training either capability alone, which is precisely the synergy that decoupled designs forfeit. Chapter 12 further pushed the model architecture of native unified multimodal models to a minimalist design: TUNA-2 removes the pretrained vision encoders altogether, including the VAE, and operates directly on pixel embeddings produced by simple patch embedding layers. TUNA-2 demonstrates that the encoder-free design matched its encoder-based counterpart on generation while consistently outperforming it on understanding, especially on perception-intensive, fine-grained visual tasks. This suggests that pretrained visual encoders, while being treated as indispensable for a very long time, may in fact constrain what end-to-end multimodal pretraining can learn, and a unified visual representation space produced by simple patch embeddings may be a possible solution to unified models.

13.2 Future Directions

Closing the loop between the three axes. The most direct continuation of this thesis is a model that can read and write within the same long visual world. Such a system would ingest long videos with the efficiency of VAMBA, represent them using the shared visual representation of TUNA, and generate continuations, edits, or entirely new segments conditioned on the accumulated visual history. Achieving this requires generation to become history-aware in a way that current clip-level generators are not: a newly generated video clip should preserve the identities, spatial relationships, and events established throughout the preceding history. The first-frame conditioning studied in Chapter 4 can be viewed as the shortest-horizon instance of this idea, and generalizing from a single key frame to an entire video history is, in my view, a central open problem for visual world models [24, 177].

Persistent memory for world models. Long-horizon generation faces the same quadratic complexity barrier that motivated VAMBA, and similar architectural solutions may apply. Linear-time and hybrid sequence models are therefore promising directions for visual generation, as they could allow a model to condition on thousands of previously generated frames rather than relying on a short sliding window. Purely recurrent sequence models, however, provides only a lossy summary of the history. This makes hybrid memory architectures particularly compelling: a compressive state could capture the overall evolution of a scene, while an explicit, retrievable memory could preserve entities and events that must remain precisely consistent. This memory challenge is shared by both understanding and generation, providing another motivation to address it within a unified model rather than solving it independently for each capability.

Data and evaluation for unified models. The lessons of Chapters 8 and 9 transfer directly to unified models, where both training data and evaluation benchmarks remain largely underexplored. On the data side, unified models currently lack interleaved understanding and generation data that span long temporal horizons, and synthesizing such supervision from existing datasets, rather than collecting it at prohibitive cost, is a promising direction. On the evaluation side, current assessments of unified models largely reuse understanding-only and generation-only benchmarks, which measure the two capabilities in isolation and therefore reveal little about whether unification itself is effective. Existing benchmarks rarely target interleaved settings in which a model must alternate between understanding and generating visual content within a single task, such as generating an image, inspecting its own output, and revising it accordingly. Without such measurements, it will remain difficult to determine whether unified models are genuinely integrating understanding and generation, or simply becoming better at each capability independently.

From visual worlds to actionable worlds. Finally, the visual worlds studied in this thesis are observed and generated, but not acted upon. Extending foundation models with action conditioning, real-time and streaming operation, and grounding in physical dynamics could transform a video generator into an interactive simulator that an agent or robot can query, explore, and plan within. This direction inherits the full agenda developed above, since an interactive world model must be controllable, remain consistent over long horizons, and both perceive and synthesize visual content. Such models could enable important applications across domains including physical AI, by providing simulated environments for training robots and autonomous vehicles; entertainment, through interactive gaming and virtual worlds; and scientific simulation, where learned world models could support the exploration and prediction of complex dynamic systems.

13.3 Closing Remarks

The gap between a model that captions an image and one that can understand and generate a coherent visual world cannot be closed in a single step. It requires progress along several different directions. This thesis advances three of them: enabling generative models to follow rich conditioning signals faithfully; scaling visual understanding to hours-long videos while developing the data and evaluation methods needed to measure that progress reliably; and building a shared representation that supports both understanding and generation. The specific systems presented in this thesis will inevitably be surpassed as the field continues to develop. What may prove more lasting is the perspective behind them: visual foundation models are limited not only by their scale, but also by the narrow ways in which they interact with the visual world. The evidence presented in this thesis suggests that carefully broadening these interactions is essential for moving visual foundation models beyond processing isolated fragments and toward modelling coherent worlds.

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[2] Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras, 2025.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[5] Jie An, Songyang Zhang, Harry Yang, Sonal Gupta, Jia-Bin Huang, Jiebo Luo, and Xi Yin. Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. *arXiv preprint arXiv:2304.08477*, 2023.

[6] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

[7] Anthropic. Introducing claude opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, April 2026. Accessed: 2026-04-24.

[8] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022.

[9] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

[10] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

[12] Jianhong Bai, Tianyu He, Yuchi Wang, Junliang Guo, Haoji Hu, Zuozhu Liu, and Jiang Bian. Uniedit: A unified tuning-free framework for video motion and appearance editing. *arXiv preprint arXiv:2402.13185*, 2024.

[13] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[14] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[15] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[16] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021.

[17] Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pages arXiv–2506, 2025.

[18] Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Fuyu-8b: A multimodal architecture for ai agents. Adept blog post, https://www.adept.ai/blog/fuyu-8b/, 2023.

[19] Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025.

[20] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.

[21] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023.

[22] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei Efros, and Tero Karras. Generating long videos of dynamic scenes. *Advances in Neural Information Processing Systems*, 35:31769–31781, 2022.

[23] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[24] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. https://openai.

com/research/video-generation-models-as-world-simulators, 2024. OpenAI technical report.

[25] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[26] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[27] Meng Cao, Pengfei Hu, Yingyao Wang, Jihao Gu, Haoran Tang, Haoze Zhao, Jiahua Dong, Wangbo Yu, Ge Zhang, Ian Reid, et al. Video simpleqa: Towards factuality evaluation in large video language models. *arXiv preprint arXiv:2503.18923*, 2025.

[28] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[29] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23206–23217, 2023.

[30] Keshigeyan Chandrasegaran, Agrim Gupta, Lea M Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Fei-Fei Li. Hourvideo: 1-hour video-language understanding. *Advances in Neural Information Processing Systems*, 37:53168–53197, 2024.

[31] Keshigeyan Chandrasegaran, Agrim Gupta, Lea M Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Fei-Fei Li. Hourvideo: 1-hour video-language understanding. *Advances in Neural Information Processing Systems*, 37:53168–53197, 2025.

[32] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[33] Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. Oneig-bench: Omni-dimensional nuanced evaluation for image generation. *arXiv preprint arXiv:2506.07977*, 2025.

[34] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.

[35] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

[36] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. *arXiv preprint arXiv:2401.09047*, 2024.

[37] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

[38] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[39] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024.

[40] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*, 2024.

[41] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video

understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2025.

[42] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025.

[43] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[44] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024.

[45] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models, 2023.

[46] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481*, 2023.

[47] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[48] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[49] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. *arXiv preprint arXiv:2310.20700*, 2023.

[50] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

[51] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[52] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[53] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[54] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023.

[55] Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Fine-grained open domain image animation with motion guidance. *arXiv preprint arXiv:2311.12886*, 2023.

[56] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[57] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

[58] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36:2252–2274, 2023.

[59] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[60] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[61] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[62] Haiwen Diao, Yufeng Cui, Xiaotong Li, Yueze Wang, Huchuan Lu, and Xinlong Wang. Unveiling encoder-free vision-language models. *Advances in Neural Information Processing Systems*, 37:52545–52567, 2024.

[63] Haiwen Diao, Mingxuan Li, Silei Wu, Linjun Dai, Xiaohua Wang, Hanming Deng, Lewei Lu, Dahua Lin, and Ziwei Liu. From pixels to words–towards native vision-language primitives at scale. *arXiv preprint arXiv:2510.14979*, 2025.

[64] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[65] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[66] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[67] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[68] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021.

[69] Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. https://huggingface.co/datasets/HuggingFaceFV/finevideo, 2024.

[70] Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, and Hui Wang. Video-ccam: Enhancing video-language understanding with causal cross-attention masks for short and long videos. *arXiv preprint arXiv:2408.14023*, 2024.

[71] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

[72] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. Stylevideogan: A temporal generative model using a pretrained stylegan. *arXiv preprint arXiv:2107.07224*, 2021.

[73] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025.

[74] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

[75] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118, 2025.

[76] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.

[77] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[78] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *European Conference on Computer Vision*, pages 102–118. Springer, 2022.

[79] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22930–22941, 2023.

[80] Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025.

[81] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023.

[82] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *CoRR*, abs/1705.06830, 2017.

[83] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[84] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023.

[85] Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. Zamba: A compact 7b ssm hybrid model. *arXiv preprint arXiv:2405.16712*, 2024.

[86] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[87] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[88] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[89] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[90] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, volume 33, pages 1474–1487, 2020.

[91] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

[92] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

[93] Jing Gu, Yilin Wang, Nanxuan Zhao, Tsu-Jui Fu, Wei Xiong, Qing Liu, Zhifei Zhang, He Zhang, Jianming Zhang, HyunJoon Jung, and Xin Eric Wang. Photoswap: Personalized subject swapping in images, 2023.

[94] Yuchao Gu, Yipin Zhou, Bichen Wu, Licheng Yu, Jia-Wei Liu, Rui Zhao, Jay Zhangjie Wu, David Junhao Zhang, Mike Zheng Shou, and Kevin Tang. Videoswap: Customized video subject swapping with interactive semantic point correspondence. *arXiv preprint*, 2023.

[95] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

[96] Jiaming Han, Hao Chen, Yang Zhao, Hanyu Wang, Qi Zhao, Ziyan Yang, Hao He, Xiangyu Yue, and Lu Jiang. Vision as a dialect: Unifying visual understanding and generation via text-aligned representations. *arXiv preprint arXiv:2506.18898*, 2025.

[97] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[98] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[99] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022.

[100] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[101] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[102] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[103] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[104] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[105] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[106] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[107] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.

[108] Emiel Hoogeboom and Tim Salimans. Blurring diffusion models. *arXiv preprint arXiv:2209.05557*, 2022.

[109] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[110] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[111] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. *arXiv preprint arXiv:2311.17982*, 2023.

[112] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[113] Ziyuan Huang, DanDan Zheng, Cheng Zou, Rui Liu, Xiaolong Wang, Kaixiang Ji, Weilong Chai, Jianxin Sun, Libin Wang, Yongjie Lv, Taozhi Huang, Jiajia Liu, Qingpei Guo, Ming Yang, Jingdong Chen, and Jun Zhou. Ming-univision: Joint image understanding and generation with a unified continuous tokenizer. *arXiv preprint arXiv:2510.06590*, 2025.

[114] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[115] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

[116] Md Mohaiminul Islam, Tushar Nagarajan, Huiyu Wang, Gedas Bertasius, and Lorenzo Torresani. Bimba: Selective-scan compression for long-range video question answering. *arXiv preprint arXiv:2503.09590*, 2025.

[117] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2758–2766, 2017.

[118] Hyeonho Jeong and Jong Chul Ye. Ground-a-video: Zero-shot grounded video editing using text-to-image diffusion models. *arXiv preprint arXiv:2310.01107*, 2023.

[119] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[120] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv preprint arXiv:2405.01483*, 2024.

[121] Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, et al. Token-efficient long video understanding for multimodal llms. *arXiv preprint arXiv:2503.04130*, 2025.

[122] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17191–17202, 2025.

[123] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *arXiv preprint arXiv:2407.06358*, 2024.

[124] Garrett Kamradt. Llmtest_needleinahaystack. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2024. Accessed: 2024-10-24.

[125] Koray Kavukcuoglu. Gemini 2.5: Our most intelligent ai model, March 2025. Accessed: 2025-05-08.

[126] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

[127] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023.

[128] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[129] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[130] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[131] Max Ku, Tianle Li, Kai Zhang, Yujie Lu, Xingyu Fu, Wenwen Zhuang, and Wenhu Chen. Imagenhub: Standardizing the evaluation of conditional image generation models. In *The Twelfth International Conference on Learning Representations*, 2024.

[132] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2403.14468.

[133] Wing-Fung Ku, Wan-Chi Siu, Xi Cheng, and H. Anthony Chan. Intelligent painter: Picture composition with resampling diffusion model, 2022.

[134] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36:71683–71702, 2023.

[135] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024.

[136] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2206–2217, 2023.

[137] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[138] Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*, 2024.

[139] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[140] Dongxu Li, Yudong Liu, Haoning Wu, Yue Wang, Zhiqi Shen, Bowen Qu, Xinyao Niu, Guoyin Wang, Bei Chen, and Junnan Li. Aria: An open multimodal native mixture-of-experts model. *arXiv preprint arXiv:2410.05993*, 2024.

[141] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

[142] Han Li, Xinyu Peng, Yaoming Wang, Zelin Peng, Xin Chen, Rongxiang Weng, Jingang Wang, Xunliang Cai, Wenrui Dai, and Hongkai Xiong. Onecat: Decoder-only auto-regressive model for unified understanding and generation. *arXiv preprint arXiv:2509.03498*, 2025.

[143] Hao Li, Changyao Tian, Jie Shao, Xizhou Zhu, Zhaokai Wang, Jinguo Zhu, Wenhan Dou, Xiaogang Wang, Hongsheng Li, Lewei Lu, et al. Synergen-vl: Towards synergistic image understanding and generation with vision experts and token folding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29767–29779, 2025.

[144] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[145] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[146] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pages 237–255. Springer, 2024.

[147] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[148] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

[149] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

[150] Tianle Li, Max Ku, Cong Wei, and Wenhu Chen. Dreamedit: Subject-driven image editing, 2023.

[151] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024.

[152] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

[153] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2025.

[154] Yunxin Li, Xinyu Chen, Baotian Hu, Longyue Wang, Haoyuan Shi, and Min Zhang. Videovista: A versatile benchmark for video understanding and reasoning. *arXiv preprint arXiv:2406.11303*, 2024.

[155] Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2779–2790, 2025.

[156] Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, et al. Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. *arXiv preprint arXiv:2312.17681*, 2023.

[157] Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. Mogao: An omni foundation model for interleaved multi-modal generation. *arXiv preprint arXiv:2505.05472*, 2025.

[158] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

[159] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

[160] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[161] Haokun Lin, Teng Wang, Yixiao Ge, Yuying Ge, Zhichao Lu, Ying Wei, Qingfu Zhang, Zhenan Sun, and Ying Shan. Toklip: Marry visual tokens to clip for multimodal comprehension and generation, 2025.

[162] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019.

[163] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5404–5411, 2024.

[164] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[165] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next. https://llava-vl.github.io/blog/2024-01-30-llava-next/, 2024. Accessed: 2025-02-14.

[166] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[167] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[168] Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024.

[169] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li St-llm. Large language models are effective temporal learners. *arXiv preprint arXiv:2404.00308*, 2024.

[170] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. *arXiv:2303.04761*, 2023.

[171] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. *arXiv preprint arXiv:2303.04761*, 2023.

[172] Xiangrui Liu, Yan Shu, Zheng Liu, Ao Li, Yang Tian, and Bo Zhao. Video-xl-pro: Reconstructive token compression for extremely long video understanding. *arXiv preprint arXiv:2503.18478*, 2025.

[173] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[174] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*, 2024.

[175] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model 2024. *arXiv preprint arXiv:2401.10166*, 2024.

[176] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

[177] Zhiheng Liu, Xueqing Deng, Shoufa Chen, Angtian Wang, Qiushan Guo, Mingfei Han, Zeyue Xue, Mengzhao Chen, Ping Luo, and Linjie Yang. Worldweaver: Generating long-horizon video worlds via rich perception. *arXiv preprint arXiv:2508.15720*, 2025.

[178] Zhiheng Liu, Weiming Ren, and Wenhu Chen. Tuna-2: Pixel embeddings beat vision encoders for multimodal understanding and generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026. arXiv:2604.24763.

[179] Zhiheng Liu, Weiming Ren, and Wenhu Chen. Tuna: Taming unified visual representations for native unified multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2512.02014.

[180] Zhiheng Liu, Weiming Ren, Haozhe Liu, Zijian Zhou, Shoufa Chen, Haonan Qiu, Xiaoke Huang, Zhaochong An, Fanny Yang, Aditya Patel, et al. Tuna: Taming unified visual representations for native unified multimodal models. *arXiv preprint arXiv:2512.02014*, 2025.

[181] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution. *arXiv preprint arXiv:2409.12961*, 2024.

[182] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[183] Gen Luo, Xue Yang, Wenhan Dou, Zhaokai Wang, Jiawen Liu, Jifeng Dai, Yu Qiao, and Xizhou Zhu. Mono-internvl: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24960–24971, 2025.

[184] Hui Lv and Qianru Sun. Video anomaly detection and explanation via large language models. *arXiv preprint arXiv:2401.05702*, 2024.

[185] Winfried Lötzsch, Max Reimann, Martin Büßemeyer, Amir Semmo, Jürgen Döllner, and Matthias Trapp. Wise: Whitebox image stylization by example-based learning. *ECCV*, 2022.

[186] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *European Conference on Computer Vision*, pages 417–435. Springer, 2024.

[187] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. *arXiv preprint arXiv:2502.20321*, 2025.

[188] Wentao Ma, Weiming Ren, Ge Zhang, and Wenhu Chen. Videoeval-pro: Robust and realistic long video understanding evaluation. *Transactions on Machine Learning Research (TMLR)*, 2026. arXiv:2505.14640.

[189] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. Janusflow: Harmonizing

autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7739–7751, 2025.

[190] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[191] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279, 2022.

[192] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

[193] Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023.

[194] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

[195] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.

[196] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

[197] OpenAI. Chatgpt. https://openai.com/index/chatgpt/, 2023.

[198] OpenAI. Gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[199] OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence, July 2024. Accessed: 2025-05-08.

[200] OpenAI. Introducing gpt-4.1 in the api, April 2025. Accessed: 2025-05-08.

[201] OpenAI. Introducing gpt-5.4. https://openai.com/index/introducing-gpt-5-4/, March 2026. Accessed: 2026-04-24.

[202] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[203] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. Codef: Content deformation fields for temporally consistent video processing. *arXiv preprint arXiv:2308.07926*, 2023.

[204] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3170–3180, 2023.

[205] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

[206] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[207] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[208] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv:2303.09535*, 2023.

[209] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. *arXiv preprint arXiv:2310.15169*, 2023.

[210] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer

for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2545–2555, 2025.

[211] A Yang Qwen, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengpeng Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint*, 2024.

[212] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[213] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[214] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[215] Weiming Ren, Wentao Ma, Huan Yang, Cong Wei, Ge Zhang, and Wenhu Chen. Vamba: Understanding hour-long videos with hybrid mamba-transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2503.11579.

[216] Weiming Ren, Wentao Ma, Huan Yang, Cong Wei, Ge Zhang, and Wenhu Chen. Vamba: Understanding hour-long videos with hybrid mamba-transformers. *arXiv preprint arXiv:2503.11579*, 2025.

[217] Weiming Ren, Harry Yang, Ge Zhang, Cong Wei, Xinrun Du, Stephen Huang, and Wenhu Chen. Consisti2v: Enhancing visual consistency for image-to-video generation. *arXiv preprint arXiv:2402.04324*, 2024.

[218] Weiming Ren, Huan Yang, Jie Min, Cong Wei, and Wenhu Chen. Vista: Enhancing long-duration and high-resolution video understanding by video spatiotemporal augmentation. *arXiv preprint arXiv:2412.00927*, 2024.

[219] Weiming Ren, Huan Yang, Jie Min, Cong Wei, and Wenhu Chen. Vista: Enhancing long-duration and high-resolution video understanding by video spatiotemporal augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2412.00927.

[220] Weiming Ren, Huan Yang, Ge Zhang, Cong Wei, Xinrun Du, Wenhao Huang, and Wenhu Chen. Consisti2v: Enhancing visual consistency for image-to-video generation. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2402.04324.

[221] Alex Rogozhnikov. Einops: Clear and reliable tensor manipulations with einstein-like notation. In *International Conference on Learning Representations*, 2021.

[222] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[223] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[224] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[225] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[226] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[227] Team Seawead, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, et al. Seaweed-7b: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*, 2025.

[228] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

[229] Yan Shu, Peitian Zhang, Zheng Liu, Minghao Qin, Junjie Zhou, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. *arXiv preprint arXiv:2409.14485*, 2024.

[230] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[231] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[232] Vasu Singla, Kaiyu Yue, Sukriti Paul, Reza Shirkavand, Mayuka Jayawardhana, Alireza Ganjdanesh, Heng Huang, Abhinav Bhatele, Gowthami Somepalli, and Tom Goldstein. From pixels to prose: A large dataset of dense image captions. *arXiv preprint arXiv:2406.10328*, 2024.

[233] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[234] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[235] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[236] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[237] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2443–2449, 2021.

[238] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[239] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

[240] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[241] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021.

[242] Hao Tang, Chenwei Xie, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. Unilip: Adapting clip for unified multimodal understanding, generation and editing. *arXiv preprint arXiv:2507.23278*, 2025.

[243] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[244] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[245] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[246] Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

[247] Wan-Video team. Wan: Open and advanced large-scale video generative models (wan2.2), 2025. GitHub repository, Apache-2.0 License.

[248] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. *arXiv preprint arXiv:2104.15069*, 2021.

[249] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9568–9578, 2024.

[250] Tony Cheng Tong, Sirui He, Zhiwen Shao, and Dit-Yan Yeung. G-veval: A versatile metric for evaluating image and video captions using gpt-4o. *arXiv preprint arXiv:2412.13647*, 2024.

[251] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[252] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[253] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[254] Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. *Advances in Neural Information Processing Systems*, 36:46830–46855, 2023.

[255] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, June 2023.

[256] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[257] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation, 2019.

[258] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[259] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[260] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[261] Haochen Wang, Anlin Zheng, Yucheng Zhao, Tiancai Wang, Zheng Ge, Xiangyu Zhang, and Zhaoxiang Zhang. Reconstructive visual instruction tuning. *arXiv preprint arXiv:2410.09575*, 2024.

[262] Haofan Wang, Qixun Wang, Xu Bai, Zekui Qin, and Anthony Chen. Instantstyle: Free lunch towards style-preserving in text-to-image generation. *arXiv preprint arXiv:2404.02733*, 2024.

[263] Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models. *arXiv preprint arXiv:2406.11230*, 2024.

[264] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024.

[265] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[266] Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, and Anthony Chen. Instantid: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*, 2024.

[267] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22958–22967, 2025.

[268] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024.

[269] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[270] Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, et al. Needle in a multimodal haystack. *arXiv preprint arXiv:2406.07230*, 2024.

[271] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *arXiv preprint arXiv:2306.02018*, 2023.

[272] Xiang Wang, Zhifei Zhang, He Zhang, Zhe Lin, Yuqian Zhou, Qing Liu, Shiwei Zhang, Yijun Li, Shaoteng Liu, Haitian Zheng, et al. Hbridge: H-shape bridging of heterogeneous experts for unified multimodal understanding and generation. *arXiv preprint arXiv:2511.20520*, 2025.

[273] Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024.

[274] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[275] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023.

[276] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

[277] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[278] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.

[279] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.

[280] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *The Thirteenth International Conference on Learning Representations*, 2024.

[281] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *International Conference on Learning Representations*, volume 2025, pages 259–271, 2025.

[282] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

[283] Luis Wiedmann, Orr Zohar, Amir Mahla, Xiaohan Wang, Rui Li, Thibaud Frere, Leandro von Werra, Aritra Roy Gosthipaty, and Andrés Marafioti. Finevision: Open data is all you need. *arXiv preprint arXiv:2510.17269*, 2025.

[284] Bichen Wu, Ching-Yao Chuang, Xiaoyan Wang, Yichen Jia, Kapil Krishnakumar, Tong Xiao, Feng Liang, Licheng Yu, and Peter Vajda. Fairy: Fast parallelized instruction-guided video-to-video synthesis. *arXiv preprint arXiv:2312.13834*, 2023.

[285] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*, 2021.

[286] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[287] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025.

[288] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[289] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*, 2024.

[290] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2025.

[291] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.

[292] Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024.

[293] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. *arXiv preprint arXiv:2503.21979*, 2025.

[294] Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. *arXiv preprint arXiv:2312.07537*, 2023.

[295] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified

foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

[296] xAI. Grok-1.5 vision preview. Company news post.

[297] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021.

[298] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13294–13304, 2025.

[299] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[300] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025.

[301] Rongchang Xie, Chen Du, Ping Song, and Chang Liu. Muse-vl: Modeling unified vlm through semantic discrete encoding. *arXiv preprint arXiv:2411.17762*, 2024.

[302] Rongchang Xie, Chen Du, Ping Song, and Chang Liu. Muse-vl: Modeling unified vlm through semantic discrete encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24135–24146, 2025.

[303] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023.

[304] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017.

[305] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[306] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[307] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[308] Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, et al. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*, 2025.

[309] Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

[310] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[311] Jiawei Yang, Tianhong Li, Lijie Fan, Yonglong Tian, and Yue Wang. Latent denoising makes good tokenizers. In *The Fourteenth International Conference on Learning Representations*, 2025.

[312] Shuai Yang, Yifan Zhou, Ziwei Liu, , and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *ACM SIGGRAPH Asia Conference Proceedings*, 2023.

[313] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[314] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[315] Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl3: Towards long image-sequence understanding

in multi-modal large language models. In *The Thirteenth International Conference on Learning Representations*, 2024.

[316] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

[317] Howard Yen. Long-context language modeling with parallel context encoding. Master's thesis, Princeton University, 2024.

[318] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[319] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[320] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134, 2019.

[321] Liping Yuan, Jiawei Wang, Haomiao Sun, Yuchen Zhang, and Yuan Lin. Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding. *arXiv preprint arXiv:2501.07888*, 2025.

[322] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[323] Zhengrong Yue, Haiyu Zhang, Xiangyu Zeng, Boyu Chen, Chenting Wang, Shaobin Zhuang, Lu Dong, KunPeng Du, Yi Wang, Limin Wang, et al. Uniflow: A unified pixel flow tokenizer for visual understanding and generation. *arXiv preprint arXiv:2510.10575*, 2025.

[324] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with

localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[325] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, and Jinhyung Kim. Videomix: Rethinking data augmentation for video classification. *arXiv preprint arXiv:2012.03457*, 2020.

[326] Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. *arXiv preprint arXiv:2311.10982*, 2023.

[327] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, 2023.

[328] Boqiang Zhang, Lei Ke, Ruihan Yang, Qi Gao, Tianyuan Qu, Rossell Chen, Dong Yu, et al. Penguin-vl: Exploring the efficiency limits of vlm with llm-based vision encoders. *arXiv preprint arXiv:2603.06569*, 2026.

[329] David Junhao Zhang, Dongxu Li, Hung Le, Mike Zheng Shou, Caiming Xiong, and Doyen Sahoo. Moonshot: Towards controllable video generation and editing with multimodal conditions. *arXiv preprint arXiv:2401.01827*, 2024.

[330] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[331] Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In Yansong Feng and Els Lefever, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, Singapore, December 2023. Association for Computational Linguistics.

[332] Hongyi Zhang. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[333] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[334] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.

[335] Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. *arXiv preprint arXiv:2404.01258*, 2024.

[336] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*, 2025.

[337] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023.

[338] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023.

[339] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[340] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

[341] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

[342] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

[343] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

[344] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

[345] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[346] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[347] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

[348] Muzhi Zhu, Yang Liu, Zekai Luo, Chenchen Jing, Hao Chen, Guangkai Xu, Xinlong Wang, and Chunhua Shen. Unleashing the potential of the diffusion model in few-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 37:42672–42695, 2024.

[349] Ran Zuo, Haoxiang Hu, Xiaoming Deng, Cangjun Gao, Zhengming Zhang, Yukun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. Scenediff: Generative scene-level image retrieval with text and sketch using diffusion models. 2024.