Learning at Test Time: Adapting Models with Synthetic Data and Environment Interaction

by

Arthur Haonan Chen

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2026

© Arthur Haonan Chen 2026

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 the following two publications, which appear as Chapters 3 and 4, respectively.

Chapter 3 is based on: Arthur Chen and Victor Zhong. "SynQuE: Estimating Synthetic Dataset Quality Without Annotations." *Transactions on Machine Learning Research (TMLR)*, 2026. I was the lead author of this work: I formulated the problem together with my supervisor, designed and implemented the proxy metrics and the LENS method, conducted all experiments, and led the writing of the manuscript. Victor Zhong supervised the project and contributed to the problem formulation, experimental design, and editing of the manuscript.

Chapter 4 is based on: Arthur Chen, Zuxin Liu, Jianguo Zhang, Akshara Prabhakar, Zhiwei Liu, Shelby Heinecke, Silvio Savarese, Victor Zhong, and Caiming Xiong. "Test-Time Adaptation for LLM Agents via Environment Interaction." *International Conference on Learning Representations (ICLR)*, 2026. This work was completed during my internship at Salesforce AI Research. I was the lead author: I designed and implemented the syntactic alignment and dynamics grounding methods, conducted the experiments, and led the writing of the manuscript. The co-authors at Salesforce AI Research provided research mentorship, infrastructure, and feedback on the manuscript; Victor Zhong and Caiming Xiong jointly advised the project.

The framing chapters of this thesis (Chapters 1, 2, 5, and 6) were written solely by me for this thesis.

Abstract

Machine learning models are trained before deployment, yet the context they are deployed into—the data distribution they will serve, or the environment they will act in—is often unavailable during training. This thesis studies *test-time context mismatch*: the setting in which information needed for reliable model behavior is only revealed at deployment time. It investigates how two kinds of test-time information, limited unannotated real data and environment interaction, can be converted into signals for adaptation.

The first contribution addresses mismatch in the training data. We introduce and formalize Synthetic Dataset Quality Estimation (SynQuE), the problem of ranking synthetic datasets by their expected real-world task performance using only limited unannotated real data. We establish the first comprehensive benchmark for this problem by adapting distribution- and diversity-based distance measures as proxy metrics, and we propose Lens, a novel proxy that leverages large language model (LLM) reasoning to characterize the differences between synthetic and real data through natural-language rubrics. Across sentiment analysis, text-to-SQL parsing, image classification, and web navigation, SynQuE proxies correlate with real task performance; on text-to-SQL, selecting the top-3 synthetic datasets by proxy score raises accuracy from 30.4% to 38.4% on average over indiscriminate selection, and Lens consistently outperforms other proxies on complex, long-horizon tasks.

The second contribution addresses mismatch in the deployment environment. We identify two failure modes of LLM agents in novel environments—syntactic misunderstanding of environment-specific formats and semantic misunderstanding of state-transition dynamics—and propose an annotation-free adaptation strategy for each. Online syntactic alignment learns a lightweight adaptation vector during deployment that aligns the agent's output distribution with the environment's syntax at roughly 3% latency overhead. Deployment-time dynamics grounding uses persona-driven exploration to build an in-context world model of the environment's causal dynamics before task execution. Both strategies improve performance across function-calling and web-navigation benchmarks; on the WebArena multi-site split, dynamics grounding raises the agent's success rate from 2% to 23%.

Finally, the thesis develops a unified view of these two adaptation routes, comparing the signals they consume and the timescales they operate on, and showing that both replace labeled supervision with structure available at test time. This view positions test-time context as a practical resource for adaptation precisely in the privacy-sensitive and low-resource settings where labels, demonstrations, and retraining are unavailable.

Acknowledgements

Thank you to everyone who has supported me throughout my research journey with their time, encouragement, and patience. In particular, I would like to thank Prof. Victor Zhong for introducing me to the world of AI agents, and for his tremendous guidance and support throughout this journey, extending far beyond my research to my personal and professional development – and even teaching me how to use exclamation marks correctly. I would like to thank Prof. Shai Ben-David for his insightful discussions and valuable feedback on my first research project. I would also like to thank Prof. Jimmy Lin for giving me my first research opportunity and helping me develop my research infrastructure skills. Last but not least, I would like to thank my parents for their unwavering support and endless encouragement throughout my academic journey. Their belief in me has sustained me through every challenge, and I am forever grateful for everything they have done for me without ever expecting anything in return.

Table of Contents

Author's Declaration ii

Statement of Contributions iii

Abstract iv

Acknowledgements v

List of Figures x

List of Tables xii

List of Abbreviations xvi

List of Symbols xviii

1 Introduction 1

1.1 Motivation: Models are Deployed into Unseen Contexts 1

1.2 Test-time Context Mismatch 2

1.3 Thesis Statement 2

1.4 Contributions 2

1.5 Organization 3

2 Background and Problem Formulation **4**
2.1 From distribution shift to test-time context mismatch 4
2.2 Test-time information as an adaptation signal 5
2.3 Two adaptation routes studied in this thesis 5
2.4 Why privacy-sensitive and low-resource settings matter 6
2.5 Why agents need environment-specific adaptation 6
2.6 Summary 7

3 Selecting Synthetic Data without Annotation **8**
3.1 Introduction 9
3.2 Related Work 10
3.3 The **Syn**thetic Dataset **Qu**ality **E**stimation Problem 13
3.4 SynQuE Proxy Metrics 13
3.4.1 Representation-Based Proxy Metrics 14
3.4.2 LLM-Evaluated Normalized Score (Lens) 16
3.5 Experiments 18
3.5.1 Results Analysis 22
3.5.2 Ablation Studies and Cost Analysis 23
3.6 Practitioner Guidelines 24
3.7 Conclusions, Limitations and Future Work 25
3.8 Broader Impact Statement 25

4 Grounding LLM Agents in New Environments **27**
4.1 Introduction 28
4.2 Related Work 29
4.3 Grounded Test-Time Adaptation Strategies 31
4.3.1 Syntactic Alignment 32
4.3.2 Dynamics Grounding 33

4.4 Experiments . . . 35
4.4.1 Setup . . . 35
4.4.2 Results and Analysis . . . 39
4.4.3 Ablations . . . 41
4.5 Limitations and Future Work . . . 42
4.6 Conclusions . . . 42

5 A Unified View of Learning at Test Time **43**
5.1 What is learned at test time? . . . 43
5.2 Two kinds of test-time signals . . . 44
5.3 Two adaptation timescales . . . 44
5.4 Two forms of supervision-free adaptation . . . 45
5.5 Practical implications . . . 45
5.6 Toward combined systems . . . 46

6 Conclusion **47**
6.1 Summary . . . 47
6.2 Main findings . . . 47
6.3 Future directions . . . 48

References **50**

A Additional Material for SynQuE **69**
A.1 Visualization of ambiguous image . . . 69
A.2 Additional Ablations . . . 70
A.2.1 Ablation Study on LENS Preferential Bias . . . 70
A.2.2 LENS with Different Backbone Models and Sizes . . . 70
A.2.3 Ablation Study on LENS Rubric Design . . . 71
A.2.4 Ablation Study on $\mathbf{MMD}^2$ Kernel Functions . . . 71

A.2.5 Ablation Study on **MDM** with Different Number of Clusters 74
A.2.6 Ablation Study on Encoder Choice for Representation-Based Metrics 74
A.2.7 Correlation of **PAD** Predictions from Different Classifiers vs. F1 Scores 74
A.2.8 Trade-Off Analysis Between LENS and Diversity Measure 75
A.2.9 Analysis on Using Hybrid Proxies for Dataset Selection 76
A.2.10 Ablation on the Number of Samples Used for Scoring on Simple Task 78
A.2.11 Correlation Improvement with De-Hallucination Prompt 79
A.2.12 Ablation on the Size of Synthetic Dataset for SYNQUE Proxies . . . 80
A.2.13 Ablation on Instance-Level Data Selection 80
A.3 Additional Experimental Details 81
A.3.1 Data Curation 81
A.3.2 SYNQUE Scoring 82
A.3.3 Dataset Size Descriptions 83
A.3.4 Standard deviation of correlation coefficients 84
A.3.5 Scatter Plot of LENS Scores and Task Utility 84
A.3.6 SynQuE proxy cost and efficiency analysis 89
A.3.7 Example Rubrics 90
A.3.8 LLM Usage 91

B Additional Material for GTTA **102**
B.1 Experiment details 102
B.2 Prompts used in WebArena experiments 102
B.3 Prompts used in BFCLv3 experiments 114
B.4 Additional results 121
B.5 Environment dynamics analysis 123

List of Figures

3.1 SynQuE uses synthetic data and unlabeled samples of real data to estimate synthetic data quality. Proxy scores are used to rank and select the datasets that lead to the best task performance. 11

4.1 **Overview of syntactic alignment (SA).** This figure includes an example of web navigation shopping task to illustrate how the agent adapts to new environment. (1) At the start of each episode, we initialize an adaptation vector δ as a zero vector and construct inputs to the LLM agent. (2) During task execution, the agent receives environment instructions and observations. (3) At each step, we update the adaptation vector using cross-entropy loss on the current input, and apply the adaptation vector as a bias to the LLM's final hidden layer. This enables rapid alignment to environment-specific observation and action formats. (4) The LLM agent takes a new action with the updated vector, which shifts the model's output distribution to better match the test-time environment. 32

4.2 **Overview of dynamics grounding (DG).** This figure includes an example of web navigation shopping task to illustrate how the pipeline generates environment dynamics in language. (1) We synthesize diverse exploration tasks based on personas using environment descriptions. (2) An exploration agent interacts with the environment to collect interaction logs of state transitions. (3) An LLM extracts and summarizes environment dynamics from these logs. (4) A reasoning model filters less informative rules, which are then used to augment the agent's context during evaluation, enabling more transition-aware decision making. 34

A.1 Image Visualization 69

A.2 Pearson/Spearman correlation coefficients of the hybrid score vs. alpha. Both correlation coefficients are monotonically decreasing as alpha increases (more LENS impact and less **MDM** impact). 75

List of Tables

2.1 Summary of adaptation routes studied in the thesis. 6

3.1 Simplified LENS prompt templates. Appendix A.3 contains detailed prompts for each task. 15

3.2 Top-3 task performance for all candidate proxies of SYNQUE. Top-3 task performance is computed by averaging task performance of synthetic datasets chosen using each proxy metric. Improvements are calculated based on increase over average performance of all synthetic datasets. 20

3.3 Spearman (left) and Pearson (right) correlation scores of SYNQUE proxy metrics. LENS uses a fraction of samples for rubric compilation except for Web Navigation tasks. 21

3.4 Division of the 15 selected ImageNet classes into three 5-class splits for image classification tasks. Each row corresponds to one split used in our experiments. 21

3.6 Spearman (left) and Pearson (right) correlations with different number of real samples for scoring Text2SQL. LENS uses debiased 32B scoring. Note that results in table 3.3 use 30 real samples. "Comp." denotes Computers. 22

3.5 Spearman (left) and Pearson (right) correlations with PERPLEXITY scoring on the BIRD Text2SQL benchmark 22

4.1 Number of tasks per website in the WebArena benchmark. The benchmark consists of six websites, including a multi-site category for tasks that require interacting across multiple websites (from the six sites), for a total of 812 tasks. 36

4.2 Main results on WebArena, WebVoyager, BFCLv3, and Tau-Bench benchmarks. We report task success rates (%) for each model and adaptation method. For Tau-Bench, we average runs across 5 seeds and use a custom more stable codebase. Both test-time adaptation strategies improve performance. "N/A" indicates not applicable—AWM requires training a web-specific world model on state transitions, which is not applicable to Tau-Bench (conversational only) or BFCLv3 (no explicit web-like state transition data); Dynamics grounding does not operate in conversational Tau-Bench as there is no explicit and fixed state transition rules in the environment. . . . 37

4.3 Success rates (%) on the WebArena benchmark across different websites and models. Both adaptation strategies improve performance over baselines. and dynamics grounding (DG) surpasses WMA on `GPT-4o mini`. DG improves success rate substantially on complex multi-site split. 39

4.4 Relative latency gain (%) of PA on BFCLv3 multi-turn. LR fixed to 0.1. . 40

4.5 Ablation on exploration policy and dynamics extractor backbones on WebArena. Here we use different backbones of exploration policy and dynamics extractor. We found for dynamics grounding—using the same LLM agent (itself) improves the same as using a stronger one. 41

5.1 Two kinds of test-time signals and what each adapts. 44

A.1 We use two non-Qwen scoring models on tasks where synthetic data was generated by `Qwen2.5` models. The consistent positive correlations of the debiased scores demonstrate that LENS is robust to this potential bias. . . 70

A.5 Pearson / Spearman correlation coefficients of $\mathbf{MMD}^2$ with different kernel functions. 71

A.2 Comparison of original and debiased LENS scores for ranking synthetic datasets across various backbone models and sizes. For GPT-4.1, sentiment analysis results are reported only due to cost constraints. 72

A.3 LENS with vs. without a rubric on sentiment analysis, showing the rubric's effectiveness. We use `Qwen2.5-7B-Instruct` for scoring. 73

A.4 Bottom: Varying the number of rubric points, where 10 points provides a good balance. 73

A.6 Pearson / Spearman coefficients of **MDM** with different number of clusters (K=3, 5, 10, 20) on sentiment analysis. 74

A.7 Pearson / Spearman correlation coefficients of proxies on sentiment analysis with BGE-M3 as the encoder. 74

A.8 Correlation Coefficients (Pearson / Spearman) of **PAD** predictions from different classifiers vs. F1 scores. 75

A.9 Top 10 hybrid proxies by Spearman (ρ) and Pearson (r) correlation for dataset selection in the Sentiment Analysis (upper block) and Text2SQL (lower block) domains. Each row details a specific weighted combination of **PAD**, **MMD**^{2}, **MDM**, LENS, and MAUVE, alongside its resulting correlation to downstream task performance. For baseline comparison, $\hat{\rho}$ and $\hat{r}$ denote the maximum correlation achieved by the best-performing individual metric from table 3.3, whereas ρ and r denote the correlation achieved by the hybrid combination. 77

A.10 Pearson / Spearman correlation between the LENS score and the test F1 score for different metrics with differet number of real instances used for scoring (100 and 200). 78

A.11 Spearman and Pearson correlations for baseline and with de-hallucination prompt (w/ DHP), across different models. 79

A.12 Scaling analysis of SYNQUE proxies for sentiment analysis. Table entries show Pearson/Spearman correlation coefficients, in the form "Pearson/Spearman", for each proxy at different synthetic dataset size multiples. We report results using debiased LENS, evaluated with `Qwen2.5-(7B/32B)-Instruct` scorers. A size multiple of n indicates that n synthetic datasets (each containing 999 samples) are merged, yielding $n \times 999$ total samples. 80

A.13 Instance-level downstream task performance (%) on sentiment analysis for synthetic datasets selected by each SYNQUE proxy. Debiased LENS uses `Qwen2.5-7B-Instruct` as the backbone model for scoring 81

A.14 Dataset sizes for synthetic ($\mathcal{D}_{\text{s}}$), rubric selection, real subset ($\mathcal{U}_{\text{r}}$), and full real dataset ($\mathcal{D}_{\text{r}}$) for all tasks. 83

A.15 Number of synthetic datasets available per task. 83

A.16 Standard deviation of Spearman (left) and Pearson (right) correlation coefficients of SYNQUE proxy metrics across 5 seeds. **MDM** only scores on synthetic datasets therefore no change in input data. 84

A.17 Comparison of computational cost across different proxy methods in sentiment analysis. Columns: **Emb.** (embedding latency), **Score** (scoring latency), **Total** (end-to-end latency), and **Tokens** (total tokens for LENS, including input and output tokens). Latencies reported in seconds per synthetic dataset (999 samples), with standard deviation in parentheses. LENS uses Qwen2.5 models. 90

B.1 Success rate (SR) of syntactic alignment across different Qwen2.5 instruct model sizes (7B, 14B, 32B) and training hyperparameters. Results are shown for varying numbers of training iterations and two learning rates (LR=0.1 and LR=1.0). Syntactic alignment adaptation generally improves SR over the baseline, with larger models and moderate learning rates yielding higher gains. 121

B.2 Number of environment dynamics per website before and after filtering with `GPT-4.1` as the exploration policy and dynamics extractor. Most environment dynamics are filtered out. 122

B.3 Number of functions available in each BFCLv3 environment. 122

List of Abbreviations

API application programming interface

BFCL Berkeley Function Calling Leaderboard

DG dynamics grounding

GTTA Grounded Test-Time Adaptation

Lens LLM-Evaluated Normalized Score

LLM large language model

LoRA low-rank adaptation

MDM mean distance to medoid

MMD maximum mean discrepancy

MRR mean reciprocal rank

PAD Proxy-A-Distance

SA syntactic alignment

SQL Structured Query Language

SynQuE Synthetic Dataset Quality Estimation

TTA test-time adaptation

VLM vision-language model

WMA world-model-augmented (agent)

List of Symbols

$\mathcal{D}_{\text{r}}$ Real dataset for the target task, assumed scarce and unannotated

$\mathcal{D}_{\text{s}}$ A candidate synthetic dataset

δ Adaptation vector applied as a bias to the final hidden representations of an LLM during syntactic alignment

$E_{\textbf{clean}}$ Filtered set of environment dynamics discovered through exploration, used as an in-context world model

$\mathcal{I}$ LLM agent input, comprising task instruction, current observation, and action history

M Task performance measure of a model on an evaluation dataset

Q Proxy function scoring a synthetic dataset against unannotated real samples

$\mathcal{U}_{\text{r}}$ Small collection of unannotated real-world inputs available at test time

Chapter 1

Introduction

1.1 Motivation: Models are Deployed into Unseen Contexts

Modern machine learning models are typically trained prior to deployment; however, the details of the deployment context are often unavailable during training. This creates a mismatch between what the model has learned during training and what it needs to know at test time.

This mismatch is pervasive in practice. A company may be unwilling or unable to share labeled private data before deployment, so the model provider never observes the distribution the model will ultimately serve. A hospital or bank may operate over proprietary data distributions, workflows, and regulatory constraints that cannot leave the organization [Tan et al., 2024, Jordan and Mitchell, 2015]. A web agent may encounter an unseen website with unfamiliar user-interface elements and hidden state-transition rules [Zhou et al., 2024a, He et al., 2024]. Similarly, a function-calling agent may face application programming interfaces (APIs) or response formats it has never observed during training [Patil et al., 2025, Yao et al., 2024]. In each case, the information required for reliable behavior exists—it is simply not available until the model is deployed.

1.2 Test-time Context Mismatch

Test-time context mismatch occurs when information needed for reliable model behavior is unavailable during training and is only revealed at deployment or evaluation time. Two important forms of mismatch can be distinguished:

Data mismatch: The target data distribution is not fully known during training. For example, a sentiment model may be trained largely on synthetic or out-of-domain data because annotated in-domain data is scarce or private [Ye et al., 2022, Li et al., 2023d].

Environment mismatch: The syntax, interface, or dynamics of the deployment environment are not fully known during training. For example, an LLM agent may misinterpret a website's observation format or mispredict the consequences of its own actions [Yang et al., 2024b, Gur et al., 2023, Chae et al., 2025].

These forms of context mismatch correspond directly to the two main contributions in this thesis.

1.3 Thesis Statement

This thesis studies how machine learning models can adapt using information revealed at test time. It focuses on two sources of test-time information: limited unannotated real data, which can guide synthetic data selection; and environment interaction, which can ground large language models (LLMs) in unfamiliar deployment environments.

1.4 Contributions

The main contributions of this thesis are:

1. Introduction of test-time context mismatch as a unifying perspective for studying post-training adaptation.
2. Formalization of synthetic dataset quality estimation as the problem of ranking synthetic datasets using limited unannotated real data.

3. Presentation of SynQuE (Chapter 3), a framework and benchmark for selecting synthetic datasets without annotations, including representation-based proxies and Lens, an LLM-based proxy for complex tasks.

4. Study of test-time adaptation for LLM agents in novel agentic environments (Chapter 4), identifying syntactic and semantic mismatch as two key failure modes.

5. Presentation of two deployment-time adaptation methods for agents:

   (a) **Syntactic alignment**, which adapts output distributions to environment-specific syntax.

   (b) **Dynamics grounding**, which builds an in-context world model through exploration.

The two primary contributions have been published as peer-reviewed papers: SynQuE (with Victor Zhong), which forms Chapter 3, and Grounded Test-Time Adaptation (with collaborators at Salesforce AI Research and the University of Waterloo), which forms Chapter 4. A detailed account of authorship is given in the Statement of Contributions.

1.5 Organization

The remainder of this thesis is organized as follows. Chapter 2 reviews background on distribution shift and test-time adaptation, and formulates test-time context mismatch. Chapters 3 and 4 present the two main research contributions on synthetic data selection and agent environment adaptation, respectively. Chapter 5 develops a unified view of the two adaptation routes, comparing their signals, timescales, and practical trade-offs. Chapter 6 concludes and suggests directions for future work.

Chapter 2

Background and Problem Formulation

This chapter connects the two research contributions of this thesis under a common problem formulation. It is deliberately concise: detailed treatments of related work specific to synthetic data selection and to agent adaptation appear in Chapters 3 and 4, respectively.

2.1 From distribution shift to test-time context mismatch

The standard experimental paradigm in machine learning studies how model performance degrades under *distribution shift*—when the distribution at test time differs from that seen during training [Quinonero-Candela et al., 2022, Ben-David et al., 2010]. Classical domain adaptation theory bounds the target-domain risk by the source-domain risk plus a measure of divergence between the two domains [Ben-David et al., 2006, 2010], and a large body of work on test-time adaptation updates model parameters at inference time to reduce this gap [Wang et al., 2021, Sun et al., 2020, Niu et al., 2022].

The setting studied in this thesis is sharper than standard distribution shift in several ways. First, the target context is not merely a shifted version of the training distribution; it may involve entirely new interfaces, schemas, or interaction protocols. Second, the target context may be unavailable before deployment: privacy constraints or the sheer novelty of the environment mean that no sample of it can be collected in advance. Third, labels may be unavailable even after deployment, ruling out supervised fine-tuning and labeled validation sets. Finally, the relevant information may be *interactive* rather than static: an agent must act in the environment to observe how it responds [Zhou et al., 2024a, Chae

et al., 2025]. Together, these properties define what Chapter 1 termed *test-time context mismatch*.

2.2 Test-time information as an adaptation signal

Although the deployment context is unknown at training time, it is not unknowable: once deployment begins, a rich set of signals becomes observable. In this thesis, we define "test-time information" broadly to include unannotated real inputs; synthetic datasets generated for the target setting; environment observations; action histories; state transitions; API responses; website structure; and task instructions. None of these signals require human annotation, yet each carries information about the deployment context that the training corpus lacked.

This observation leads to the main claim of the thesis:

> *Adaptation is possible when test-time information*
> *can be converted into a useful signal for model behavior.*

The two contributions of this thesis are two instantiations of this claim, differing in which test-time signal they exploit and in how the signal is converted into adapted behavior.

2.3 Two adaptation routes studied in this thesis

The thesis explores two adaptation routes, summarized in Table 2.1.

In the first route, studied in Chapter 3, the test-time signal is a small collection of unannotated real inputs. The signal is converted into adaptation by scoring candidate synthetic training datasets against the real sample and selecting the dataset expected to yield the best downstream performance. In the second route, studied in Chapter 4, the test-time signal consists of environment observations and state transitions gathered through interaction. The signal is converted into adaptation either parametrically, by aligning the agent's output distribution with the environment's syntax, or non-parametrically, by grounding the agent's planning context in discovered environment dynamics.

Table 2.1: Summary of adaptation routes studied in the thesis.

Route	Test-time signal	Adaptation mechanism	Chapter
Synthetic-data-based	Limited unannotated real data	Select synthetic dataset for training	3
Interaction-based	Environment observations and transitions	Align syntax and ground dynamics	4

2.4 Why privacy-sensitive and low-resource settings matter

Data scarcity presents particular challenges for tasks that require specialized expertise, such as text-to-SQL parsing over proprietary databases or autonomous navigation, where data collection is costly and slow [Yang et al., 2024a, Xie et al., 2024]. Privacy constraints—especially in domains such as healthcare and finance—further complicate data acquisition, limiting the amount of real data available for learning and adaptation [Tan et al., 2024, Jordan and Mitchell, 2015]. Large generative models offer a way out by synthesizing annotated training data [Ye et al., 2022, Xu et al., 2024, Geng et al., 2025], but the quality of synthetic datasets varies widely, and training on a poorly chosen dataset can hurt rather than help [Geng et al., 2025, Li et al., 2023d, Huang et al., 2025]. Selecting among candidate synthetic datasets using only a handful of unannotated real examples is therefore a practical necessity in exactly the settings where real data is hardest to obtain. This motivates the SynQuE framework developed in Chapter 3.

2.5 Why agents need environment-specific adaptation

LLM-based agents inherit broad priors from pre-training, but those priors systematically diverge from any particular deployment environment [Chen et al., 2024, Yang et al., 2025]. Chapter 4 argues that agents often fail in novel environments due to a mismatch between their pre-trained knowledge and the deployed environment's syntax and dynamics: the agent may emit actions in the wrong format for the environment's interface [Yang et al., 2024b, Gur et al., 2023], or mispredict how the environment state changes in response to its actions [Chae et al., 2025, Ha and Schmidhuber, 2018]. Collecting annotated demonstrations for every new environment is expensive [Wang et al., 2024b, Xu et al., 2024], and training a separate world model per environment is computationally heavy [Chae et al., 2025].

Bridging this gap therefore necessitates environment-specific adaptation mechanisms that operate at deployment time using only the agent's own interactions.

2.6 Summary

The next two chapters instantiate this view: Chapter 3 studies adaptation through synthetic data selection, and Chapter 4 studies adaptation through environment interaction. Chapter 5 then returns to the unified perspective and compares the two routes directly.

Chapter 3

Selecting Synthetic Data without Annotation

This chapter is based on the following publication: Arthur Chen and Victor Zhong. "SynQuE: Estimating Synthetic Dataset Quality Without Annotations." Transactions on Machine Learning Research (TMLR), 2026. The paper is presented here in its entirety; supplementary material appears in Appendix A.

Chapter Overview

We introduce and formalize the **Synthetic Dataset Quality Estimation** (SynQuE) problem: ranking synthetic datasets by their expected real-world task performance using only limited unannotated real data. This addresses a critical and open challenge where data is scarce due to collection costs or privacy constraints. We establish the first comprehensive benchmarks for this problem by introducing and evaluating proxy metrics that choose synthetic data for training to maximize task performance on real data. We introduce the first proxy metrics for SynQuE by adapting distribution and diversity-based distance measures to our context via embedding models. To address the shortcomings of these metrics on complex planning tasks, we propose Lens, a novel proxy that leverages large language model reasoning. Our results show that SynQuE proxies correlate with real task performance across diverse tasks, including sentiment analysis, Text2SQL, web navigation, and image classification, with Lens consistently outperforming others on complex tasks by capturing nuanced characteristics. For instance, on text-to-SQL parsing, training on the

top-3 synthetic datasets selected via SynQuE proxies can raise accuracy from 30.4% to 38.4 (+8.1)% on average compared to selecting data indiscriminately. This work establishes SynQuE as a practical framework for synthetic data selection under real-data scarcity and motivates future research on foundation model-based data characterization and fine-grained data selection. We release our code[1].

3.1 Introduction

Data scarcity hinders effective machine learning, especially for tasks requiring specialized expertise like autonomous navigation or natural language interfaces, where data collection is costly and slow [Xie et al., 2024, Yang et al., 2024a]. In sensitive domains such as healthcare and finance [Tan et al., 2024, Jordan and Mitchell, 2015], privacy concerns further complicate data acquisition. Large generative models have emerged as capable synthetic data generators, producing annotated data for tasks like policy learning [Xu et al., 2024], Text2SQL [Yang et al., 2024a], sentiment analysis [Ye et al., 2022, Li et al., 2023d], and image classification [Geng et al., 2025]. While synthetic data can improve real-world performance under scarcity, results vary widely depending on task and data quality [Huang et al., 2025, Geng et al., 2025].

Can we distinguish between high-quality synthetic data that improves real-world task performance and low-quality data that offers little benefit, *without any annotated real data* and *without costly model training*? Crucially, increasing the size of synthetic datasets does not always lead to better downstream performance as it does with real data; in some cases, larger synthetic datasets can even degrade performance, exhibiting inverse scaling trends [Geng et al., 2025, Li et al., 2023d, Setlur et al., 2024, Gao et al., 2022, Møller et al., 2023]. Therefore, selecting a synthetic dataset from a pool of datasets to train on to optimize downstream performance is important.

We introduce **Syn**thetic Dataset **Qu**ality **E**stimation, or SynQuE, the problem of ranking multiple synthetic datasets by quality using only limited unannotated samples of real data. A synthetic dataset A is of higher quality than B if a model trained on A outperforms one trained on B on a real-world test set. This ability is crucial when real data annotation is costly or infeasible. For example, in text-to-SQL parsing, SynQuE helps select the synthetic dataset that yields better generalization from a small set of unannotated real queries. Similarly, for intelligent web agents, it identifies the synthetic interactions that produce agents performing best on real navigation tasks.

[1] https://github.com/r2llab/SynQuE

This work makes three main contributions. 1) We formalize the SynQuE problem and establish the first comprehensive benchmark. As part of this, we introduce and evaluate a suite of *proxy metrics*: computable scores that estimate a synthetic dataset's quality using only a subset of unannotated real data. We adapt proxy metrics using established distributional measures such as mean distance to medoids [Cox et al., 2021], diversity measures such as Proxy-A-Distance [Ben-David et al., 2006], and divergence measures such as MAUVE [Pillutla et al., 2021] — none have been systematically evaluated for the purpose of synthetic data selection. 2) We propose **L**LM-**E**valuated **N**ormalized **S**core (Lens), a novel proxy measure that leverages large language model (LLM) reasoning to create *dataset rubrics* that highlight difference between synthetic data and real data in language. 3) We conduct a comprehensive experiment across diverse domains in sentiment analysis, Text2SQL, web navigation, and image classification in order to evaluate how well these proxy metrics are able to select synthetic data to maximize performance on real test data.

Our empirical evaluation shows that SynQuE proxies exhibit moderate to strong correlation with real task performance across diverse domains including sentiment analysis, Text-to-SQL, image classification, and web navigation. While the best proxy varies by task, most can effectively predict downstream performance without *any* labeled real data, enabling practical synthetic data selection that outperforms indiscriminate synthetic data selection. We find that the reliability of proxies depends on task complexity, with higher variance on noisy data like synthetic images. Among the proxies, Lens, leveraging LLM reasoning and a principled debiasing strategy, consistently achieves superior gains on complex tasks like web navigation by capturing nuanced task details beyond embedding-based metrics. These results establish SynQuE as a robust framework for selecting high-quality synthetic data and motivate future work on stronger foundation model methods to characterize data and perform fine-grained, example-level data selection.

3.2 Related Work

Data synthesis Synthesizing training data with generative models is a promising way to address data scarcity by leveraging instruction-following abilities [Touvron et al., 2023, Ouyang et al., 2022] and vast pre-trained knowledge [Long et al., 2024, Honovich et al., 2022, Mishra et al., 2022]. This has been applied across domains such as text classification [Ye et al., 2022], Text-to-SQL [Yang et al., 2024a, Lei et al., 2024, Li et al., 2024], planning [Sun et al., 2024, Hu et al., 2024, Xu et al., 2024, Murty et al., 2025], and computer vision [Geng et al., 2025, Li et al., 2022]. While synthetic data often improves downstream

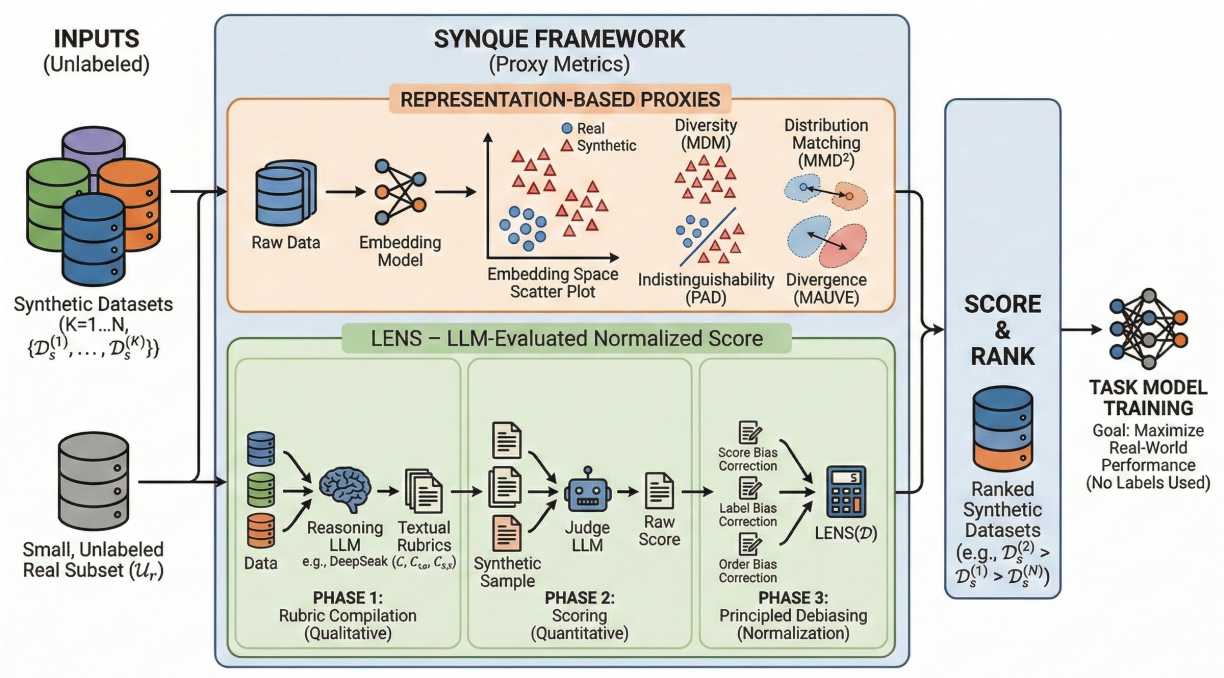


Figure 3.1: SynQuE uses synthetic data and unlabeled samples of real data to estimate synthetic data quality. Proxy scores are used to rank and select the datasets that lead to the best task performance.

models [Liu et al., 2024b, Ye et al., 2022], challenges remain in ensuring quality due to issues such as hallucination [Huang et al., 2025], mode collapse [Goodfellow et al., 2014, Durall et al., 2020, Shumailov et al., 2024], and counterfactual artifacts [Li et al., 2023d, Yu et al., 2023]. Our work introduces the SynQuE framework for selecting synthetic data to maximize task performance without access to large amounts of real data and without training task models.

Distributional and diversity metrics Metrics like Proxy-$\mathcal{A}$-Distance (**PAD**) estimate domain divergence by training classifiers to distinguish source and target data [Ben-David et al., 2010, 2006, Quinonero-Candela et al., 2022]. While effective in some NLP tasks [Elsahar and Gallé, 2019], **PAD** struggles in noisy, complex settings such as agent planning [He et al., 2024]. Diversity metrics like mean-distance-to-medoids (**MDM**)[Cox et al., 2021] and DCScore[Zhu et al., 2025] approximate the coverage of synthetic data. However, these measures do not capture common synthetic data factual inconsistencies from generative model hallucinations [Huang et al., 2025, Li et al., 2023c, Gunjal et al., 2023], which can harm downstream performance [Geng et al., 2025, Yu et al., 2023, Casco-Rodriguez et al., 2023, Li et al., 2023d, Hataya et al., 2023, Briesch et al., 2023].

Data selection and weighting A growing body of work focuses on selecting high-value training samples. Some approaches utilize scaling laws by training regression models on short runs to predict the best data mixture [Liu et al., 2025, Magnusson et al., 2025]. Others employ influence functions or gradient-based heuristics to select influential subsets [Zhang et al., 2024a, Hu et al., 2025b], or utilize small real datasets to learn instance-level weights for synthetic data [Kuo et al., 2025]. Crucially, these methods focus on *instance-level* selection or weighting, often requiring access to labeled validation sets to compute gradients or train selection models. In contrast, SynQuE addresses *dataset-level* ranking in a zero-shot, gradient-free manner, making it suitable for "cold-start" settings where no labeled data exists. Finally, another group of works try to distinguish between human and machine text using divergence measures [Pillutla et al., 2021] or LLM-as-a-judge [Gu et al., 2025, Krumdick et al., 2025, Zheng et al., 2023]. Neither of these two techniques have been studied for synthetic data selection. Furthermore, the latter also suffers from limitations of generative models previously mentioned such as hallucination. Our work establishes the first systematic benchmark and results for synthetic data selection in a practical, fully unsupervised regime.

Dataset Ranking and Selection Most prior work has focused on data selection at the sample-level, aiming to curate optimal subsets of individual data points to reduce training time or labeling costs. For example, Kaushal et al. [2018] leverage submodular functions to maximize diversity and coverage in data subsets, while active learning frameworks like BatchBALD [Kirsch et al., 2019] acquire batches of informative points based on Bayesian mutual information. However, ranking or selecting at the dataset-level is equally important, as it accounts for dependencies and structure among samples within a dataset.

Recent methodologies have expanded selection to this macro scale. For instance, Nadas et al. [2025] introduce a hierarchical framework designed to retrieve entire real-world datasets from massive, decentralized sources under strict computational constraints. Similarly, previous work on dataset retrieval, such as [Chapman et al., 2020], studies data set ranking using keyword-based queries and defines data set quality primarily in terms of search relevance.

In contrast to both instance-level subset curation and the macro-level routing or retrieval of real data, our setting addresses the emerging challenge of selecting *synthetic* datasets under *label scarcity*. We are concerned with ranking candidate model-generated datasets according to their potential downstream utility when annotated validation data is unavailable. Rather than optimizing search relevance or structural efficiency across data silos, our notion of quality relies on unannotated target data to estimate real-world task performance – a fundamentally different objective.

3.3 The Synthetic Dataset Quality Estimation Problem

We define the SynQuE problem and establish notation. Let $\mathcal{D}_{\text{r}} = \{x_{\text{r}}^{(i)}, y_{\text{r}}^{(i)}\}_{i=1}^{n_{\text{r}}}$ be a real dataset, which we assume is scarce and for which labels are unavailable for validation. Instead, we only have access to a small, unannotated collection of real-world inputs $\mathcal{U}_{\text{r}} = \{x_i\}_{i=1}^{m_{\text{r}}} \in \mathcal{D}_{\text{r}}$. Our goal is to use $\mathcal{U}_{\text{r}}$ to estimate the quality of K synthetic datasets $\{\mathcal{D}_{\text{s}}^{(1)}, \ldots, \mathcal{D}_{\text{s}}^{(K)}\}$. These datasets might be generated by different methods and thus vary in quality. We aim to select the synthetic dataset $\mathcal{D}_{\text{s}}^{*}$ that yields the best model performance on $\mathcal{D}_{\text{r}}$, *without using labeled real data or training models on any synthetic dataset.*

Consider a model $f(\cdot; \theta_k)$ trained on the synthetic dataset $\mathcal{D}_{\text{s}}^{(k)}$ with parameters θ_k. Let $M(f, \mathcal{D}_e)$ denote model f's performance on the evaluation dataset $\mathcal{D}_e$, for instance task completion rate or accuracy. The ideal synthetic dataset $\mathcal{D}_{\text{s}}^{*}$ satisfies:

$$\mathcal{D}_{\text{s}}^{*} = \text{argmax}_{\mathcal{D}_{\text{s}}^{(k)}} M\left(f\left(\cdot; \theta_k\right), \mathcal{D}_e\right) \tag{3.1}$$

In practice, Eq 3.1 is infeasible because it requires labeled real data for evaluation and extensive model training across K synthetic datasets. Instead, SynQuE seeks *proxy metrics* $Q(\mathcal{D}_{\text{s}}^{(k)}, \mathcal{U}_{\text{r}})$—computable using only the synthetic dataset $\mathcal{D}_{\text{s}}^{(k)}$ and unannotated real samples $\mathcal{U}_{\text{r}}$—that correlate strongly with true downstream performance. The SynQuE problem thus reduces to designing or learning a proxy function $Q : (\mathcal{D}_{\text{s}}^{(k)}, \mathcal{U}_{\text{r}}) \to \mathbb{R}$ such that a higher score indicates the synthetic dataset $\mathcal{D}_{\text{s}}^{(k)}$ is more likely to produce better real-world task performance. This formulation enables efficient and effective use of synthetic data in low-resource or privacy-sensitive scenarios. For ease of exposition, we refer to the score produced by the proxy function Q as the **SynQuE score**.

3.4 SynQuE Proxy Metrics

In this section, we introduce the proxy metrics designed for SynQuE. To avoid circular reasoning, we explicitly ground these proxies in machine learning theory, linking them directly to our downstream objective in eq. (3.1) through two primary indicators: **distribution matching** and **diversity**.

We ground our similarity metrics (**PAD**, **MMD**^{2}) in domain adaptation theory [Ben-David et al., 2006]. Theoretical bounds show that downstream target risk is upper-bounded by source training risk plus domain divergence. Therefore, ranking synthetic datasets by their divergence from real data mathematically targets the upper bound of the downstream error defined in eq. (3.1).

Similarly, we ground our diversity proxy (**MDM**) in embedding space coverage. If a synthetic dataset lacks diversity (e.g., due to mode collapse), its empirical support is artificially restricted. Models trained on this limited support fail to generalize to the broader real-world target distribution, directly increasing target risk.

We adapt these traditional distance and diversity measures for SynQuE by embedding raw text or image data into continuous representations. Finally, for complex, long-horizon tasks where representation-based metrics fail, we introduce Lens. This LLM-based measure operates directly on raw data, providing deep contextual understanding without relying on a universal embedding model.

3.4.1 Representation-Based Proxy Metrics

Mean Distance to Medoid Our first proxy metric adapts Mean Distance to Medoid (**MDM**), a measure of dataset diversity [Cox et al., 2021, Laliberté and Legendre, 2010, Lehman and Stanley, 2011, Risi et al., 2009]. The rationale is that high-diversity synthetic datasets may offer broader coverage in the embedding space and thus yield higher performance on real data. For SynQuE, **MDM** characterizes this diversity by measuring the sparsity of data points around medoids. Given $\mathcal{D}_\text{s}$ of size n_s, we compute K medoids using clustering algorithms such as kMedoids [2]. Let $\tilde{x}_k$ be the k-th medoid and $\mathcal{C}_k$ be the set of points assigned to this medoid. We aggregate the Euclidean distances from all points within their corresponding cluster: $\textbf{MDM} = \frac{1}{n_s}\sum_{k=1}^{K}\sum_{x_i \in \mathcal{C}_k} d(x_i, \tilde{x}_k)$. Intuitively, if points within each cluster are centered around the medoid, **MDM** will be small, indicating the dataset is less diverse in the embedding space. In contrast, high-diversity synthetic datasets to have broader coverage, and therefore higher performance on real data. A higher **MDM** score suggests greater diversity, so we use it directly as the SynQuE score.

Maximum Mean Discrepancy Next, we propose a proxy based on Maximum Mean Discrepancy ($\textbf{MMD}^2$), a nonparametric test that assesses whether two samples originate from the same distribution [Gretton et al., 2012, Borgwardt et al., 2006, Lu et al., 2022, Li et al., 2015]. As a SynQuE proxy, we use $\textbf{MMD}^2$ to measure the discrepancy between the synthetic dataset and the real data distribution. Given n_s synthetic input samples $x_i \in \mathcal{D}_\text{s}$ of size n_s and m_r unannotated real input samples $y_i \in \mathcal{U}_\text{r}$, $\textbf{MMD}^2$ quantifies the distance between these two empirical distributions in a reproducing kernel Hilbert space using a

[2] https://pypi.org/project/kmedoids/

Listing 3.1: Rubric prompt

```
You are shown samples from
datasets A and B. Give up
to 10 points describing
how dataset B is similar
to dataset A.

Samples from dataset A:
${location A}

Samples from dataset B:
${location B}
```

Listing 3.2: Scorer Prompt

```
Given similarities and differences be
-tween datasets, how likely is the given
sample from dataset ${prediction}? Choo
-se from very unlikely, unlikely,
unsure, likely, and very likely.

Similarities: ${similarities}

Differences: ${differences}

Sample: ${sample}
```

Table 3.1: Simplified LENS prompt templates. Appendix A.3 contains detailed prompts for each task.

kernel function $k(\cdot)$.

$$\mathbf{MMD}^2 = \frac{1}{{m_\mathrm{r}}^2}\sum_{i,j=1}^{m_\mathrm{r}} k(x_i, x_j) + \frac{1}{{n_\mathrm{s}}^2}\sum_{i,j=1}^{n_\mathrm{s}} k(y_i, y_j) - \frac{2}{m_\mathrm{r} n_\mathrm{s}}\sum_{i,j=1}^{n_\mathrm{s}, m_\mathrm{r}} k(x_i, y_j) \tag{3.2}$$

A smaller $\mathbf{MMD}^2$ score indicates the synthetic data distribution is closer to the real one. To maintain consistency with other proxies where higher is better, we use $-\mathbf{MMD}^2$ as the SYNQUE score.

Proxy-A-Distance Our third representation-based proxy adapts Proxy-A-Distance (**PAD**), a discriminative measure from domain adaptation that quantifies the divergence between two distributions [Ben-David et al., 2006, Elsahar and Gallé, 2019]. The **PAD** proxy measures how well a classifier can discriminate between samples from the synthetic dataset $\mathcal{D}_\mathrm{s}$ and real dataset $\mathcal{D}_\mathrm{r}$. We compute $\mathbf{PAD} = 1 - 2\mathcal{E}(G)$ by training a binary domain classifier $G : x \to [0, 1]$ (e.g., a linear SVM, a multi-layer perceptron) to distinguish between synthetic and real inputs (i.e. we do not assume access to labels), where the error of the classifier $\mathcal{E}$ can be computed as:

$$\mathcal{E}\left(G\right) = 1 - \frac{1}{n_\mathrm{s} + m_\mathrm{r}} \sum_{x_i \in \mathcal{D}_\mathrm{s}, \mathcal{U}_\mathrm{r}} \left| G\left(x_i\right) - \mathbb{I}\left(x_i \in \mathcal{D}_\mathrm{s}\right)\right| \tag{3.3}$$

A higher classification error implies the datasets are not easily separable, indicating lower divergence and thus higher synthetic data quality. For consistency with other divergence metrics, use $-\mathbf{PAD}$ as the SYNQUE score.

MAUVE As an established generative baseline to compare against our representation-based proxies, we evaluate MAUVE [Pillutla et al., 2021], a widely used metric that quantifies the divergence between *text* distributions. MAUVE summarizes both Type I and Type II errors. Given a synthetic data distribution Q and a real data distribution P, Type I error means Q generates text that is unlikely under P (unrealistic samples), while Type II error means Q fails to generate text that is plausible under P (lacks diversity). MAUVE captures both error types in a single score, which approaches 1 as the distributions become more similar. Since a higher score indicates better alignment with real data, we utilize the MAUVE score directly as a baseline dataset ranking metric to compare against our proposed SYNQUE proxies.

3.4.2 LLM-Evaluated Normalized Score (LENS)

The representation-based proxies described so far rely on high-quality continuous representations of inputs. In low-resource settings where such representations are unavailable, or in long-horizon settings where it is intractable to compress a long sequence of observations and states into a compact, fixed-size representation, these representation-based proxies may fall short. To address this, we introduce **L**LM-**E**valuated **N**ormalized **S**core (LENS), a novel method that leverages LLMs as zero-shot discriminators. LENS first derives a language **rubric** describing the similarities and differences between samples of unannotated real data $\mathcal{U}_\text{r}$ and inputs from the synthetic dataset $\mathcal{D}_\text{s}$. A subsequent (potentially smaller) LLM then scores how likely each synthetic example is to belong to the real dataset, guided by the rubric. The average score across the synthetic dataset is used as the final SYNQUE score. The intuition is similar to that behind **PAD**: a higher classification error by the rubric-guided scorer implies higher synthetic data quality. We now detail how LENS is computed.

Rubric compilation Given real input samples $\mathcal{U}_\text{r}$, we collect an equal number of samples $\mathcal{U}_\text{s}$ from the synthetic dataset $\mathcal{D}_\text{s}$. Both collections are given to a reasoning LLM (e.g. `DeepSeek R1` or `o4-mini`) to generate three sets of **characteristic descriptions**: commonalities (C), differences of real from synthetic ($C_{\text{r,s}}$), and differences of synthetic from real ($C_{\text{s,r}}$). Listing 3.1 shows a simplified rubric compilation prompt template. Our design is backed by the principled idea of approximating domain divergence through discriminator error [Ben-David et al., 2006]: LENS's scoring is motivated by **PAD**, where the error of a classifier (here, the LLM-based scorer) reflects the distance between distributions. Unlike **PAD**, however, LENS does not require a pretrained encoder to map samples into

fixed-length representations; instead, it operates directly on the native data format and characterizes differences using language rubrics.

Principled Debiasing and Scoring We now describe how to compute the score of a synthetic dataset. A key challenge of scoring is in mitigating LLM biases. We identified three primary sources:

1. **Order Bias:** The set of differences an LLM derives when comparing A to B can differ significantly from when comparing B to A.
2. **Label Bias:** When asked how likely an example x belongs to A or B, an LLM may score both as "very likely", a contradiction.
3. **Score Bias:** LLMs may have an inherent preference for certain score values (e.g., "likely") regardless of the input.

To address these systematically, we employ a *minimal design* involving four scoring permutations for each sample. Specifically, we evaluate the likelihood of each sample belonging to the target dataset (real or synthetic) under both difference rubrics ($C_{\mathrm{s,r}}$ and $C_{\mathrm{r,s}}$). This yields four distinct scores *per sample* x: $g_{\mathrm{r}|C_{\mathrm{s,r}}}(x)$, $g_{\mathrm{s}|C_{\mathrm{s,r}}}(x)$, $g_{\mathrm{r}|C_{\mathrm{r,s}}}(x)$, $g_{\mathrm{s}|C_{\mathrm{r,s}}}(x)$.

We formalize the LLM scoring function as $g_{\mathcal{D}|C} : \mathcal{X} \to \{0, 1, 2, 3, 4\}$, where $x \in \mathcal{X}$ denotes a sample from the input space. The output score (0-4) indicates how likely the example x belongs to dataset $\mathcal{D}$ given characteristic descriptions C in natural language.

First, to mitigate **score bias**, we compute baseline scores by averaging the LLM's judgments on real inputs $x \in \mathcal{U}_{\mathrm{r}}$ for each of the four permutations. For instance, the baseline for scoring an example as real, given the description of how synthetic differs from real, is:

$$z_{\mathrm{r}|C_{\mathrm{s,r}}} = \mathbb{E}\left[g_{\mathrm{r}|C_{\mathrm{s,r}}}(x)\right] \approx \frac{1}{n_{\mathrm{r}}} \sum_{i=1}^{n_{\mathrm{r}}} g_{\mathrm{r}|C_{\mathrm{s,r}}}(x_i) \tag{3.4}$$

We then compute a *score-debiased* score h for each synthetic sample by normalizing its raw score against this baseline:

$$h_{\mathrm{r}|C_{\mathrm{s,r}}}(x) = \frac{g_{\mathrm{r}|C_{\mathrm{s,r}}}(x)}{\max(\epsilon, z_{\mathrm{r}|C_{\mathrm{s,r}}})} \tag{3.5}$$

Here, ϵ is a small constant to avoid division by zero. Intuitively, this scores-debiased score expresses how much the LLM scores the example compared to how it usually scores real

examples. Similarly, the score-debiased score for the synthetic label is computed as:

$$h_{\mathrm{s}|C_{\mathrm{s,r}}}(x) = \frac{g_{\mathrm{s}|C_{\mathrm{s,r}}}(x)}{\max(\epsilon, z_{\mathrm{s}|C_{\mathrm{s,r}}})} \tag{3.6}$$

where $z_{\mathrm{s}|C_{\mathrm{s,r}}} = \frac{1}{n_{\mathrm{r}}} \sum_{i=1}^{n_{\mathrm{r}}} g_{\mathrm{s}|C_{\mathrm{s,r}}}(x_i)$. Next, to compute a **label-debiased** score, we normalize the LLM's preference for the "real" label over the "synthetic" label for each synthetic example:

$$p_{\mathrm{r}|C_{\mathrm{s,r}}}(x) = \frac{h_{\mathrm{r}|C_{\mathrm{s,r}}}(x)}{h_{\mathrm{r}|C_{\mathrm{s,r}}}(x) + h_{\mathrm{s}|C_{\mathrm{s,r}}}(x) + \epsilon} \tag{3.7}$$

Finally, to create an **order-debiased** error score for a synthetic sample $x \in \mathcal{D}_{\mathrm{s}}$, we average the label-debiased scores indicating a "real" prediction using both sets of difference descriptions ($C_{\mathrm{s,r}}$ and $C_{\mathrm{r,s}}$):

$$\hat{p}_{\mathrm{s}}(x) = \frac{1}{2}\left[p_{\mathrm{r}|C_{\mathrm{s,r}}}(x) + p_{\mathrm{r}|C_{\mathrm{r,s}}}(x)\right] \tag{3.8}$$

Conversely, we compute a symmetrical error score for each real sample $x \in \mathcal{U}_{\mathrm{r}}$, representing the likelihood it is misclassified as synthetic. Using corresponding baselines normalized against the synthetic dataset, this is calculated as:

$$\hat{p}_{\mathrm{r}}(x) = \frac{1}{2}\left[p_{\mathrm{s}|C_{\mathrm{s,r}}}(x) + p_{\mathrm{s}|C_{\mathrm{r,s}}}(x)\right] \tag{3.9}$$

The final LENS score is the combined, weighted empirical mean of these classification errors across both datasets, capturing the overall indistinguishability between the real and synthetic samples:

$$\textsc{Lens}(\mathcal{D}_{\mathrm{s}}) = \frac{1}{n_{\mathrm{s}} + n_{\mathrm{r}}}\left(\sum_{i=1}^{n_{\mathrm{s}}} \hat{p}_{\mathrm{s}}(x_i) + \sum_{j=1}^{n_{\mathrm{r}}} \hat{p}_{\mathrm{r}}(x_j)\right) \tag{3.10}$$

3.5 Experiments

We choose four diverse tasks spanning different machine learning domains to examine how well each candidate proxy metric extrapolates to real data performance on tasks with varying complexities and modalities. For LENS, we incorporate `Deepseek-R1` to generate 10 points about similar and different characteristics $C_{\mathrm{s,r}}$ between synthetic data samples $\mathcal{U}_{\mathrm{s}}$ and real

data samples $\mathcal{U}_{\text{r}}$. We then use `Qwen2.5-32B-Instruct` [Qwen et al., 2025] and `8B` to score synthetic examples according to the rubric. For image domain, we use OpenAI `o4-mini` to compile rubrics and `Qwen2.5-VL-32B-Instruct` to score. For representation-based metrics **PAD**, **MDM**, and **MMD**^{2}, we use state-of-the-art `qte-Qwen2-7B-Instruct` to embed text inputs and `E5-V` [Jiang et al., 2024] to embed image inputs for proxy scoring. We use XGBoost [Chen and Guestrin, 2016] to compute **PAD** and polynomial kernel for **MMD**^{2} [Gretton et al., 2012]. We include additional kernel ablations in Appendix A.5. We use the official release[3] from MAUVE, with the default hyperparameter setting for MAUVE calculation.

We also include an experiment with perplexity-based metric as additional baseline in Text2SQL (see table 3.5). PERPLEXITY, inspired by scaling law methods [Magnusson et al., 2025, Liu et al., 2025], fine-tunes a model on each synthetic data set and measures its perplexity on the unannotated real data subset; a lower perplexity is expected to indicate higher quality.

We use Pearson [Pearson and Galton, 1997] and Spearman rank [Spearman, 1904] correlation coefficients to measure how strongly task performance and proxy scores are related. Pearson focuses on *predictability* by capturing linear relationships, while Spearman focuses on *trend* by evaluating whether the relationship between variables is consistently increasing or decreasing (i.e., monotonic), regardless of the exact shape. To reduce variance in correlation analysis across different sample subsets, for all tasks, we construct subsets $\mathcal{U}_{\text{r}}$ by sampling with five different seeds. Final correlation scores are averaged across seeds.

Sentiment Analysis Recent work shows LLMs overfit widely-used datasets due to data contamination [Balloccu et al., 2024, Sainz et al., 2023, Oren et al., 2023]. To mitigate this, we evaluate on a domain-specific financial tweets sentiment dataset[4]. We create 32 synthetic class-balanced datasets (998 samples each) using eight prompt types: zero-shot, zero-shot with background knowledge, with train-time or test-time stock ticker info, and few-shot variants. We use `Qwen2.5-7B-Instruct`, `Qwen2.5-32B-Instruct`, `Llama3.1-8B-Instruct`, and `Llama3.3-70B-Instruct` [Grattafiori et al., 2024] for each prompt type. Background knowledge uses detailed guideline instructions for better task alignment. Stock tickers are sampled one at a time for synthesis. Details are in section A.3. We train task models using XGBoost and evaluate F1 score on a 2,388-item test set. Rubrics are compiled by randomly sampling 200 points from each real and synthetic dataset.

[3] https://pypi.org/project/mauve-text/

[4] https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment

Table 3.2: Top-3 task performance for all candidate proxies of SynQuE. Top-3 task performance is computed by averaging task performance of synthetic datasets chosen using each proxy metric. Improvements are calculated based on increase over average performance of all synthetic datasets.

		Top-3 Ranked Average Task Performance across Proxy Metrics							
Tasks	Test	Lens 7B		Lens 32B		**PAD**	**MMD^2**	**MDM**	**Mauve**
	mean	debiased	biased	debiased	biased				
Sentiment	49.6	50.5 +0.8	51.2 +1.6	52.0 +2.4	51.0 +1.4	55.3 +5.7	54.7 +5.1	54.2 +4.6	54.6 +4.9
Text2SQL									
Computer	45.8	46.6 +0.7	46.4 +0.6	48.3 +2.5	46.3 +0.5	48.3 +2.5	47.4 +1.6	48.2 +2.3	48.3 +2.5
Apps	30.4	34.7 +4.4	36.3 +5.9	33.8 +3.4	35.2 +4.9	33.5 +3.2	38.4 +8.1	33.9 +3.5	38.4 +8.1
Movies	37.3	41.0 +3.7	41.4 +4.1	43.8 +6.5	44.7 +7.4	44.2 +6.9	43.0 +5.7	46.9 +9.6	44.6 +7.3
Average	37.8	40.8 +2.9	41.4 +3.5	42.0 +4.1	42.1 +4.3	42.0 +4.2	42.9 +5.1	43.0 +5.2	43.8 +6.0
Image									
Split 1	57.2	57.3 +0.2	56.7 -0.4	56.4 -0.8	53.4 -3.7	55.9 -1.3	57.3 +0.1	57.0 -0.1	56.0 -1.1
Split 2	55.8	55.3 -0.4	55.8 +0.0	56.2 +0.4	56.0 +0.2	55.4 -0.4	54.5 -1.3	54.8 -1.0	56.3 +0.5
Split 3	57.7	59.1 +1.4	58.7 +1.0	60.2 +2.5	57.0 -0.7	58.2 +0.6	64.1 +6.5	52.2 -5.5	58.4 +0.7
Average	56.9	57.2 +0.4	57.0 +0.2	57.6 +0.7	55.5 -1.4	56.5 -0.4	58.6 +1.8	54.7 -2.2	56.9 +0
WebNav	25.8	26.5 +0.7	26.3 +0.5	26.3 +0.5	26.0 +0.2	25.7 -0.1	26.5 +0.7	25.8 -0.1	26.3 +0.5

Text2SQL We evaluate SynQuE on Text2SQL using three DBs from the BIRD benchmark (Movies, App Store, Computer Students — we denote the last two as Apps and Computers) [Li et al., 2023a]. We synthesize 1,000 data points with 4 prompt types: zero-shot with background knowledge (guidelines and schema), zero-shot with test-time info (random table rows), and few-shot (three examples). `Qwen2.5-7B-Instruct` and `Llama3.1-8B-Instruct` models are used for dataset generation. Task models are finetuned from `Qwen2.5-Coder-1.5B-Instruct` following CodeS[5] and evaluated using execution accuracy on the real test set. For rubrics, we sample 30 points per synthetic and real dataset. Real data sizes are 60, 69, and 164 for Apps, Computers, and Movies respectively.

Image Classification In addition to text-only settings, we evaluate SynQuE on image classification using synthetic datasets curated from `unmet-promise`[6]. These datasets are created with different prompts using Stable Diffusion 1.1 and 1.5: label, label plus physical relation, and label plus background description [Geng et al., 2025]. Images are mapped to ImageNet classes [Deng et al., 2009] via caption analysis A.3, then filtered with a vision-

[5] https://github.com/RUCKBReasoning/codes
[6] https://huggingface.co/datasets/scottgeng00/unmet-promise

Table 3.3: Spearman (left) and Pearson (right) correlation scores of SynQuE proxy metrics. Lens uses a fraction of samples for rubric compilation except for Web Navigation tasks.

Tasks	Lens 7B debiased		Lens 7B biased		Lens 32B debiased		Lens 32B biased		**PAD**		**MMD^2**		**MDM**		Mauve	
Sentiment	.25	.33	.26	.17	.38	.26	.24	.23	.53	.65	.45	.67	.68	.85	.53	.57
Text2SQL																
Computer	.19	.13	.10	.10	.41	.45	.18	.23	.46	.69	.33	.85	.39	.63	.24	.78
Apps	.38	.37	.42	.49	.46	.40	.55	.61	.43	.42	.53	.79	.44	.56	.74	.52
Movies	.41	.50	.41	.26	.50	.46	.56	.47	.50	.64	.38	.46	.61	.41	.65	.68
Average	.33	.33	.31	.28	.46	.43	.43	.44	.46	.58	.41	.70	.48	.53	.55	.66
Image																
Split 1	-.18	-.19	-.30	-.35	-.28	-.28	-.68	-.67	-.06	-.05	.66	.52	-.37	-.27	-.04	-.20
Split 2	.02	-.04	.14	.05	.20	.05	.20	.05	.20	.31	.09	.17	.03	-.32	.03	.13
Split 3	-.10	-.01	-.15	-.03	.31	.33	.37	.44	.02	-.15	.26	.21	-.54	-.76	.46	.34
Average	-.09	-.08	-.10	-.11	.08	.03	-.04	-.06	.05	.04	.33	.30	-.30	-.45	.15	.09
WebNav	.15	.17	.11	.18	.15	.15	.08	.09	.11	.08	-.02	.06	-.11	-.08	-.09	-.10

language model to remove noisy labels. Data cleaning details are provided in section A.3. The final set includes 15 classes with 300 images each. Due to limited samples, the 15-class task is split into three 5-class tasks (table 3.4). We train ResNet-50 [He et al., 2015] from scratch for 50 epochs, early stopping on 10% validation data. Evaluation uses mean reciprocal rank (MRR) for finer performance measurement. Rubrics are constructed from 100 sampled images per real and synthetic data, consistent across SynQuE methods.

Web Navigation Our fourth task evaluates SynQuE on agentic web navigation planning using WebVoyager [He et al., 2024] and synthetic data from NNetNav [Murty et al., 2025]. Inputs include task objectives, current step observations (accessibility tree), and past actions; targets are actions leading to task success. WebVoyager has 15 websites; we exclude Google Flights and Booking which are no longer feasible [Zhou et al., 2024c, Murty et al., 2025], leaving 13 sites with 557 tasks. Each site forms a test domain split into 5 synthetic subsets. Models are fine-tuned with LoRA [Hu et al., 2021] on `Qwen2.5-7B-Instruct`. We use all synthetic and 20 real samples per method.

Table 3.4: Division of the 15 selected ImageNet classes into three 5-class splits for image classification tasks. Each row corresponds to one split used in our experiments.

Splits	ImageNet classes
1	bra, mask, lion, cloak, tank
2	hammer, backpack, stage, throne, tray
3	plate, desk, kimono, shield, church

Table 3.6: Spearman (left) and Pearson (right) correlations with different number of real samples for scoring Text2SQL. LENS uses debiased 32B scoring. Note that results in table 3.3 use 30 real samples. "Comp." denotes Computers.

	$\lvert\mathcal{U}_r\rvert = 25$										$\lvert\mathcal{U}_r\rvert = 50$									
	LENS		**PAD**		**MMD²**		**MDM**		**Mauve**		**LENS**		**PAD**		**MMD²**		**MDM**		**Mauve**	
Comp.	.22	.22	.36	.65	.28	.80	.39	.65	.19	.65	.64	.38	.51	.78	.34	.87	.39	.62	.87	.78
Apps	.29	.48	-.20	-.23	.40	.77	.44	.56	.65	.70	.64	.66	.68	.76	.57	.80	.44	.56	.32	.65
Movies	.33	.48	.33	.36	.52	.57	.60	.40	-.14	-.35	.43	.49	.57	.37	.81	.56	.67	.45	.81	.60
Average	.28	.39	.16	.26	.40	.71	.47	.54	.23	.33	.57	.51	.59	.64	.57	.74	.50	.55	.67	.68

3.5.1 Results Analysis

SYNQUE proxies correlate with task performance and improve selection. table 3.3 shows that SYNQUE proxy metrics demonstrate moderate to strong correlation with downstream task performance. To show the practical utility of these proxies, we simulate selecting the top 3 datasets based on each metric's score and compare their average task performance against the mean performance of all available synthetic datasets. As shown in table 3.2, nearly all proxy metrics significantly improve dataset selection over selecting synthetic datasets non-discriminately (i.e. uniform selection). This demonstrates that SYNQUE is an effective framework for maximizing real-data performance, despite not having access to labeled real data and only a limited sample of real data. We also conduct an experiment with PERPLEXITY on Text2SQL, to examine the effectiveness as a potential proxy. As shown in table 3.5, it correlates poorly with even text data, therefore we conclude that scaling methods would not work under our setting, where no annotated data is available for evaluation to build a regression model that predicts the best data mixture.

Performance on ambiguous image data shows high variance. The synthetic image classification data contains significant visual variability and label ambiguity, especially in Split 2 between classes like "stage" and "throne" (fig. A.1a, fig. A.1b). This confuses most proxy metrics, resulting in inconsistent correlations across splits (table 3.3). However, table 3.2 shows that when used for selection, several proxies (e.g. debiased LENS 32B, **MMD²**) still improve average task performance.

Table 3.5: Spearman (left) and Pearson (right) correlations with PERPLEXITY scoring on the BIRD Text2SQL benchmark

	PERPLEXITY	
Computers	-.31	-.33
Apps	-.25	-.29
Movies	.24	.31
Average	-.11	-.10

Using more real samples improves correlation. As shown in table 3.6, increasing the number of unannotated real

samples m_r consistently leads to stronger correlations for all
proxy metrics. This indicates that even a modest increase in available real-world data can significantly improve the reliability of synthetic data quality estimation.

LENS excels on complex, long-horizon tasks. As shown in both tables, the 32B debiased LENS is the only proxy that consistently achieves positive correlation and improves top-3 task performance across all tasks and splits. Its advantage is particularly pronounced in web navigation, a complex planning task where representation-based metrics struggle. LENS leverages LLM reasoning over rich, structured inputs like accessibility trees to generate interpretable rubrics. For instance, an example characteristic point for the website "Wolfram Alpha" is: "Dataset B tasks focus on data retrieval (e.g. temperature anomalies, moon phases) while Dataset A emphasizes applied computational problem-solving". These specific nuances in long text (e.g. instructions, state observations) are difficult to capture using general-purpose dense vector embedders, which explains why LENS outperforms representation-based proxy metrics on complex, abstract tasks like web navigation. This method does exhibit weaker correlation in image classification. We hypothesize that this stems from the inability of VLMs to capture meaning characteristics descriptions in batches of images during rubric generation, and VLM rubric generation will likely improve as VLMs improve in quality.

3.5.2 Ablation Studies and Cost Analysis

Cost-Effectiveness of SYNQUE Proxies A key motivation for SYNQUE is efficiency. Our representation-based proxies require a one-time embedding computation, after which scoring all datasets is nearly instantaneous (e.g., 19 seconds for MMD on 32 datasets). LENS, using modern LLM serving frameworks, is also highly efficient, taking ~ 15 seconds per dataset with a 32B model on a H200 GPU. In contrast, perplexity-based data selection, inspired by scaling-law studies [Liu et al., 2025], require training many (e.g., 512 1M models used in their experiment) small models on the mixture of all synthetic datasets, a significantly more costly procedure, yet yield weaker correlations (table 3.3). This highlights the practical advantage of the SYNQUE framework.

Larger scorers lead to stronger correlations We find that larger scoring models yield stronger correlations between LENS and task performance, as shown in table 3.3. Intuitively, this is expected because larger models generally possess more robust instruction-following

and reasoning capabilities, enabling them to better assess data quality and align proxy scores with downstream performance.

LENS is robust to preferential bias in LLM training data. To address concerns that an LLM evaluator might favor data it generated, we tested LENS with different scoring models on data generated by `Qwen2.5`. The results, detailed in table A.1, show that performance is consistent across evaluators, including those distinct from models used to generate the synthetic data, indicating minimal preferential bias.

Principled debiasing and rubrics are critical for LENS. As illustrated in table 3.3, the correlations between LENS and task performance consistently increase when debiasing is applied, indicating that raw scores may be systematically biased and do not reliably reflect true data quality. Once debiasing is introduced, the correlation becomes strongly positive and consistently outperforms the biased scores. This demonstrates that debiasing effectively corrects for these systematic errors and aligns LENS scores with actual task performance. Further ablations show that using a rubric consistently improves correlation over a zero-shot baseline (table A.3), and that 10 rubric points generally offer the best trade-off between specificity and generality (table A.4).

3.6 Practitioner Guidelines

Based on our empirical evaluation including ablation studies across diverse domains and experiment setups, we offer the following guidelines for practitioners utilizing the SYNQUE framework to select synthetic data.

- **Simple, One-shot NLP Tasks (e.g., Sentiment Analysis):** Representation-based metrics like **PAD** and **MDM** serve as effective and *computationally efficient* defaults. These metrics require only a one-time embedding computation and demonstrate remarkable robustness, maintaining strong correlations with task performance even with as few as 100 real samples.
- **Complex, Long-horizon Tasks (e.g., Agentic, Large-Scale Datasets):** We highly recommend utilizing LENS. General-purpose dense vector embedders often struggle to capture the specific nuances and sequential logic present in long texts like state observations; LENS overcomes this by leveraging LLM reasoning to generate interpretable rubrics over structured inputs.

- **Vision Tasks:** Selecting high-quality data is particularly challenging due to significant visual variability and label ambiguity. In these settings, distribution matching via $\mathbf{MMD}^2$ provides a solid baseline that can still improve dataset selection, though it should be paired with explicit data cleaning pipelines – such as utilizing vision-language models to filter out noisy or hallucinated images.
- **Safeguarding Against Generative Issues:** To safeguard against common generative issues like mode collapse, **MDM** should be utilized to measure dataset diversity. If a synthetic dataset lacks diversity, its empirical support is artificially restricted; therefore, prioritizing higher MDM scores ensures broader coverage in the embedding space and better generalization to the real-world target distribution.

3.7 Conclusions, Limitations and Future Work

We formalized the SynQuE problem of ranking synthetic datasets by their impact on real-world task performance using limited unannotated real data. Our comprehensive evaluation established that various proxies can reliably predict downstream performance, offering a cost-effective alternative to full model training. We proposed Lens, a novel proxy leveraging LLM reasoning and principled debiasing, which consistently outperforms others on challenging, long-horizon tasks. Overall, SynQuE offers a robust framework for synthetic data selection when labeled real data is scarce.

SynQuE assumes that real data is scarce, a setting not all deployments face. While Lens performs well on the complex tasks studied, its effectiveness should be validated on more diverse tasks. Additionally, we experiment with limited-size LLMs due to resource constraints. Future work should explore 1) scaling Lens to larger sizes and different architectures, especially strong VLMs, to assess generality and improvements. 2) using rubric feedback to guide LLMs in synthesizing more realistic data, and 3) developing fine-grained, example-level proxy use to directly improve task model training.

3.8 Broader Impact Statement

While utilizing synthetic datasets offers a promising solution to data scarcity, it could introduce ethical considerations regarding the potential amplification of biases. It is crucial to clarify that our SynQuE framework does not select datasets by directly optimizing for downstream test performance, as such labels are strictly inaccessible during the selection

phase. Instead, proxies such as **PAD**, **MMD**^{2}, and MAUVE evaluate synthetic data based strictly on its **unsupervised distributional similarity** to a small real-world reference set; test performance is utilized solely as a post-hoc oracle to evaluate the success of these proxies. Consequently, the risk of amplifying known biases depends entirely on the composition of this real-world anchor set. If the reference data contains historical or representational biases, similarity-based selection will inherently prioritize synthetic datasets that reflect and perpetuate those same biases. To mitigate this risk, we strongly advise practitioners to rigorously audit their real-world reference samples for representational fairness and equity prior to applying SYNQUE, ensuring that the anchor data aligns with desired ethical standards.

Chapter 4

Grounding LLM Agents in New Environments

This chapter is based on the following publication: Arthur Chen, Zuxin Liu, Jianguo Zhang, Akshara Prabhakar, Zhiwei Liu, Shelby Heinecke, Silvio Savarese, Victor Zhong, and Caiming Xiong. "Test-Time Adaptation for LLM Agents via Environment Interaction." International Conference on Learning Representations (ICLR), 2026. Work done during an internship at Salesforce AI Research. The paper is presented here in its entirety; supplementary material appears in Appendix B.

Chapter Overview

Large language model (LLM)-based agents struggle to generalize to novel and complex environments, such as unseen websites or new sets of functions, due to a fundamental mismatch between their pre-training and test-time conditions. This challenge stems from two distinct failure modes: a syntactic misunderstanding of environment-specific components like observation formats, and a semantic misunderstanding of state-transition dynamics, which are only revealed at test time. To address these issues, we propose two distinct strategies for adapting LLM agents by leveraging environment-specific information from interaction that is available during deployment. First, an *online syntactic alignment* (SA) method parameterizes environmental nuances by learning a lightweight adaptation vector that biases the model's output distribution, enabling rapid alignment with an environment response format. Second, a *deployment-time dynamics grounding* (DG) method employs

a persona-driven exploration phase to systematically probe and learn the environment's causal dynamics before task execution, equipping the agent with an in-context world model. We evaluate these strategies across diverse agentic benchmarks, including function calling and web navigation. Our empirical results show the effectiveness of both strategies across all benchmarks with minimal computational cost. We find that dynamics grounding is particularly effective in complex environments where unpredictable dynamics pose a major obstacle, demonstrating a robust path toward more generalizable and capable LLM-based agents. For example, on the WebArena multi-site split, this method increases the agent's success rate from 2% to 23%. We release our code[1].

4.1 Introduction

Large language model (LLM)-based AI agents have demonstrated remarkable capabilities across diverse tasks [Li et al., 2023a, Zhou et al., 2024a, Patil et al., 2025, Xie et al., 2024], yet they often fail to generalize when deployed in novel agentic environments such as web navigation [Zhou et al., 2024a, He et al., 2024] and function calling [Patil et al., 2025, Yao et al., 2024]. These failures reflect two core challenges—the systematic mismatch between the agent's pre-trained knowledge and the specific syntax and dynamics of the deployed environment [Yang et al., 2025, Thil et al., 2024]. Although both mismatches stem from a lack of prior knowledge, we distinguish them operationally because they require different adaptation mechanisms (syntactic vs. semantic). To illustrate these two failure patterns, we consider an LLM agent tasked with booking a flight on a previously unseen travel website:

1. **Syntactic Mismatch**: the LLM agent's prior knowledge does not align with environment-specific information, such as observation structure [Yang et al., 2024b, Gur et al., 2023] and the environment's unique syntax [Lei et al., 2024, Chen et al., 2024]. This causes parsing and context-understanding issues: a pre-trained LLM agent might try `click("Search")` or target `destination`, while the new site exposes `Go` and `dest_field`, producing invalid actions and immediate failures. [Yang et al., 2024b, Gur et al., 2023, Lei et al., 2024, Chen et al., 2024]

2. **Semantic Mismatch**: the agent lacks an accurate, environment-specific causal model of state transitions, so it cannot predict the consequences of actions. For example, the agent may expect `click("Go")` to show flight results, but the site instead opens a date-confirmation pop-up; without knowing this transition, the agent cannot form the correct multi-step plan and fails. [Forrester, 1971, Chae et al., 2025, Zhang et al., 2024b]

[1] https://github.com/r2llab/GTTA

Current approaches to this generalization gap are ill-suited for rapid adaptation in novel environments. In addition, many methods require human-annotated or LLM-annotated demonstrations [Wang et al., 2024b, Murty et al., 2025, Xu et al., 2024], which can be resource-intensive or rely on LLM's prior knowledge about the environment. On the other hand, to address semantic mismatch, explicit world modeling like [Chae et al., 2025] involves a heavyweight pipeline of extensive data collection and fine-tuning a separate, specialized model that is computationally expensive and struggles to generalize without being retrained. Both paradigms present significant overhead, highlighting the need for more efficient strategies that can ground an agent using only the information available at test time.

To address the two failure modes and close the generalization gap, we propose two annotation-free strategies that operate at deployment time. (1) Online syntactic alignment (SA)—a lightweight module that learns a small per-interaction bias (an adaptation vector applied to late features) to quickly align an agent's output distribution with environment-specific syntax. Observing `Go` and `dest_field` on the page, for example, lets the adapter steer action generation toward the site's actual element names. (2) Dynamics grounding (DG)—a short, persona-guided exploration phase that iteratively collects in-context rules about state transitions (e.g., clicking `Go` opens a date pop-up) and supplies those rules as context for downstream planning. Both methods require only test-time observations and no trajectory annotations or expensive fine-tuning.

We evaluate our methods on function-calling and web-navigation benchmarks. Both approaches yield consistent improvements: syntactic alignment gives efficient, steady gains across sites, while dynamics grounding produces particularly large improvements on environments with unfamiliar transitions. For example, on the WebArena multi-site split [Zhou et al., 2024a], dynamics grounding raises `GPT-4.1` success from 2% to 23%. Taken together, syntactic and semantic test-time grounding substantially narrow the deployment generalization gap and offer a practical path toward more robust LLM agents.

4.2 Related Work

Test-Time Adaptation Originating in computer vision, test-time adaptation (TTA) addresses distributional shifts between training and testing data by adapting models at inference time [Wang et al., 2021, Niu et al., 2022, Sun et al., 2020]. Our syntactic alignment builds directly on methods that update model parameters or steering vectors at test time to minimize an unsupervised objective, such as entropy minimization or self-supervised learning [Wang et al., 2021, Hu et al., 2025a, Li et al., 2023b]. Our syntactic alignment aligns

with steering vector research [Tan et al., 2025, Sinii et al., 2025, Panickssery et al., 2024, Turner et al., 2024] in its use of latent space biases to shift output distributions. However, standard steering approaches generally apply modulation vectors to shift high-level traits [Tan et al., 2025], such as honesty [Zou et al., 2025]. We distinguish our work by tailoring this mechanism to the dynamic nature of agentic tasks: our vector is treated as a temporary parameter, updated online using the environment's response as a self-supervisory signal, and reset per episode to ensure precise, context-dependent alignment without interference across tasks. While TTA for LLMs is emerging for tasks like math reasoning [Zuo et al., 2025] and few-shot learning [Akyürek et al., 2025], its application to the unique, interactive, and stateful challenges of LLM agents remains underexplored. Our work is the first to systematically apply this paradigm to adapt agents to novel environment formats without supervised trajectories.

Environment Modeling for Agents Recent work has focused on incorporating environment dynamics into LLM-based planning by training a parameterized world model [Fang et al., 2025, Chae et al., 2025, Ding et al., 2026, Qiao et al., 2024], or using an LLM to predict the next state using continuously updated environment rules [Zhou et al., 2024b]. Such approaches typically require collecting a large set of observations for learning a world model, and can be computationally expensive [Chae et al., 2025]. In contrast, our dynamics grounding introduces a lightweight, deployment-time pipeline to generate a temporary, in-context world model from a few exploratory interactions. This aligns with a broader trend of using in-context learning for adaptation, including methods like agent workflow memory [Wang et al., 2024b] and retrieval-augmented generation (RAG) for planning [Luo et al., 2023, Kagaya et al., 2024, Wang et al., 2024a]. However, our contribution lies in the systematic and automated process of generating this knowledge via persona-driven exploration, removing the need for pre-existing annotated demonstrations or a separate, trained dynamics model.

LLM-Based Agents Large language models (LLMs) have become the planning backbone of a wide range of agentic systems due to their strong instruction-following capabilities and generalization across diverse tasks [Ouyang et al., 2022, Mishra et al., 2022, Zhou et al., 2024a]. Their ability to interpret complex instructions and generate coherent action plans has enabled their integration into applications such as web navigation [Zhou et al., 2024a, He et al., 2024, Lù et al., 2024, Deng et al., 2023] and function calling [Patil et al., 2025, Yao et al., 2024]. Despite recent advances, LLM agents struggle to comprehend the specific syntactical input structure of different environments, such as in web navigation [Yang et al., 2024b, Gur et al., 2023] and text-to-SQL [Lei et al., 2024]. This is due to the fundamental

mismatch between the LLM pre-training corpus and the specific agentic environment [Chen et al., 2024, Ben-David et al., 2010].

4.3 Grounded Test-Time Adaptation Strategies

In this section, we detail our two adaptation strategies which mitigate syntactic and semantic gaps. We first formalize the problem setup, which is designed to reflect realistic deployment scenarios and is defined by three key constraints:

1. **No annotated trajectories or offline data.** The agent cannot access expert demonstrations or pre-collected data from the target environment. It must learn from scratch during interaction. For our running travel example, the agent has never seen travel-example.com and has no pre-existing dataset of successful bookings on that site.

2. **Online, streaming adaptation.** The agent receives tasks one at a time and must adapt on-the-fly without seeing a full batch of test instances in advance. For example, it must adapt to the website's quirks while handling a single request (e.g. "Book a flight from New York to Saskatoon") before seeing any future tasks.

3. **Permitted test-time interaction.** The agent is allowed to interact with the environment to gather information before receiving a specific task, but without supervision. For instance, it can perform a "blind" exploration of travel-example.com to discover that a button triggers a calendar pop-up, but it does not know what kind of tasks the user will ultimately ask it to execute.

This setting mirrors constraints in practical benchmarks (e.g., BIRD [Li et al., 2023a], WebArena [Zhou et al., 2024a]) and necessitates methods that can leverage only unsupervised, test-time interaction. To formulate this challenge, at test-time the LLM input $\mathcal{I}$ is constructed as

$$\mathcal{I} = [p; o; \{a\}_{i=1}^{T-1}] \tag{4.1}$$

where p denotes the task instruction (e.g., rules, task description), o denotes the current environment observation (e.g., accessibility tree observation, response of function call), and $\{a\}_{i=1}^{T-1}$ denotes the sequence of past actions taken by the LLM-based agent up to the current step. For environments with short or simple observations (e.g., tool-use scenarios), we instead construct the model input using all previous interactions between the agent and the environment, which may include both environment responses and agent actions. With this formulation of the challenge in mind, we describe two adaptation methods in the remainder of this section.

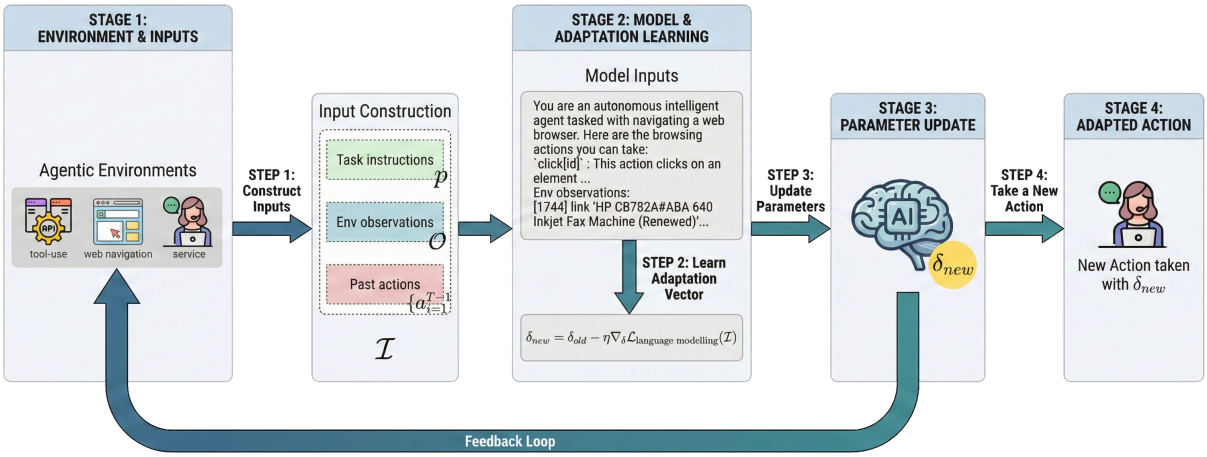


Figure 4.1: **Overview of syntactic alignment (SA).** This figure includes an example of web navigation shopping task to illustrate how the agent adapts to new environment. (1) At the start of each episode, we initialize an adaptation vector δ as a zero vector and construct inputs to the LLM agent. (2) During task execution, the agent receives environment instructions and observations. (3) At each step, we update the adaptation vector using cross-entropy loss on the current input, and apply the adaptation vector as a bias to the LLM's final hidden layer. This enables rapid alignment to environment-specific observation and action formats. (4) The LLM agent takes a new action with the updated vector, which shifts the model's output distribution to better match the test-time environment.

4.3.1 Syntactic Alignment

To address syntactic mismatches that arise in novel environments, we propose *syntactic alignment*. The objective is to parametrically adapt the model's output distribution at test time by treating the current context (i.e. task instructions, observations, and action history) as a self-supervisory signal. This allows the model to align with local syntactic patterns, such as unique UI element labels or response formats, without requiring a deep understanding of the environment's causal dynamics.

Our approach introduces a single lightweight adaptation vector $\delta \in \mathbb{R}^d$, where d is the hidden dimension of the language model. This vector acts as an additive bias to the final hidden representations $H \in \mathbb{R}^{n \times d}$ before the final projection layer. At each step of an episode, the adapted logits are computed as:

$$\text{logits}' = (H + \delta) W_{\text{LM}}^T \tag{4.2}$$

where $W_{\text{LM}} \in \mathbb{R}^{|V| \times d}$ is the model's output projection matrix and $|V|$ is the vocabulary size.

This vector δ is updated at each turn by performing one gradient descent step to minimize the language modeling loss of the current input context $\mathcal{I}$ according to the following:

$$\delta_{\text{new}} \leftarrow \delta_{\text{old}} - \eta \nabla_{\delta} \mathcal{L}_{\text{CE}} \left(f_{\theta,\delta}(\mathcal{I}_{1:n-1}),\ \mathcal{I}_{2:n} \right) \tag{4.3}$$

where $f_{\theta,\delta}$ is the LLM parameterized by its fixed weights θ and the adaptable vector δ, and η is the learning rate. $\mathcal{I}_{1:n-1}$ denotes the input subsequence consisting of tokens 1 through $n-1$, and $\mathcal{I}_{2:n}$ denotes the corresponding next-token targets obtained by shifting the sequence by one position. The loss in Eq. 4.3 is therefore computed by predicting each token in $\mathcal{I}_{2:n}$ from its preceding context $\mathcal{I}_{1:n-1}$. This update encourages the model to internalize the syntactic features present in the immediate context by adapting its output distribution to the environment context.

To illustrate this in practice, consider our agent on the unseen website `travel-example.com`. The agent's input context $\mathcal{I}$ contains the accessibility tree, which includes strings such as `[145] <button>Go` and `dest_field`.

1. **Initialization**: At the episode's start, the adaptation vector is initialized as a zero vector, $\delta \leftarrow 0$.
2. **Prediction**: Without adaptation, the model's logits would likely favor the action `click(“Search”)`, based on its pre-trained priors.
3. **Adaptation**: We compute the language modeling loss on the context $\mathcal{I}$ (Eq. 4.3). Because the strings `Go` and `dest_field` are present in $\mathcal{I}$, the gradient update to δ will shift its values to favor their tokens.
4. **Correction**: In the subsequent generation step, the adapted logits (Eq. 4.2) now assign a higher probability to the syntactically correct action, `click(“Go”)`.

This process is computationally efficient, and steps 2-4 are repeated at every model prediction within an episode. To prevent catastrophic forgetting and ensure that adaptation is specific to the current environment, δ is reset to zero at the beginning of each new episode. This lightweight update aligns the model's pre-trained distribution with the specific distribution of the test-time environment.

4.3.2 Dynamics Grounding

LLM-based agents often fail in novel environments due to a **semantic mismatch**: they lack a “world model” to predict the outcomes of their actions [Ha and Schmidhuber, 2018, Chae

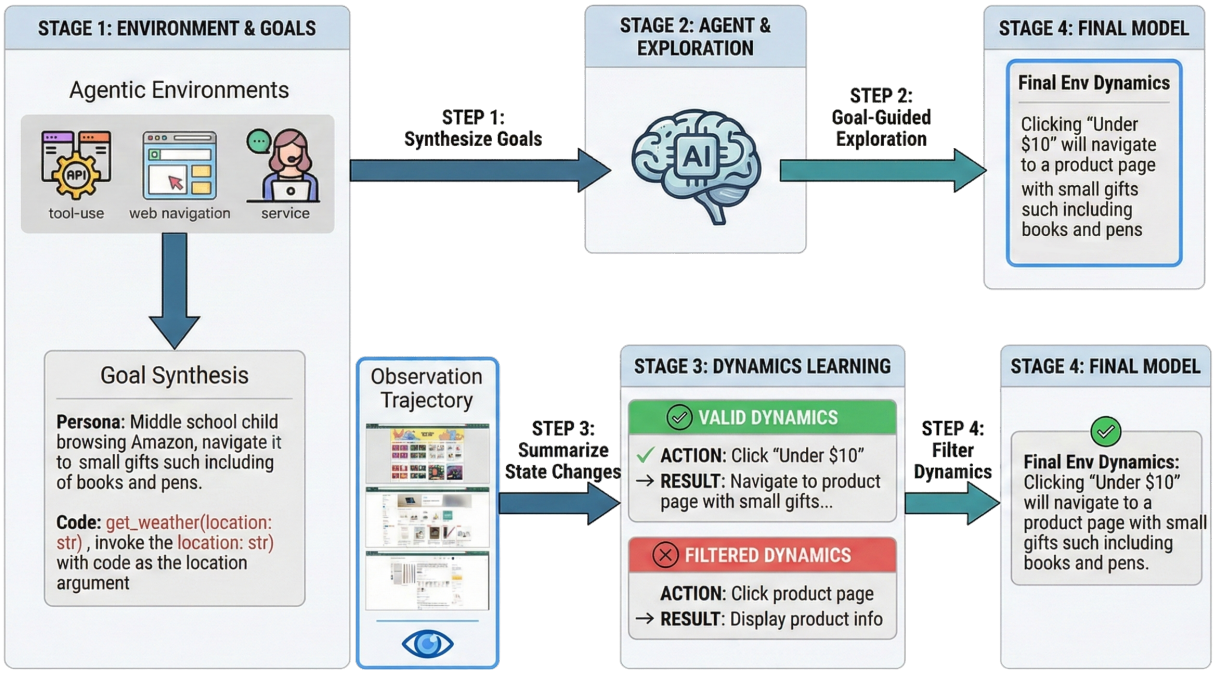


Figure 4.2: **Overview of dynamics grounding (DG).** This figure includes an example of web navigation shopping task to illustrate how the pipeline generates environment dynamics in language. (1) We synthesize diverse exploration tasks based on personas using environment descriptions. (2) An exploration agent interacts with the environment to collect interaction logs of state transitions. (3) An LLM extracts and summarizes environment dynamics from these logs. (4) A reasoning model filters less informative rules, which are then used to augment the agent's context during evaluation, enabling more transition-aware decision making.

et al., 2025]. To address this, we propose *dynamics grounding*, a strategy that proactively discovers the causal dynamics of an environment **before the agent encounters any user tasks during deployment**. The objective is to construct an in-context world model using natural language by performing a one-time, automated exploration. The cost of this discovery phase is amortized over all subsequent tasks performed in that environment.

Our method follows a four-step pipeline to generate a concise set of environment dynamics, denoted as E_{clean}, for a given new environment.

1. **Persona/Exploratory Goal Synthesis.** An LLM is prompted with a high-level description of the environment (e.g., website purpose, API documentation) to generate a set of N diverse, exploratory "personas" or "goals". Personas frame open-ended tasks

designed to probe for non-obvious interactions. For our `travel-example.com` scenario, a synthesized persona might be: *"As a first-time user, I want to see what happens if I try to search for a flight without selecting a date."*. We prefer personas over a less-structured exploration policy (e.g. maximizing state novelty or coverage) to guide the agent to explore complex, multi-step interactions that a naive exploration strategy might miss, thus yielding more semantically rich dynamics.

2. **Exploration and on-the-fly Dynamics Extraction** For each persona or goal, an LLM agent interacts with the environment. The agent is instructed to take novel actions to maximize its discovery of new state transitions. After each action is executed and the system transitions into the next state, we take (observation, action, new observation) to summarize into a concise human-readable rule e. Immediately after, the list of generated rules $\{e\}_{i=1}^{T}$ is appended to the agent to encourage taking actions that have not been taken before. In our example, the agent would execute `click(‘‘Go’’)` and log the transition from the main page to a state where a calendar pop-up is active. Then the transition on `travel-example.com` would be extracted as the dynamic: Executing the action `click(‘Go’)` on the main page causes a calendar modal to appear for date selection.

3. **Filtering and Consolidation.** The set of all extracted dynamics, $E = \{e\}_{i=1}^{M}$, is passed to a reasoning model. This model filters out trivial (e.g., "typing in a text field shows text") or repetitive rules, producing a final, clean set of dynamics, E_{clean}.

During task execution at test time, we augment the input context $\mathcal{I}$ with discovered dynamics:

$$\mathcal{I}' = [\mathcal{I}; E_{\text{clean}}] \tag{4.4}$$

By leveraging the LLM's in-context learning capabilities, this explicit knowledge of the environment's state-transition patterns guides the agent to make more informed and transition-aware decisions. In our example, with the calendar pop-up dynamic in its context, the agent correctly anticipates the outcome of its click and plans its subsequent actions accordingly, successfully navigating the semantic mismatch. Example prompts for exploration and dynamics extraction are shown in section B.2 and section B.3.

4.4 Experiments

4.4.1 Setup

We evaluate our approach on three realistic agentic benchmarks: BFCLv3 [Patil et al., 2025], WebArena [Zhou et al., 2024a], and Tau-Bench [Yao et al., 2024], which together cover both web navigation and conversational function calling tasks. As for baselines, we evaluate open-source `Qwen2.5-14B-Instruct` [Qwen et al., 2025] model and closed-source `GPT-4.1`. On WebArena, we additionally compare to the World-model-augmented (WMA) agent from [Chae et al., 2025] as a baseline. This approach first collects exploration demonstrations, then trains a `Llama-3.1-8B-Instruct` [Grattafiori et al., 2024] model to predict the next state given an action (i.e., a learned world model). During evaluation, this world model is used to provide next-state predictions to the agent on-the-fly, helping the agent reason about the consequences of its actions. In this comparison, we use the same `GPT-4o mini` as the original WMA experiments. Aside from the syntactic alignment and dynamics grounding experiments, we provide ablation studies in section 4.4.3.

Table 4.1: Number of tasks per website in the WebArena benchmark. The benchmark consists of six websites, including a multi-site category for tasks that require interacting across multiple websites (from the six sites), for a total of 812 tasks.

Website	**Tasks**
Shopping	187
Shopping Admin	182
GitLab	180
Map	109
Reddit	106
Multi-site	48
Total	812

For all dynamics grounding experiments, we leverage `GPT-4.1` as an exploration policy to extract environmental dynamics. For each environment, we conduct 10 exploration episodes to gather a diverse set of dynamics. For environment dynamics filtering, we apply `o3`[2] to remove repetitive and trivial environment dynamics. We also include ablation studies on the number of exploration episodes and the effect of filtering in section 4.4.3.

Before delving into the results, we first introduce the benchmarks and describe the specific experimental setups used for experiments.

WebArena The WebArena benchmark consists of five self-hosted websites and a total of 812 tasks. Many tasks require interacting across multiple websites, which poses additional challenges for LLM-based agents. For evaluation, WebArena employs a string-matching-based mechanism to assess correctness, providing more stable and objective results compared to benchmarks using LLMs to score, such as [He et al., 2024, Gou et al., 2025]. For dynamics grounding, we first synthesize 10 personas via prompt instruction B.1 for each website using GPT-4o [OpenAI et al., 2024] and with seed personas from NNetNav [Murty et al., 2025]. Then for each persona, we roll-out a `GPT-4.1` agent to explore the environment

[2] https://platform.openai.com/docs/models/o3

guided by the persona description, with the exploration prompt instruction B.2 derived from NNetNav, and with a maximum exploration budget of 30 steps. To extract environment dynamics from interaction logs, we summarize each state transition pair (observation, action taken, new observation) with prompt instruction B.3, and concatenate all summarized state transition pairs to obtain the list of environmental dynamics $E = \{e\}_{i=1}^{M}$ for each website. Finally, we utilize the `o3` model to remove repetitive and trivial environment dynamic entries from the list via prompt instruction B.4, resulting in E_{clean}. We provide all prompts we use in chapter B. With syntactic alignment, we reset the adaptation vector after each episode and we use a learning rate of 0.1 and training step of 1 for all experiments. During evaluation, we use the text-based accessibility tree as the observation type and run experiments using BrowserGym[3] and AgentLab[4] with a maximum step of 30.

Table 4.2: Main results on WebArena, WebVoyager, BFCLv3, and Tau-Bench benchmarks. We report task success rates (%) for each model and adaptation method. For Tau-Bench, we average runs across 5 seeds and use a custom more stable codebase. Both test-time adaptation strategies improve performance. "N/A" indicates not applicable—AWM requires training a web-specific world model on state transitions, which is not applicable to Tau-Bench (conversational only) or BFCLv3 (no explicit web-like state transition data); Dynamics grounding does not operate in conversational Tau-Bench as there is no explicit and fixed state transition rules in the environment.

Model	**Task**			
	WebArena	BFCLv3	Tau-Bench	
			Airline	Retail
GPT-4.1	30.0	55.5	-	-
GPT-4.1 (+ DG)	35.0 (+5.0)	64.0 (+8.5)	N/A	N/A
GPT-4o mini	12.0	-	-	-
GPT-4o mini (+ WMA)	13.5 (+1.5)	N/A	N/A	N/A
GPT-4o mini (+ DG)	18.0 (+6.0)	-	N/A	N/A
Qwen2.5-14B-Instruct	17.0	18.5	21.6	43.3
Qwen2.5-14B-Instruct (+ SA)	18.0 (+1.0)	20.0 (+1.5)	25.2 (+3.6)	44.9 (+1.6)
Qwen2.5-14B-Instruct (+ DG)	20.0 (+3.0)	22.0 (+3.5)	N/A	N/A
Qwen2.5-14B-Instruct (hybrid)	21.0 (+4.0)	21.0 (+2.5)	N/A	N/A

[3] https://github.com/ServiceNow/BrowserGym
[4] https://github.com/ServiceNow/AgentLab

BFCLv3 BFCLv3 [Patil et al., 2025] is a benchmark with various realistic APIs to evaluate the functions calling abilities of LLM. It contains 8 different API domains. We provide the number of APIs for each environment in table B.3. We choose the multi-turn-base split to assess the agent's multi-turn function calling capabilities. This split is multi-turn and multi-step, meaning that instead of just handling a single tool call in a one-off request, models have to carry out sequences of function calls across multiple dialogue turns. BFCLv3 uses state-based evaluation (checking the system's resulting state) and response-based evaluation (checking whether the call sequences are valid and minimally sufficient) to assess correctness, therefore being stable and deterministic. For dynamics grounding, we synthesize 10 exploration tasks with `GPT-4o` conditioned on the function documentation for each domain via prompt instruction B.7. Concretely, we instruct the LLM to think about *what functions it wants to explore in order to find out unpredictable outcomes or interesting behaviors, given the function documentation.* We use a `GPT-4.1` agent to execute the exploration task and we collect interaction logs for environment dynamics extraction. During exploration, we ask the exploration agent to decide if the steps taken so far are sufficient to answer the exploration task with prompt instruction B.8. The environment dynamics are extracted using prompt instruction B.9. We follow the same procedure as in WebArena to remove repetitive and trivial environment dynamics with prompt instruction B.10. For syntactic alignment, we reset adaptation vector at each turn (when a new user query arrives).

Tau-Bench Tau-Bench [Yao et al., 2024] is a realistic function calling benchmark that simulates real-world service agents interacting with an LLM-simulated user. It comprises two domains: airline (50 tasks) and retail (115 tasks). The airline domain is generally more challenging than retail, as it features a much larger database, increasing the complexity of the tasks. Tau-bench's use of an LLM-simulated user agent introduces significant variance in evaluation results, as the simulated human may drift from the original instruction or be influenced by the agent's responses over multiple conversational turns. To address this, we adopt the improved evaluation codebase from [Prabhakar et al., 2025], which substantially reduces reward variance and improves evaluation fidelity. Specifically, we (1) fix incorrect labels, (2) set the temperature of the LLM-simulated user agent to 0 for deterministic responses, and (3) employ a Best-of-N (N=5) sampling strategy combined with a self-critique mechanism, allowing the simulated user to better adhere to the task instruction and avoid being misled by the agent's outputs. This protocol has been shown to improve consistency in agent performance evaluation across multiple trials according to [Prabhakar et al., 2025]. Tau-Bench consists solely of API-based tasks without observable state transitions, meaning there are no environment dynamics to extract. As a result, dynamics grounding, which

relies on extracting and leveraging such dynamics, is not applicable to this benchmark. Therefore, we only evaluate the syntactic alignment method on Tau-Bench. To further reduce variance, we report results averaged over 5 random seeds in table 4.2.

4.4.2 Results and Analysis

Table 4.3: Success rates (%) on the WebArena benchmark across different websites and models. Both adaptation strategies improve performance over baselines. and dynamics grounding (DG) surpasses WMA on `GPT-4o mini`. DG improves success rate substantially on complex multi-site split.

	Website						
Model	Gitlab	Shop Admin	Shop	Reddit	Map	Multi	Avg
GPT-4.1	48.0%	31.0%	37.0%	29.0%	28.0%	2.0%	30.0%
GPT-4.1 (+DG)	45.0%	27.0%	33.0%	42.0%	31.0%	23.0%	35.0%
GPT-4o mini	9.0%	12.0%	18.0%	8.0%	12.0%	12.0%	12.0%
GPT-4o mini (+WMA)	–	–	–	–	–	–	13.5%
GPT-4o mini (+DG)	18.0%	14.0%	26.0%	14.0%	19.0%	12.0%	18.0%
Qwen2.5-14B-Instruct	20.0%	14.0%	22.0%	13.0%	13.0%	6.0%	17.0%
Qwen2.5-14B-Instruct (+SA)	16.0%	14.0%	24.0%	25.0%	16.0%	6.0%	18.0%
Qwen2.5-14B-Instruct (+DG)	23.0%	13.0%	24.0%	23.0%	20.0%	10.0%	20.0%
Qwen2.5-14B-Instruct (Hybrid)	26.0%	13.0%	24.0%	31.0%	19.0%	4.0%	21.0%

Both Adaptation Strategies Consistently Improve Performance As shown in table 4.2, both syntactic alignment (SA) and dynamics grounding (DG) improve performance across models and tasks. dynamics grounding yields substantial improvements for `GPT-4.1`, which is better able to leverage the provided dynamics due to its superior instruction-following abilities. It is also evident in table 4.2 that our `GPT-4o mini` surpasses WMA that uses a learned world model by a large margin. For `Qwen2.5-14B-Instruct`, both methods provide gains, highlighting the general applicability of test-time adaptation. The improvements of dynamics grounding are especially significant for models with stronger instruction-following abilities (e.g., GPT-4.1), likely because they can better leverage the provided dynamics context.

Both Adaptation Strategies are computationally efficient The two proposed strategies are computationally efficient, though they have different profiles. syntactic alignment is highly efficient for real-time use. As shown in table 4.4, a single update step adds only a 3% latency overhead. In contrast, dynamics grounding is a one-time, deployment-time investment designed to be far more lightweight than traditional world-modeling approaches like the WMA baseline. For example, while WMA requires synthesizing 870 tasks to collect trajectories for training a separate `Llama-3.1-8B` model, our method uses only 50 exploratory rollouts and requires no additional model training. This streamlined process avoids WMA's costly training phase and its significant inference overhead, where action simulation takes approximately 140 seconds per step. The total upfront cost for our dynamics grounding procedure on a single WebArena website is approximately 7M tokens, a cost that is amortized across all subsequent tasks. This investment pays significant dividends where environmental dynamics are unpredictable. For instance, on the challenging multi-site split, the success rate of our agent increases from 2% to 23%. This demonstrates the value of efficiently learning an environment's dynamics at test time to compensate for the limitations of an agent's pre-training.

Table 4.4: Relative latency gain (%) of PA on BFCLv3 multi-turn. LR fixed to 0.1.

Step	**Latency Gain ↓**
0	0%
1	3.0%
2	5.9%
3	8.2%
4	12.6%
5	15.6%

Effectiveness Depends on Environment Complexity and Method Synergy The smaller gains for DG on simpler websites like "Shopping" suggest that when an environment's dynamics align with common sense priors (e.g., clicking 'search' shows a search bar), the explicit dynamics provide less new information, and the overhead of a longer context may slightly hinder performance. In these simple settings, longer input on dynamics may negatively impact the model's effectiveness, as prior work has shown that excessively long contexts can hurt performance [Beltagy et al., 2020, Liu et al., 2024a]. We further analyze this phenomenon and provide exemplar environment dynamics in section B.5. Furthermore, our test of a naive hybrid approach showed mixed results, underperforming the dynamics grounding method alone on BFCLv3 (21.0% vs 22.0%). This suggests a simple combination is insufficient and may create interference between the two adaptation signals. A key challenge for future work is developing a more principled integration of semantic guidance and distributional alignment.

Table 4.5: Ablation on exploration policy and dynamics extractor backbones on WebArena. Here we use different backbones of exploration policy and dynamics extractor. We found for dynamics grounding—using the same LLM agent (itself) improves the same as using a stronger one.

Task Model	Exploration Policy	Dynamics Extractor	SR
GPT-4o mini (baseline)	–	–	12.0
GPT-4o mini	GPT-4.1	GPT-4.1	18.0 (+6.0)
GPT-4o mini (self)	GPT-4o mini	GPT-4o mini	19.0 (+7.0)
Qwen2.5-14B-Instruct (baseline)	–	–	17.0
Qwen2.5-14B-Instruct	GPT-4.1	GPT-4.1	20.0 (+3.0)
Qwen2.5-14B-Instruct (self)	Qwen2.5-14B-Instruct	Qwen2.5-14B-Instruct	20.0 (+3.0)

4.4.3 Ablations

Dynamics grounding enables self-improvement We ablate the choice of backbone LLMs for exploration and dynamics extraction in our dynamics grounding method (table 4.5). Results show that using the same model for both components (self-improvement) performs as well as using a stronger model, indicating that our approach is robust to the choice of exploration policy and dynamics extractor.

Effect of filtering environmental dynamics We conduct an ablation experiment on BFCLv3 and find that filtering improves success rate across exploration budgets. Specifically, with 10 episodes, success rate rises from 61.0 to 64.0 (+3.0).

Syntactic alignment is robust to different hyperparameters As shown in table B.1, training with different learning rates generally improves performance across a range of reasonable hyperparameter values. However, using extreme hyperparameters can degrade performance. For example, in the case of the 7B model, applying a very high learning rate of 1.0 or training for 5 steps leads to reduced success rates, highlighting the importance of careful hyperparameter selection for optimal adaptation.

Larger models benefit from a larger learning rate When testing with a very large learning rate (i.e., learning rate of 1.0), we find that larger models (i.e., 14B and 32B models) benefit more with a larger learning rate. We attribute this to the fact that the dimension of the adaptation vector for 14B and 32B models is the same (i.e., 5120), and is much larger compared to the 7B model's (i.e., 3584).

4.5 Limitations and Future Work

Our study is primarily focused on the Qwen2.5 model family for syntactic alignment; future work should validate these findings across a broader range of open-source architectures. Our dynamics grounding method currently operates in environments with explicit state transitions, where testbeds such as user-oriented conversational tasks are not explored. Additionally, our online syntactic alignment method does not currently normalize updates by the hidden dimension size, which could improve robustness to hyperparameters.

The most promising direction for future work is the development of a principled method for integrating our two adaptation strategies as our analysis suggests that simply combining them is suboptimal. A more advanced approach could involve a meta-controller that assesses an environment's complexity to dynamically decide whether to rely on efficient online adaptation for simple tasks or to deploy the more costly dynamics grounding for complex, unfamiliar environments.

4.6 Conclusions

In this work, we presented a comparative study of two distinct strategies for adapting LLM agents to novel, complex environments. We identified two primary failure modes—syntactic and semantic mismatches—and investigated targeted solutions for each. The first, syntactic alignment (SA), is an efficient technique for aligning an agent's output distribution to the local format of an environment. The second, dynamics grounding (DG), is an effective and efficient method for providing the agent with a causal world model through proactive dynamics extraction.

Our evaluations on diverse agentic benchmarks demonstrate that both strategies significantly improve performance. Online adaptation provides consistent, low-cost gains, while dynamics grounding is critical for success in complex settings where an agent's pre-trained knowledge fails. Our findings highlight that a multi-faceted approach to adaptation is necessary for robust agent generalization and point toward exciting future directions in creating more intelligent and integrated adaptation systems.

Chapter 5

A Unified View of Learning at Test Time

The two preceding chapters presented two seemingly different systems: a framework for ranking synthetic datasets (Chapter 3) and a pair of adaptation strategies for LLM agents (Chapter 4). This chapter argues that they are two instances of the same underlying idea, and draws out the dimensions along which they differ.

5.1 What is learned at test time?

Both contributions study how to extract useful information from the deployment context and convert it into improved model behavior. What differs is the object being learned. SynQuE learns *which synthetic dataset is likely to transfer*: given a small unannotated sample of real data, its proxy metrics estimate how well a model trained on each candidate dataset would perform on the real task. The agent adaptation methods of Chapter 4 instead learn *how the environment behaves and how its syntax should be used*: syntactic alignment (SA) captures the environment's surface form in an adaptation vector, while dynamics grounding (DG) captures the environment's causal structure as in-context rules.

In both cases, no ground-truth labels are observed at test time. The deployment context itself—real inputs in one case, environment responses in the other—acts as the supervisory signal.

5.2 Two kinds of test-time signals

The two routes differ first in the *kind* of signal they consume, summarized in Table 5.1. SynQuE consumes a *distributional* signal: a static sample of unannotated real inputs whose value lies in describing the target distribution. The agent methods consume an *interaction* signal: observations and state transitions that only exist because the agent acts in the environment.

Table 5.1: Two kinds of test-time signals and what each adapts.

Signal type	**Chapter**	**What is observed**	**What is adapted**
Distributional	3	Unannotated real data	Synthetic data selection for training
Interaction	4	Environment observations and transitions	Agent syntax, planning context, world model

5.3 Two adaptation timescales

The two routes also operate on different timescales, which explains why they take such different mechanistic forms.

SynQuE adapts in a slower, iterative process:

1. Generate synthetic data.
2. Score each dataset using unannotated real data.
3. Refine and rescore synthetic data as needed.
4. Once satisfactory, train or fine-tune the model.

The agent methods adapt through a faster deployment loop:

1. Observe the environment;
2. Update the adaptation vector or collect dynamics;

3. Execute tasks with grounded context.

The slow loop pays a training cost once, before deployment, and is appropriate when the bottleneck is the training data itself. The fast loop operates during deployment, either per-step (SA's per-turn gradient update) or as a one-time exploration phase whose cost is amortized over all subsequent tasks (DG). This distinction explains why the two contributions belong together despite using different mechanisms: they occupy two ends of a spectrum of *when* test-time information is converted into behavior, not two unrelated problem settings.

5.4 Two forms of supervision-free adaptation

A strong unifying point is that both works deliberately avoid expensive labels. SynQuE avoids annotated real validation data: its proxies rank datasets using only unannotated real inputs, sidestepping the labeled validation sets that scaling-law and regression-based selection methods require [Liu et al., 2025, Magnusson et al., 2025]. The agent methods avoid annotated trajectories and expensive environment-specific fine-tuning: SA uses the agent's own input context as a self-supervisory signal, and DG uses unsupervised exploration rather than expert demonstrations [Wang et al., 2024b, Chae et al., 2025]. In both cases, supervision is replaced by structure: the assumption that proximity to the real distribution predicts transfer, and the assumption that observed state transitions predict future environment behavior.

5.5 Practical implications

The unified view yields concrete guidance on where each method fits. Use SynQuE when the main bottleneck is choosing useful synthetic training data under real-data scarcity—for example, when several synthesis pipelines are available and training on all of their outputs is infeasible. Use environment interaction when the bottleneck is unfamiliar environment syntax or dynamics—for example, when an agent is deployed to a new website or API surface. In privacy-sensitive deployments, both methods reduce dependence on labeled private data: SynQuE requires only a small unannotated sample that can be scored locally, and the agent methods require only interaction logs that never leave the deployment environment.

5.6 Toward combined systems

The two routes are complementary, and their combination is a natural next step. Several directions follow directly from this thesis:

- grounded verifier synthesis, in which the exploration machinery of Chapter 4 is used to discover observable evidence of task completion and synthesize executable verifiers, whose outcomes then serve as reward signals for adapting agents and models to new environments;
- agents that explore environments and synthesize training data from discovered dynamics, closing the loop between interaction and data generation;
- SynQuE-style scoring for agent trajectories, ranking exploration logs by their expected value for downstream adaptation;
- Lens-style rubrics to evaluate environment interaction logs, bringing LLM-based data characterization to the interactive setting;
- dynamics-grounded synthetic data generation, in which extracted environment rules constrain the synthesizer to produce transition-consistent data;
- local, privacy-preserving test-time adaptation inside organizations, combining both routes without exporting any private data.

The first of these directions deserves emphasis, because it extends the thesis claim one step further. The signals studied in this thesis inform *what to train on* (Chapter 3) and *how to act* (Chapter 4); an interaction-grounded verifier would additionally provide a signal for *whether behavior succeeded*, turning test-time information into supervision for online improvement. Chapter 6 describes this ongoing work and revisits the remaining directions in the context of the thesis's overall findings.

Chapter 6

Conclusion

6.1 Summary

This thesis studied how models can adapt after training by using information revealed at test time. We framed this challenge as *test-time context mismatch*: the information a model needs for reliable behavior in its deployment context is often unavailable during training, whether because real data is private or scarce, or because the deployment environment's syntax and dynamics are novel. We investigated two adaptation routes. Chapter 3 introduced SynQuE, which converts a small unannotated sample of real data into a signal for selecting synthetic training datasets. Chapter 4 introduced two deployment-time strategies—syntactic alignment and dynamics grounding—which convert environment interaction into adapted agent behavior. Chapter 5 showed that both routes instantiate the same principle: adaptation is possible when test-time information can be converted into a useful signal for model behavior.

6.2 Main findings

The experiments in this thesis support five main findings.

First, limited unannotated real data can guide synthetic data selection. SynQuE proxy metrics correlate with downstream task performance across sentiment analysis, text-to-SQL, image classification, and web navigation; on text-to-SQL, selecting the top-3 synthetic datasets by proxy score raises average accuracy from 30.4% to 38.4% compared to indiscriminate selection.

Second, LLM reasoning can help assess synthetic data quality for complex tasks. LENS, which builds language rubrics contrasting synthetic and real data and applies principled debiasing, is the only proxy that consistently achieves positive correlation across all tasks, with the largest advantage on long-horizon web navigation where embedding-based metrics struggle.

Third, LLM agents fail in new environments due to both syntactic and semantic mismatch. Distinguishing these two failure modes matters because they call for different mechanisms: output-distribution alignment for the former, and world-model grounding for the latter.

Fourth, environment interaction can provide the syntax and dynamics needed for better agent behavior. Syntactic alignment yields consistent gains at roughly 3% latency overhead, and dynamics grounding produces large improvements where dynamics are unpredictable—most notably raising the WebArena multi-site success rate from 2% to 23%—while remaining far cheaper than training a separate world model.

Fifth, and most generally, test-time context is a valuable resource for adaptation when labels, demonstrations, or full retraining are unavailable.

6.3 Future directions

Several directions follow from this work. We first describe the direction we are actively pursuing, which extends the central idea of this thesis, and then outline further directions.

From grounded dynamics to grounded verification. The methods of Chapter 4 convert environment interaction into knowledge that *describes* the environment: its syntax and its state-transition dynamics. Our ongoing work extends this recipe to knowledge that *evaluates* behavior in the environment: executable verifiers of task completion. Task verification is itself a deployment bottleneck. Determining whether an agent actually completed a task today requires hand-written, task-specific evaluators that are expensive to author and maintain, or LLM-as-a-judge scoring that is prompt-sensitive, unstable, and prone to judging surface text rather than real environment state. Yet most practical agent tasks are state-changing workflows that leave observable traces—a submitted form updates a backend record, a changed setting appears in a configuration file, a completed report exists at a path. We therefore study *grounded verifier synthesis*: given a natural-language task specification and black-box access to an environment, actively explore the environment to discover observable evidence of task completion, synthesize executable postcondition checks

over that evidence, and stress-test the resulting verifiers against perturbations and negative trajectories. As in dynamics grounding, the system does not assume a complete symbolic description of the environment; grounding is obtained through interaction. The central challenges are identifying predicates that are specific enough to detect real completion yet robust to interface variation, abstaining when a task leaves no observable trace, and guarding against spurious predicates and verifier hacking—false positives are more damaging than false negatives, since they corrupt evaluation and invite reward exploitation.

Ultimately, we aim to use the outcomes of synthesized verifiers as *reward signals* for adapting agents and machine learning models to new environments. This would complete the arc of this thesis. Chapter 3 converted test-time information into a criterion for selecting training data, and Chapter 4 converted it into aligned syntax and planning context; grounded verification would convert it into supervision itself, enabling online improvement—through reinforcement learning or trajectory selection—in environments where labels, demonstrations, and hand-written reward functions are unavailable. Verifier-scored trajectories would also connect the two routes directly: outcome signals from synthesized verifiers could annotate exploration data for training, and SynQuE-style proxies could select among the resulting datasets.

Further directions. Beyond verification, several directions remain. A first is to combine synthetic data selection with environment interaction, for example by using exploration-derived dynamics to generate training data and SynQuE proxies to select among the generated datasets. A second is to develop safer and cheaper exploration strategies, since dynamics grounding—and verifier synthesis built on it—relies on unconstrained interaction with the environment. A third is to build example-level synthetic data selection methods that move beyond ranking whole datasets to weighting or filtering individual examples. A fourth is to use environment-derived rules to guide data synthesis, constraining generators to produce data consistent with observed transitions. Finally, studying privacy-preserving local adaptation would bring these methods to the organizational settings—healthcare, finance, proprietary software environments—that motivated this thesis in the first place.

References

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The Surprising Effectiveness of Test-Time Training for Few-Shot Learning, March 2025. URL http://arxiv.org/abs/2411.07279. arXiv:2411.07279 [cs].

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs, 2024. URL https://arxiv.org/abs/2402.03927. Version Number: 2.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The Long-Document Transformer, December 2020. URL http://arxiv.org/abs/2004.05150. arXiv:2004.05150 [cs].

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of Representations for Domain Adaptation. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/hash/b1b0432ceafb0ce714426e9114852ac7-Abstract.html.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, May 2010. ISSN 1573-0565. doi: 10.1007/s10994-009-5152-4. URL https://doi.org/10.1007/s10994-009-5152-4.

Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49–e57, July 2006. ISSN 1367-4811, 1367-4803. doi: 10.1093/bioinformatics/btl242. URL https://academic.oup.com/bioinformatics/article/22/14/e49/228383.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large Language Models Suffer From Their Own Output: An Analysis of the Self-Consuming Training Loop, 2023. URL https://arxiv.org/abs/2311.16822. Version Number: 2.

Josue Casco-Rodriguez, Sina Alemohammad, Lorenzo Luzi, Ahmed Imtiaz, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-Consuming Generative Models go MAD. *LatinX in AI at Neural Information Processing Systems Conference 2023*, December 2023. doi: 10.52591/lxai202312101. URL https://research.latinxinai.org/papers/neurips/2023/pdf/Josue_CascoRodriguez.pdf. Conference Name: LatinX in AI at Neural Information Processing Systems Conference 2023.

Hyungjoo Chae, Namyoung Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web Agents with World Models: Learning and Leveraging Environment Dynamics in Web Navigation, March 2025. URL http://arxiv.org/abs/2410.13232. arXiv:2410.13232 [cs].

Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. Dataset search: a survey. *The VLDB Journal*, 29(1):251–272, January 2020. ISSN 0949-877X. doi: 10.1007/s00778-019-00564-x. URL https://doi.org/10.1007/s00778-019-00564-x.

Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, August 2016. doi: 10.1145/2939672.2939785. URL http://arxiv.org/abs/1603.02754. arXiv:1603.02754 [cs].

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-FLAN: Designing Data and Methods of Effective Agent Tuning for Large Language Models, 2024. URL https://arxiv.org/abs/2403.12881. Version Number: 1.

Samuel Rhys Cox, Yunlong Wang, Ashraf Abdul, Christian Von Der Weth, and Brian Y. Lim. Directed Diversity: Leveraging Language Embedding Distances for Collective Creativity in Crowd Ideation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–35, Yokohama Japan, May 2021. ACM. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445782. URL https://dl.acm.org/doi/10.1145/3411764.3445782.

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, June 2009. doi: 10.1109/CVPR.2009.5206848. URL https://ieeexplore.ieee.org/document/5206848.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a Generalist Agent for the Web, December 2023. URL http://arxiv.org/abs/2306.06070. arXiv:2306.06070 [cs].

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, Fengli Xu, and Yong Li. Understanding World or Predicting Future? A Comprehensive Survey of World Models. *ACM Computing Surveys*, 58(3):1–38, February 2026. ISSN 0360-0300, 1557-7341. doi: 10.1145/3746449. URL https://dl.acm.org/doi/10.1145/3746449.

Ricard Durall, Avraam Chatzimichailidis, Peter Labus, and Janis Keuper. Combating Mode Collapse in GAN training: An Empirical Analysis using Hessian Eigenvalues, December 2020. URL http://arxiv.org/abs/2012.09673. arXiv:2012.09673 [cs].

Hady Elsahar and Matthias Gallé. To Annotate or Not? Predicting Performance Drop under Domain Shift. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2163–2173, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1222. URL https://aclanthology.org/D19-1222/.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. WebEvolver: Enhancing Web Agent Self-Improvement with Coevolving World Model, 2025. URL https://arxiv.org/abs/2504.21024. Version Number: 2.

Jay W. Forrester. Counterintuitive behavior of social systems. *Theory and Decision*, 2 (2):109–140, December 1971. ISSN 1573-7187. doi: 10.1007/BF00148991. URL https://doi.org/10.1007/BF00148991.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. Self-Guided Noise-Free Data Generation for Efficient Zero-Shot Learning, May 2022. URL https://www.semanticscholar.org/paper/Self-Guided-Noise-Free-Data-Generation-for-Learning-Gao-Pi/6f7e03e4ccd26c762090e25dc5d2eb1e1f8c641d.

Scott Geng, Cheng-Yu Hsieh, Vivek Ramanujan, Matthew Wallingford, Chun-Liang Li, Pang Wei Koh, and Ranjay Krishna. The Unmet Promise of Synthetic Training Images:

Using Retrieved Real Images Performs Better, January 2025. URL http://arxiv.org/abs/2406.05184. arXiv:2406.05184 [cs].

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jiménez Gutiérrez, Yiheng Shu, Chan Hee Song, Jiaman Wu, Shijie Chen, Hanane Nour Moussa, Tianshu Zhang, Jian Xie, Yifei Li, Tianci Xue, Zeyi Liao, Kai Zhang, Boyuan Zheng, Zhaowei Cai, Viktor Rozgic, Morteza Ziyadi, Huan Sun, and Yu Su. Mind2Web 2: Evaluating Agentic Search with Agent-as-a-Judge, July 2025. URL http://arxiv.org/abs/2506.21506. arXiv:2506.21506 [cs].

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi,

Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz,

Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin

Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The Llama 3 Herd of Models, November 2024. URL http://arxiv.org/abs/2407.21783. arXiv:2407.21783 [cs].

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. ISSN 1533-7928. URL http://jmlr.org/papers/v13/gretton12a.html.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A Survey on LLM-as-a-Judge, March 2025. URL http://arxiv.org/abs/2411.15594. arXiv:2411.15594 [cs].

Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and Preventing Hallucinations in Large Vision Language Models, 2023. URL https://arxiv.org/abs/2308.06394. Version Number: 3.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis, 2023. URL https://arxiv.org/abs/2307.12856. Version Number: 4.

David Ha and Jürgen Schmidhuber. World Models, March 2018. URL http://arxiv.org/abs/1803.10122. arXiv:1803.10122 [cs].

Ryuichiro Hataya, Han Bao, and Hiromi Arai. Will Large-scale Generative Models Corrupt Future Datasets? *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20498–20508, October 2023. doi: 10.1109/ICCV51070.2023.01879. URL https://ieeexplore.ieee.org/document/10376575/. Conference Name: 2023 IEEE/CVF International Conference on Computer Vision (ICCV) ISBN: 9798350307184 Place: Paris, France.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models, June 2024. URL http://arxiv.org/abs/2401.13919. arXiv:2401.13919 [cs].

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. URL http://arxiv.org/abs/1512.03385. arXiv:1512.03385 [cs].

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor, December 2022. URL http://arxiv.org/abs/2212.09689. arXiv:2212.09689 [cs].

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs].

Mengkang Hu, Pu Zhao, Can Xu, Qingfeng Sun, Jianguang Lou, Qingwei Lin, Ping Luo, and Saravan Rajmohan. AgentGen: Enhancing Planning Abilities for Large Language Model based Agent via Environment and Task Generation, November 2024. URL http://arxiv.org/abs/2408.00764. arXiv:2408.00764 [cs].

Yang Hu, Xingyu Zhang, Xueji Fang, Zhiyang Chen, Xiao Wang, Huatian Zhang, and Guojun Qi. SLOT: Sample-specific Language Model Optimization at Test-time, May 2025a. URL http://arxiv.org/abs/2505.12392. arXiv:2505.12392 [cs].

Yuzheng Hu, Pingbang Hu, Han Zhao, and Jiaqi W. Ma. Most Influential Subset Selection: Challenges, Promises, and Beyond, January 2025b. URL http://arxiv.org/abs/2409.18153. arXiv:2409.18153 [cs].

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, 43(2):1–55, March 2025. ISSN 1046-8188, 1558-2868. doi: 10.1145/3703155. URL https://dl.acm.org/doi/10.1145/3703155.

Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-V: Universal Embeddings with Multimodal Large Language Models, July 2024. URL http://arxiv.org/abs/2407.12580. arXiv:2407.12580 [cs].

M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, July 2015. doi: 10.1126/science.aaa8415. URL https://www.science.org/doi/10.1126/science.aaa8415.

Tomoyuki Kagaya, Thong Jing Yuan, Yuxuan Lou, Jayashree Karlekar, Sugiri Pranata, Akira Kinose, Koki Oguri, Felix Wick, and Yang You. RAP: Retrieval-Augmented Planning with Contextual Memory for Multimodal LLM Agents, 2024. URL https://arxiv.org/abs/2402.03610. Version Number: 1.

Vishal Kaushal, Anurag Sahoo, Khoshrav Doctor, Narasimha Raju, Suyash Shetty, Pankaj Singh, Rishabh Iyer, and Ganesh Ramakrishnan. Learning From Less Data: Diversified Subset Selection and Active Learning in Image Classification Tasks, May 2018. URL http://arxiv.org/abs/1805.11191. arXiv:1805.11191 [cs].

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/95323660ed2124450caaac2c46b5ed90-Abstract.html.

Michael Krumdick, Charles Lovering, Varshini Reddy, Seth Ebner, and Chris Tanner. No Free Labels: Limitations of LLM-as-a-Judge Without Human Grounding, March 2025. URL http://arxiv.org/abs/2503.05061. arXiv:2503.05061 [cs].

Hsun-Yu Kuo, Yin-Hsiang Liao, Yu-Chieh Chao, Wei-Yun Ma, and Pu-Jen Cheng. Not All LLM-Generated Data Are Equal: Rethinking Data Weighting in Text Classification, March 2025. URL http://arxiv.org/abs/2410.21526. arXiv:2410.21526 [cs].

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention, September 2023. URL http://arxiv.org/abs/2309.06180. arXiv:2309.06180 [cs].

Etienne Laliberté and Pierre Legendre. A distance-based framework for measuring functional diversity from multiple traits. *Ecology*, 91(1):299–305, 2010. ISSN 1939-9170. doi: 10.1890/08-2244.1. URL https://onlinelibrary.wiley.com/doi/abs/10.1890/08-2244.1. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1890/08-2244.1.

Joel Lehman and Kenneth O. Stanley. Abandoning Objectives: Evolution Through the Search for Novelty Alone. *Evolutionary Computation*, 19(2):189–223, June 2011. ISSN 1063-6560. doi: 10.1162/EVCO_a_00025. URL https://ieeexplore.ieee.org/abstract/document/6793380.

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. Spider 2.0: Evaluating Language Models on Real-World Enterprise Text-to-SQL Workflows, November 2024. URL http://arxiv.org/abs/2411.07763. arXiv:2411.07763 [cs].

Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21298–21308, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.02064. URL https://ieeexplore.ieee.org/document/9878775/.

Haoyang Li, Jing Zhang, Hanbing Liu, Ju Fan, Xiaokang Zhang, Jun Zhu, Renjie Wei, Hongyan Pan, Cuiping Li, and Hong Chen. CodeS: Towards Building Open-source Language Models for Text-to-SQL, February 2024. URL http://arxiv.org/abs/2402.16347. arXiv:2402.16347.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs, November 2023a. URL http://arxiv.org/abs/2305.03111. arXiv:2305.03111 [cs].

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. *Advances in Neural Information Processing Systems*, 36:41451–41530, December 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating Object Hallucination in Large Vision-Language Models, 2023c. URL https://arxiv.org/abs/2305.10355. Version Number: 3.

Yujia Li, Kevin Swersky, and R. Zemel. Generative Moment Matching Networks, February 2015. URL https://www.semanticscholar.org/paper/2904a9932f4cd0f0886121dc1f2d4aaac0455176.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations, 2023d. URL https://arxiv.org/abs/2310.07849. Version Number: 2.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, February 2024a. ISSN 2307-387X. doi: 10.1162/tacl_a_00638.

URL https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00638/119630/Lost-in-the-Middle-How-Language-Models-Use-Long.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. RegMix: Data Mixture as Regression for Language Model Pre-training, January 2025. URL http://arxiv.org/abs/2407.01492. arXiv:2407.01492 [cs].

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. APIGen: Automated Pipeline for Generating Verifiable and Diverse Function-Calling Datasets, June 2024b. URL http://arxiv.org/abs/2406.18518. arXiv:2406.18518.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey, June 2024. URL http://arxiv.org/abs/2406.15126. arXiv:2406.15126 [cs].

Siyu Lu, Jialiang Guo, Shan Liu, Bo Yang, Mingzhe Liu, Lirong Yin, and Wenfeng Zheng. An Improved Algorithm of Drift Compensation for Olfactory Sensors. *Applied Sciences*, 12(19):9529, September 2022. ISSN 2076-3417. doi: 10.3390/app12199529. URL https://www.mdpi.com/2076-3417/12/19/9529.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning, 2023. URL https://arxiv.org/abs/2310.01061. Version Number: 2.

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. WebLINX: Real-World Website Navigation with Multi-Turn Dialogue, September 2024. URL http://arxiv.org/abs/2402.05930. arXiv:2402.05930 [cs].

Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. DataDecide: How to Predict Best Pretraining Data with Small Experiments, July 2025. URL http://arxiv.org/abs/2504.11393. arXiv:2504.11393 [cs].

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 3470–3487, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.244. URL https://aclanthology.org/2022.acl-long.244.

Shikhar Murty, Hao Zhu, Dzmitry Bahdanau, and Christopher D. Manning. NNetNav: Unsupervised Learning of Browser Agents Through Environment Interaction in the Wild, February 2025. URL http://arxiv.org/abs/2410.02907. arXiv:2410.02907 [cs].

Anders Giovanni Møller, Jacob Aarup Dalsgaard, Arianna Pera, and L. Aiello. The Parrot Dilemma: Human-Labeled vs. LLM-augmented Data in Classification Tasks, April 2023. URL https://www.semanticscholar.org/paper/The-Parrot-Dilemma%3A-Human-Labeled-vs.-LLM-augmented-M%C3%B8ller-Dalsgaard/73051b7b25ef972c15ea8e7a221f4361991facbe.

Mihai Nadas, Laura Diosan, and Andreea Tomescu. Synthetic Data Generation Using Large Language Models: Advances in Text and Code. *IEEE Access*, 13:134615–134633, 2025. ISSN 2169-3536. doi: 10.1109/ACCESS.2025.3589503. URL http://arxiv.org/abs/2503.14023. arXiv:2503.14023 [cs].

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient Test-Time Model Adaptation without Forgetting, 2022. URL https://arxiv.org/abs/2204.02610. Version Number: 2.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke,

Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, March 2024. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto.

Proving Test Set Contamination in Black Box Language Models, 2023. URL https://arxiv.org/abs/2310.17623. Version Number: 2.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, March 2022. URL http://arxiv.org/abs/2203.02155. arXiv:2203.02155 [cs].

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via Contrastive Activation Addition, July 2024. URL http://arxiv.org/abs/2312.06681. arXiv:2312.06681 [cs].

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Karl Pearson and Francis Galton. VII. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, January 1997. doi: 10.1098/rspl.1895.0041. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspl.1895.0041.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers, November 2021. URL http://arxiv.org/abs/2102.01454. arXiv:2102.01454 [cs].

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. APIGen-MT: Agentic Pipeline for Multi-Turn Data Generation via Simulated Agent-Human Interplay, July 2025. URL http://arxiv.org/abs/2504.03601. arXiv:2504.03601 [cs].

Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Agent Planning with World Knowledge Model, 2024. URL https://arxiv.org/abs/2405.14205. Version Number: 4.

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, June 2022. ISBN 978-0-262-54587-7. Google-Books-ID: MBZuEAAAQBAJ.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report, January 2025. URL http://arxiv.org/abs/2412.15115. arXiv:2412.15115 [cs].

Sebastian Risi, Sandy D. Vanderbleek, Charles E. Hughes, and Kenneth O. Stanley. How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 153–160, New York, NY, USA, July 2009. Association for Computing Machinery. ISBN 978-1-60558-325-9. doi: 10.1145/1569901.1569923. URL https://dl.acm.org/doi/10.1145/1569901.1569923.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark, 2023. URL https://arxiv.org/abs/2310.18018. Version Number: 1.

Erich Schubert and Peter J. Rousseeuw. Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms. In Giuseppe Amato, Claudio Gennaro, Vincent Oria, and Miloš Radovanović, editors, *Similarity Search and Applications*, pages 171–187, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32047-8. doi: 10.1007/978-3-030-32047-8_16.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RL on Incorrect Synthetic Data Scales the Efficiency of LLM Math Reasoning by Eight-Fold, 2024. URL https://arxiv.org/abs/2406.14532. Version Number: 1.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631 (8022):755–759, July 2024. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-024-07566-y. URL https://www.nature.com/articles/s41586-024-07566-y.

Viacheslav Sinii, Alexey Gorbatovski, Artem Cherepanov, Boris Shaposhnikov, Nikita Balagansky, and Daniil Gavrilov. Steering LLM Reasoning Through Bias-Only Adaptation, October 2025. URL http://arxiv.org/abs/2505.18706. arXiv:2505.18706 [cs].

C. Spearman. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1):72, January 1904. ISSN 00029556. doi: 10.2307/1412159. URL https://www.jstor.org/stable/1412159?origin=crossref.

Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, Ben Kao, Guohao Li, Junxian He, Yu Qiao, and Zhiyong Wu. OS-Genesis: Automating GUI Agent Trajectory Construction via Reverse Task Synthesis, December 2024. URL http://arxiv.org/abs/2412.19723. arXiv:2412.19723 [cs].

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9229–9248. PMLR, November 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Daniel Tan, David Chanin, Aengus Lynch, Dimitrios Kanoulas, Brooks Paige, Adria Garriga-Alonso, and Robert Kirk. Analyzing the Generalization and Reliability of Steering Vectors, May 2025. URL http://arxiv.org/abs/2407.12404. arXiv:2407.12404 [cs].

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. Large Language Models for Data Annotation and Synthesis: A Survey, December 2024. URL http://arxiv.org/abs/2402.13446. arXiv:2402.13446 [cs].

Lucas-Andrei Thil, Mirela Popa, and Gerasimos Spanakis. Navigating WebAI: Training Agents to Complete Web Tasks with Large Language Models and Reinforcement Learning. *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, pages 866–874, April 2024. doi: 10.1145/3605098.3635903. URL https://dl.acm.org/doi/10.1145/3605098.3635903. Conference Name: SAC '24: 39th ACM/SIGAPP Symposium on Applied Computing ISBN: 9798400702433 Place: Avila Spain.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, February 2023. URL http://arxiv.org/abs/2302.13971. arXiv:2302.13971 [cs].

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering Language Models With Activation Engineering, October 2024. URL http://arxiv.org/abs/2308.10248. arXiv:2308.10248 [cs].

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-time Adaptation by Entropy Minimization, March 2021. URL http://arxiv.org/abs/2006.10726. arXiv:2006.10726 [cs].

Junjie Wang, Mingyang Chen, Binbin Hu, Dan Yang, Ziqi Liu, Yue Shen, Peng Wei, Zhiqiang Zhang, Jinjie Gu, Jun Zhou, Jeff Z. Pan, Wen Zhang, and Huajun Chen. Learning to Plan for Retrieval-Augmented Large Language Models from Knowledge Graphs, 2024a. URL https://arxiv.org/abs/2406.14282. Version Number: 3.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent Workflow Memory, September 2024b. URL http://arxiv.org/abs/2409.07429. arXiv:2409.07429 [cs].

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments, May 2024. URL http://arxiv.org/abs/2404.07972. arXiv:2404.07972 [cs].

Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. AgentTrek: Agent Trajectory Synthesis via Guiding Replay with Web Tutorials, December 2024. URL http://arxiv.org/abs/2412.09605. arXiv:2412.09605 [cs].

Jiaxi Yang, Binyuan Hui, Min Yang, Jian Yang, Junyang Lin, and Chang Zhou. Synthesizing Text-to-SQL Data from Weak and Strong LLMs, August 2024a. URL http://arxiv.org/abs/2408.03256. arXiv:2408.03256 [cs].

Ke Yang, Jiateng Liu, John Wu, Chaoqi Yang, Yi R. Fung, Sha Li, Zixuan Huang, Xu Cao, Xingyao Wang, Yiquan Wang, Heng Ji, and Chengxiang Zhai. If LLM Is the Wizard, Then Code Is the Wand: A Survey on How Code Empowers Large Language Models to Serve as Intelligent Agents, 2024b. URL https://arxiv.org/abs/2401.00812. Version Number: 2.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. AgentOccam: A Simple Yet Strong Baseline for LLM-Based Web Agents, May 2025. URL http://arxiv.org/abs/2410.13825. arXiv:2410.13825 [cs].

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains, June 2024. URL http://arxiv.org/abs/2406.12045. arXiv:2406.12045 [cs].

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.801. URL https://aclanthology.org/2022.emnlp-main.801.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. Large Language Model as Attributed Training Data Generator: A Tale of Diversity and Bias, 2023. URL https://arxiv.org/abs/2306.15895. Version Number: 2.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Jiantao Qiu, Lei Cao, Ju Fan, Ye Yuan, Guoren Wang, and Conghui He. Harnessing Diversity for Important Data Selection in Pretraining Large Language Models, October 2024a. URL http://arxiv.org/abs/2409.16986. arXiv:2409.16986 [cs].

Yanzhe Zhang, Tao Yu, and Diyi Yang. Attacking Vision-Language Computer Agents via Pop-ups, November 2024b. URL http://arxiv.org/abs/2411.02391. arXiv:2411.02391 [cs] version: 1.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, December 2023. URL http://arxiv.org/abs/2306.05685. arXiv:2306.05685 [cs].

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A Realistic Web Environment for Building Autonomous Agents, April 2024a. URL http://arxiv.org/abs/2307.13854. arXiv:2307.13854 [cs].

Siyu Zhou, Tianyi Zhou, Yijun Yang, Guodong Long, Deheng Ye, Jing Jiang, and Chengqi Zhang. WALL-E: World Alignment by Rule Learning Improves World Model-based LLM Agents, 2024b. URL https://arxiv.org/abs/2410.07484. Version Number: 2.

Yifei Zhou, Qianlan Yang, Kaixiang Lin, Min Bai, Xiong Zhou, Yu-Xiong Wang, Sergey Levine, and Erran Li. Proposer-Agent-Evaluator(PAE): Autonomous Skill Discovery For Foundation Model Internet Agents, December 2024c. URL http://arxiv.org/abs/2412.13194. arXiv:2412.13194 [cs].

Yuchang Zhu, Huizhe Zhang, Bingzhe Wu, Jintang Li, Zibin Zheng, Peilin Zhao, Liang Chen, and Yatao Bian. Measuring Diversity in Synthetic Datasets, February 2025. URL http://arxiv.org/abs/2502.08512. arXiv:2502.08512 [cs].

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation Engineering: A Top-Down Approach to AI Transparency, March 2025. URL http://arxiv.org/abs/2310.01405. arXiv:2310.01405 [cs].

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-Time Reinforcement Learning, June 2025. URL http://arxiv.org/abs/2504.16084. arXiv:2504.16084 [cs].

Appendix A

Additional Material for SynQuE

A.1 Visualization of ambiguous image



(a) Sample synthetic images of class "stage"



(b) Sample synthetic images of class "throne"

Figure A.1: Visualization of synthetic images from second split for classes (a) "stage" and (b) "throne"

A.2 Additional Ablations

A.2.1 Ablation Study on LENS Preferential Bias

Table A.1: We use two non-Qwen scoring models on tasks where synthetic data was generated by `Qwen2.5` models. The consistent positive correlations of the debiased scores demonstrate that LENS is robust to this potential bias.

LENS (`Granite-8B`)	Debiased		Biased	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.21	.30	-.04	-.02
Text2SQL	.22	.20	.19	.23
LENS (`Ministral-8B`)	Debiased		Biased	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.28	.39	.52	.31
Text2SQL	.32	.29	.16	.16

A.2.2 LENS with Different Backbone Models and Sizes

As shown in table A.2, LENS demonstrates robust ranking capabilities across a diverse range of backbone models and sizes. However, the impact of the debiasing step varies significantly by task complexity and model capability:

- **Text2SQL (Complex Generation):** Debiasing consistently improves or maintains correlation strength across nearly all models. For instance, with `Ministral-8B`, debiasing doubles the Spearman correlation (.16 $\rightarrow$ .32), and it yields the highest overall performance with `Qwen2.5-32B` (.46). This suggests that for complex structural tasks, removing distributional priors remains beneficial even for capable models.
- **Sentiment Analysis (Simple Classification):** The results are more nuanced. For models with severe initial bias, such as `Granite-8B-Instruct`, debiasing is critical, correcting a negative correlation (-.04) to a positive one (.21). However, for highly

capable or larger models (e.g., `GPT-4.1`, `Granite-4.0 (32B)`, `Gemma3-27B`), debiasing tends to degrade performance. We hypothesize that these stronger models are already well-calibrated for simple classification tasks; consequently, our analytical debiasing term may introduce unnecessary noise that overrides the model's accurate intrinsic priors.

A.2.3 Ablation Study on LENS Rubric Design

Our ablation results from table A.3 demonstrate the effectiveness of the rubric component of LENS. Table A.4 demonstrates that generating too few or too many points will degrade the performance of LENS.

A.2.4 Ablation Study on MMD^2 Kernel Functions

Table A.5: Pearson / Spearman correlation coefficients of $\mathbf{MMD}^2$ with different kernel functions.

Task	Polynomial	RBF	Laplacian	Linear	Sigmoid
Sentiment	.67 / .45	.67 / .45	**.78** / **.60**	.67 / .45	.67 / .46
Text2SQL (avg)	.51 / .43	.52 / .44	**.70** / **.51**	.52 / .44	.53 / .44
Image (avg)	.22 / .30	.22 / .30	.21 / .30	.22 / .30	.22 / .30
WebNav	.06 / -.02	.06 / -.01	.05 / -.04	.06 / -.02	.05 / -.03

Based on the ablation results from table A.5, the Laplacian kernel performs best on text tasks, but all kernels show limited effectiveness on more complex domains.

Table A.2: Comparison of original and debiased LENS scores for ranking synthetic datasets across various backbone models and sizes. For GPT-4.1, sentiment analysis results are reported only due to cost constraints.

LENS (Qwen2.5-7B-Instruct)	Debiased		Original	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.25	.33	.26	.17
Text2SQL	.33	.33	.31	.28
LENS (Qwen2.5-32B-Instruct)	Debiased		Original	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.38	.26	.24	.23
Text2SQL	.46	.43	.43	.44
LENS (Ministral-8B-Instruct)	Debiased		Original	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.28	.39	.52	.31
Text2SQL	.32	.29	.16	.16
LENS (Granite-8B-Instruct)	Debiased		Original	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.21	.30	-.04	-.02
Text2SQL	.22	.20	.19	.23
LENS (Granite-4.0-h-small) (32B)	Debiased		Original	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.16	.39	.50	.44
Text2SQL	.32	.36	.40	.48
LENS (Gemma3-12B-Instruct)	Debiased		Original	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.10	.13	.30	.18
Text2SQL	.30	.34	.25	.25
LENS (Gemma3-27B-Instruct)	Debiased		Original	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.17	.33	.37	.31
Text2SQL	.31	.38	.31	.27
LENS (GPT-4.1-2025-04-14)	Debiased		Original	
Task	Spearman	Pearson	Spearman	Pearson
Sentiment	.07	.30	.28	.37

Table A.3: LENS with vs. without a rubric on sentiment analysis, showing the rubric's effectiveness. We use `Qwen2.5-7B-Instruct` for scoring.

LENS Model	**w/ Rubric**	**Spearman**	**Pearson**
Qwen2.5-7B-Instruct	NO	0.23	0.21
Qwen2.5-32B-Instruct	NO	0.32	0.13
Qwen2.5-7B-Instruct	YES	0.25	0.33
Qwen2.5-32B-Instruct	YES	0.38	0.26

Table A.4: Bottom: Varying the number of rubric points, where 10 points provides a good balance.

# Rubric Points	**Sentiment Analysis**		**Text2SQL (Avg)**	
(Debiased)	Spearman	Pearson	Spearman	Pearson
5	0.14	0.19	0.38	0.30
10	0.25	0.33	0.33	0.33
15	-0.06	0.05	0.23	0.21

A.2.5 Ablation Study on MDM with Different Number of Clusters

Table A.6: Pearson / Spearman coefficients of **MDM** with different number of clusters (K=3, 5, 10, 20) on sentiment analysis.

K=3	**K=5**	**K=10**	**K=20**
.85/ .68	.86/ .68	.86/ .69	.86/ .69

The table with varying number of medoid clusters for **MDM** shows that changing the number of clusters has negligible effect on the correlation between **MDM** and F1 score.

A.2.6 Ablation Study on Encoder Choice for Representation-Based Metrics

The results with BGE-M3[1] in table A.7 demonstrate that representation-based metrics are measuring fundamental distribution shifts rather than artifacts of a specific encoder, therefore indicating the effectiveness of representation-based metrics.

Table A.7: Pearson / Spearman correlation coefficients of proxies on sentiment analysis with BGE-M3 as the encoder.

Mauve	**PAD**	**MMD^2**	**MDM**
.60/ .65	.75/ .65	.82/ .67	.86/ .74

A.2.7 Correlation of PAD Predictions from Different Classifiers vs. F1 Scores

In summary, according to table A.8, classifier choice plays a critical role in **PAD** prediction performance, with the random forest model demonstrating the highest correlation with F1 scores. This may be due to the ensemble nature of both the random forest and the sentiment analysis classifier (XGBoost), possibly enhancing alignment between predictions and F1 outcomes, while the MLP shows weaker correspondence.

[1] https://huggingface.co/BAAI/bge-m3

Table A.8: Correlation Coefficients (Pearson / Spearman) of **PAD** predictions from different classifiers vs. F1 scores.

Classifier	Correlation Coefficients (P/S)
Logistic Regression	.65 / .53
Random Forest	.75 / .64
2-layer MLP	.56 / .43

A.2.8 Trade-Off Analysis Between LENS and Diversity Measure

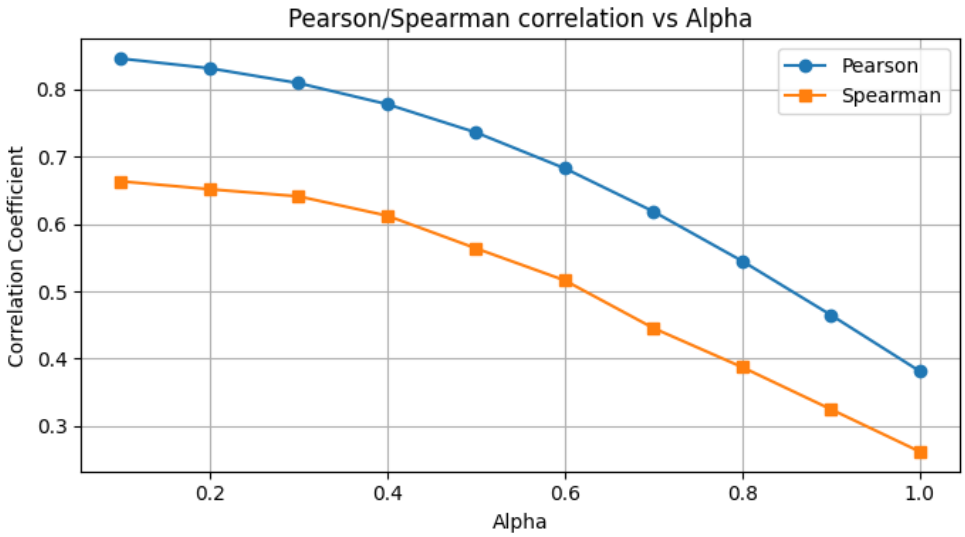


Figure A.2: Pearson/Spearman correlation coefficients of the hybrid score vs. alpha. Both correlation coefficients are monotonically decreasing as alpha increases (more LENS impact and less **MDM** impact).

To further understand the interplay between LENS and the diversity measure, we conducted an experiment using a "hybrid" score, defined as $\alpha \cdot \text{LENS} + (1 - \alpha) \cdot \textbf{MDM}$. As shown in fig. A.2, our analysis reveals that increasing α – thereby placing greater weight on LENS and less on diversity – results in a monotonic decrease in both the Pearson and Spearman correlation coefficients. This trend suggests that the LENS and diversity measures do not demonstrate synergistic effects when combined.

A.2.9 Analysis on Using Hybrid Proxies for Dataset Selection

To understand the synergistic potential of different synthetic data metrics, we evaluated over 7,700 linear combinations (6 settings for each proxy, ranging from 0.0 to 1.0 with a step size of 0.2, resulting in 6^5 combinations) of our five primary proxies. We expanded this experiment to include both the Sentiment Analysis and Text2SQL "movie_platform" domains (as this domain provides the most instances and is therefore a more stable Text2SQL domain) to analyze how hybrid performance scales with task complexity. Here are the core findings based on results from table A.9:

- **Accurate Dataset Ranking Requires Synergy:** While individual metrics establish a baseline, combining them yields strict improvements, particularly for rank-ordering. In Sentiment Analysis, blending metrics increases the maximum single-proxy Spearman correlation ($\max \rho$) from 0.68 to 0.72. In the more complex Text2SQL domain, the synergistic gains are significant: the hybrid approach boosts the maximum single-proxy Spearman correlation ($\max \rho$) from 0.65 to 0.89, and the maximum Pearson correlation ($\max r$) from 0.68 to 0.79.
- **The Optimal Blend Varies from Task to Task:** The optimal hybrid composition shifts significantly depending on the nature of the task – there is no single combination of proxies that performs the best across tasks. For simple stylistic tasks like Sentiment Analysis, representation-based metrics dominate the top combinations, with **MDM** acting as a primary anchor. However, for structurally complex reasoning tasks like Text2SQL, LENS becomes a critical component for both Pearson and Spearman correlations, consistently requiring heavy weights (between 0.6 and 1.0) to achieve optimal utility prediction.
- **Proxies Capture Orthogonal Blind Spots:** The necessity of this blend confirms that these metrics measure fundamentally distinct properties. A synthetic dataset might exhibit excellent diversity but drift out-of-domain, or it might perfectly mimic the real distribution's style while suffering from severe mode collapse. Furthermore, representation metrics may miss nuanced logical errors in complex queries that LLM-based reasoning (LENS) catches. The top-performing hybrid proxies act as a counterbalance, successfully penalizing datasets that fail on any front.

Overall, if the goal is to select the absolute best synthetic dataset for downstream training, practitioners must use a holistic scoring approach that demands broad embedding coverage (i.e., diversity), strict distributional alignment, and – particularly for complex tasks – explicit reasoning-based evaluation.

Table A.9: Top 10 hybrid proxies by Spearman (ρ) and Pearson (r) correlation for dataset selection in the Sentiment Analysis (upper block) and Text2SQL (lower block) domains. Each row details a specific weighted combination of **PAD**, **MMD^2**, **MDM**, LENS, and MAUVE, alongside its resulting correlation to downstream task performance. For baseline comparison, $\hat{\rho}$ and $\hat{r}$ denote the maximum correlation achieved by the best-performing individual metric from table 3.3, whereas ρ and r denote the correlation achieved by the hybrid combination.

Spearman							**Pearson**						
$\hat{\rho}$	ρ	**PAD**	**MMD^2**	**MDM**	**LENS**	**MAUVE**	$\hat{r}$	r	**PAD**	**MMD^2**	**MDM**	**LENS**	**MAUVE**
Sentiment Analysis													
0.68	0.72	0.0	0.4	0.8	0.0	0.2	0.85	0.85	0.0	0.0	0.2	0.0	0.0
0.68	0.72	0.0	0.6	1.0	0.0	0.4	0.85	0.85	0.0	0.0	0.4	0.0	0.0
0.68	0.72	0.0	0.4	0.6	0.0	0.2	0.85	0.85	0.0	0.0	0.6	0.0	0.0
0.68	0.71	0.0	0.6	0.8	0.0	0.4	0.85	0.85	0.0	0.0	0.8	0.0	0.0
0.68	0.71	0.2	0.6	1.0	0.0	0.6	0.85	0.85	0.0	0.0	1.0	0.0	0.0
0.68	0.71	0.0	0.6	1.0	0.0	0.2	0.85	0.85	0.2	0.0	1.0	0.0	0.2
0.68	0.71	0.0	0.8	1.0	0.0	0.6	0.85	0.85	0.2	0.0	0.8	0.0	0.2
0.68	0.71	0.2	0.4	0.8	0.0	0.4	0.85	0.85	0.0	0.0	1.0	0.0	0.2
0.68	0.71	0.0	0.8	1.0	0.0	0.4	0.85	0.85	0.0	0.2	1.0	0.0	0.2
0.68	0.71	0.0	0.2	1.0	0.0	0.0	0.85	0.84	0.0	0.0	0.8	0.0	0.2
Text2SQL													
0.65	0.89	0.6	0.8	0.4	0.6	0.2	0.68	0.79	0.0	1.0	0.2	0.8	0.2
0.65	0.89	0.6	0.6	0.4	0.6	0.2	0.68	0.79	0.0	0.8	0.2	0.8	0.2
0.65	0.89	0.2	0.2	0.6	0.8	0.2	0.68	0.79	0.0	0.6	0.2	0.8	0.2
0.65	0.89	0.0	0.2	0.6	0.8	0.2	0.68	0.79	0.0	0.4	0.2	0.8	0.2
0.65	0.89	0.2	0.4	0.4	0.6	0.2	0.68	0.79	0.0	0.2	0.2	0.8	0.2
0.65	0.89	0.4	1.0	0.4	0.6	0.2	0.68	0.79	0.0	0.0	0.2	0.8	0.2
0.65	0.89	0.2	0.2	0.4	0.6	0.2	0.68	0.79	0.0	1.0	0.2	1.0	0.2
0.65	0.89	0.4	0.6	0.4	0.6	0.2	0.68	0.79	0.0	0.8	0.2	1.0	0.2
0.65	0.89	0.4	0.4	0.4	0.6	0.2	0.68	0.79	0.0	0.6	0.2	1.0	0.2
0.65	0.89	0.6	0.2	0.4	0.6	0.2	0.68	0.79	0.0	0.4	0.2	1.0	0.2

A.2.10 Ablation on the Number of Samples Used for Scoring on Simple Task

Table A.10: Pearson / Spearman correlation between the LENS score and the test F1 score for different metrics with differet number of real instances used for scoring (100 and 200).

$\lvert\mathcal{U}_\text{r}\rvert = 100$					$\lvert\mathcal{U}_\text{r}\rvert = 200$				
LENS	**PAD**	**MMD**^{2}	**MDM**	MAUVE	LENS	**PAD**	**MMD**^{2}	**MDM**	MAUVE
-.03/-.13	.65/.47	.68/.44	.85/.70	.54/.51	.26/.38	.65/.53	.67/.45	.85/.68	.57/.53

We investigate the sensitivity of LENS proxies to the size of the unannotated real dataset $|\mathcal{U}_r|$ for the sentiment analysis task.As shown in table A.10, representation-based metrics (**PAD**, **MMD**^{2}, **MDM**, MAUVE) demonstrate remarkable robustness in this domain, maintaining strong correlations (e.g., **MDM** Spearman 0.70) even with as few as 100 samples. This aligns with our hypothesis that embedding spaces effectively capture distributional shifts in simpler, stylistic tasks. In contrast, LENS requires a larger sample size to stabilize for this specific task 5, exhibiting weak correlations at $|\mathcal{U}_r| = 100$ (Pearson -0.03, Spearman -0.13) before recovering significantly to a positive Spearman correlation of 0.38 at $|\mathcal{U}_r| = 200$. This behavior highlights a distinct complexity trade-off: while representation metrics are highly data-efficient for stylistic domains, they fail on complex, long-horizon tasks such as Web Navigation. Conversely, LENS proves robust in those more complex domains even with very limited unlabled real data.

A.2.11 Correlation Improvement with De-Hallucination Prompt

Table A.11: Spearman and Pearson correlations for baseline and with de-hallucination prompt (w/ DHP), across different models.

	Baseline	w/ DHP (Delta)
Qwen2.5-7B-Instruct		
Spearman	0.25	0.15 (−0.10)
Pearson	0.33	0.30 (−0.03)
Gemma-3-12B-It		
Spearman	0.10	0.19 (+0.09)
Pearson	0.13	0.24 (+0.11)
Ministral-8B-Instruct-2410		
Spearman	0.28	0.53 (+0.25)
Pearson	0.39	0.49 (+0.10)
Granite3.3-8B-Instruct		
Spearman	0.21	0.23 (+0.02)
Pearson	0.30	0.32 (+0.02)

As shown in Table A.11, incorporating an explicit de-hallucination prompt during the LLM-based scoring phase produces highly model-dependent effects – four out of three models improve correlations.

To incorporate de-hallucination into the LENS scoring prompt, we add the following sentence:

```
If you find the headline is hallucinated (i.e., not a real headline),
answer "unsure".
```

We find that the impact of de-hallucination prompts on correlation metrics is highly model-dependent: while some models benefit noticeably with improved alignment (e.g., substantial gains in proxy reliability for certain models), others show little change or even reduced correlations. This indicates that prompt-based de-hallucination provides inconsistent results across models, suggesting that more robust, systematic approaches are needed to reliably reduce hallucinations during evaluation.

A.2.12 Ablation on the Size of Synthetic Dataset for SynQuE Proxies

Table A.12: Scaling analysis of SynQuE proxies for sentiment analysis. Table entries show Pearson/Spearman correlation coefficients, in the form "Pearson/Spearman", for each proxy at different synthetic dataset size multiples. We report results using debiased Lens, evaluated with `Qwen2.5-(7B/32B)-Instruct` scorers. A size multiple of n indicates that n synthetic datasets (each containing 999 samples) are merged, yielding $n \times 999$ total samples.

Size	LENS-7B	LENS-32B	MDM	MMD^2	PAD	Mauve
1	.33/.25	.26/.38	.85/.68	.67/.45	.65/.53	.57/.53
2	.17/.21	.11/.16	.69/.52	.55/.50	.51/.43	.35/.36
4	.17/.12	.38/.30	-.70/-.81	-.27/-.42	-.25/-.43	-.31/-.32
8	.04/.04	.49/.44	-.51/-.20	.62/.40	.04/.40	-.20/.24

We perform a scaling experiment on sentiment analysis to evaluate how well SynQuE proxies maintain their effectiveness as the size of synthetic datasets increases. Sentiment analysis provides the most abundant synthetic training data (32 datasets), enabling us to systematically merge n datasets together, $n \in \{1, 2, 4, 8\}$.

From the results in table A.12, we observe that increasing the synthetic dataset size generally leads to performance degradation across all proxies. In particular, when the size multiple reaches 4, Lens remains capable of achieving a positive correlation, suggesting greater robustness to growing dataset size compared to other methods. It is important to note that, throughout these experiments, the number of synthetic samples chosen for rubric compilation in Lens is fixed at 200, which means that as dataset size increases, the relative proportion of data represented in the scoring rubric decreases, potentially limiting its ability to capture the full variance present in larger datasets.

A.2.13 Ablation on Instance-Level Data Selection

We explore whether SynQuE proxies can be leveraged for *instance-level* data selection, as opposed to the default *dataset-level* ranking. In this ablation, for sentiment analysis, we aggregate all deduplicated synthetic datasets into a single pool containing 29,235 instances. From this pool, for each class, we select the top 333 instances according to each proxy score, yielding a final synthetic dataset of 999 samples (333 per class).

Table A.13: Instance-level downstream task performance (%) on sentiment analysis for synthetic datasets selected by each SynQuE proxy. Debiased Lens uses `Qwen2.5-7B-Instruct` as the backbone model for scoring

Proxy	**Average**	Lens	**PAD**	**MMD^2**	**MDM**	Mauve
Result	49.6	54.8	55.9	53.2	n/a	52.7

The results in table A.13 reveal: *discriminative* proxies – namely **PAD** and Lens – achieve higher downstream task performance when selecting at the instance level, compared to their performance with dataset-level selection. Specifically, incorporating comparisons to table 3.2, instance-level selection improves **PAD**'s performance (55.9% vs. 55.3%) and Lens's performance (54.8% vs. 50.5%) over the dataset-level setting.

In contrast, for the other proxies such as **MMD^2** and Mauve, switching from dataset-level to instance-level selection leads to reduced task performance, indicating that these methods are less suitable for fine-grained instance selection. We also note that **MDM** is not applicable at the instance level, since it measures diversity across a set of samples and cannot score individual ones – hence it is marked as "n/a" in the table.

Overall, these results demonstrate that discriminative proxies can better exploit instance-level selection, whereas distributional measures suffer from a loss of signal in this setting.

A.3 Additional Experimental Details

A.3.1 Data Curation

Sentiment analysis For sentiment analysis, we used a cleaned version of the original validation split as the test set, removing URLs from the data. Synthetic samples are generated and validated to ensure each output is non-empty, alphabetic 'headline' and a 'sentiment' label restricted to the values '0', '1', or '2'. Only samples meeting these structural and content criteria were retained for downstream analysis.

Text2SQL During Text2SQL data synthesis, we validate generated question–SQL pairs by ensuring both fields are non-empty, contain alphanumeric content, and are formatted as a dictionary with 'question' and 'SQL' keys. For non-zero-shot generations, SQL queries are executed against the target database; pairs failing execution are discarded. This process enforces correct structure, meaningful content, and SQL executability.

Image classification We classify each image in the `unmet-promise` dataset into one of three categories: *label_ only*, *label_ relation*, or *label_ background*, based on its caption. Captions are lowercased and stemmed. If a class-specific background keyword appears in the caption, the image is assigned to *label_ background*. If a relation keyword is present, it is assigned to *label_ relation*. Images not matching either are assigned to *label_ only*. We balance the number of samples per category and class, and store the processed data for downstream tasks. To further ensure fidelity of images, we use `Qwen2.5-VL-7B-Instruct` to filter noisy images. Prompt used for filtering is provided in listing A.6. For image classification, we create a test set for each split by selecting 150 samples that are balanced across all labels.

Web navigation We first group individual trajectories into tasks. Then we use different seeds to create 5 disjoint subsets with equal amount of tasks. We fine-tune using LoRA rank of 64 and the open-instruct [2] fine-tuning codebase.

A.3.2 SynQuE Scoring

For **PAD**, we reserve 20% of all embeddings as a holdout test set to train the classifier, and compute the classification error on this set to obtain the final **PAD** score. For $\mathbf{MMD}^2$, we generate the proxy score using a polynomial kernel [3] with degree 3 and Coef0 parameter 1. For **MDM**, we use Fasterpam [Schubert and Rousseeuw, 2019] to compute k medoids in the embeddings, setting k to 3 for sentiment analysis and 5 for other tasks, using Euclidean distance for clustering; **MDM** is then calculated by averaging the Euclidean distance from each data point to its corresponding medoid. For Lens, the prompts used for rubric compilation and scoring are provided in section A.3.8 and section A.3.8, respectively. For Mauve calculation, we use the default hyperparameter, the same embedding model as the other representation-based proxies (*qte-Qwen2-7B-Instruct*), and the scaling factor is 5. For Perplexity calculation in Text2SQL, we first reformat each Text2SQL question as a string: `"question": <sample>`. We then use `Qwen2.5-7B` to compute the perplexity, considering only the tokens corresponding to the question text (i.e., the <sample> portion), and excluding the prompt tokens (such as `"question":`).

[2] https://github.com/allenai/open-instruct

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.polynomial_kernel.html

A.3.3 Dataset Size Descriptions

Table A.14: Dataset sizes for synthetic ($\mathcal{D}_s$), rubric selection, real subset ($\mathcal{U}_r$), and full real dataset ($\mathcal{D}_r$) for all tasks.

Task	Synthetic	LENS Rubric Compilation	Real Subset	Full Real
Sentiment Analysis	999	400	200	2388
		Text-to-SQL		
Movie Platform	1000	60	30	164
App Store	1000	60	30	60
Computer Student	1000	60	30	69
Image Classification	1500	200	100	150
		Web Navigation		
Allrecipes	79	99	20	45
Amazon	63	83	20	41
Apple	70	90	20	43
ArXiv	80	100	20	43
BBC	69	89	20	42
Coursera	72	92	20	42
dictionary	54	74	20	43
ESPN	62	82	20	44
GitHub	71	91	20	41
Google Maps	75	95	20	41
Google Search	72	92	20	43
Hugging Face	76	96	20	43
Wolfram Alpha	66	86	20	46

Table A.15: Number of synthetic datasets available per task.

Task	Number of Synthetic Datasets
Sentiment Analysis	32
Text-to-SQL	8
Image Classification	6
Web Navigation	65

A.3.4 Standard deviation of correlation coefficients

Table A.16: Standard deviation of Spearman (left) and Pearson (right) correlation coefficients of SynQuE proxy metrics across 5 seeds. **MDM** only scores on synthetic datasets therefore no change in input data.

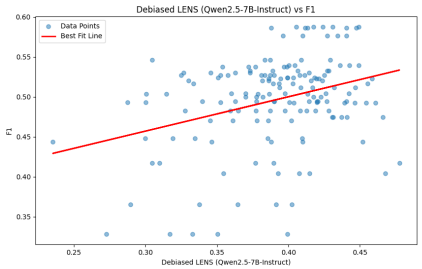
Tasks	Lens 7B		Lens 32B		PAD		MMD		MDM	
	debiased	biased	debiased	biased						
Sentiment	.23	.17 .12	.14	.09 .11	.07	.07 .02	.02	.02	.00	.00
Text2SQL										
Computer	.48	.39 .38	.44	.21 .21	.18	.14 .04	.08	.03	.00	.00
Apps	.38	.34 .25	.21	.22 .22	.22	.14 .36	.48	.05	.00	.00
Movies	.32	.32 .33	.24	.22 .24	.11	.14 .07	.13	.05	.00	.00
Average	.39	.35 .32	.30	.22 .22	.17	.14 .15	.23	.04	.00	.00
Image										
Split 1	.47	.48 .51	.51	.39 .38	.28	.22 .08	.06	.00	.00	.00
Split 2	.43	.47 .43	.42	.40 .41	.43	.53 .11	.08	.00	.00	.00
Split 3	.52	.39 .47	.39	.40 .42	.29	.22 .07	.07	.00	.00	.00
Average	.47	.44 .47	.44	.40 .40	.33	.32 .09	.07	.00	.00	.00
WebNav	.15	.17 .11	.13	.09 .11	.15	.19 .04	.02	.02	.02	.00 .00

A.3.5 Scatter Plot of Lens Scores and Task Utility

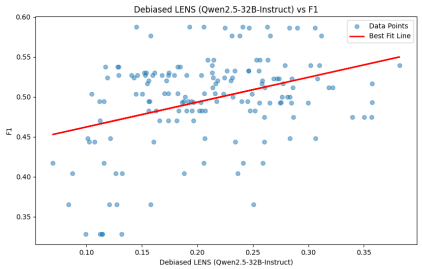
Sentiment Analysis

Scatter plots of debiased SynQuE scores and baselines vs. F1 utility on sentiment analysis datasets. Note that we use the raw scores of $\mathbf{MMD}^2$ and **PAD** – we negate them to have

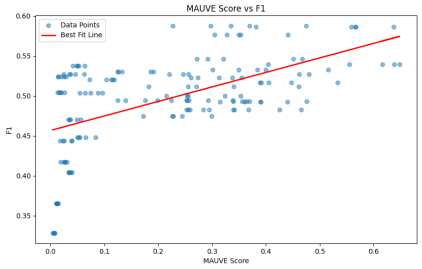
a positive correlation with task utility for dataset selection.



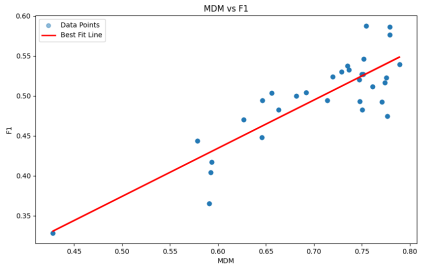
(a) Debiased LENS (`Qwen2.5-7B-Instruct`) vs. F1



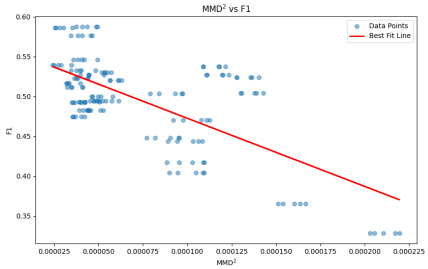
(b) Debiased LENS (`Qwen2.5-32B-Instruct`) vs. F1



(c) MAUVE vs. F1



(d) **MDM** vs. F1



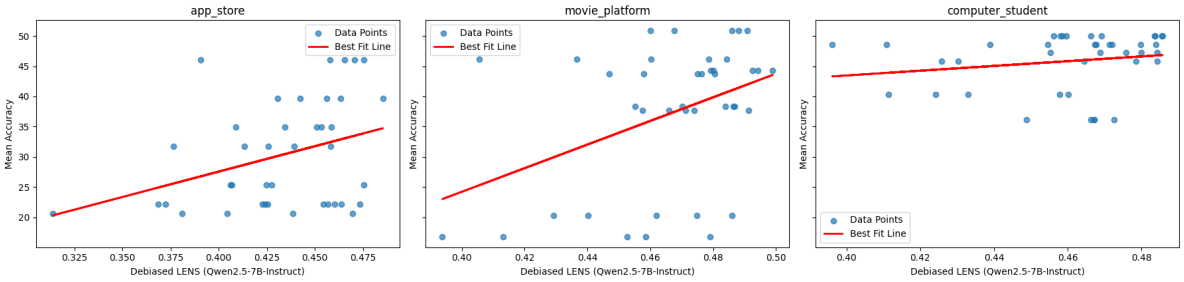
(e) **MMD**^{2} vs. F1



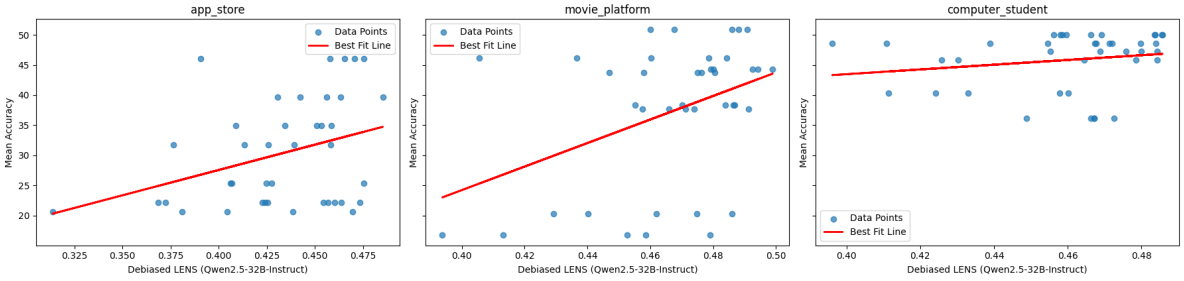
(f) **PAD** vs. F1

Text-to-SQL

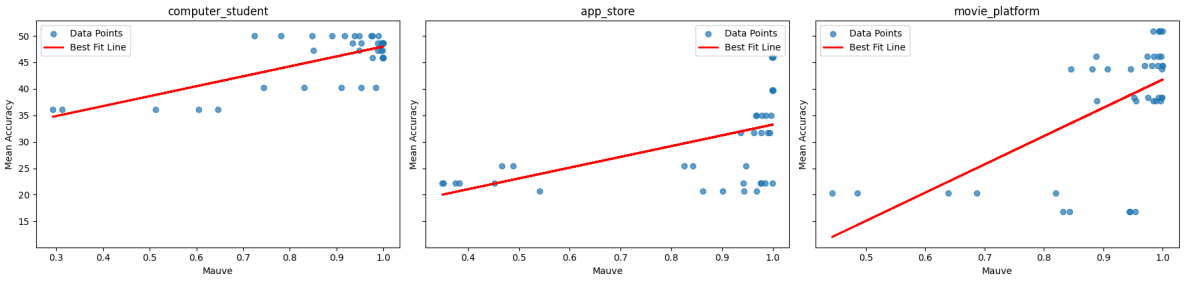
Below are scatter plots of SynQuE scores vs. accuracy utility on text-to-SQL datasets. For **MMD**^{2} and **PAD**, we use raw scores and negate them for positive correlation with task utility for dataset selection.



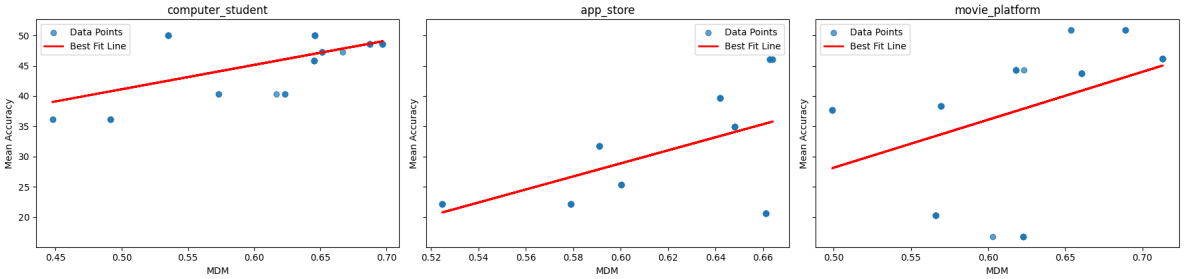
(a) Debiased Lens (`Qwen2.5-7B-Instruct`) vs. Accuracy



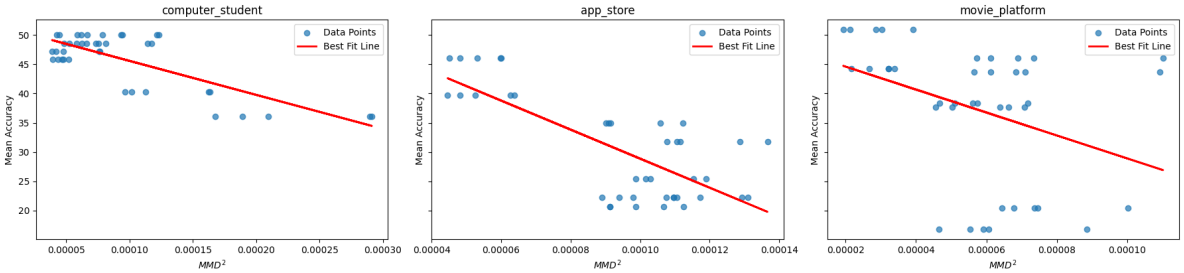
(b) Debiased Lens (`Qwen2.5-32B-Instruct`) vs. Accuracy



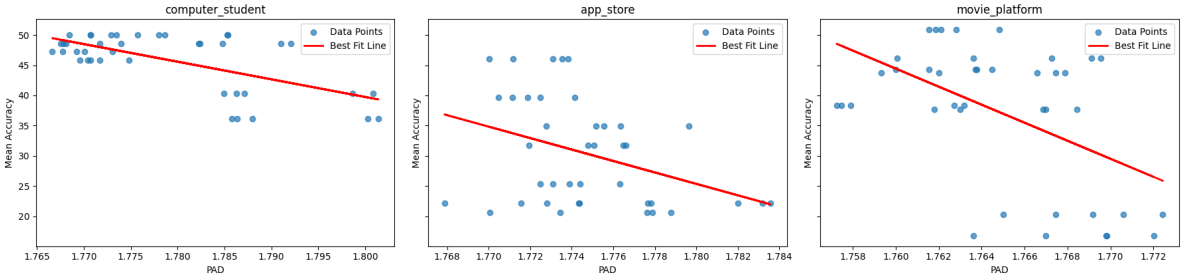
(c) Mauve vs. Accuracy



(d) **MDM** vs. Accuracy



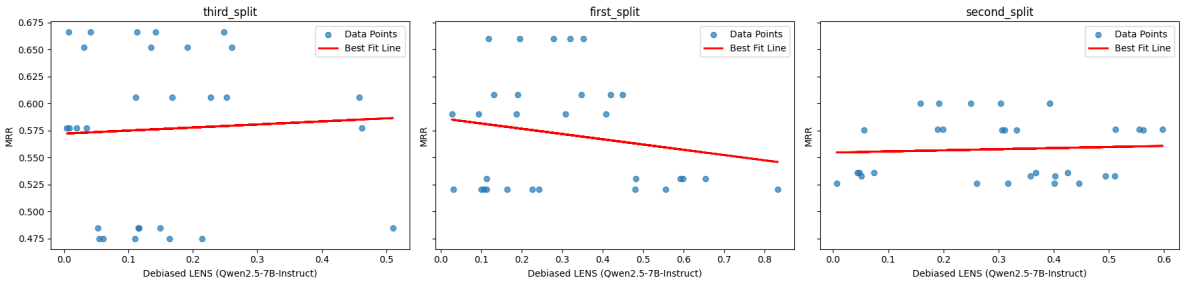
(e) $\mathbf{MMD}^2$ vs. Accuracy



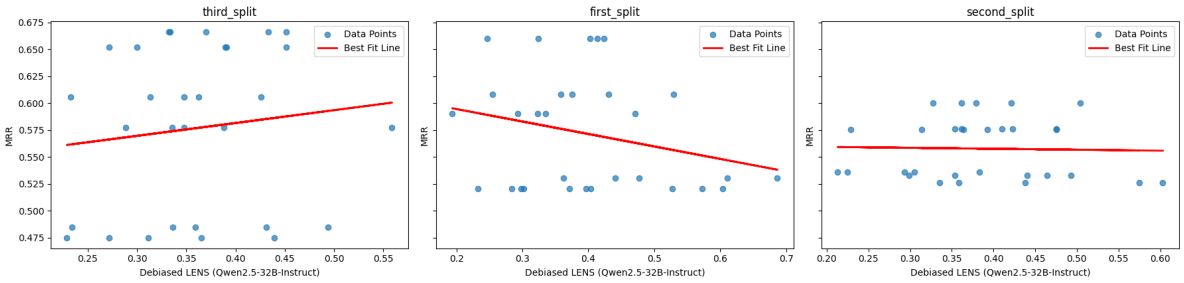
(f) **PAD** vs. Accuracy

Image Classification

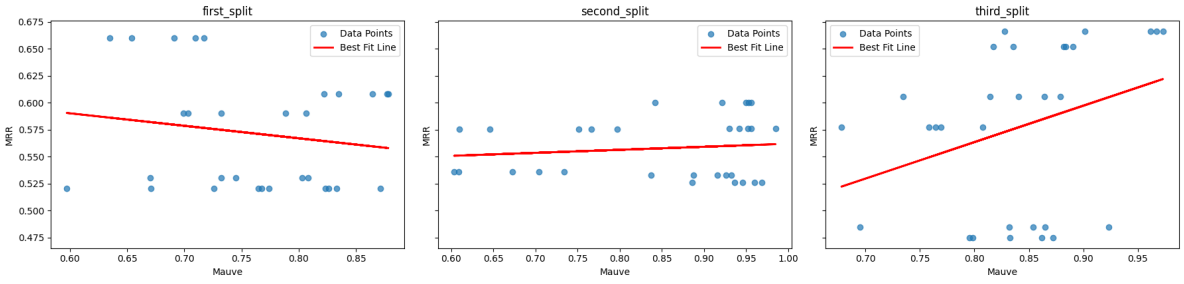
Below are scatter plots SynQuE scores vs. MRR utility on image classification datasets. For **MMD**^{2} and **PAD**, we use raw scores and negate them for positive correlation with task utility for dataset selection.



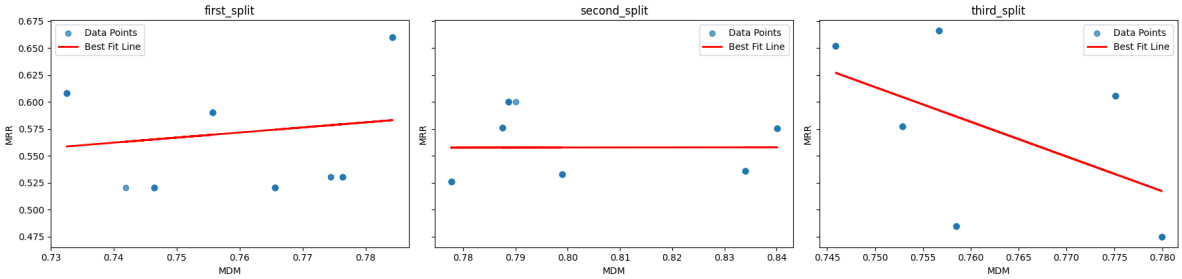
(a) Debiased Lens (`Qwen2.5-7B-Instruct`) vs. MRR



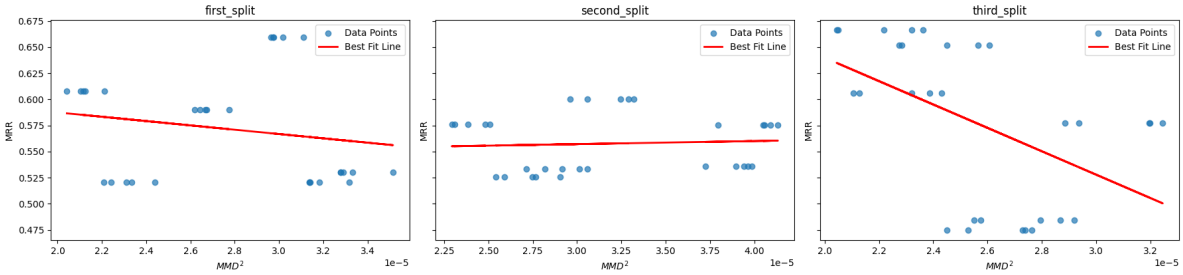
(b) Debiased Lens (`Qwen2.5-32B-Instruct`) vs. MRR



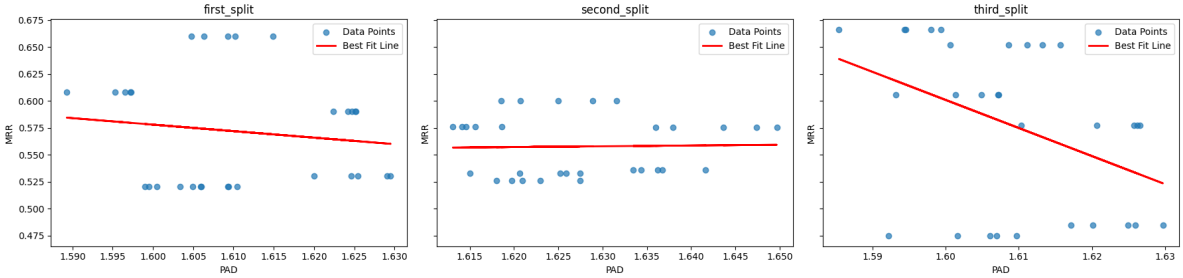
(c) Mauve vs. MRR



(d) **MDM** vs. MRR



(e) **MMD**^{2} vs. MRR



(f) **PAD** vs. MRR

A.3.6 SynQuE proxy cost and efficiency analysis

All computational cost analyses were performed on a system equipped with an Nvidia RTX Pro 6000 Blackwell Server Edition GPU[4] and a 12-core Intel(R) Xeon(R) Gold 5415+ CPU[5].

[4] https://www.nvidia.com/en-us/data-center/rtx-pro-6000-blackwell-server-edition/

[5] https://www.intel.com/content/www/us/en/products/sku/232373/intel-xeon-gold-5415-processor-22-5m-cache-2-90-ghz/specifications.html

Table A.17: Comparison of computational cost across different proxy methods in sentiment analysis. Columns: **Emb.** (embedding latency), **Score** (scoring latency), **Total** (end-to-end latency), and **Tokens** (total tokens for LENS, including input and output tokens). Latencies reported in seconds per synthetic dataset (999 samples), with standard deviation in parentheses. LENS uses Qwen2.5 models.

Method	**Emb. (s/$\mathcal{D}$)**	**Score (s/$\mathcal{D}$)**	**Total (s/$\mathcal{D}$)**	**Num. of Tokens**
LENS(7B)	–	90.52 ± 4.84	90.52 ± 4.84	9264 ± 440
LENS(32B)	–	271.25 ± 20.54	271.25 ± 20.54	9264 ± 440
PAD	6.08 ± 0.66	1.99 ± 0.29	8.07 ± 0.72	–
MDM	6.08 ± 0.66	0.24 ± 0.01	6.32 ± 0.66	–
MMD^2	6.08 ± 0.66	0.04 ± 0.01	6.12 ± 0.66	–
MAUVE	6.08 ± 0.66	106.43 ± 6.30	112.51 ± 6.33	–

Across all sentiment analysis datasets, we observe that **MMD^2** offers the lowest end-to-end computational cost, closely followed by **MDM** and **PAD**, which also demonstrate strong efficiency. Notably, while LENS relies on LLM-based scoring—which substantially increases latency compared to embedding-based proxies—its runtime in the 7B setting (90.52 s/$\mathcal{D}$) is still considerably lower than that of MAUVE (112.51 s/$\mathcal{D}$). This suggests that, although LLM scoring introduces additional overhead, LENS can remain competitive in throughput relative to existing generative evaluation methods when using smaller-scale LLMs.

A.3.7 Example Rubrics

Listing A.1: Example characteristic descriptions $C_{\mathrm{s,r}}$

```
"Dataset B consistently specifies the analyst behind actions...",
"Dataset B maintains strict financial focus without political or
    entertainment tangents present in A...",
"Dataset B entries always directly connect stock movements to specific
    analyst actions...",
"Dataset B shows more frequent price target amount disclosures...",
"Dataset B uses standardized financial terminology consistently...",
"Dataset B maintains neutral tone in earnings reports...",
"Dataset B focuses exclusively on institutional analyst
    perspectives...",
"Dataset B headlines strictly follow '[Analyst] [Action] on [Ticker]
```

```
    [Rationale]' structure...",
"Dataset B contains no social media tags/hashtags...",
"Dataset B shows higher frequency of ETF coverage..."
```

Listing A.2: Example characteristic descriptions $C_{r,s}$

```
"Dataset B includes headlines without stock tickers ...",
"Dataset B contains non-financial news ...",
"Dataset B incorporates social media-style commentary...",
"Dataset B includes international/non-English company names...",
"Dataset B references non-institutional analysts/sources...",
"Dataset B features headlines about dividends...",
"Dataset B includes legal/regulatory actions unrelated to markets...",
"Dataset B uses technical trading jargon...",
"Dataset B contains macroeconomic commentary without stock links...",
"Dataset B includes non-company-specific index/currency forecasts..."
```

A.3.8 LLM Usage

Model serving We use vLLM [Kwon et al., 2023] to serve open source models such as for data synthesis and dataset scoring. For `Llama3.3-70B-Instruct` model, we use Ollama [6] Q4_K quantized version to construct synthetic datasets for sentiment analysis. We use 2 * Nvidia A40 48GB GPUs for other models for synthesis and scoring. To improve LLMs' generation throughput, we use vLLM's batched inference feature and enable prefix-caching to further improve generation efficiency.

LLM hyperparameter For LENS rubric compilation and scoring, we set temperature to 0 and top_p to 0.95.

Data Synthesis Prompts

Listing A.3: Zero-shot prompt used for sentiment analysis dataset generation.

```
Generate three realistic financial news headlines for sentiment
   analysis.

Guidelines for Generating Headlines:
```

[6] https://ollama.com/

```
Sentiment Labeling:
Each headline must be assigned a sentiment label based on its tone:
    -   Bearish (0): Indicates negative sentiment about a stock or
       market trend.
    -   Bullish (1): Indicates positive sentiment about a stock or
       market trend.
    -   Neutral (2): Indicates neutral or informational tone.

Now, generate three new financial news headlines following these
   guidelines. Please use JSON format and generate one type of each
    sentiment label (0, 1, 2) in your response.
```

Listing A.4: Zero-shot with background knowledge prompt used for sentiment analysis dataset generation.

```
Generate three realistic financial news headlines about stock
   tickers following real-world financial reporting for sentiment
   analysis.

Guidelines for Generating Headlines:

1. Format & Style:
    -   Headlines must be concise and mimic real financial news.
    -   Use sentence case formatting (capitalize only the first
       word and proper nouns).
    -   Some headlines should start with a stock ticker (e.g.,
       $AAPL -), while others should begin with the company name or
        a broader market trend.

2. Ticker Inclusion:
    -   At least one headline should include a stock ticker (e.g.,
       $TSLA - or $NVDA -).
    -   Some headlines should refer to companies by name instead of
        tickers (e.g., "Alphabet and Meta see price targets cut at
       Barclays").

3. Common Financial Themes:

Ensure headlines reflect realistic financial news topics,
including:
    -   Stock downgrades/upgrades
    -   Price target adjustments
    -   Market trends/economic outlook
```

```
    -   Company performance concerns
    -   Company news
    -   Company announcements
    -   Company events

4. Source Attribution:
    -   When relevant, mention an investment firm, analyst, or
       research group (e.g., Morgan Stanley, Barclays, Oppenheimer)
       .
    -   Do not fabricate research firms-use only well-known
       institutions.

5. Sentiment Labeling:

Each headline must be assigned a sentiment label based on its tone:
    -   Bearish (0): Indicates negative sentiment about a stock or
       market trend.
    -   Bullish (1): Indicates positive sentiment about a stock or
       market trend.
    -   Neutral (2): Indicates neutral or informational tone.

Sentiment Labeling:
Each headline must be assigned a sentiment label based on its tone:
    - Bearish (0): Indicates negative sentiment about a stock or
       market trend.
    - Bullish (1): Indicates positive sentiment about a stock or
       market trend.
    - Neutral (2): Indicates neutral or informational tone.

Now, generate three new financial news headlines following these
   guidelines. Please use JSON format and generate one type of each
    sentiment label (0, 1, 2) in your response.
```

Listing A.5: Zero-shot with background and stock ticker information prompt used for sentiment analysis dataset generation.

```
Generate three realistic financial news headlines about stock
   tickers following real-world financial reporting for sentiment
   analysis.

Guidelines for Generating Headlines:

1. Format & Style:
    -   Headlines must be concise and mimic real financial news.
```

```
    -   Use sentence case formatting (capitalize only the first
       word and proper nouns).
    -   Some headlines should start with a stock ticker (e.g.,
       $AAPL -), while others should begin with the company name or
        a broader market trend.

2. Ticker Inclusion:
    -   At least one headline should include a stock ticker (e.g.,
       $TSLA - or $NVDA -).
    -   Some headlines should refer to companies by name instead of
        tickers (e.g., "Alphabet and Meta see price targets cut at
       Barclays").

3. Common Financial Themes:

Ensure headlines reflect realistic financial news topics, including
   :
    -   Stock downgrades/upgrades
    -   Price target adjustments
    -   Market trends/economic outlook
    -   Company performance concerns
    -   Company news
    -   Company announcements
    -   Company events

4. Source Attribution:
    -   When relevant, mention an investment firm, analyst, or
       research group (e.g., Morgan Stanley, Barclays, Oppenheimer)
       .
    -   Do not fabricate research firms-use only well-known
       institutions.

5. Sentiment Labeling:

Each headline must be assigned a sentiment label based on its tone:
    -   Bearish (0): Indicates negative sentiment about a stock or
       market trend.
    -   Bullish (1): Indicates positive sentiment about a stock or
       market trend.
    -   Neutral (2): Indicates neutral or informational tone.

Now, generate three new financial news headlines about stock
   tickers: {stock_ticker} following these guidelines. Please use
   JSON format and generate one type of each sentiment label (0, 1,
```

```
    2) for diversity.
```

Data Cleaning Prompts

Listing A.6: Prompt used for filtering noisy synthetic images

```
You are a helpful assistant that filters out an image. You will be
   given an image and its corresponding text caption.

You should return true if the primary object in the image is not a
   ${label} in common sense. Return false otherwise.

Image:
{image}

Caption:
{caption}
```

LENS Rubric Compilation Prompts

Listing A.7: Rubric compilation prompt used in sentiment analysis (commonalities)

```
You are a world class data analyst on financial news headline. You
   will be given some financial news headline samples from dataset
   A and dataset B. Based on the provided similar characteristics
   between them, list how B is similar to A. Return {num} points as
    a JSON list of strings. Please focus on specific and granular
   similarities between the two datasets, your generated
   characteristic points should apply to all the samples from the
   two datasets.

Samples from A:
{A}

Samples from B:
{B}
```

Listing A.8: Rubric compilation prompt used in sentiment analysis (differences)

```
You are a world class data analyst on financial news headlines. You
    will be given some financial news headline samples from dataset
    A and dataset B. Based on the provided similar characteristics
   between them, list how B is {feedback} A. Please focus on
   granular differences between the two datasets, your generated
   characteristic points should apply to all the samples from the
   corresponding dataset (A or B). Return {num} points as a JSON
   list of strings.

Similar characteristics between A and B:
{similar_points}

Samples from A:
{A}

Samples from B:
{B}
```

Listing A.9: Rubric compilation prompt used in Text2SQL (commonalities)

```
You are a world class data analyst on database queries in natural
   language. Given samples from dataset A and dataset B, list how B
    is {feedback} A. Return {num} points as a JSON list of strings.
    Please focus on specific and granular similarities between the
   two datasets, your generated characteristic points should apply
   to all the samples.

Question samples from A:
{A}

Question samples from B:
{B}
```

Listing A.10: Rubric compilation prompt used in Text2SQL (differences)

```
You are a world class data analyst on database queries in natural
   language. Given query samples from dataset A and dataset B.
   Based on the provided similar characteristics between them, list
    how B is {feedback} A. Return {num} points as a JSON list of
   strings. Please focus on specific and granular differences
   between the two datasets, your generated characteristic points
   should apply to all the samples from the corresponding dataset (
   A or B).

Similar characteristics between A and B:
```

```
{similar_points}

Question samples from A:
{A}

Question samples from B:
{B}
```

Listing A.11: Rubric compilation prompt used in image classification (commonalities)

```
Below are {num_image} images from dataset B:
{B}

Below are {num_images} images from dataset A:
{A}

Given some samples from image classification dataset A and dataset
   B, list how dataset B is similar to dataset A. Return ${
   num_points} points that summarize the similar characteristics of
    the two datasets. Focus on the characteristics of the image in
   terms of how they are structured, styled, or captured (e.g.,
   lighting, background, composition, etc.) rather than the image
   specifications such as resolution, size, etc. Your generated
   characteristic points should apply to all the samples from the
   corresponding dataset (A or B). Output should be a JSON list of
   strings.

Your listed points:
```

Listing A.12: Rubric compilation prompt used in image classification (differences)

```
Below are {num_image} images from dataset B:
{B}

Below are {num_images} images from dataset A:
{A}

Given some samples from image classification dataset A and dataset
   B, list how dataset B is different from dataset A. Similar
   characteristics are provided below for reference. Return ${
   num_points} points that summarize the characteristics of the two
    datasets (e.g., dataset A is ... dataset B is ...). Focus on
   the characteristics of the images in terms of how they are
   structured, styled, or captured (e.g., lighting, background,
   composition, etc.) rather than the image specifications such as
```

```
   resolution, size, etc. Your generated characteristic points
   should apply to all the samples from the corresponding dataset (
   A or B). Output should be a JSON list of strings.

Similar characteristics:
${similar_characteristics}

Your listed points:
```

Listing A.13: Rubric compilation prompt used in web navigation (commonalities)

```
Given some samples of web navigation tasks and the web
   accessibility tree of dataset A and dataset B, list how B is {
   feedback} A. Return {num} points that summarize the
   characteristics of the two datasets. The two accessibility trees
    generated from the same website are provided. Please only list
   out characteristics that are related to the proposed web
   navigation tasks, the accessibility trees are provided to only
   help you understand the context of the proposed tasks, therefore
    do not mention the accessibility tree in your response. Please
   focus on granular and specific characteristics, and your
   generated characteristic points should apply to all the samples.
    Output should be a JSON list of strings.

Accessibility Tree for dataset A:
{A_tree}

Sampled web navigation tasks from dataset A:
{A}

Accessibility Tree for dataset B:
{B_tree}

Sampled web navigation tasks from dataset B:
{B}
```

Listing A.14: Rubric compilation prompt used in web navigation (differences)

```
Given some samples of proposed web navigation tasks and the web
   accessibility tree of dataset A and dataset B. Based on the
   similar characteristics between them. List how B is {feedback} A
   . Return {num} points that summarize the characteristics of the
   two datasets. The two accessibility trees of the same website
   are provided. Please only list out characteristics that are
   related to the proposed web navigation tasks, the accessibility
```

```
   trees are only provided to help you understand the context of
   the proposed tasks. Therefore do not list any characteristics
   that are related to the accessibility tree in your response.
   Please focus on granular and specific characteristics, and your
   generated characteristic points should apply to all samples in
   corresponding dataset (A or B). Output should be a JSON list of
   strings.

Similar characteristics between A and B:
{similar_points}

Accessibility Tree for dataset A:
{A_tree}

Sampled web navigation tasks from dataset A:
{A}

Accessibility Tree for dataset B:
{B_tree}

Sampled web navigation tasks from dataset B:
{B}
```

LENS Scoring Prompts

Listing A.15: Scorer prompt used in sentiment analysis

```
You are given similarities and differences between two financial
   news headline datasets A and B.

Your task is to judge how likely is the given financial news
   headline comes from dataset {prediction}. Answer your judgement
   with one of the following strings: "very unlikely", "unlikely",
   "unsure", "likely", and "very likely".

Similar characteristics between dataset A and B:
{similar_characteristics}

Differences between dataset A and B:
{differences}

Financial news headline sample to be judged:
{example}
```

```
Your judgement in JSON format:
```

Listing A.16: Scorer prompt used in Text2SQL

```
You are given similarities and differences between datasets A and B
    about database queries in natural language.

Your task is to judge how likely is the given database query in
   natural language comes from dataset {prediction}. Answer your
   judgement with one of the following strings: "very unlikely", "
   unlikely", "unsure", "likely", and "very likely".


Similar characteristics between dataset A and B:
{similar_characteristics}

Differences between dataset A and B:
{differences}

Natural language database query to be judged:
{example}

Your judgement in JSON format:
```

Listing A.17: Scorer prompt used in image classification

```
You are given similarities and differences between datasets A and B
   .

Your task is to judge how likely is the given image comes from
   dataset {prediction}. Answer your judgement with one of the
   following strings: "very unlikely", "unlikely", "unsure", "
   likely", and "very likely".

Format:
{format_instructions}

Similar characteristics between dataset A and B:
{similar_characteristics}

Differences between dataset A and B:
{differences}
```

```
Image to be judged:
{image}

your judgement in JSON format:
```

Listing A.18: Scorer prompt used in web navigation

```
You are given similar and different characteristics between two
   datasets A and B consisting of web navigation tasks.

Your objective is to judge how likely is the given web browsing
   task comes from dataset {prediction}. Answer your judgement with
    one of the following strings: "very unlikely", "unlikely", "
   unsure", "likely", and "very likely".

Format:
{format_instructions}

Similar characteristics between dataset A and B:
{similar_characteristics}

Different characteristics between dataset A and B:
{differences}

Web navigation task to be judged:
{example}

Your judgement in JSON format:
```

Appendix B

Additional Material for GTTA

B.1 Experiment details

WebArena We follow the BrowserGym guide[1] to setup WebArena enviornment. We use AWS EC2 instance with a `t3a.xlarge` instance. For all exploration episodes of dynamics grounding, we use a temperatue of 1.0. We use a temperature of 0.0 for all task evaluation experiments.

BFCLv3 For BFCLv3, we follow the official setup[2] and consistently using a temperature of 0.

Tau-bench We adopt the codebase form [Prabhakar et al., 2025] and use Best-of-N (N=5) strategy for LLM-simulated user agent, so each trajectory will be scored by an LLM judge and the reponse that best adheres with user intent will be select, therefore drastically mitigating variance in average reward.

B.2 Prompts used in WebArena experiments

[1] https://github.com/ServiceNow/BrowserGym/blob/main/browsergym/webarena/README.md
[2] https://github.com/ShishirPatil/gorilla/tree/main/berkeley-function-call-leaderboard

Listing B.1: prompt used to synthesize personas for WebArena

```
You are an expert at creating realistic user personas given a
   website. You are given the name of a website and some example
   outputs for the website, generate a diverse list of ${n_personas
   } personas, each with a unique "persona" name and a 1-2 sentence
    "description" of their typical behavior, interests, or
   motivations on that site.

Format your output as a JSON array, where each element is an object
    with two fields:
- "persona": a short, descriptive name for the persona (e.g., "
   Casual Browser", "Tech Enthusiast")
- "description": a brief description of how this persona typically
   uses the website

Now, generate a list of personas in the specified JSON format. You
   are given the name of the website and some example outputs for
   the website.
Website: ${website}
Example outputs: ${examples}
```

Listing B.2: prompt used for persona-driven exploration in WebArena

```
You are an autonomous intelligent agent tasked with navigating a
   web browser.  Your objective is to simulate a task that a person
    might perform, by interacting with the browser through the use
   of specific actions.

Here's the information you'll have:

The current web page's accessibility tree: This is a simplified
   representation of the webpage, providing key information.
The current web page's URL: This is the page you're currently
   navigating.
The open tabs: These are the tabs you have open.
The previous action: This is the action you just performed. It may
   be helpful to track your progress.
Trajectory: This is a sequence of natural language descriptions of
   the agent's interaction with the web-browser.
Person Description: The description of a specific kind of person
   whose task you are supposed to simulate.
Environment dynamics: Descriptions of how states transition.

The actions you can perform fall into several categories:
```

```
Page Operation Actions:
‘click [id]‘: This action clicks on an element with a specific id
   on the webpage.
‘type [id] [content] [press_enter_after=0|1]‘: Use this to type the
    content into the field with id. By default, the "Enter" key is
   pressed after typing unless press_enter_after is set to 0.
‘hover [id]‘: Hover over an element with id.
‘press [key_comb]‘:  Simulates the pressing of a key combination on
    the keyboard (e.g., Ctrl+v).
‘scroll [direction=down|up]‘: Scroll the page up or down.

Tab Management Actions:
‘new_tab‘: Open a new, empty browser tab.
‘tab_focus [tab_index]‘: Switch the browser’s focus to a specific
   tab using its index.
‘close_tab‘: Close the currently active tab.

URL Navigation Actions:
‘goto [url]‘: Navigate to a specific URL.
‘go_back‘: Navigate to the previously viewed page.
‘go_forward‘: Navigate to the next page (if a previous ’go_back’
   action was performed).

Completion Action:
‘stop ["done"]‘: Issue this action when you are done.

Homepage:
If you want to visit other websites, check out the homepage at http
   ://homepage.com. It has a list of websites you can visit.

To be successful, it is very important to follow the following
   rules:
1. You should only issue an action that is valid given the current
   observation
2. You should only issue one action at a time.
3. You should follow the examples to reason step by step and then
   issue the next action.
4. Generate the action in the correct format. Start with a "In
   summary, the next action I will perform is" phrase, followed by
   action inside ‘‘‘‘‘‘. For example, "In summary, the next action
   I will perform is ‘‘‘click [1234]‘‘‘".
5. To generate an interesting task, make sure you issue atleast 4
   actions before stopping. More interesting tasks typically
```

```
   involve more interactions with the browser.
6. You can issue atmost 40 actions before stopping, but feel free
   to output the stop action early if you want to stop exploring.
   Don't generate anything after stop.
Here are the list of environmental dynamics of this environment:

OBSERVATION:
[1744] link 'HP CB782A#ABA 640 Inkjet Fax Machine (Renewed)'
                [1749] StaticText '$279.49'
                [1757] button 'Add to Cart'
                [1760] button 'Add to Wish List'
                [1761] button 'Add to Compare'
URL: http://onestopmarket.com/office-products/office-electronics.
   html
ENVIRONMENTAL DYNAMICS:
[{'initial_state': 'Homepage of onestopmarket.com, with the button
   Add to Cart visible', 'action': 'click [1757] where the button
   is purchase'}]
PREVIOUS ACTION: None
Let's think step-by-step. This page lists the information of HP
   Inkjet Fax Machine, which is the product identified in the
   objective. Its price is $279.49. I have checked the
   environmental dynamics and see that stopping with this answer
   from this state has not been done before, so this action will
   contribute to exploration diversity. I think I have achieved the
    objective. In summary, the next action I will perform is ```
   stop [$279.49]```
OBSERVATION:
[164] textbox 'Search' focused: True required: False
[171] button 'Go'
[174] link 'Find directions between two points'
[212] heading 'Search Results'
[216] button 'Close'
URL: http://openstreetmap.org
ENVIRONMENTAL DYNAMICS:
[{'initial_state': 'OpenStreetMap homepage with search box and Go
   button visible', 'action': 'click [171] to submit a search'}, {'
   initial_state': 'OpenStreetMap homepage with search box focused
   ', 'action': 'type [164] [parks near NY] [1]'}]
PREVIOUS ACTION: None
Let's think step-by-step. This page has a search box whose ID is
   [164]. According to the nominatim rule of openstreetmap, I can
   search for restaurants near a location by typing "restaurants
   near" followed by the location. I can submit my typing by
```

```
   pressing Enter afterwards. Before proceeding, I will check the
   environmental dynamics to see if a similar action has already
   been performed from this state. I see that there is no previous
   action where, starting from the current state, someone searched
   for "restaurants near CMU" in the search box. This means my
   action will contribute to exploration diversity. In summary, the
    next action I will perform is ```type [164] [restaurants near
   CMU] [1]```
OBSERVATION:
${OBSERVATION}
Person Description:
${PERSON DESCRIPTION}
URL: ${URL}
PREVIOUS ACTION: ${PREVIOUS ACTION}
ENVIRONMENTAL DYNAMICS:
${ENVIRONMENTAL DYNAMICS}
```

Listing B.3: prompt used to extract environment dynamics from observations

```
You are given the output of an action taken by an autonomous
   intelligent agent navigating a web browser. Your objective is to
    produce a description of the changes made to the state of the
   browser. Specifically, you need to produce the following
   elements:
1) Initial state: provide a concise description of the browser's
   state before the action was taken (for example: "GitHub homepage
   " or "GitHub repository page").
2) Environmental dynamics: clearly describe the specific changes
   that occurred in the browser's state as a result of the action (
   for example: "The search bar expanded and is able to consume
   input").

Here's the information you'll have:
- **Initial state of the browser as an accessibility tree:**
  This is a simplified representation of the webpage, providing key
      information.
- **Final state of the browser:**
  This is the accessibility tree representation after the agent
     took the action.
- **The action taken by the web agent:**
  The agent can take actions that fall under the following
     categories (with descriptions):
**Page Operation Actions:**
- `click [id]`: Clicks on an element with a specific id on the
   webpage.
```

```
- `type [id] [content] [press_enter_after=0|1]`: Types the content
   into the field with id. By default, the "Enter" key is pressed
   after typing unless `press_enter_after` is set to 0.
- `hover [id]`: Hovers over an element with id.
- `press [key_comb]`: Simulates pressing a key combination on the
   keyboard (e.g., Ctrl+v).
- `scroll [direction=down|up]`: Scrolls the page up or down.

**Tab Management Actions:**
- `new_tab`: Opens a new, empty browser tab.
- `tab_focus [tab_index]`: Switches the browser's focus to a
   specific tab using its index.
- `close_tab`: Closes the currently active tab.

**URL Navigation Actions:**
- `goto [url]`: Navigates to a specific URL.
- `go_back`: Navigates to the previously viewed page.
- `go_forward`: Navigates to the next page (if a previous 'go_back'
    action was performed).

**Completion Action:**
- `stop [answer]`: Issue this action when you believe the task is
   complete. If the objective is to find a text-based answer,
   provide the answer in the bracket. If you believe the task is
   impossible to complete, provide the answer as "N/A" in the
   bracket.

**To be successful, it is very important to follow these rules:**
1. Explicitly think about the various features on the website and
   how the interaction with the website changed these features.
2. Provide the description of changes in one or two sentences.
3. If there is no change, your description of the changes in state
   should be "no change".
4. Strictly follow the JSON format in your response: {"
   initial_state": <description of initial state>, "
   environmental_dynamics": <description of environmental change>}

Here are some output examples for some random tasks:
1. {"initial_state": "Google homepage with search bar visible", "
   environmental_dynamics": "A modal search dialog opened up,
   bringing up a focused search input with suggestions, allowing
   the user to search or jump to various GitHub sections."}
2. {"initial_state": "Wikipedia article page", "
   environmental_dynamics": "The page scrolled down, revealing the
```

```
   'History' section."}
3. {"initial_state": "Amazon product page", "environmental_dynamics
   ": "A confirmation message appeared indicating the item was
   added to the cart."}
```

Listing B.4: prompt used to remove repeatitive or trivial environment dynamics in WebArena

```
You are given a list of environmental dynamics of an environment
   collected by an autonomous web browsing agent. You are tasked to
    remove repetitive and trivial environmental dynamics. Each of
   the point consists of the following:
1) initial_state: provide a concise description of the browser's
   state before the action was taken (for example: "GitHub homepage
   " or "GitHub repository page").
2) action: action taken to make the environmental change
2) environmental_dynamics: clearly describe the specific changes
   that occurred in the browser's state as a result of the action (
   for example: "The search bar expanded and is able to consume
   input").

To be successful, it is important to follow these rules:
1. entries with trivial enviornmental dynamics, such as "when I
   scroll down, the page will reveal more content" should be
   removed. Trivial environmental dynamics are those that do not
   reflect a meaningful or unique change in the environment, such
   as simple scrolling, pagination, or toggling the visibility of
   already available content without introducing new information or
    interface elements.

2. Remove entries where the environmental_dynamics only describe
   the expansion or collapse of text (e.g., "The full abstract
   expanded and is now visible"), unless the expansion reveals new
   interface elements or options that were not previously
   accessible.

3. Remove entries where the environmental_dynamics only describe
   navigation to a different section of the same page (e.g., "The
   page scrolled to the 'References' section"), unless this
   navigation results in a substantial change in the available
   interface or content.

4. Keep entries where the environmental_dynamics describe:
   - Unexpected or spurious state change of executing an action at
      the initial state.
   - Keep entries where the environmental_dynamics indicate that
```

```
      performing an action did not result in any change to the
      browser's state, even though a change would normally be
      expected (for example, clicking a button that should open a
      menu but nothing happens).
   - The appearance of new controls, filters, or options in the
      interface.
   - Navigation to a new page or a substantially different view (e.
      g., from search results to a detailed article page).
   - The addition or removal of search fields, filters, or other
      interactive elements.
   - Any change that enables new actions or reveals new types of
      information not previously accessible.

5. When in doubt, prefer to remove entries that are repetitive (e.g
   ., multiple entries describing the same type of abstract
   expansion) or that do not add new information about the
   environment's capabilities or state.

Your output should be a cleaned list of environmental dynamics,
   with only the non-trivial, non-repetitive, and meaningful
   changes retained. Please output in the original JSON format.
```

Listing B.5: prompt used for evaluation in WebArena

```
You are an autonomous intelligent agent tasked with navigating a
   web browser. You will be given web-based tasks. These tasks will
    be accomplished through the use of specific actions you can
   issue.

Here's the information you'll have:
The user's objective: This is the task you're trying to complete.
The current web page's accessibility tree: This is a simplified
   representation of the webpage, providing key information.
The current web page's URL: This is the page you're currently
   navigating.
The open tabs: These are the tabs you have open.
The previous actions: These are all the action you have performed.
   It may be helpful to track your progress.

The actions you can perform fall into several categories:

Page Operation Actions:
`click [id]`: This action clicks on an element with a specific id
   on the webpage.
`type [id] [content] [press_enter_after=0|1]`: Use this to type the
```

```
    content into the field with id. By default, the "Enter" key is
   pressed after typing unless press_enter_after is set to 0.
‘hover [id]‘: Hover over an element with id.
‘press [key_comb]‘:  Simulates the pressing of a key combination on
    the keyboard (e.g., Ctrl+v).
‘scroll [down|up]‘: Scroll the page up or down.

Tab Management Actions:
‘new_tab‘: Open a new, empty browser tab.
‘tab_focus [tab_index]‘: Switch the browser’s focus to a specific
   tab using its index.
‘close_tab‘: Close the currently active tab.

URL Navigation Actions:
‘goto [url]‘: Navigate to a specific URL.
‘go_back‘: Navigate to the previously viewed page.
‘go_forward‘: Navigate to the next page (if a previous ’go_back’
   action was performed).

Completion Action:
‘stop [answer]‘: Issue this action when you believe the task is
   complete. If the objective is to find a text-based answer,
   provide the answer in the bracket. If you believe the task is
   impossible to complete, provide the answer as "N/A" in the
   bracket.

Homepage:
If you want to visit other websites, check out the homepage at http
   ://homepage.com. It has a list of websites you can visit.

To be successful, it is very important to follow the following
   rules:
1. You should only issue an action that is valid given the current
   observation
2. You should only issue one action at a time.
3. You should follow the examples to reason step by step and then
   issue the next action.
4. You are strictly forbidden from issuing a goto action to a URL
   that is not on the homepage.
5. Generate the action in the correct format. Start by reasoning
   about the current situation. End with "In summary, the next
   action I will perform is" phrase, followed by action inside
   ‘‘‘‘‘‘. For example, "Let’s think step-by-step. Given the
   current state, I need to click on the like button which has id
```

```
   1234. In summary, the next action I will perform is ```click
   [1234]```".
6. Issue stop action when you think you have achieved the objective
   . Don't generate anything after stop.

Here are some example outputs for some random tasks:
1. Let's think step-by-step. This page list the information of HP
   Inkjet Fax Machine, which is the product identified in the
   objective. Its price is $279.49. I think I have achieved the
   objective. I will issue the stop action with the answer. In
   summary, the next action I will perform is ```stop [$279.49]```
2. Let's think step-by-step. This page has a search box whose ID is
    [164]. According to the nominatim rule of openstreetmap, I can
   search for the restaurants near a location by "restaurants near
   ". I can submit my typing by pressing the Enter afterwards. In
   summary, the next action I will perform is ```type [164] [
   restaurants near CMU] [1]```
```

Listing B.6: prompt used for evaluation on WebArena with environment dynamics

```
You are an autonomous intelligent agent tasked with navigating a
   web browser. You will be given web-based tasks. These tasks will
    be accomplished through the use of specific actions you can
   issue.

Here's the information you'll have:
The user's objective: This is the task you're trying to complete.
The current web page's accessibility tree: This is a simplified
   representation of the webpage, providing key information.
The current web page's URL: This is the page you're currently
   navigating.
The open tabs: These are the tabs you have open.
The previous actions: These are all the action you have performed.
   It may be helpful to track your progress.
The environmental dynamics: You will also be given the
   environmental dynamics of the environment. You can use the
   environmental dynamics to predict the next action. Each
   environmental dynamics entry describes a state change in the
   environment and contains the following fields:
- "initial_state": The state of the environment before the action
   is taken.
- "action": The action performed by the agent.
- "environmental_dynamics": A description of how the environment
   changes as a result of the action.
```

```
The actions you can perform fall into several categories:

Page Operation Actions:
‘click [id]‘: This action clicks on an element with a specific id
   on the webpage.
‘type [id] [content] [press_enter_after=0|1]‘: Use this to type the
    content into the field with id. By default, the "Enter" key is
   pressed after typing unless press_enter_after is set to 0.
‘hover [id]‘: Hover over an element with id.
‘press [key_comb]‘:  Simulates the pressing of a key combination on
    the keyboard (e.g., Ctrl+v).
‘scroll [down|up]‘: Scroll the page up or down.

Tab Management Actions:
‘new_tab‘: Open a new, empty browser tab.
‘tab_focus [tab_index]‘: Switch the browser’s focus to a specific
   tab using its index.
‘close_tab‘: Close the currently active tab.

URL Navigation Actions:
‘goto [url]‘: Navigate to a specific URL.
‘go_back‘: Navigate to the previously viewed page.
‘go_forward‘: Navigate to the next page (if a previous ’go_back’
   action was performed).

Completion Action:
‘stop [answer]‘: Issue this action when you believe the task is
   complete. If the objective is to find a text-based answer,
   provide the answer in the bracket. If you believe the task is
   impossible to complete, provide the answer as "N/A" in the
   bracket.

Homepage:
If you want to visit other websites, check out the homepage at http
   ://homepage.com. It has a list of websites you can visit.

To be successful, it is very important to follow the following
   rules:
1. You should only issue an action that is valid given the current
   observation
2. You should only issue one action at a time.
3. You should follow the examples to reason step by step and then
   issue the next action.
```

```
4. You are strictly forbidden from issuing a goto action to a URL
   that is not on the homepage.
5. Generate the action in the correct format. Start by reasoning
   about the current situation. End with "In summary, the next
   action I will perform is" phrase, followed by action inside
   ``````. For example, "Let's think step-by-step. Given the
   current state, I need to click on the like button which has id
   1234. In summary, the next action I will perform is ```click
   [1234]```".
6. When selecting actions, consider the environmental dynamics -
   what state changes will occur based on your knowledge of how the
    website behaves. Use this to avoid actions that would lead to
   undesired states or to strategically choose actions that lead to
    desired outcomes.
7. Issue stop action when you think you have achieved the objective
   . Don't generate anything after stop.

Here are some example outputs for some random tasks:
1. Let's think step-by-step. This page lists the information of HP
   Inkjet Fax Machine, which is the product identified in the
   objective. Its price is $279.49. By considering the
   environmental dynamics, I can confirm that no further actions
   are needed to change the state, and I have achieved the
   objective. I will issue the stop action with the answer. In
   summary, the next action I will perform is ```stop [$279.49]```
2. Let's think step-by-step. This page has a search box whose ID is
    [164]. According to the nominatim rule of openstreetmap, I can
   search for the restaurants near a location by "restaurants near
   ". By considering the environmental dynamics, I know that typing
    this query and pressing Enter will trigger the search and
   update the results accordingly. Therefore, I can submit my
   typing by pressing the Enter afterwards. In summary, the next
   action I will perform is ```type [164] [restaurants near CMU]
   [1]```
```
```

B.3 Prompts used in BFCLv3 experiments

Listing B.7: prompt used to synthesize exploration tasks on BFCLv3

```
You are an expert at creating exploration strategies that will help
    guide an LLM to explore and understand a new function calling
   environment. Your task is to synthesize some exploration goals
   based on a list of functions and a specific environment.

## Task
Given a list of functions and an environment, create ${N} distinct
   goals that will explore diffferent aspects of the environment.
   Each goal should represent different exploration strategies and
   approaches to discovering functionality.

## Input Format
- **Environment**: The computing environment or system (e.g., "file
    system", "vehicle control", "web browser")
- **Functions**: A list of available functions/commands alongside
   with detailed descriptions (e.g., ["cd()", "ls()", "pwd()", "
   mkdir()", "rm()"])

## Output Format
Output a JSON list of goals. Each goal should be a single sentence
   that captures the exploration goal and the strategy to achieve
   it.

Format: ["goal description", "goal description", ...]

## Example Output
**Environment**: File System
**Functions**:
{"name": "touch", "description": "This tool belongs to the Gorilla
   file system. It is a simple file system that allows users to
   perform basic file operations such as navigating directories,
   creating files and directories, reading and writing to files,
   etc. Tool description: Create a new file of any extension in the
    current directory.", "parameters": {"type": "dict", "properties
   ": {"file_name": {"type": "string", "description": "The name of
   the new file in the current directory. file_name is local to the
    current directory and does not allow path."}}, "required": ["
   file_name"]}, "response": {"type": "dict", "properties": {}}}
{"name": "wc", "description": "This tool belongs to the Gorilla
   file system. It is a simple file system that allows users to
```

```
   perform basic file operations such as navigating directories,
   creating files and directories, reading and writing to files,
   etc. Tool description: Count the number of lines, words, and
   characters in a file of any extension from current directory.",
   "parameters": {"type": "dict", "properties": {"file_name": {"
   type": "string", "description": "Name of the file of current
   directory to perform wc operation on."}, "mode": {"type": "
   string", "description": "Mode of operation ('l' for lines, 'w'
   for words, 'c' for characters). ", "default": "l"}}, "required":
    ["file_name"]}, "response": {"type": "dict", "properties": {"
   count": {"type": "integer", "description": "The count of the
   number of lines, words, or characters in the file."}, "type": {"
   type": "string", "description": "The type of unit we are
   counting. [Enum]: [\"lines\", \"words\", \"characters\"]"}}}}
{"name": "ls", "description": "This tool belongs to the Gorilla
   file system. It is a simple file system that allows users to
   perform basic file operations such as navigating directories,
   creating files and directories, reading and writing to files,
   etc. Tool description: List the contents of the current
   directory.", "parameters": {"type": "dict", "properties": {"a":
   {"type": "boolean", "description": "Show hidden files and
   directories. Defaults to False. ", "default": false}}, "required
   ": []}, "response": {"type": "dict", "properties": {"
   current_directory_content": {"type": "array", "description": "A
   list of the contents of the specified directory.", "items": {"
   type": "string"}}}}}
{"name": "rm", "description": "This tool belongs to the Gorilla
   file system. It is a simple file system that allows users to
   perform basic file operations such as navigating directories,
   creating files and directories, reading and writing to files,
   etc. Tool description: Remove a file or directory.", "parameters
   ": {"type": "dict", "properties": {"file_name": {"type": "string
   ", "description": "The name of the file or directory to remove.
   "}}, "required": ["file_name"]}, "response": {"type": "dict", "
   properties": {"result": {"type": "string", "description": "The
   result of the remove operation."}}}}


Generated goals:
[
  "Explore simple ways to create a file with touch(), count words
     of the file with wc(), and remove the file with rm().",
  "Understand the "file_name" argument of the touch() function to
     see if a file can be created using a path instead of just a
```

```
    file name."
]

## Guidelines
- **Be function-specific**: Each goal should explicitly mention
   which specific functions/tools will be used and in what sequence
- **Incorporate function descriptions into goal formulation**: You
   should examine the given function descriptions (i.e., input or
   output arguments) to propose goals to solve confusing or
   ambiguous function descriptions or environmental dynamics.
- **Ensure clarity and achievability**: Each goal should be a
   single, clear action that can be completed with the available
   functions

## Your Task
Now synthesize goal descriptions for the environment and functions
   provided in the format of a JSON list. Please do not leave any
   comment.
```

Listing B.8: user prompt used in BFCLv3 exploration

```
Do you think you have well-explored all functions that you are
   interested in, particularly those with ambiguous input formats/
   structures or potentially unexpected outputs?

Consider stopping (###STOP) if:
- You have thoroughly tested functions with unclear parameter
   requirements
- You have explored functions that might produce unexpected or
   variable output formats
- You have sufficient understanding of how to handle edge cases and
    errors

Continue exploring (###CONTINUE) if:
- There are still functions with ambiguous input specifications
   that need clarification
- You haven't fully tested functions that could have unpredictable
   outputs
- You need more exploration to understand parameter constraints and
    error handling

Please output ###STOP if you believe you have comprehensive
   coverage, or ###CONTINUE if more exploration is needed.
```

Listing B.9: prompt used to extract environment dynamics from interaction logs of BFCLv3

```
You are an expert data analyst specializing in function calling
   environments. Your task is to analyze interaction logs between a
    function calling bot and an environment to extract
   environmental dynamics based on a given exploration goal.

## Task
Given an exploration goal and an interaction log, extract
   environmental dynamics that describe how the environment changes
    in response to bot actions. Each environmental dynamic should
   capture the cause-and-effect relationship between actions and
   environmental changes.

## Environmental Dynamic Structure
Each environmental dynamic must include these three components:

1. **initial_state**: The relevant state of the environment before
   the action
   - Include context like current directory, variable values, file
      states, or any pertinent environmental conditions
   - Be specific about what matters for understanding the action's
      impact

2. **action_taken**: The exact function call or action performed by
    the bot
   - Include the function name and all parameters with their values
   - Use the exact syntax from the interaction log

3. **environmental_dynamics**: How the environment changes as a
   result of the action
   - Describe what was created, modified, deleted, or revealed
   - Include any state transitions, data changes, or new
      information made available
   - Be specific about the observable effects

## Guidelines
- **Output format**: Generate your response as a JSON array of
   objects
- **Completeness**: Extract dynamics for every significant action
   in the log
- **Accuracy**: Base descriptions strictly on what's observable in
   the interaction log
- **Clarity**: Use precise, unambiguous language for each component
- **Focus**: Only include dynamics that show meaningful
```

```
   environmental changes

## Example Input
Goal: Investigate how to create a new file with touch() and confirm
    its existence using ls().

Interaction Logs:
Step 1:
Assistant: [ls()]
Environment: {"current_directory_content": []}
Step 2:
Assistant: [mkdir(dir_name="testdir")]
Environment: None
Step 3:
Assistant: [cd(folder="testdir")]
Environment: {"current_working_directory": "testdir"}
Step 4:
Assistant: [pwd()]
Environment: {"current_working_directory": "///testdir"}

## Example Output
[
  {
    "initial_state": "The current directory contains no files.",
    "action_taken": "touch(file_name=\"testfile.txt\")",
    "environmental_dynamics": "A new file named \"testfile.txt\" is
        created in the current directory."
  },
  {
    "initial_state": "The current directory contains the file \"
       testfile.txt\".",
    "action_taken": "ls()",
    "environmental_dynamics": "The environment reveals the contents
        of the current directory, showing that \"testfile.txt\"
       exists."
  }
]
```

Listing B.10: prompt used to filter environment dynamics of BFCLv3

```
You are an expert data analyst specializing in function calling
   environments. Your task is to filter environmental dynamics to
   remove insignificant or environment-specific details while
   preserving meaningful behavioral patterns.
```

```
## Task
Given a JSON array of environmental dynamics, identify and remove
   dynamics that are:
1. **Very environment-specific**: Contain specific data, names, or
   values that are unique to a particular environment instance/
   setup
2. **Insignificant**: Describe trivial or non-meaningful changes
   that don't contribute to understanding function behavior (e.g.,
   ls() in a file system)
3. **Redundant**: Duplicate similar patterns already captured in
   other dynamics

## Filtering Criteria

### Remove dynamics that contain:
- **Specific data values**: Names, IDs, file contents, user lists,
   or any concrete data that varies between environments
- **Environment-specific paths**: Absolute paths, specific
   directory names, or location-dependent information
- **Instance-specific identifiers**: User names, session IDs,
   timestamps, or other unique identifiers
- **Trivial state changes**: Minor formatting changes, temporary
   states, or cosmetic updates
- **Redundant information**: Multiple dynamics describing the same
   behavioral pattern with different specific values

### Keep dynamics that describe:
- **General behavioral patterns**: How functions typically behave
   across different environments
- **Error conditions**: Standard error messages and failure modes
- **State transitions**: General state changes (logged in/out, file
    created/deleted, etc.)
- **Function capabilities**: What functions can do, not what
   specific data they return
- **Function contains meaningful arguments**: Keep dynamics where
   the function call includes arguments that affect behavior or
   state, as these illustrate how input parameters influence
   outcomes.

## Guidelines
- **Output format**: Return a JSON array containing only the
   filtered dynamics
- **Preserve structure**: Keep the same JSON structure for
   remaining dynamics
```

````
- **Generalize descriptions**: Replace specific values with general
    descriptions where possible
- **Maintain accuracy**: Don't change the core meaning of the
   dynamics, only remove specificity
- **Focus on patterns**: Prioritize dynamics that show reusable
   behavioral patterns

## Examples

### Remove (Very environment-specific that it contains data entries
    specific to a environment instance):
```json
{
    "initial_state": "No user is logged into the system, and a list
        of users needs to be identified.",
    "action_taken": "list_users()",
    "environmental_dynamics": "Provides the list of available users
       : Alice, Bob, Catherine, and Daniel."
}
```

### Keep (General pattern and error message):
```json
{
    "initial_state": "No user is currently logged in.",
    "action_taken": "search_messages(keyword=\"test\")",
    "environmental_dynamics": "An error message is returned
       indicating that no user is currently logged in."
}
```

### Remove (Trivial transition):
```json
{
    "initial_state": "In a directory with existing files.",
    "action_taken": "ls()",
    "environmental_dynamics": "Returns a list of files and
       directories in the current location."
}
```

## Instructions
1. Analyze each environmental dynamic in the input JSON array
2. Apply the filtering criteria to determine if it should be kept
````

```
   or removed
3. For dynamics that are kept, generalize any remaining specific
   details
4. Return the filtered JSON array with only the significant,
   generalizable dynamics
```

B.4 Additional results

Table B.1: Success rate (SR) of syntactic alignment across different Qwen2.5 instruct model sizes (7B, 14B, 32B) and training hyperparameters. Results are shown for varying numbers of training iterations and two learning rates (LR=0.1 and LR=1.0). Syntactic alignment adaptation generally improves SR over the baseline, with larger models and moderate learning rates yielding higher gains.

Model	**Number of Training Iterations**	**SR (LR=0.1)**	**SR (LR=1.0)**
Qwen2.5-7B-Instruct	Baseline (0)	9.00	-
Qwen2.5-7B-Instruct	1	9.50	9.50
Qwen2.5-7B-Instruct	2	9.50	8.50
Qwen2.5-7B-Instruct	3	9.50	9.00
Qwen2.5-7B-Instruct	4	10.00	8.50
Qwen2.5-7B-Instruct	5	10.00	8.50
Qwen2.5-14B-Instruct	Baseline (0)	18.5	-
Qwen2.5-14B-Instruct	1	20.00	20.00
Qwen2.5-14B-Instruct	2	20.00	20.00
Qwen2.5-14B-Instruct	3	20.00	21.00
Qwen2.5-14B-Instruct	4	19.50	20.00
Qwen2.5-14B-Instruct	5	19.00	19.00
Qwen2.5-32B-Instruct	Baseline (0)	26.00	-
Qwen2.5-32B-Instruct	1	25.50	26.50
Qwen2.5-32B-Instruct	2	26.50	26.50
Qwen2.5-32B-Instruct	3	26.50	27.00
Qwen2.5-32B-Instruct	4	26.50	27.00
Qwen2.5-32B-Instruct	5	26.00	26.50

Table B.2: Number of environment dynamics per website before and after filtering with `GPT-4.1` as the exploration policy and dynamics extractor. Most environment dynamics are filtered out.

Website	**Number of Env Dynamics**	**Number of Env Dynamics (after filtering)**
Shopping	334	52
Shopping Admin	334	50
GitLab	334	53
Map	334	18
Reddit	334	23

Table B.3: Number of functions available in each BFCLv3 environment.

Environment	**Num of Functions**
VehicleControlAPI	22
GorillaFileSystem	18
TravelAPI	17
MathAPI	17
TwitterAPI	14
MessageAPI	10
TicketAPI	9
TradingBot	22

B.5 Environment dynamics analysis

Here we provide some concrete examples of environment dynamics that are removed by our filtering mechanism. Exploration policy and dynamics extractor is `GPT-4.1`.

- **No Change Dynamics**

```
{
  "initial_state": "Magento Admin product edit page, with the '
     Country of Manufacture' dropdown expanded and no country
     selected.",
  "action": "click [1405] where [1405] is United States",
  "environmental_dynamics": "no change"
}
```

- **Basic Dropdown/Combobox Expansion**

```
{
  "initial_state": "Magento Admin product edit page for 'Sprite
      Stasis Ball 65 cm', with the 'Country of Manufacture'
     combobox (notice-WPNSU51) collapsed and not focused.",
  "action": "click [1168] where [1168] is notice-WPNSU51",
  "environmental_dynamics": "The 'Country of Manufacture'
     combobox became focused and expanded, showing its list of
     country options."
}
```

- **Basic Dialog Open/Close**

```
{
  "initial_state": "Magento Admin product edit page for 'Sprite
      Stasis Ball 65 cm', with the 'Add Attribute' button
     visible but no attribute selection dialog open.",
  "action": "click [720] where [720] is Add Attribute",
  "environmental_dynamics": "Clicking the 'Add Attribute'
     button opened the 'Add Attribute' dialog, displaying
     available product attributes for selection and actions
     such as 'Create New Attribute'."
}
```

- **Standard Form Field Visibility**

```
{
  "initial_state": "Magento Admin product detail page for '
     Sprite Stasis Ball 65 cm' with all product attribute forms
      visible, including a collapsed 'Country of Manufacture'
     dropdown (combobox).",
  "action": "click [1267] where [1267] is notice-BKW2147",
  "environmental_dynamics": "The 'Country of Manufacture'
     combobox was clicked, causing attribute fields for '
     Country of Manufacture' and all subsequent grouped
     attributes to become visible."
}
```

- **Basic Button Focus**

```
{
  "initial_state": "Magento Admin product edit page with the '
     Add Attribute' dialog open; in the attributes table, the '
     Options' column header includes a checkbox and an 'Options
     ' button, neither focused nor expanded.",
  "action": "click [2372] where [2372] is ",
  "environmental_dynamics": "The 'Options' button in the '
     Options' column header of the Add Attribute dialog became
     focused and expanded a dropdown list with a single option,
      'Select All', now visible below the button."
}
```