Using Domain Adaptation to Improve Water Quality Modeling with Scarce Data

by

Chi-Chung Cheung

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, 2024

© Chi-Chung Cheung 2024

Author's Declaration

I hereby declare that I am the sole author of this 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.

Abstract

Water Quality (WQ) modelling is important not just to the conservation of ecosystems, but also to the welfare of modern human society. However, collecting enough high-quality data to use for training WQ prediction models is difficult. Unlike hydrology, current WQ collecting methods are constrained by cost, spatial coverage, and temporal sparsity.

This thesis explores using Domain Adaptation (DA) to overcome the data scarcity problem. By treating the different WQ measuring locations as different domains, high-resolution data from other locations can be used to better model a target location that has sparse data. The chosen DA method is inspired by domain-invariant (DI) representation learning. The model itself consists of (1) an f submodel representing the DI portion, and (2) one g submodel per domain representing the domain-variant portion.

Within the context of this thesis, the main findings are as follows:

1. DA can be successfully applied in the context of WQ modeling
2. The optimal model sizes are different between the full DA method and just the pretraining.
3. Using a station's basin was not a good measure of similarity.
4. At a high number of domains, further increasing the number of domains did not increase model performance.
5. Simply adding the outputs of f and g (i.e. $f(x) + g(x)$) did not perform as well as passing the output of f through g (i.e. $g(f(x))$).

These findings support the effectiveness of using DA in WQ modelling as well as present various considerations that affect the final performance. Furthermore, these findings are relevant to not only this particular DA method but also to DA in general.

Acknowledgements

I would like to first thank Professor Anita Layton for giving me this precious opportunity to complete this thesis under her valuable guidance.

Next, I would like to thank Professor Nandita Basu and Tyler Hampton for providing their expertise on water quality modelling.

Finally, I would like to thank my family whose unending support during these years has greatly motivated me to undertake this program and persevere until the end. I am especially grateful to my brother who helped me edit this thesis until 3 am.

eaaaww://fvhsah.il/MuX2Ilp+SjS

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

List of Figures viii

List of Tables x

List of Abbreviations xi

1 Introduction 1

1.1 Water Quality Modelling 1

1.2 Data Scarcity 2

1.3 Domain Adaptation 2

1.4 Gap and Thesis Objective 3

2 Literature Review 5

2.1 Water Quality 5

2.2 Water Quality Modeling 6

2.3 Domain Adaptation 9

2.4 Domain Adaptation for Water Quality Modeling 11

3 Methodology **12**

3.1 Data . . . 13

3.2 Model Training . . . 17

3.3 Experimental Design . . . 19

3.3.1 Model Size . . . 19

3.3.2 Choice of Source Domains . . . 20

3.3.3 Method to Utilize f and g . . . 23

4 Results and Analysis **25**

4.1 Validity of the DA Method . . . 25

4.2 Effect of Decreasing Temporal Sparsity . . . 27

4.3 Model Sizes . . . 29

4.3.1 Source Domain Pretraining . . . 29

4.3.2 Target Domain Fine-tuning . . . 30

4.4 Choice of Source Domains . . . 33

4.4.1 Source Domain Pretraining . . . 33

4.4.2 Target Domain Fine-tuning . . . 35

4.5 Method to Utilize f and g . . . 39

4.6 Evaluation Under Different Metrics . . . 42

5 Discussion and Future Work **46**

5.1 Experiments . . . 46

5.1.1 Quality of Experiments . . . 46

5.1.2 Main Findings . . . 47

5.2 Broader Context . . . 48

6 Conclusion **50**

References **51**

APPENDICES **59**

A Extra Information on Water Quality Data **60**

A.1 Pearson Correlation Coefficient between Flow and the Other Water Quality Paramters 60

B Extra Plots **62**

B.1 Validity of the DA Method 62

B.2 Model Size 64

B.3 Choice of Source Domains 66

B.3.1 Investigating Sets With Multiple Domains 66

B.3.2 Investigating Sets With a Single Domain 71

B.4 Method to Utilize f and g 76

List of Figures

2.1 LSTM Diagram 7

3.1 General DA Method 13
3.2 Sparsity Handling 16
3.3 Obtaining Samples from Time Series 17
3.4 General Model Structure 18
3.5 Maps 22
3.6 Default Model 23
3.7 Alternative Model 24

4.1 Baseline f32,2 g8,2 26
4.2 Example Water Quality Prediction 27
4.3 Decreasing Temporal Sparsity Phenomenon 28
4.4 Pretraining Validation Loss by Number of Parameters 29
4.5 Pretraining Validation Loss for Size 30
4.6 Fine-Tuning Validation Loss for Size 32
4.7 Fine-Tuning Validation Loss Difference for Size 33
4.8 Pretraining Validation Loss for Choice of Source Domains 35
4.9 Fine-Tuning Validation Loss for Sets with Multiple Domains 36
4.10 Fine-Tuning Validation Loss Difference for Sets with Multiple Domains . . 37
4.11 Fine-Tuning Validation Loss for Sets with One Domain 38

4.12 Fine-Tuning Validation Loss Difference for Sets with One Domain 39
4.13 Fine-Tuning Validation Loss for Different Utilization of f and g 40
4.14 Fine-Tuning Validation Loss Difference for Different Utilization of f and g 41
4.15 Adding Outputs of f and g Results in Negative Validation Loss Difference 42
4.16 NSE for Different Model Sizes 43
4.17 KGE for Different Model Sizes 44

5.1 Future Method of Utilizing f and g 49

B.1 Appendix Baseline 63
B.2 Appendix Validation Loss for Model Size 64
B.3 Appendix Validation Loss Difference for Model Size 65
B.4 Appendix Validation Loss for Sets with Multiple Domains 67
B.5 Appendix Validation Loss Difference for Sets with Multiple Domains . . . 69
B.6 Appendix Validation Loss for Sets with One Domain 72
B.7 Appendix Validation Loss Difference for Sets with One Domain 74
B.8 Appendix Validation Loss for Different Utilization of f and g 77
B.9 Appendix Validation Loss Difference for Different Utilization of f and g . . 79

List of Tables

3.1 Station Data Information 14

3.2 Experimented Sets of Source Domains. 21

4.1 Optimal Model Size Parameters 31

4.2 Interpretation of Typical Metric Values 42

4.3 Metric Statistics of Runs 45

A.1 Pearson Correlation Coefficient (PCC) between Flow and the Other WQ Paramters 61

List of Abbreviations

ADAM Adaptive Moment Estimation 18

CDOM Chromophoric Dissolved Organic Matter 5

CL Chloride 13

CNN Convolutional Neural Network 48

COND Conductivity 13, 44, 46

CORAL Correlation Alignment 11

DA Domain Adaptation iii, 2–4, 9, 11–13, 19, 25–27, 32, 36, 39, 40, 47, 48, 50, 63

DI domain-invariant iii, 3, 9–12

DL Deep Learning 2, 6, 48

DO Dissolved Oxygen 2, 5

DRCN Deep Reconstruction Convolutional Network 11

DSD Domain-invariant Sequence Distance 11

DV domain-variant 3

HTLP Heidelberg Tributary Loading Program 13

KGE Kling–Gupta Efficiency 7, 8, 42, 44–46

LOADEST Load Estimator 6

LSTM Long Short-Term Memory 6, 7, 17, 18, 48

MDA Multi-source Domain Adaptation 20

MMD Maximum Mean Discrepancy 11

MSE Mean Squared Error 7, 8, 11, 18

NCWQR National Center for Water Quality Research 13

NO23 Nitrite + Nitrate 13

NSE Nash–Sutcliffe Efficiency 7, 8, 42–46

PCC Pearson Correlation Coefficient x, 61

PWQMN Provincial (Stream) Water Quality Monitoring Network 6

S4 Structured State Space Sequence Model 48

SI Total Silica 13

SO4 Sulfate 13, 44

SRP Soluble Reactive Phosphorus 13, 44

STD standard deviation 14

TKN Total Kjeldahl Nitrogen 13, 44

TP Total Phosphorus 13, 44

TSS Total Suspended Solids 2, 5, 13, 25, 27, 44

WQ Water Quality iii, x, 1–8, 11–13, 17, 18, 23, 25, 27, 28, 45, 46, 48–50, 61

WRTDS Weighted Regressions on Time, Discharge, and Season 6

Chapter 1

Introduction

1.1 Water Quality Modelling

WQ is defined as the physical, chemical, thermal, and biological properties of water [53]. For example, nitrogen concentration is considered a WQ parameter but the flow rate of water is not. Studying WQ is an essential task in the conservation of bodies of water. Not only is this important to the conservation of wildlife [18], but it also has a direct impact on human society, through the public water supply [34], agriculture [32], fishery [63], and manufacturing industry [58].

One notable area where studying WQ is important is eutrophication, the excessive plant and algal growth due to the availability of various growth factors, notably nitrogen and phosphorous [9]. Eutrophication causes great harm to the aquatic ecosystem by decreasing the oxygen supply in the water. Annually, nitrogen and phosphorus pollution in freshwater costs at least $2.4 billion in the U.S. alone [70]. Modelling WQ well allows people to better understand ecologically harmful events such as eutrophication. With this knowledge, scientists, politicians, and other relevant stakeholders can use these models to better predict and prevent these events through well-informed decisions and policies.

However, modelling WQ is difficult partly because the processes behind its regulation are complex. Using process-based models is especially difficult because the processes must be sufficiently comprehensive and well-understood to be accurately modelled [75]. As a result, processed-based models are currently computationally expensive, especially on large spatio-temporal scales, and require parameter re-calibration even when the spatio-temporal setting is varied slightly.

An alternative to process-based models is Deep Learning (DL) models. Without relying on feature engineering, these models can extract hidden relationships within data. Currently, given equal computation power, these models achieve similar or even better performance than purely process-based ones [75]. However, DL models require significantly more data and do not address the issue of the differences among WQ measuring sites.

1.2 Data Scarcity

The high-quality data required to train DL models is not always available, which is a significant issue in WQ modelling using DL [3]. Compared to hydrology, WQ measurements are often lower resolution and the number of measurement sites is less since WQ-specialized equipment is required [75]. In addition, not all stations measure WQ to a consistent degree of comprehensiveness. To illustrate, 83% of global data for Total Suspended Solids (TSS) (one of the most common WQ variables) comes from only 17% of all the sampled rivers in the world, predominantly in North America. Newer sites will have made even fewer measurements. Finally, some WQ variables are measured much more frequently than others. For example, globally, there are more than 50% more data points for TSS compared to the second most measured WQ variable Dissolved Oxygen (DO).

One approach to surmount the data scarcity challenge is to use DA to borrow data from similar domains, in this case, other WQ measuring stations. The challenge of how to incorporate the additional data effectively is explored in the next section, section 1.3.

1.3 Domain Adaptation

DA is the technique of training a model with source domain data to perform the same task in a target domain. It is a subcategory of transfer learning. A model trained in the source domain may have trouble when predicting in the target domain because of dataset shift.

Dataset shift refers to changes in the input-output joint distributions between different datasets. Dataset shift can be divided into 3 components:

1. Covariate shift, where the distribution of X is different
2. Prior probability shift, where the distribution of Y is different
3. Concept drift, where the relation of X and Y is different

Often, dataset shift refers to the difference between training and test datasets. However, when the domain is explicitly changed, the effects of the drift become even more significant. Consider the case of WQ. Not only will different sites have different climates, terrains, biodiversities, etc., but the stations themselves could also capture different WQ parameters, at differing frequencies, using various devices.

In modelling WQ, each WQ-measuring station is considered a single domain. The source domains must be similar enough to the target domain or else learning domain-invariant features can increase error [5]. However, the source domains should also be varied enough so that as a whole, they capture most of the target domain's characteristics. With multiple source domains, the DA method also needs to balance the weighting of each domain and extract the transferable correlations.

One framework for understanding DA is to divide the data distribution into two parts: the domain-invariant part, which is the shared underlying behaviour of all domains of interest, and the domain-variant part. In this thesis, the proposed DA model is composed of (1) an invariant submodel f and (2) separate variant submodels g for each of the source and target domains. To make a prediction for a domain, f and that domain's g will be used.

The model is first pretrained with the source domain data. Then, the pretrained invariant submodel and the target domain's variant submodel are used when fine-tuning the model in the target domain.

1.4 Gap and Thesis Objective

Although there are many studies on different DA techniques, there is a gap in empirical studies focusing on specific configurations of a DA method. One such area is the sizes between the different submodels within a model(e.g. feature extractor and classifier). The optimal model sizes and what sizes lead to overfitting have not been deeply investigated. Another area is the way different submodels are utilized. This area is concerned with the relationship between the invariant and variant portions.

Furthermore, although there are some theoretical frameworks and empirical studies on the choice of source domains, there is a limited number of studies concerning the choice of source domains for WQ modelling. For example, the characteristics that make a set of source stations desirable are unknown. After all, applying DA to WQ modelling with limited data is a relatively new field.

In summary, this thesis investigates the use of an invariant-variant-based DA method to improve the prediction of temporally sparse WQ data. Specifically, this thesis will address the three gaps: the sizes of the submodels, the choice of source domains, and the way to utilize the different submodels. To fully capture the effects of these three aspects, they are investigated under varying temporal sparsities of data.

Chapter 2

Literature Review

2.1 Water Quality

Different WQ parameters may be measured depending on the specific use case. When studying eutrophication, nitrogen and phosphorous are measured since they are factors that cause algae growth, and DO is measured to assess the degree of eutrophication [27]. When studying pollution, WQ parameters measured include various types of heavy metals, petroleum compounds, organic compounds, biological toxins, fluoride, and chloride [50].

As opposed to hydrology, where technology has enabled high-frequency wide-area measurements to be made with satellites and drones [38], only the fraction of WQ parameters that affect water appearance can be measured through remote sensing [2]. These parameters include TSS, CDOM, and Chlorophyll-a. However, other parameters which are not easily detectable are measured with sensors or through lab analysis of water samples. Traditionally, WQ is measured by collecting water samples according to a schedule, and then sending them to a lab to be analyzed. However, this is expensive, labour-intensive [2], and leads to a low temporal resolution [39]. Low-cost WQ sensors (normally costing under US$200) only measure basic WQ parameters and suffer from durability and accuracy issues [12]. Furthermore, many of these low-cost sensors lack information on their accuracy, durability, and reliability for specific applications [12]. However, organizations cannot afford more expensive ones for every body of water. This forces organizations to deal with reduced spatial resolution for their WQ measurements.

Because of the lack of high-quality and affordable solutions to measure WQ, many WQ datasets lack temporal resolution for many sites. For example, in a typical WQ dataset,

such as the Provincial (Stream) Water Quality Monitoring Network (PWQMN) [45], typical values for WQ data resolution range from once per year to once per month, depending on the WQ feature and at what period of time the measurements were taken.

2.2 Water Quality Modeling

Statistical and process-based models have often been used for WQ modelling. Two well-known ones are the LOADEST [55] and the WRTDS [24]. These are regression-based programs that estimate various constituent loads using the time series of streamflow, initial constituent concentration, and additional data variables.

However, because it is challenging to include all the relevant factors, DL has been often used instead to capture the complex temporal trends. To handle the time-series nature of the data, Long Short-Term Memory (LSTM)s [25] are ubiquitously used [56, 76, 28, 49, 68, 15].

The LSTM equations are defined as follows:

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}),\\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}),\\
g_t &= tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}),\\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}),\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t,\\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{2.1}
$$

where

- x_t is the input,
- h_t is the hidden state,
- c_t is the cell state,
- f_t is the forget gate,
- i_t is the input gate,
- o_t is the output gate,

- σ is the sigmoid activation function, and
- $\odot$ is the hadamard product

Figure 2.1 depicts a visualization of the LSTM.

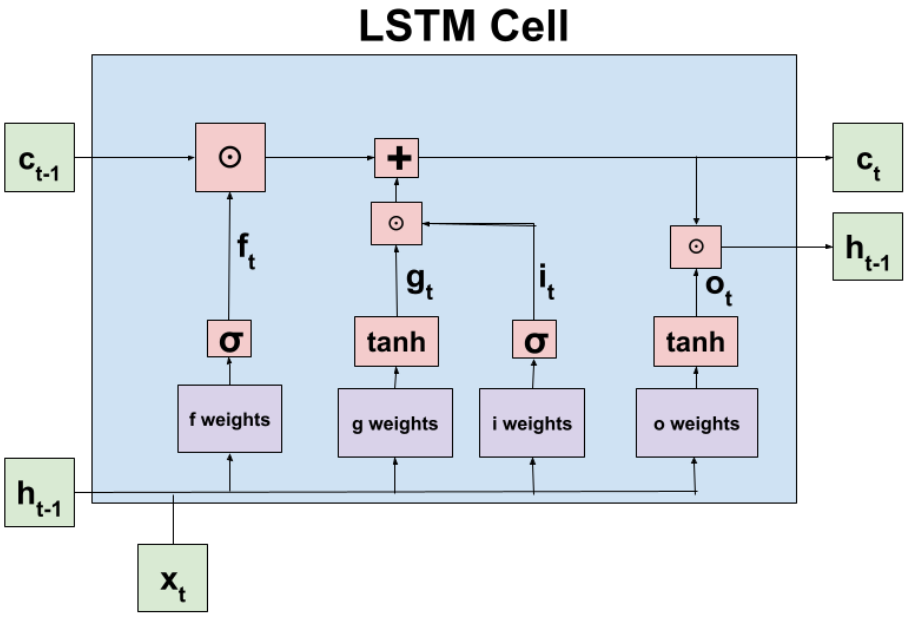


Figure 2.1: Depiction of an LSTM cell. At time t, the input x_t is fed into the LSTM. The previous hidden state h_{t-1} and previous cell state c_{t-1} are updated to h_t and c_t respectively. h_t is used as the LSTM output.

There are various metrics to measure the performance of a WQ model. Apart from measuring the absolute error with Mean Squared Error (MSE), the two main metrics for WQ models are Nash–Sutcliffe Efficiency (NSE) [41] and Kling–Gupta Efficiency (KGE) [21].

The equation for NSE is

$$NSE = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\bar{y}_i - y_i)^2} \tag{2.2}$$

where

- y_t is the output and time t,
- n is the number of timesteps,
- $\hat{}$ means the variable is approximated

In essence, NSE is the mean normalized version of MSE. However, if the MSE is less than σ, then the NSE would become negative. At one extreme, if the actual observations are constant, then the NSE would become $-\infty$ [23]. Furthermore, it is possible for the model whose predictions are only off by a constant to perform worse than the mean of the observed values.

An alternative to address these issues is KGE, whose equation is

$$KGE = 1 - \sqrt{(PCC - 1)^2 + (\alpha - 1)^2 + (\beta - 1)^2} \tag{2.3a}$$

$$\alpha = \frac{\mu_{\hat{y}}}{\mu_y} \tag{2.3b}$$

$$\beta = \frac{\sigma_{\hat{y}}}{\sigma_y} \tag{2.3c}$$

$$PCC = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.3d}$$

where

- x is the input,
- μ is the mean,
- σ is the standard deviation,
- $\bar{}$ means the mean of the variable

KGE separates terms for the correlation, variability bias and mean bias.

Multiple WQ metrics are used because having good performance when using one metric does not guarantee good performance with the other metrics [37].

2.3 Domain Adaptation

There are several approaches to the DA problem. One approach is to use DI representation learning. The source and target data are mapped into a common DI feature space and then a submodel that operates in this space performs the desired task. By reducing the difference in the distribution of the domains in this new latent feature space while optimizing for the main task, the model finds what remains invariant between the domains.

Phung et al. provide a theoretical framework for learning DI representations with multiple sources [47]. They suggest two types of representations: the general DI representation [5] and the compressed DI representation [33, 35]. The following notations are only applicable within this section 2.3.

In their setting, they consider a multi-class classification problem where the goal is to map $x \in \mathcal{X}$ to the correct class $y \in \mathcal{Y}$ where $\mathcal{Y} = [C]$ and C is the number of classes. The mapping function of interest is $f : \mathcal{X} \mapsto \mathcal{Y}_\Delta$ where $\mathcal{Y}_\Delta := \{\alpha \in \mathbb{R}^C : \|\alpha\|_1 = 1 \wedge \alpha \geq= 0\}$. Specifically, $f(x)$ is a vector representing the probability of x corresponding to each class. Next, let's define a domain $\mathcal{D} := (\mathcal{P}, f)$, where $x \sim \mathcal{P}$. In other words, each domain is the pair of (1) the distribution of x and (2) the true mapping function that maps x to the corresponding y. There are K source domains $\mathcal{D}^{S,i}, 1 \leq i \leq K$ and one target domain $\mathcal{D}^T$. Let's define a mixture of source domains $\mathcal{D}^\pi = \sum_{i=1}^K \pi_i \mathcal{D}^{S,i}$, where the mixing coefficient is $\pi_i = \frac{N_i}{\sum_{j=1}^K N_j}$. Essentially, this treats all the source domains as one big source domain. Finally, let's define $\mathcal{L}(\hat{f}, f, \mathcal{P}) = \mathcal{L}(\hat{f}, \mathcal{D}) := \mathbb{E}_{x\sim\mathcal{P}}[\ell(\hat{f}(x), f(x))]$ where ℓ is a loss between function between elements of $\mathcal{Y}_\Delta$.

The DA method of learning DI representations would then be to divide f into $g : \mathcal{X} \mapsto \mathcal{Z}$ and $h : \mathcal{Z} \mapsto \mathcal{Y}$ where $\mathcal{Z}$ is the DI space. Let $\mathcal{D}_g = (\mathcal{P}_g, h)$, where $\mathcal{P}_g$ is the distribution in $\mathcal{Z}$ space after it has been mapped by g.

Now, for both general and compressed DI feature maps $g*$, the quantity that is optimized is the following:

$$\min_g \min_{\hat{h}} \sum_{i=1}^K \pi_i \mathcal{L}(\hat{h}, \mathcal{D}_g^{S,i}) \tag{2.4}$$

However, $\mathcal{P}_{g*}^T = \mathcal{P}_{g*}^\pi$ for general DI features and $\mathcal{P}_{g*}^{S,1} = \mathcal{P}_{g*}^{S,2} = ...\mathcal{P}_{g*}^{S,K}$ for DI features. Essentially, general DI features match the target and the mixture of source domain distributions while compressed DI features match the distributions of each source domain.

Finally, if general and compressed DI representations are used, the upper bounds of the loss are

$$\mathcal{L}\left(\hat{f}, \mathcal{D}^T\right) \leq \sum_{i=1}^{K} \pi_i \mathcal{L}(\hat{f}, \mathcal{D}^{S,i}) + L \max_{i\in[K]} \mathbb{E}_{\mathcal{P}^{S,i}}\left[\|\Delta p^i(y|x)\|_1\right] + L\sqrt{2} d_{1/2}(\mathcal{P}_g^T, \mathcal{P}_g^\pi) \quad (2.5a)$$

$$\mathcal{L}\left(\hat{f}, \mathcal{D}^T\right) \leq \sum_{i=1}^{K} \pi_i \mathcal{L}(\hat{f}, \mathcal{D}^{S,i}) + L \max_{i\in[K]} \mathbb{E}_{\mathcal{P}^{S,i}}\left[\|\Delta p^i(y|x)\|_1\right] + \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{L\sqrt{2\pi_j}}{K} d_{1/2}(\mathcal{P}_g^{S,j}, \mathcal{P}_g^{S,i}) + \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{L\sqrt{2\pi_j}}{K} d_{1/2}(\mathcal{P}_g^T, \mathcal{P}_g^{S,i}) \quad (2.5b)$$

respectively, where $D_{1/2}(\mathcal{P}^1, \mathcal{P}^2) = 2\int \left(\sqrt{p^1(x)} - \sqrt{p^2(x)}\right)^2 dx$ and $d_{1/2} = \sqrt{D_{1/2}}$, $\Delta p^i(y|x) := \left[|f^T(x,y) - f^{S,i}(x,y)|\right]_{y=1}^{C}$ and L is a positive constant.

Importantly, there are three terms that determine the loss of classifying the target domain data:

1. the representation discrepancy $d_{1/2}\left(\mathcal{P}_g^T, \mathcal{P}_g^\pi\right)$ for general representations and $\sum_{i=1}^{K}\sum_{j=1}^{K} \frac{L\sqrt{2\pi_j}}{K} d_{1/2}(\mathcal{P}_g^{S,j}, \mathcal{P}_g^{S,i}) + \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{L\sqrt{2\pi_j}}{K} d_{1/2}(\mathcal{P}_g^T, \mathcal{P}_g^{S,i})$ for compressed representations,
2. the label shift, $\max_{i\in[K]} \mathbb{E}_{\mathcal{P}^{S,i}}\left[\|\Delta p^i(y|x)\|_1\right]$, and
3. the general source loss, $\sum_{i=1}^{K} \pi_i \mathcal{L}(\hat{f}, \mathcal{D}^{S,i})$.

Label shift is harder to tackle because it is intrinsic to the domains, so the other two are minimized.

Practically speaking, considering both types of DI representations, the training can be done adversarially with the objective being the following:

$$\min_g \left(\min_{\hat{h}} \mathcal{L}_{gen} + + \lambda^{s-s} \min_{\hat{h}^{s-s}} \mathcal{L}_{disc}^{s-s} + \lambda^{s-t} \min_{\hat{h}^{s-t}} \mathcal{L}_{disc}^{s-t} \right) \quad (2.6)$$

where the λs represent the weighting between the two types of representation; $\hat{h}^{s-s}$ and $\hat{h}^{s-t}$ are discriminators which differentiate the domains; $\hat{h}^{s-s}$ differentiates among the source domains while $\hat{h}^{s-t}$ differentiates between the sources and the target domain.

There are various domain adversarial similarity losses one could use to align different distributions, essentially replacing the discriminators in the expression 2.6. These include MMD [20, 77], CORAL [59, 60], confusion loss [16, 65], perplexity [72], and moment distance [46]. Many of these DA techniques also use similar ideas, such as having a classifier per domain and using their aggregate as the final result [46, 77, 72].

Without hand-crafted adversarial similarity losses, another way to perform DA is to pair the main task with a secondary task. DRCNs use the task of reconstruction of target domain data as the secondary task [19]. The reasoning is that DI features should have enough information for a model to both reconstruct data from any domain and make useful predictions. Other methods use a generative and a discriminative model. The generative submodel uses source and target domain data to generate outputs in the DI space, and the discriminative submodel attempts to discriminate between the domains [26, 8, 47, 17, 69].

2.4 Domain Adaptation for Water Quality Modeling

Applying DA to WQ modelling is a relatively new field. Wang et al. propose the M2M framework [67]. They use MSE for regression and propose the DSD loss as the adversarial similarity loss. Finally, after pretraining, they also fine-tune the model not just on the target domain but also on stations within the same basin. They use these additional stations to prevent the model from overfitting on the target domain while ensuring all the stations have similar hydrological characteristics.

Chapter 3

Methodology

On a high level, the experiments were designed to investigate ways of using high-resolution source domain data to improve the modelling of a target domain that only has temporally sparse data. In this thesis, WQ data was used and the different WQ stations were framed as the various domains. Because all the WQ quality stations had high-resolution data, the sparse target domain was simulated by artificially down-sampling the WQ data.

This thesis' general DA method is depicted in figure 3.1. The prediction model consists of a single invariant submodel f and one variant submodel g per domain. The model prediction for a domain is the combination of the output of f and the output of that particular domain's g. To obtain the final model, the model is first pretrained with the data from the low-sparsity domains. Then, during fine-tuning on the sparse target domain, f is frozen and g is randomly initialized. Unlike strict DI representation learning, there is no explicit regularization for domain differences in the feature space. This thesis focuses on other aspects of the DA so unneeded independent variables were minimized.

This DA method explicitly splits the input-output relationship into its invariant and variant parts. Practically speaking, this method reduces overall computation in two ways. Firstly, the model does not need to be constantly retrained with all the source domains every time there is a new target domain of interest. Secondly, for every new target domain, only g is trained rather than the entire model.

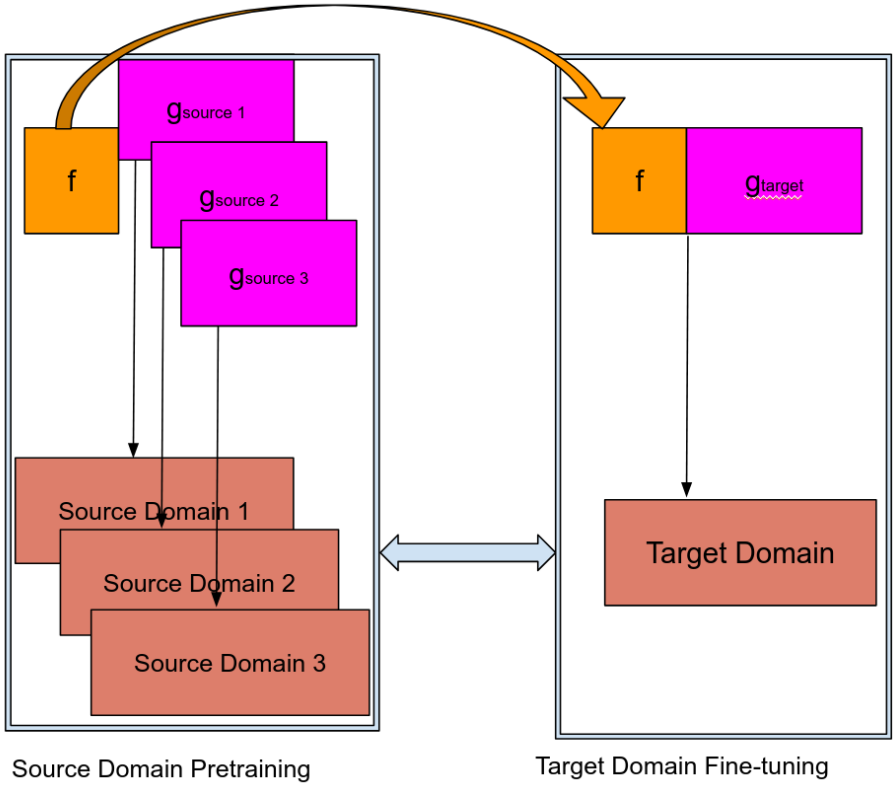


Figure 3.1: The general DA method. First, the f and multiple g_{source}s are pretrained with source domain data. With the frozen weights of f, g_{target} is fine-tuned on the target domain data.

3.1 Data

The dataset used was the Heidelberg Tributary Loading Program (HTLP) dataset [42]. It consists of 25 WQ measuring stations throughout Ohio and Michigan. Each station provides 9 WQ parameters (TSS, Total Phosphorus (TP), Soluble Reactive Phosphorus (SRP), Nitrite + Nitrate (NO23), Total Kjeldahl Nitrogen (TKN), Chloride (CL), Sulfate (SO4), Total Silica (SI), Conductivity (COND)), flow rate (not a WQ parameter), as well as the time the measurement was taken. The WQ data was supposedly gathered 3 times a day using automatic water samplers and the samples were sent back to the National Center for Water Quality Research (NCWQR) laboratories for analysis at weekly intervals. However, though the collection method implies that the samples were evenly sampled, in reality,

they are not. The unevenness in sampling is to the extent that some days have no samples while others have four samples. The stations' details are listed in table 3.1.

Station Name	Basin	Length (days)	Data Availability Average	Data Availability STD among features
Lost	Maumee	5761	0.89	0.013
Wolf	Maumee	1903	0.74	0.03
Shallow Run	Maumee	1695	0.72	0.04
Maumee	Maumee	17645	0.74	0.199
Tiffin River	Maumee	5775	0.62	0.013
Blanchard River	Maumee	5776	0.91	0.07
S. Turkeyfoot	Maumee	1954	0.83	0.035
Potato Run	Maumee	1287	0.87	0.056
West	Maumee	1946	0.82	0.034
Sandusky River	Sandusky	17744	0.70	0.201
Rock Creek	Sandusky	14826	0.82	0.203
Honey Creek	Sandusky	17275	0.77	0.222
Scioto River @ Piketon	Scioto	587	0.61	0.004
Scioto River @ Chillicothe	Scioto	9793	0.75	0.004
Grand River	N/A	6751	0.68	0.235
Vermillion River	N/A	2850	0.94	0.005
Huron River	Huron	1923	0.90	0.031
Muskingum	Muskingum	10626	0.95	0.006
Chickasaw	Chickasaw	5348	0.88	0.012
Raisin	Raisin	15031	0.59	0.162
Portage	Portage	4628	0.92	0.014
Beaver	Beaver	3513	0.40	0.003
Coldwater	Coldwater	4571	0.80	0.015
Great Miami	Great Miami	9884	0.90	0.007
Cuyahoga	Cuyahoga	15153	0.76	0.179

Table 3.1: Data availability for various rivers. Data availability is the fraction of days that have at least one measurement. Only two stations exhibit a data availability below 0.7.

The data preparation consists of several steps. First, to ensure evenly temporally-spaced

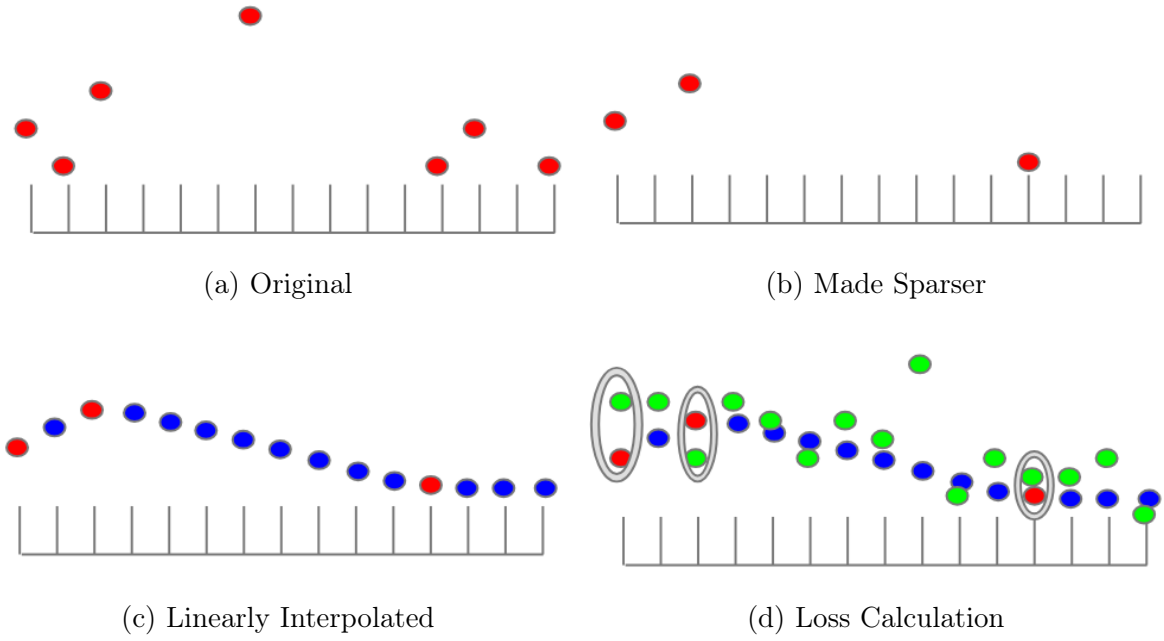
samples, the timestep size was set to be 1 day by taking the mean of the measurements made that day. Next, to reduce skewness in the data, the non-temporal features were transformed using equation 3.1:

$$v^* = log_{10}(\sqrt{v - min(v)} + 0.1) \tag{3.1}$$

This transformation was inspired by other papers that used equation 3.2 to transform streamflow and precipitation. In those papers, the authors used that equation because they found that those variables exhibited gamma distributions and it hindered training. [15, 76].

$$v^* = log_{10}(\sqrt{v} + 0.01) \tag{3.2}$$

Reducing data skewness was necessary for the model to be trained well enough for the experiments of this thesis to be conducted. The non-temporal features were then normalized by subtracting the global mean and dividing by the global standard deviation. Here, a global statistic means calculating the statistic of each feature separately but across all (25) domains. Finally, missing data was imputed by linear interpolation (figure 3.2) and an additional feature indicating whether it was linearly interpolated was added for each existing non-temporal feature.



(a) Original

(b) Made Sparser

(c) Linearly Interpolated

(d) Loss Calculation

Figure 3.2: Example of how sparsity is handled. The red dots are actual measurements; the blue dots are imputed measurements; and the green dots are the model predictions. 3.2a is the original data, with 7 measurements in 15 days. In 3.2b, the data happens to need to be made sparser, so 3 points are sampled. In 3.2c, the missing measurements are linearly interpolated. Finally, when calculating the loss of the model in 3.2d, the loss is only calculated on non-interpolated dates.

Regarding the model's usage of time information, the date was embedded into 3 temporal features (equation 3.3). The first two, t_{sin} and t_{cos}, account for seasonality. The third one, $t_{relative}$ is the measurement time relative to the global minimum and maximum times.

$$\begin{aligned} t_{sin} &= sin(2\pi \frac{t_{day\,of\,year}}{365}) \\ t_{cos} &= cos(2\pi \frac{t_{day\,of\,year}}{365}) \\ t_{relative} &= \frac{t - t_{global\,min}}{t_{global\,max} - t_{global\,min}} \end{aligned} \tag{3.3}$$

In total, the input has 23 features: 10 pairs of non-temporal features and 3 temporal features.

The time series data was split into samples of 128-day windows. For each window, the first 64 days of WQ data and 128 days of flow data were used to predict the latter 64 days of WQ data (figure 3.3 and 3.4). To simulate sparseness in other datsets, the target domain WQ features were made sparser. However, when validating, the latter 64 days were not made sparser to ensure there are as many measured data points as possible to validate the model.

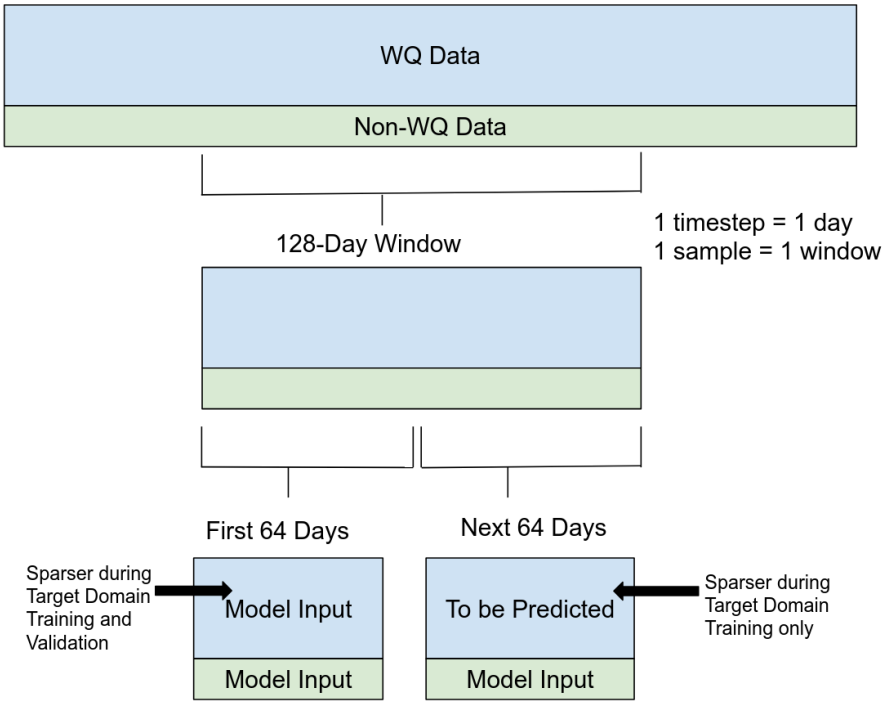


Figure 3.3: Visualization of how the time series data was split to obtain training samples. Which parts are the input and output of the model as well as which parts are made sparser are also illustrated.

3.2 Model Training

As shown in 3.4, both f and g consisted of two LSTMs. LSTMs were chosen as the model architecture because as mentioned in section 2.2, they have been widely and successfully used for WQ modelling.

The two size hyperparameters relevant to LSTMs are the hidden dimension and the

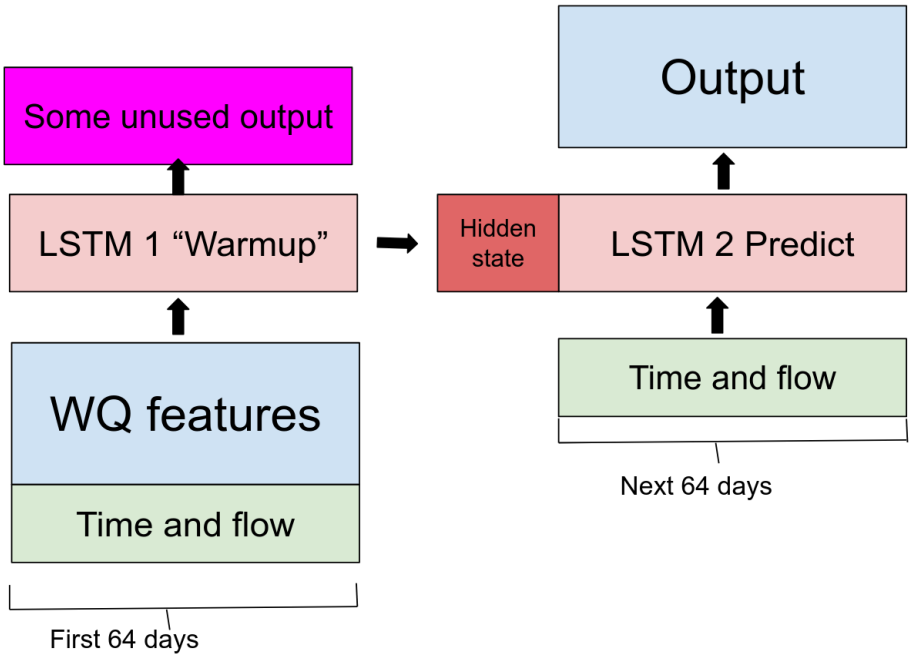


Figure 3.4: General structure of f and g. The first LSTM was fed data from the first 64 days to update the hidden state of the second LSTM. Then the second LSTM was fed the temporal and flow data to predict the next 64 days of WQ data.

number of hidden layers. The notation used is the following: fa,b gc,d, where a is the hidden dimension of f, b is the number of layers of f, c is the hidden dimension of g, and d is the number of layers of g. For example, f32,2 g8,1 means the 2 LSTMs used in f have a hidden dimension of 32 and 2 hidden layers and the LSTMs in each g have a hidden dimension of 8 and 1 hidden layer. In addition, any of the size parameters can be replaced by an underscore (e.g. f_,2 g32,2) to indicate that that size parameter is varied.

During training, the loss was calculated between the true and predicted WQ features only at the non-interpolated times so that the model does not try to fit imputed data. The losses from each domain were added together and backpropagated in every training step. Training was done until no new lowest validation loss was found within the following 30 epochs. MSE loss and the ADAM optimizer [29] were used. One batch contained 32 data windows per domain. Finally, for every training configuration, three runs were performed, each with different seeds to initialize the pseudo-random number generators. This was to reduce uncertainty in the results and ensure replicability.

3.3 Experimental Design

The three aspects of the DA method that were explored were model size, choice of source domains, and method to combine f and g. To investigate these aspects separately, other parameters were fixed to a default training method. The default model sizes are the following 7:

1. f32,2 g8,2
2. f8,2 g8,2
3. f128,2 g8,2
4. f32,1 g8,2
5. f32,2 g32,2
6. f32,2 g128,2
7. f32,2 g8,1

Essentially, these sizes are f32,2 g8,2 but with one of the size hyperparameters changed. During the target domain fitting, for every period of 64 days where the data was made sparse, the tested number of measured points were 1, 3, 6, 12, and 25. Again, this reduced number of measured data points does not apply to the second half of the validation windows. As explained in section 2.1, a typical sparseness ranges from one measurement every month to one measurement every year, so 1 data point every 64 days is a typical sparseness. This range of sparsities can test the extent to which the proposed DA technique is still useful. Furthermore, for all experiments, the performance of freezing and not freezing f were compared with validation loss difference graphs. Not freezing f was used as a baseline to assess the benefits of the DA method.

3.3.1 Model Size

Model size is hypothesized to affect the final model performance in several ways. Firstly, in the pretraining, f needs to be just large enough to capture the true invariance. If f is too small, it will only be a rough approximation of the true invariance. If f is too large, it will capture unwanted correlations that only help in the prediction of the source domains.

Secondly, the size of the model must also be appropriate for training in the target domain, whose temporal sparsity and amount of data are less.

Four size hyperparameters were explored: the hidden dimension and the number of layers for the f and g portions. While varying one of the hyperparameters, the others were fixed to their default values. The hidden dimensions explored were 2, 4, 8, 16, 32, 64, and 128, and the number of hidden layers explored were 1, 2, and 4. For the other experiments where size is not the central focus, models with only 1 or 2 layers, and 8, 32, or 128 hidden dimensions were used.

3.3.2 Choice of Source Domains

A key question is what characteristics of a set of source domains are desirable for Multi-source Domain Adaptation (MDA). One hypothesis is that domains whose distributions are more "similar" to the target domain are better source domains. This hypothesis is reasonable since if the source domains have no relation to the target domain, then the label shift will be high and pretraining will be useless or even detrimental. However, if the set of domains is not "diverse" enough, the model may learn an f that includes unwanted correlations.

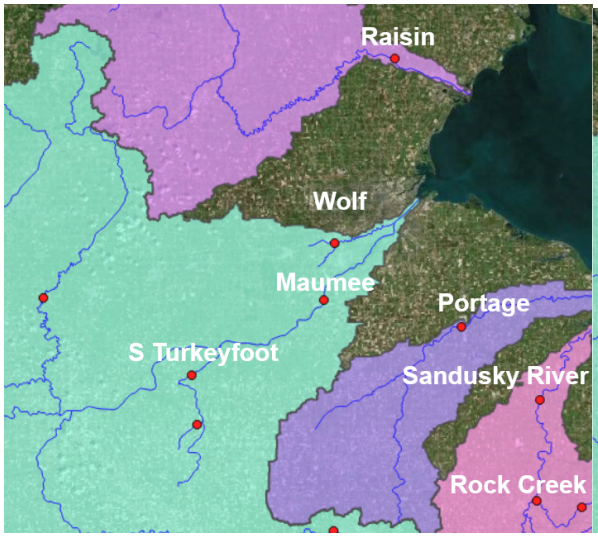
For this thesis, the idea of approximating similarity using the station's basin was tested. The rationale is that stations with the same basins would be affected by similar weather patterns, geology, and hydrology, thus, the inter-parameter relationships within a station will be similar to those of another. Wang et al. also uses the concept of using the stations from the same basin, but for fine-tuning only [67].

Apart from similarity, another possibility is that just the raw number of domains. Perhaps the origin is not important because the model will be able to precisely extract the useful correlations if there are any.

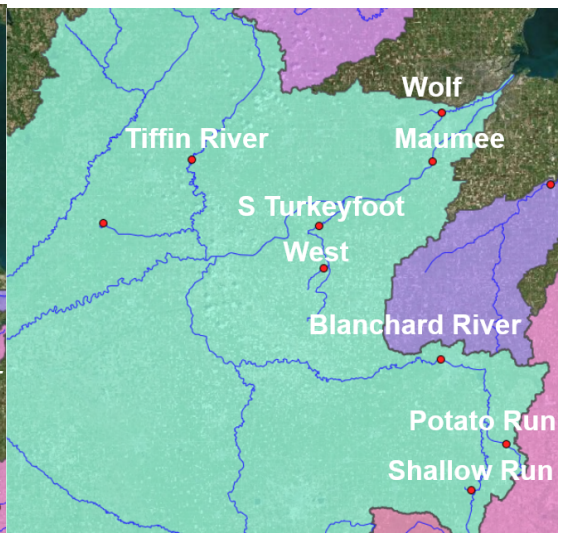
Table 3.2 summarizes the sets of source domains that were tested. Figure 3.5 shows a map of the 3 sets with multiple domains

Domain Set Name	**Description**	**Why the Set was Chosen**	**Stations**
Various	7 random sites	Default set	Maumee, Portage, Raisin, Rock Creek, S. Turkeyfoot, Sandusky River, Wolf
Maumee Basin	All 8 sites from Maumee Basin	Investigate using many stations from the same basin as the target station	Maumee, S. Turkeyfoot, Blanchard River, Potato Run, Shallow Run, Wolf, Tiffin River, West
Pretrain 14	All 8 sites of Maumee Basin plus 6 random sites	Investigate the effect of further increasing the number of source domains	Chickasaw, Honey Creek, Maumee, Raisin, Scioto River @ Chillicothe, Potato Run, Muskingum, Beaver, S. Turkeyfoot, Wolf, West, Tiffin River, Shallow Run, Blanchard River
Just Maumee	Just the Maumee site	Test using a single source from the same basin as the target domain	Maumee
Just Cuyahoga	Just the Cuyahoga site	Test using a single source from a different basin as the target domain	Cuyahoga

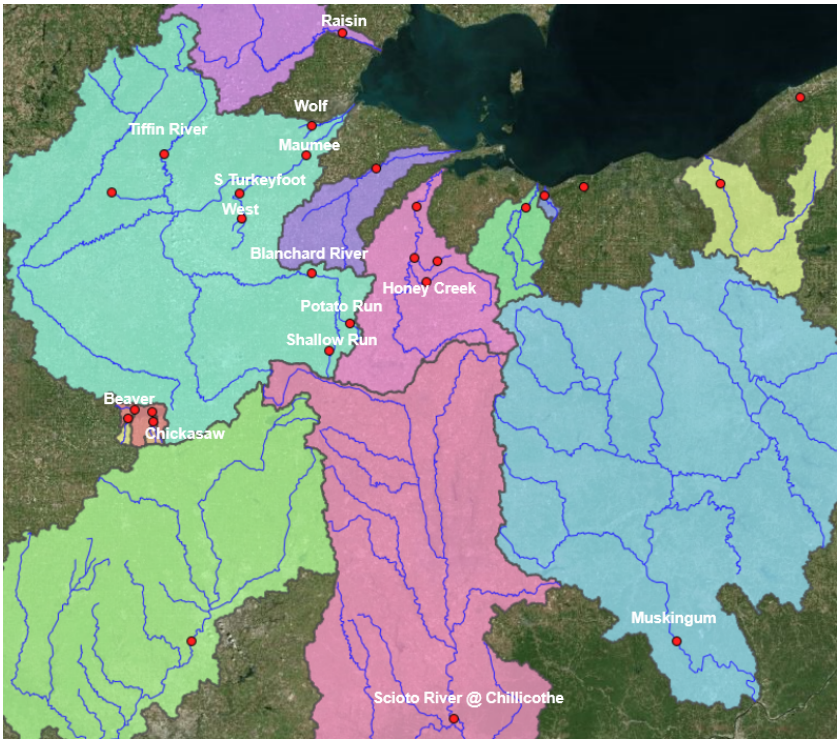
Table 3.2: Experimented sets of source domains with their description, rationale, and stations makeup.



(a) Various



(b) Maumee Basin



(c) Pretrain 14

Figure 3.5: Maps of the sets of source domains with multiple domains [42]. Shaded regions are the different basins. For example, all stations from the Maumee Basin set (figure 3.5b) come from the Maumee Basin, which is turquoise shaded.

3.3.3 Method to Utilize f and g

The relationship between the invariant and variant is of interest. As mentioned previously, the default method to utilize f and g is to feed the output of f into g (figure 3.6). This method is commonly used in transfer learning. For example, in transfer learning for image classification [62], after pretraining, only the top few layers of the model are retrained for a new task, and the rest of the model is frozen.

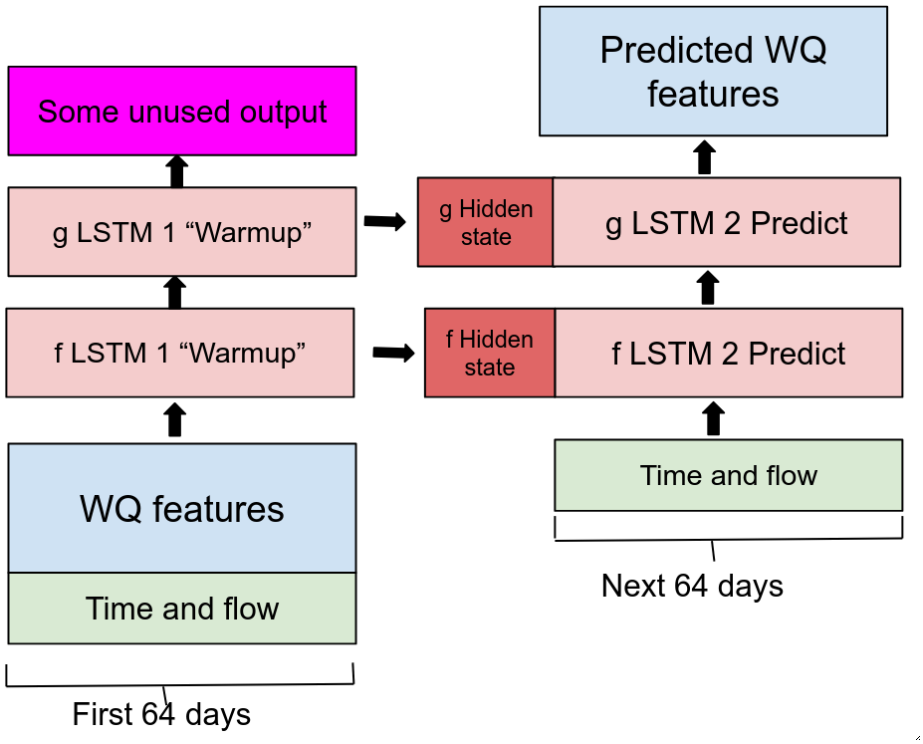


Figure 3.6: The default method to utilize f and g. Data is fed into f and the output of f is then fed into g.

The default method of utilizing f and g may not be optimal or necessary. Another method is first to feed the data into f and g, then sum the results to obtain the WQ prediction (figure 3.7). Although it is simpler and requires less computation, it may not be flexible enough because it assumes the relationship between the invariant and variant is additive.

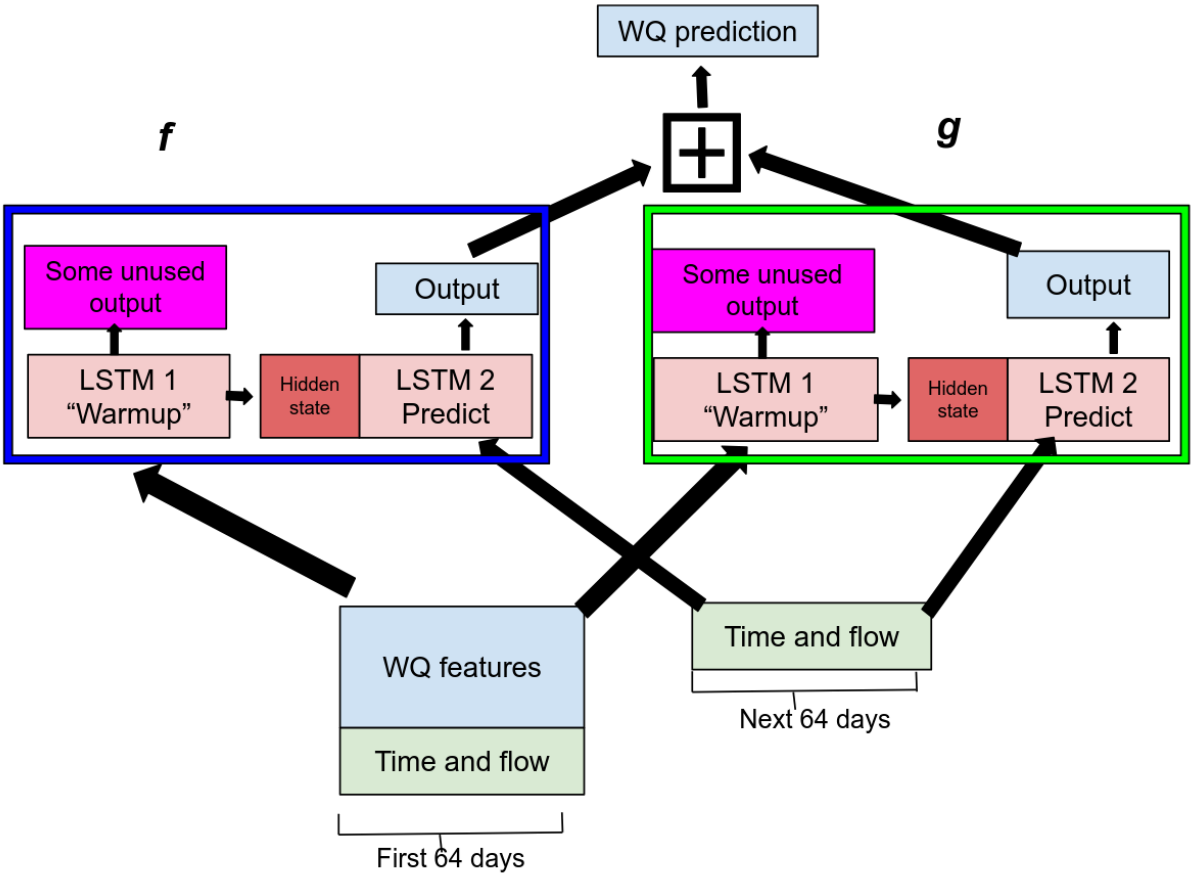


Figure 3.7: The alternative method to utilize f and g. Data is fed into each portion separately and the outputs are added together to obtain the prediction of the model.

Chapter 4

Results and Analysis

4.1 Validity of the DA Method

The validity of the DA method was first verified. The default DA method was compared to not pretraining. Specifically, in one case, the model was trained with only the temporally sparse data of the target domain. In the other case, the model was trained with both the data of the target and source domains. As shown in figure 4.1, the default method is always better at all temporal sparsities up to 25 measurements for 64 days. This observation also holds with any equivalent model sizes (figure B.1 in appendix B). Figure 4.2 plots the three models' outputs for one of the WQ features (TSS). From the graph, the default method does indeed fit the actual WQ data better.

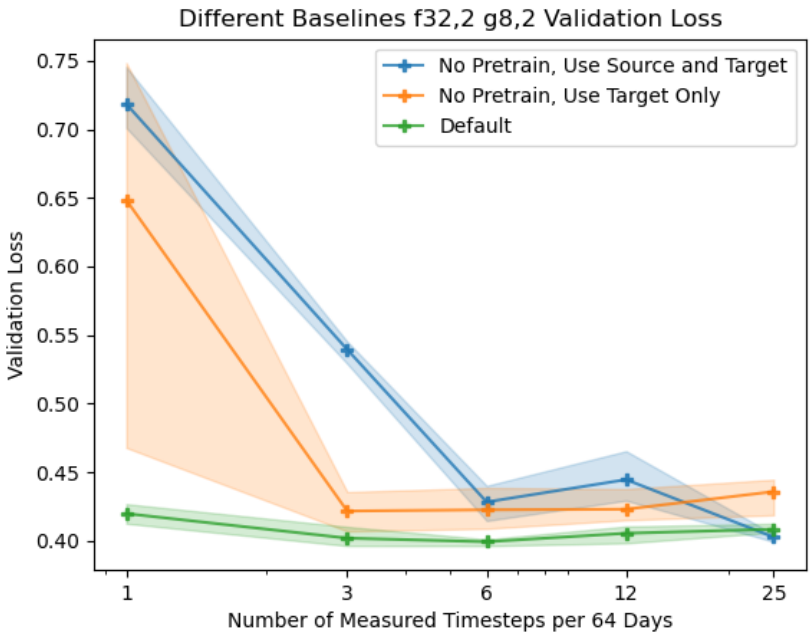
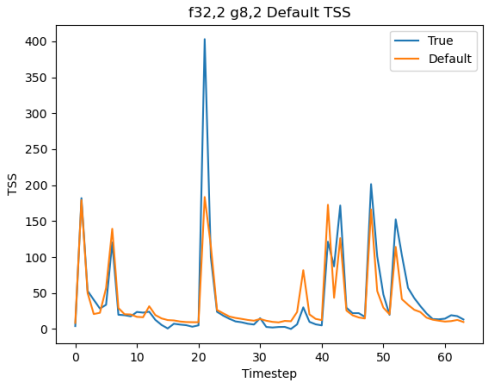
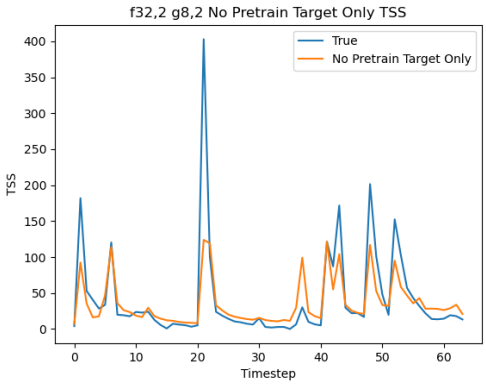


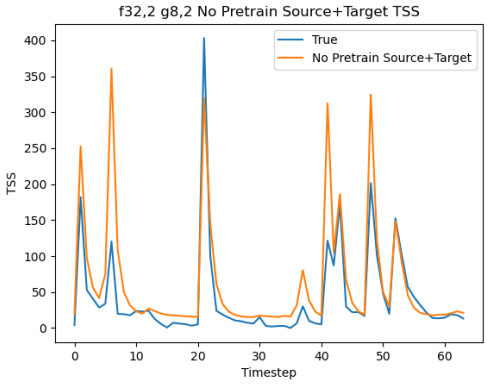
Figure 4.1: Comparing the validation loss of the default DA method versus both no pretraining baselines, at various sparsities, for the default model size of f32,2 g8,2.



(a) Default



(b) No Pretraining, Target Domain Only



(c) No Pretraining, Target Domain Only

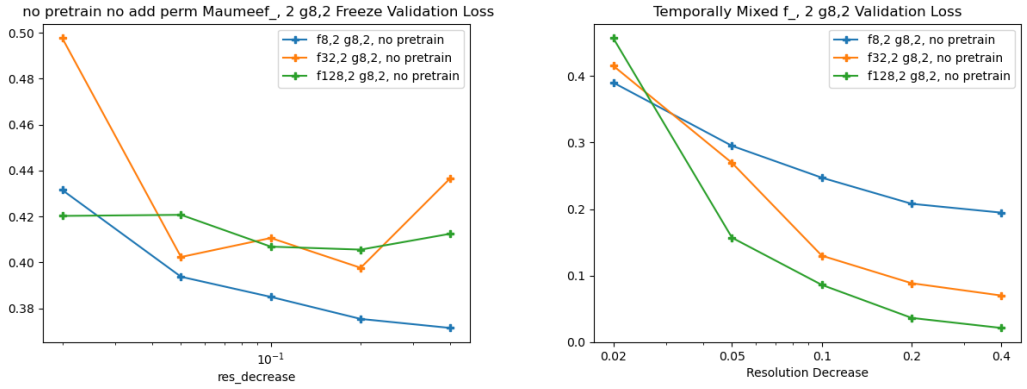
Figure 4.2: Sample WQ predictions from a random validation sample. Figure 4.2a is for the default DA method, figure 4.2b is for training only with target domain data, and figure 4.2c is for training with source and target domain data. The y-axis is the true TSS values, obtained by reversing the data processing transformations on the model outputs. The x-axis shows the timesteps in this particular window's latter 64 days. The model size was f32,2 g8,2 and the number of measured timesteps in the first 64 days is 3.

4.2 Effect of Decreasing Temporal Sparsity

In general, not just in terms of DA, the validation loss was expected to monotonically decrease as the temporal sparsity of WQ data decreased. This was expected because the

model will have more measured WQ data points not just for the input but also for the measured target values. However, in almost all cases (figure 4.1, 4.6, 4.9, 4.11, 4.13) that depict validation loss as temporal sparsity decreases, the validation loss does not monotonically decrease. Instead, as the temporal sparsity decreased, there were three stages to the loss' behaviour. First, the loss decreased as expected. Then, it increased. In the final stage, the loss' behaviour was different among runs. This phenomenon occurred no matter the size of the model, the set of source domains used, and whether the model was pretrained or not.

Furthermore, this occurred even if the target domain was switched to Maumee station and the model was not pretrained (figure 4.3a). This eliminates the possibility of pretraining or the lack of data being reasons for the phenomenon. The only case where this effect was not present occurred when the training and validation set were not separated temporally (figure 4.3b). This suggests that increasing temporal sparsity decreases the model's ability to fit the training data but not necessarily its ability to generalize to other validation examples that are separated temporally. Therefore, when analyzing the results, the model cannot necessarily be assumed to perform better if temporal sparsity is decreased. The fact that reducing temporal sparsity does not necessarily decrease validation loss must be taken into account when performing analysis of the results.



(a) (b)

Figure 4.3: Figure 4.3a depicts the validation loss of a model trained on Maumee. Figure 4.3b depicts the validation loss when training and validation data were mixed temporally.

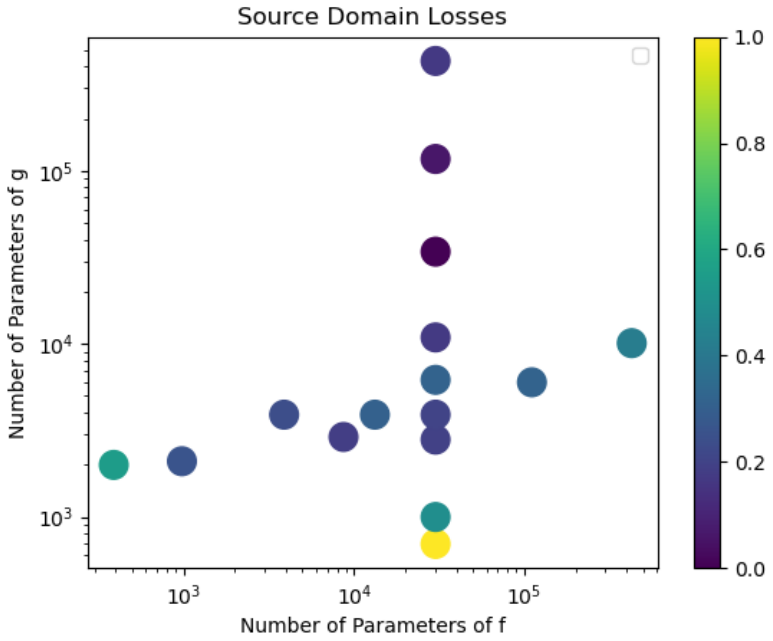


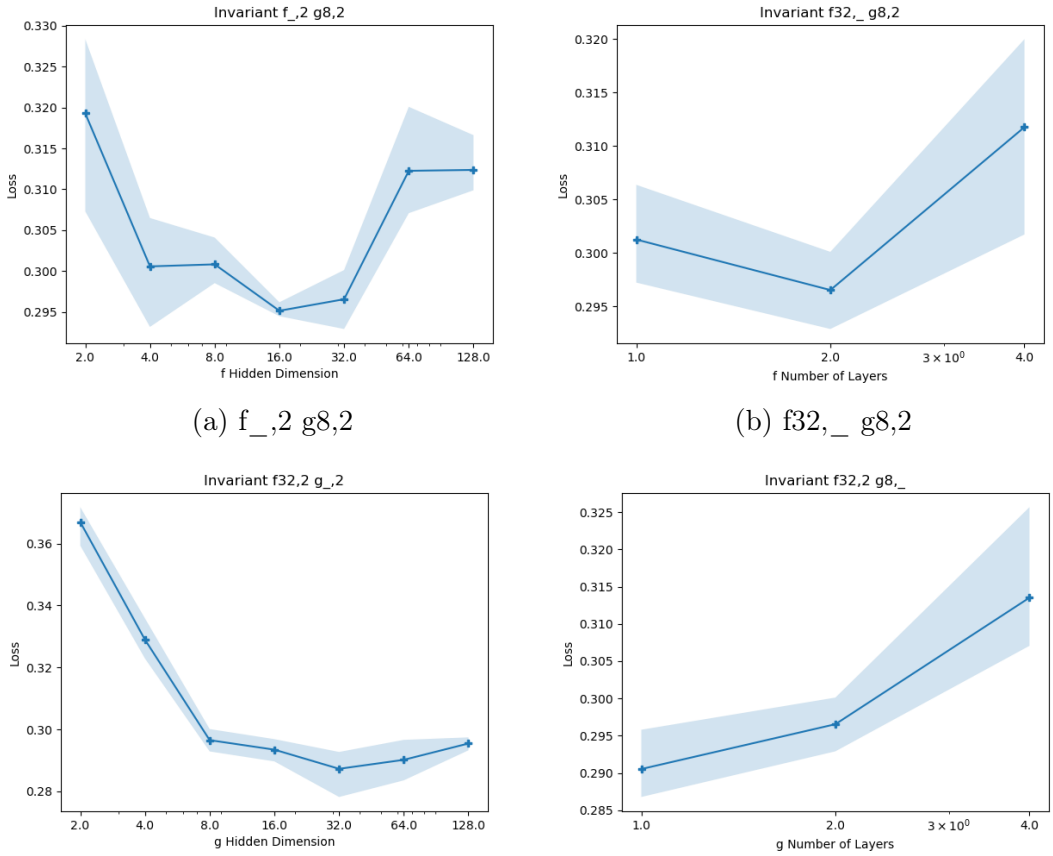
Figure 4.4: Scaled pretraining validation loss (color) with respect to number of parameters of f and g. The validation losses were divided by the maximum validation loss of all the runs.

4.3 Model Sizes

4.3.1 Source Domain Pretraining

The pretraining validation loss with respect to model size was explored to see if it helps predict the model's performance in the target domain. The first figure, figure 4.4, plots the validation losses with respect to the number of parameters of f and of g. The only conclusion that can be drawn is that the model won't fit well if g is too small.

The next figure, figure 4.5, visualizes the variation of the size parameters separately. These graphs indicate that there are optimal values for the size parameters. For f, 16 or 32 hidden dimensions and 2 layers resulted in the lowest validation loss. For g, it was 32 hidden dimensions and 1 layer.



(a) f_,2 g8,2

(b) f32,_ g8,2

(c) f32,2 g_,2

(d) f32,2 g8,_

Figure 4.5: Source domain validation loss of models where only one of the size parameters was varied at a time.

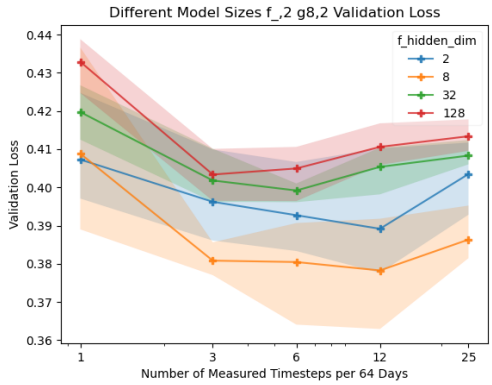
4.3.2 Target Domain Fine-tuning

Figure 4.6 depicts the validation loss in the target domain when varying each size parameter. For the most part, the relative performance between the different sizes did not change as temporal sparsity varied. For f and g, using 8 and 128 hidden dimensions performed the best respectively. For both f and g, using one layer performed the best.

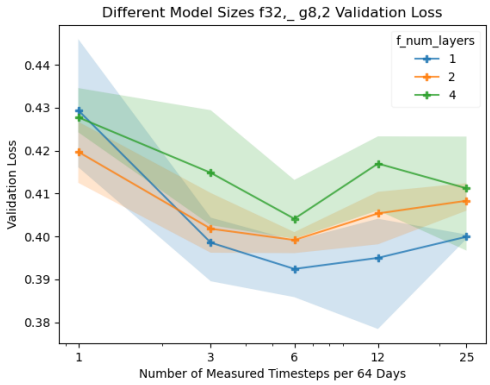
Model Size Paramater	Source Domain Pretraining	Target Domain Fine-tuning
f hidden dimension	16 or 32	8
f number of layers	2	1
g hidden dimension	32	128
g number of layers	1	1

Table 4.1: Optimal model size parameters between source domain pretraining and target domain fine-tuning. Compared to the pretraining, the optimal f for the fine-tuning was smaller and the g hidden dimension was larger

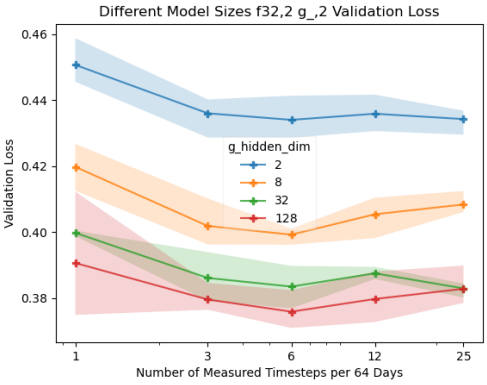
The optimal sizes for the pretraining were different from the ones for the target domain (table 4.1). Specifically, the optimal model had a comparatively smaller f and a larger hidden dimension for g. This means that when f has a smaller capacity and captures fewer correlations, the model performs better when fitting the target domain. This suggests that source domains have correlations among themselves that are not shared with the target domain. When f is larger, e.g. f32,2, it captured correlations that existed only among the source domains. If so, that would explain why a hidden dimension for g would perform better. g's extra capacity would allow the model to offset the wrong f by training on the sparse target domain data.



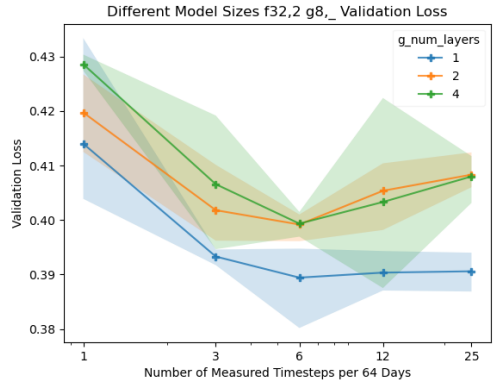
(a) f_, 2 g8,2



(b) f32, _ g8,2



(c) f32, 2 g_,2



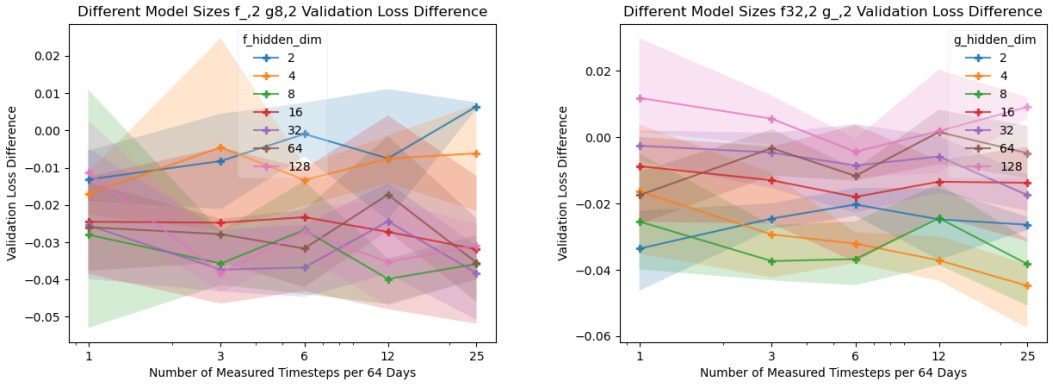
(d) f32, 2 g8,_

Figure 4.6: Validation loss of various sizes of models at different sparsities. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss. Figure 4.6a and 4.6c only show a portion of the model sizes for readability. See B.2 in appendix B for full graphs.

Figure 4.7 shows the difference in validation loss between freezing and not freezing f. Firstly, all runs except for f32,2 g128,2 and f2,2 g8,2 have negative differences, meaning freezing is indeed better in most cases. Note that this also includes the runs where the number of layers was varied (figure B.2 in appendix B).

From figure 4.7a, models with a hidden dimension greater than 4 all have lower validation loss differences than those without. This indicates that at small hidden dimensions for f, increasing the hidden dimension increases the efficacy of the DA method. Figure 4.7b

shows that a higher hidden dimension for g resulted in a larger validation loss difference. This further supports the claim that models with a larger hidden dimension for g perform better due to g's capability to fit the target domain better rather than the model learning a better f.



(a) f_, 2 g8,2 (b) f32, 2 g_,2

Figure 4.7: Validation loss difference between freezing and not freezing f for various sizes of hidden dimensions at different sparsities. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

4.4 Choice of Source Domains

4.4.1 Source Domain Pretraining

Similar to section 4.3, the pretraining losses are investigated first to see if there are any connections to the final model's performances. The details of the experimented sets are listed in table 3.2. As a reminder, Just Cuyahoga and Just Maumee are sets with only a single source domain. These sets are meant to investigate the effects of severely reducing the number of source domains. Between the two sets, Just Maumee's station shares the same basin as the target station while Just Cuyahoga's does not. Various, Maumee Basin, and Pretrain 14 are sets with 7, 8, and 14 source domains respectively. The Maumee Basin set is meant to investigate the effect of using multiple stations that all belong to the same basin as the target station. Pretrain 14 is meant to investigate having additional source domains without worrying about their relations to the target domain.

Figure 4.8 shows the validation losses from pretraining on various sets of domains. The set of source domains is the significant factor in determining pretraining validation loss; for the same model size, the set of source domains ordered by their validation loss from least to greatest is always Just Cuyahoga, Just Maumee, Various, Pretrain 14, and Maumee Basin.

Interestingly, Maumee Basin has a much higher validation loss than Various even though it has just one more domain. As well, Just Maumee has a similar validation loss to Various even though it only has one domain. Therefore, the number of domains cannot solely be used as a predictor of how low the pretraining loss will be.

The validation loss also varies with size. For the models pretrained on Maumee Basin, the validation loss varies much more with size than the other sets of source domains. Validation loss does not vary with size the same way for different sets of source domains. For example, when comparing f128,2 g8,2 to f32,2 g8,2, the validation loss increased for Maumee Basin while it decreased for Pretrain 14.

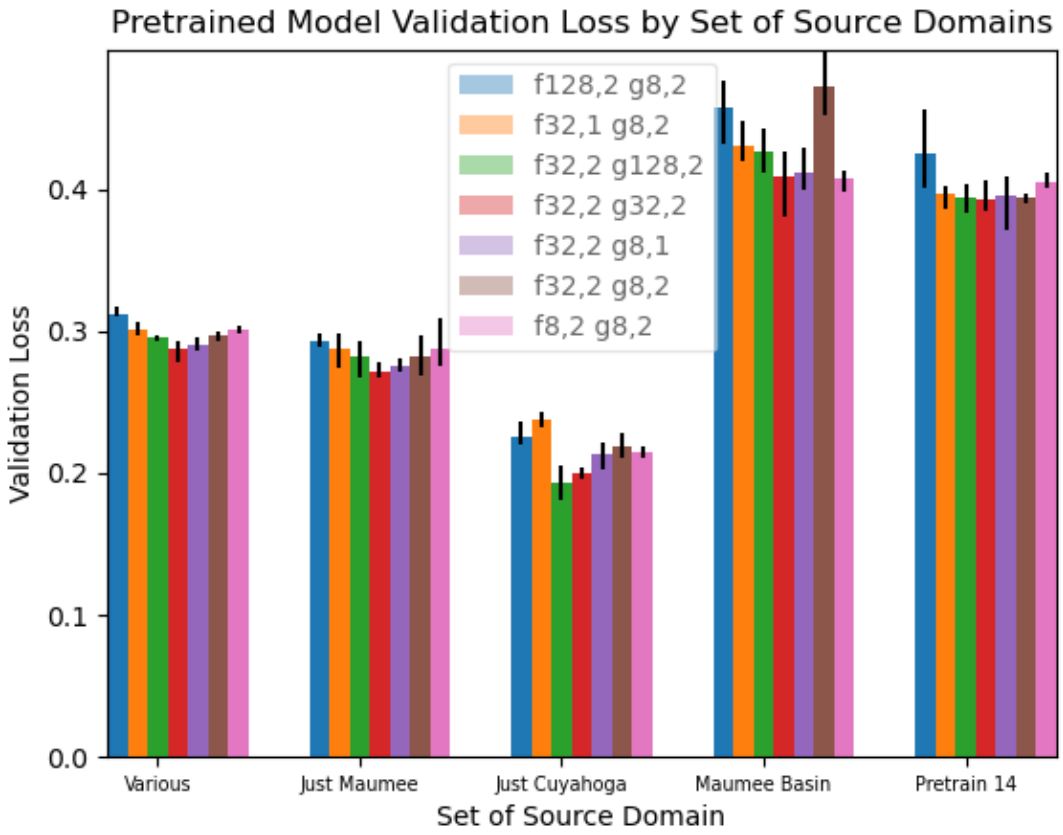
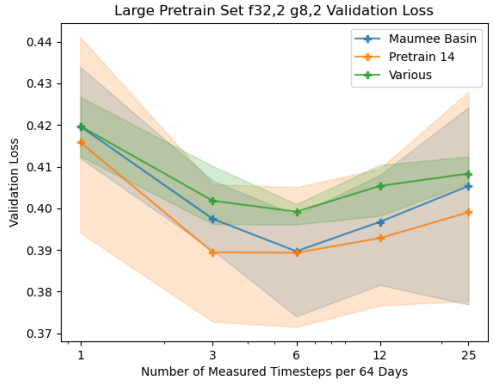


Figure 4.8: Pretraining losses for various sets of source domains and various model sizes. Error bars show the minimum and maximum validation loss of the three differently seeded runs.

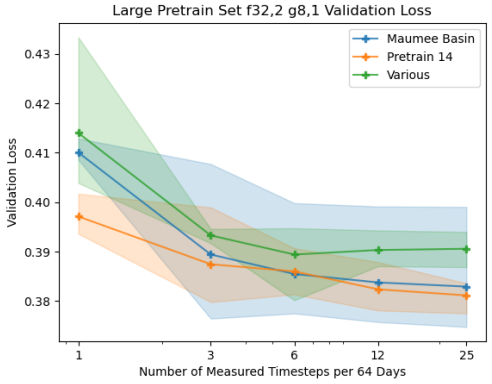
4.4.2 Target Domain Fine-tuning

Next, the target domain fine-tuning was done. Figure 4.9 depicts the validation loss using models pretrained on sets with multiple domains (Various, Pretrain 14, and Maumee Basin). Overall, for the sets with seven or more source domains, the models all perform similarly and their relative performances with each other are not consistent. This suggests that the number of domains and their similarity (basin of station) do not influence the performance when there are seven or more source domains. Furthermore, concerning the

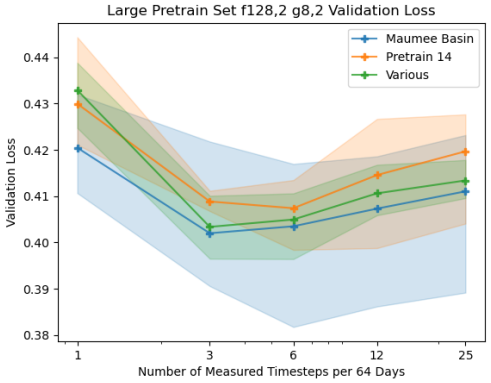
choice of source domains, there is no connection between the model's source domain fitting loss and the model's performance in the target domain.



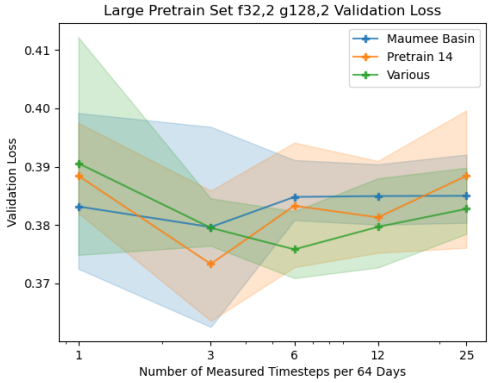
(a) f32,2 g8,2



(b) f32,2 g8,1



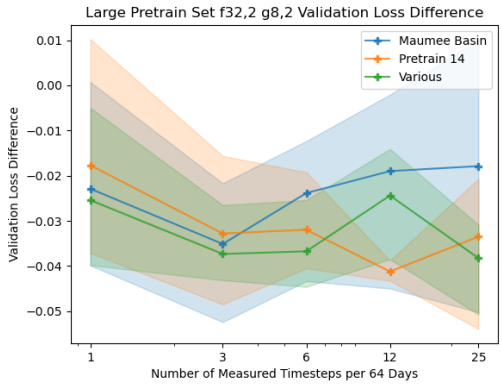
(c) f128,2 g,2



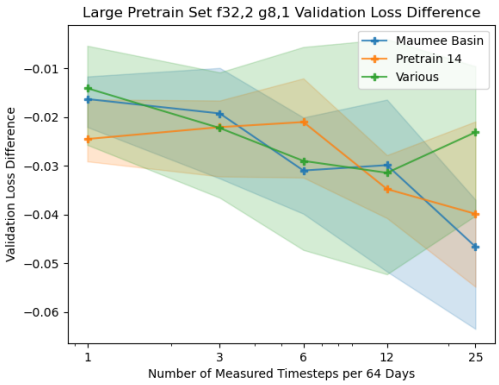
(d) f32,2 g128,2

Figure 4.9: Validation loss of sets with multiple source domains, for various sizes of models, and at different sparsities. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

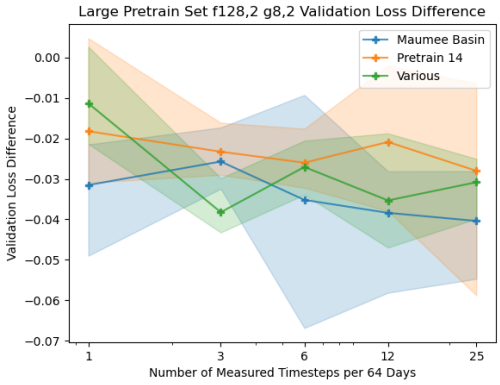
The same conclusion is drawn from the loss difference graphs (figure 4.10). Their loss differences are similar and their relative rankings of their loss differences are inconsistent. However, at the very least, the DA method remains effective since all runs except for f32,2 g128,2 have negative loss differences.



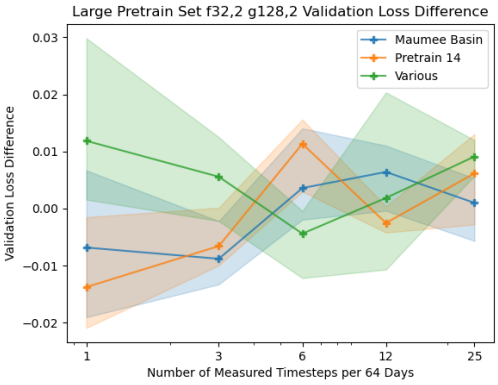
(a) f32,2 g8,2



(b) f32,2 g8,1



(c) f128,2 g,2



(d) f32,2 g128,2

Figure 4.10: Validation loss difference between freezing and not freezing f of sets with a multiple of source domains, for various sizes of models, and at different sparsities. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

Figure 4.11 depicts the validation loss of f32,2 g8,2 when the number of pretrained domains in a set is reduced to 1. As expected, the two sets with a single source domain performed poorly compared to Various. This was also true for all other model sizes. This suggests that either (1) a minimum number of distinct source domains is needed or, (2) a sufficient diversity within the source domains is crucial.

One strange observation is that in almost all runs, Just Cuyahoga was better than Just Maumee. This is surprising because the Maumee and Lost stations belong to the same

basin, while the Cuyahoga station lies far away from both stations. Thus, the Maumee station was expected to be more similar to the Lost station, and pretraining on the Maumee station data was expected to result in a lower validation loss. Furthermore, as listed in table 3.1, the two stations are very similar in terms of data quantity and availability; the Maumee station has a little more data (17645 to 15153) but its data availability is slightly less (0.74 to 0.76) and the variation in data availability among different features is slightly more (0.199 to 0.179).

The straightforward explanation is that geographical distance and the stations' basins are not a good measure of similarity in this application. More broadly speaking, another possibility is that there is another characteristic more significant than similarity in these scenarios that determines the quality of a set of source domains.

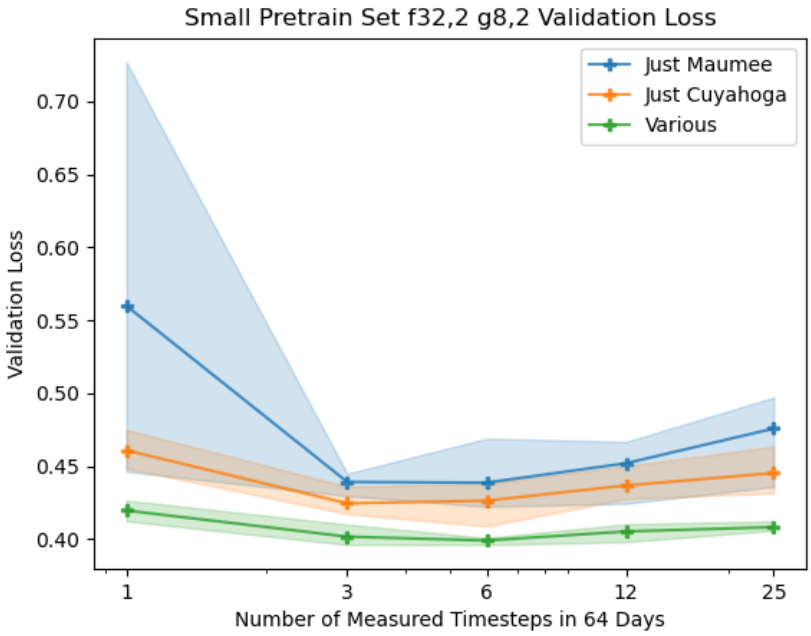


Figure 4.11: Validation loss (1) of the Various, Just Maumee, and Just Cuyahoga sets, (2) for f32,2, g8,2, and (3) at different sparsities. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

When looking at the validation loss difference (figure 4.12), Various has a lower loss difference than the sets with a single source domain. Furthermore, there were only two

cases where models using a single source domain had negative validation loss differences: when the hidden dimension of f is 128 and when the number of layers of f is 1 (figure B.7 in appendix B). In conclusion, it is necessary to have a minimum number of source domains for the DA method to be beneficial.

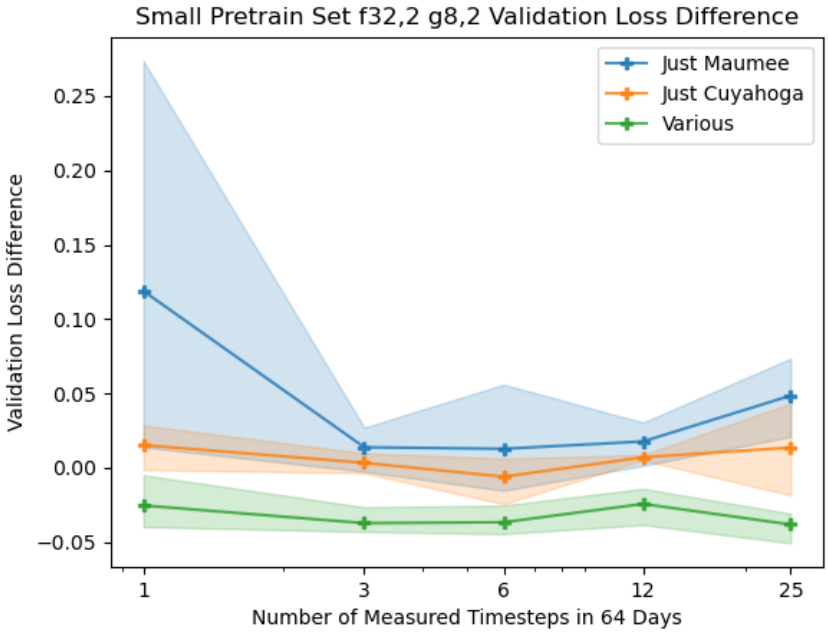


Figure 4.12: Validation loss difference between freezing and not freezing f (1) of the Various, Just Maumee, and Just Cuyahoga sets (2) for various sizes of models, and (3) at different sparsities. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

4.5 Method to Utilize f and g

In figure 4.13, adding the outputs of f and g is compared to the default method of feeding the input through f and then feeding the output through g (see figures 3.6 and 3.7). In all cases, the default method of combining f and g has a lower validation loss.

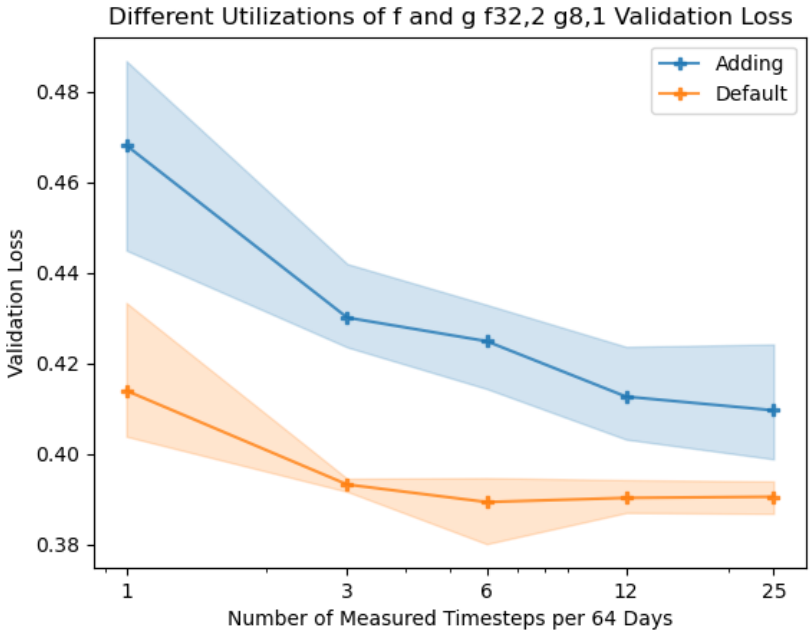


Figure 4.13: The validation loss of adding the outputs of f and g is compared to the default method. The graph displays the validation loss for f32,2 g8,1, at various sparsities. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

Furthermore, from figure 4.14, when adding the outputs of f and g, it is better not to freeze the model in all cases except f32,2 g128,2, indicating that the f learned from the pretraining is not good. From figure 4.15b, for adding the outputs of f and g, increasing the hidden dimension of g actually decreases the validation loss difference. This indicates that the relationship between the model size and performance is different between the two f and g utilization methods. Specifically, the hidden dimension of g where the DA method is most effective may be greater than 128.

In conclusion, in all tested cases, adding the outputs of f and g does not lead to a better model performance when compared to the default method. Furthermore, when the model adds the outputs of f and g, except for the case of f32,2 g128,2, the DA method is ineffective.

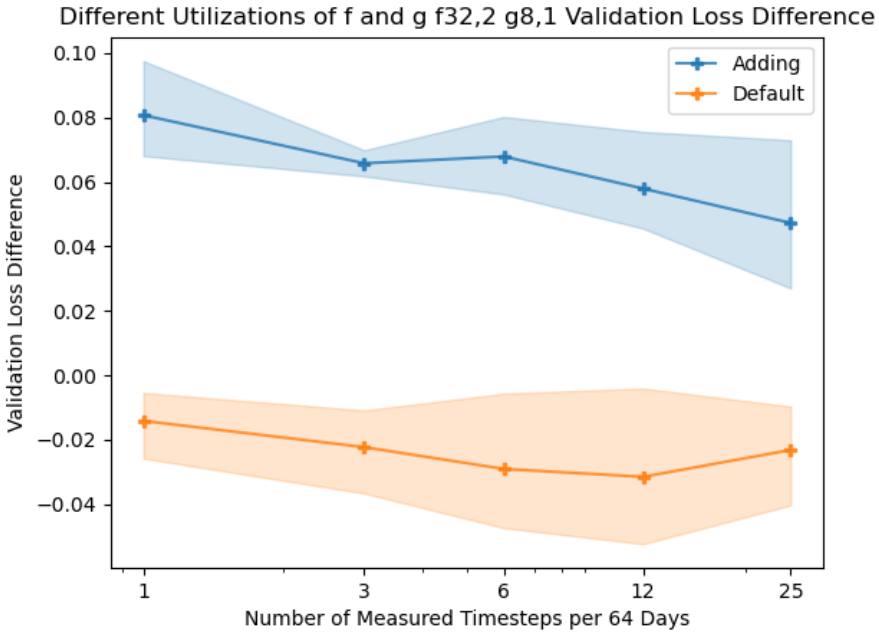
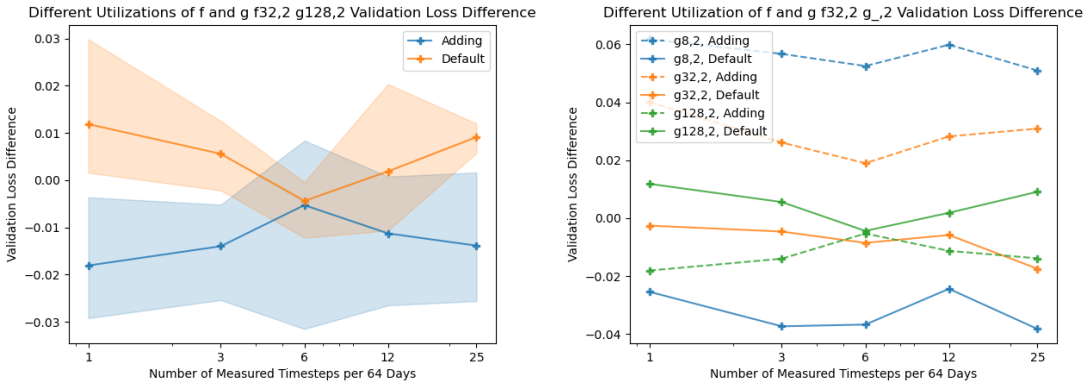


Figure 4.14: The validation loss of adding the outputs of f and g is compared to the default method. The model size shown is f32,2 g8,1. This size is representative of the other sizes. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.



(a) (b)

Figure 4.15: The case where adding the outputs of f and g achieved negative validation loss difference. For figure 4.15a, the model size is f32,2 g128,2. The results of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss. For figure 4.15b, the hidden dimension of g is varied and only the average of 3 runs is shown.

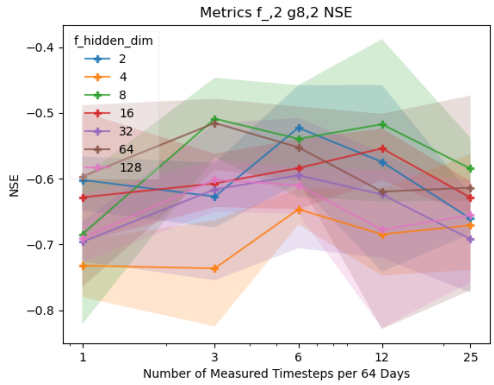
4.6 Evaluation Under Different Metrics

To find the true performance of the trained models, the NSE and KGE are calculated for the experiments performed in 4.3 for model sizes (figure 4.16 and 4.17). Note that model predictions have been transformed back to their actual measurements before the metrics are calculated. Overall, the metrics agree with the validation loss in terms of which models perform better. The optimal hidden dimensions for f and g are still 8 and 128 respectively. The optimal number of layers for f and g are still both 1. However, the NSE and KGE are low. For NSE, the negative scores indicate that the models' performances averaged across the different features are worse than using the average value of a window.

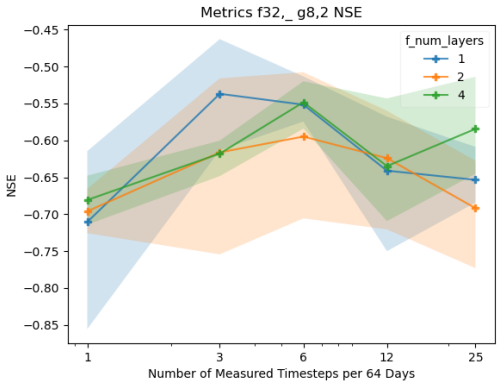
NSE	Interpretation
>0.75	Well
0.36-0.75	Satisfying
<0.36	Less Satisfying

KGE	Interpretation
>0.75	Good
0.5-0.75	Intermediate
<0.5	Poor

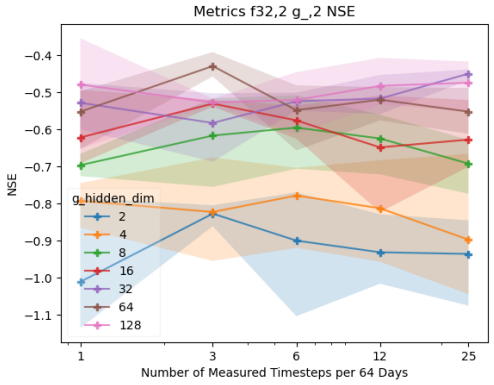
Table 4.2: Interpretation of typical NSE values (left) [14] and KGE values (right) [40] for hydrological models.



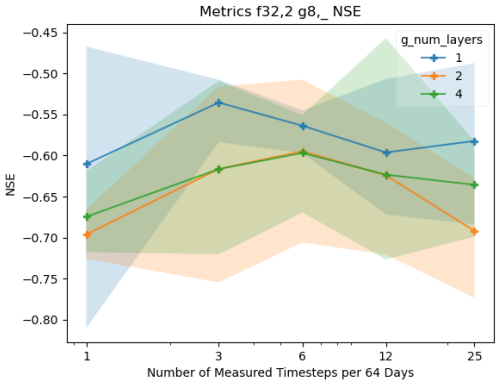
(a) f_, 2 g8,2



(b) f32, _ g8,2

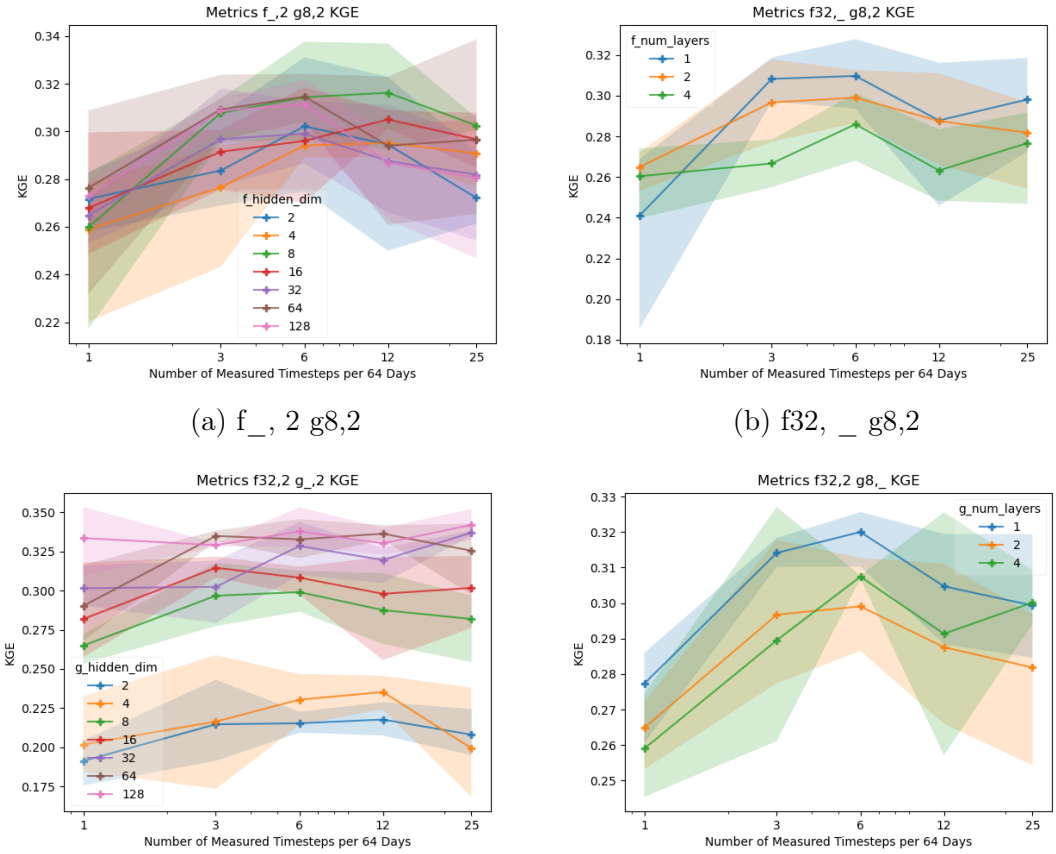


(c) f32, 2 g_,2



(d) f32, 2 g8,_

Figure 4.16: NSE of different-sized models at differing sparsities. The values represent the mean validation loss of the three runs with seeds and the envelope represents the maximum and minimum validation loss.



(a) f_, 2 g8,2

(b) f32, _ g8,2

(c) f32, 2 g_,2

(d) f32, 2 g8,_

Figure 4.17: KGE of different-sized models at differing sparsities. The value represents the mean validation loss of the three runs with different seeds and the envelope represents the maximum and minimum validation loss.

However, there is a large variation in metric scores among the different features. The maximum NSEs for five of the features (TSS, TP, SRP, TKN, and SO4) were "satisfying" and the maximum KGEs for three of the features (TP, TKN, and SO4) were intermediate (table 4.3). For these features, the model fit the target domain fairly well. On the other hand, the feature that consistently had the worst metric score was COND.

Feature	Mean NSE	Min NSE	Max NSE	Mean KGE	Min KGE	Max KGE
[TSS]	0.080	-3.037	0.460	0.298	-0.084	0.471
[TP]	0.536	0.307	0.645	0.593	0.461	0.657
[SRP]	0.244	-0.023	0.379	0.389	0.266	0.468
[NO23]	-1.440	-2.751	-0.617	0.030	-0.235	0.232
[TKN]	0.135	-0.763	0.390	0.529	0.341	0.602
[CL]	-0.493	-1.463	-0.061	0.204	-0.165	0.374
[SO4]	-0.024	-0.908	0.391	0.560	0.343	0.678
[SI]	-0.701	-1.673	-0.077	0.253	-0.327	0.498
[COND]	-4.560	-8.168	-2.606	-0.410	-0.735	-0.170

Table 4.3: The mean, minimum, and maximum NSE and KGE for each WQ feature over all sparsities and pretrained on the Various set.

Chapter 5

Discussion and Future Work

5.1 Experiments

5.1.1 Quality of Experiments

Out of all WQ features, only 3-4 were modelled to a decent extent. In addition, the model's performance for COND was especially poor, when measured by both its NSE and KGE values.

However, there are several aspects to consider regarding the general performance of the models. First, all WQ features were weighed equally during optimization. In other words, the WQ features were prioritized equally. However, if there is only a single WQ feature of interest, then it should be solely optimized, or at least heavily prioritized. In the experiments, the overall relative performances of the models under different configurations were the main interest, and so the specific WQ feature characteristics, especially their relation to flow, were not explored. However, a more detailed analysis could be conducted on each model's performance on each WQ feature to ensure none of the models are skewing their predictions to a single WQ feature.

Secondly, although there have been past models built for this geographical area that deal with similar WQ features [54], there are none that operate with a daily timestep and deal with a reduced number of measurements, and so there are no perfect benchmarks for the obtained metric values. Future work into developing sparsity-focused models would be useful for benchmarking.

To ensure the certainty of the findings, all experiments were initialized with 3 different seeds to reduce the effect of randomness from parameter initialization. The findings from the investigation on model size are credible; the relative performance between the models of different sizes did not vary that much with different seeds or sparsities. The result regarding the method of utilizing f and g is also quite definite as the default method always performed better than the addition method in nearly all cases. In contrast to the first two areas that were explored, the results from the investigation of the choice of domains were more inconclusive and chaotic than expected. Further cross-validation or different seeds could be used, as well as a more comprehensive and specific definition of similarity (see the next section 5.1.2).

Outside of the three main areas explored, one significant finding is the fact that decreasing temporal sparsity did not always decrease validation loss. Although the temporal difference between the training and validation set explains this phenomenon, a more in-depth investigation should be conducted to quantify its effect.

5.1.2 Main Findings

Overall, the DA method was successful. When compared to not pretraining, the default method was always better. When compared to not freezing f, the default method was better in the majority of relevant cases.

Ultimately, all three investigated areas—the size of f and g, the composition of the source domains, and the method used to combine f and g—significantly impacted the model's performance.

The optimal sizes for f and g were found to depend on both the source and target domain, leading to different optimal sizes during pretraining and training. Specifically, when fitting in the target domain compared to the source domains, the optimal size for f was smaller and the hidden dimension of g was larger. However, in those experiments, the size of g was not changed between the pretraining and target domain fine-tuning, meaning the optimal g size was constrained to work well with both the source and target domains. To get at understand in more detail the effects of model size, the size of g can be allowed between pretraining and fine-tuning.

Only the number of source domains was found to affect the model performance. Specifically, there is a minimum number of source domains needed for the DA method to do well. Domain similarity on the other hand was not found to improve model performance, especially in the single source domain case. However, it is not clear whether this finding is due to the effect of similarity being too insignificant or whether the basin of stations can

be used as a measure of domain similarity. Using a dataset with stations located geographically further apart can help resolve this uncertainty. When using a single source domain, if using source stations that are closer to the target results in better performance, then the effect of similarity in this thesis' experiments was too insignificant but using the station's basin as a measure of similarity is still valid. Similarity metrics based on simple statistics of the data or the static features of the sites could be used. However, in the end, it still must be evaluated by how well it correlates with the performance of a DA method. Outside of the analysis of WQ features, incorporating other available static and geographical features could help develop a more complete notion of similarity from a physical standpoint.

Finally, using the default method to combine f and g performed better than simply adding the outputs of f and g. However, this does not confirm whether the default method is optimal or necessary. The order of f and g in the default method can be explored. Another method of combining f and g is to feed both the input data, concatenate their outputs, and feed that into f_2 to obtain WQ predictions (figure 5.1). f_2, like f, is jointly trained during pretraining and frozen in the target domain fine-tuning.

5.2 Broader Context

Although this DA method was applied in the context of WQ modelling, it is a very general framework. As long as there are source and target domains and there is a need to improve target domain predictions, this DA method is applicable. In this thesis, temporal sparsity was chosen as the "quality" measure of interest but spatial resolution (e.g. density of stations in an area), noise (e.g. less accurate sensor readings), and quantity of the data could also be measures of "quality".

Regarding choosing LSTMs as the model architecture of choice, changing it to most other types of DL architecture (e.g. Convolutional Neural Network (CNN)s, transformers, Structured State Space Sequence Model (S4)s) should be fine as long as the architecture can be divided into layers. Otherwise, a new method to combine f and g may be required. Finding an effective method of combination is important as shown from the experiments, where simply adding the results together underperforms compared to the default.

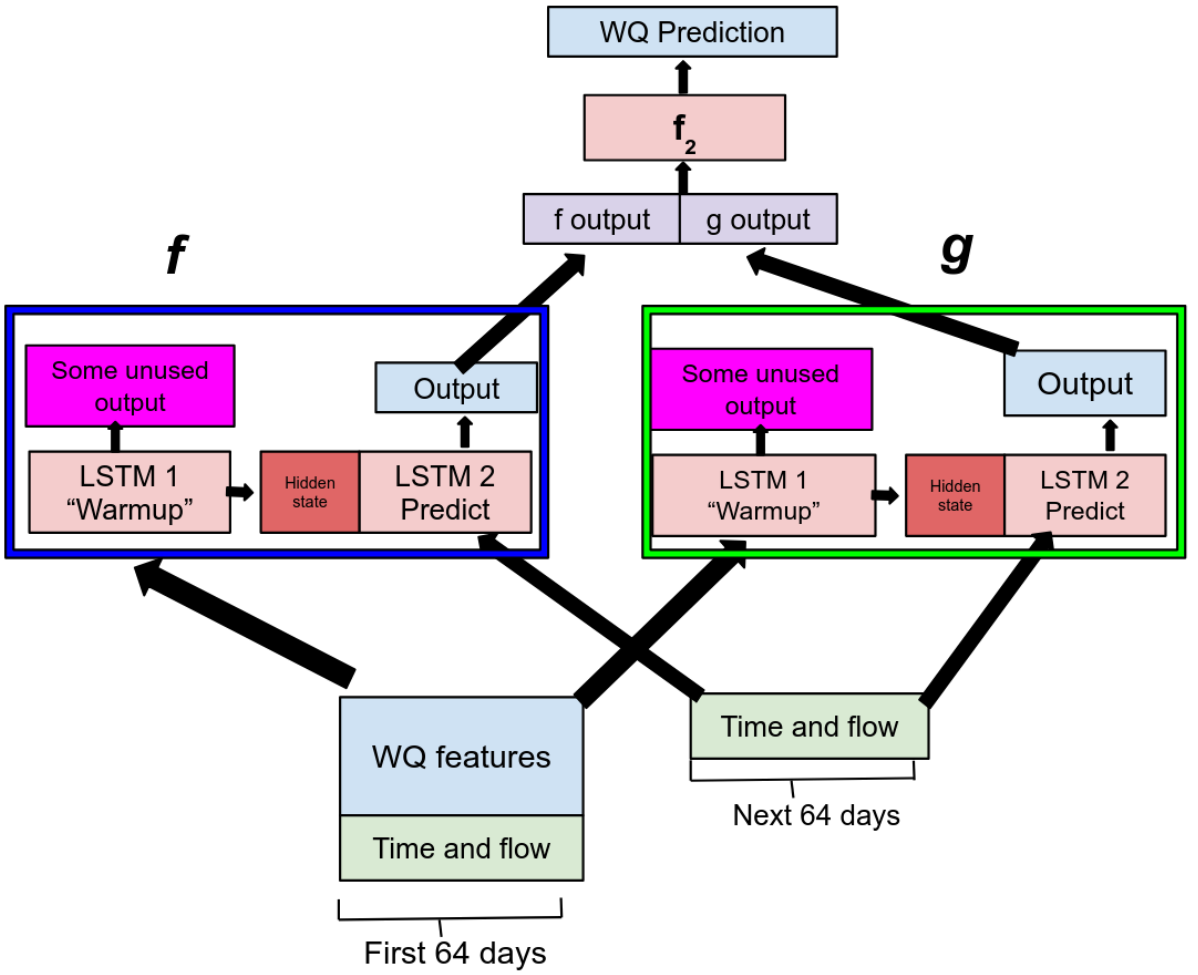


Figure 5.1: An alternative method to utilize f and g. Like the addition. Like the adding method, the input is fed into both f and g. However, the two outputs are concatenated and fed into an f_2 to obtain the final WQ prediction.

Chapter 6

Conclusion

This thesis investigated the use of an invariant-variant based DA method to better predict sparse WQ data by using source domains with WQ data that is less sparse. The 3 aspects that were investigated were model sizes, source domain composition, and method to combine f and g.

From investigating the size of the model, the optimal size is dependent on not only the source and target domains but also the relation between the domains. Specifically, the size of f needs to be adjusted depending on how much of the invariant can be transferred from the source domains to the target domain.

From investigating the choice of source domains, similarity as measured by the station's basin was not a significant factor in the final model performance. Rather, the number of source domains was the significant factor. However, once that number exceeded a certain threshold, further increasing the number of source domains or changing the similarity of the source domains did not affect performance.

Finally, the default method of combining f and g (feeding the output of f to g) performed significantly better than just adding the outputs of f and g. This confirms the relation of f and g is more than an addition in the tested scenarios.

These findings serve as guidelines for applying other similar DA techniques, particularly those that use submodels to capture the invariant and variant parts of the joint distribution. As well, this thesis encourages further investigations into addressing the data scarcity issue in WQ modelling by framing the problem as a DA problem.

References

[1] Nans Addor, Andrew J. Newman, Naoki Mizukami, and Martyn P. Clark. The camels data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, 21(10):5293–5313, October 2017.

[2] Godson Ebenezer Adjovu, Haroon Stephen, David James, and Sajjad Ahmad. Overview of the application of remote sensing in effective monitoring of water quality parameters. *Remote Sensing*, 15(7):1938, April 2023.

[3] Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, A. S. Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A. Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H. Al-Timemy, Ye Duan, Amjed Abdullah, Laith Farhan, Yi Lu, Ashish Gupta, Felix Albu, Amin Abbosh, and Yuantong Gu. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications, April 2023.

[4] E. Arandia, B. Eck, and S. McKenna. The effect of temporal resolution on the accuracy of forecasting models for total system demand. *Procedia Engineering*, 89:916–925, 2014.

[5] 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(1–2):151–175, October 2009.

[6] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *CoRR*, abs/1608.06019, 2016.

[7] Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. *arXiv*, 2022.

[8] Xilun Chen and Claire Cardie. Multinomial adversarial networks for multi-domain text classification, 2018.

[9] Michael F Chislock, Enrique Doster, Rachel A Zitomer, and Alan E Wilson. Eutrophication: causes, consequences, and controls in aquatic ecosystems. *Nature Education Knowledge*, 4(4):10, 2013.

[10] Dan C. Cireşan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and Jürgen Schmidhuber. High-performance neural networks for visual object classification. *arXiv*, 2011.

[11] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv*, 2019.

[12] Edson Tavares de Camargo, Fabio Alexandre Spanhol, Juliano Scholz Slongo, Marcos Vinicius Rocha da Silva, Jaqueline Pazinato, Adriana Vechai de Lima Lobo, Fábio Rizental Coutinho, Felipe Walter Dafico Pfrimer, Cleber Antonio Lindino, Marcio Seiji Oyamada, and Leila Droprinchinski Martins. Low-cost water quality sensors for iot: A systematic review. *Sensors*, 23(9):4424, April 2023.

[13] Xiao e Yang, Xiang Wu, Hu lin Hao, and Zhen li He. Mechanisms and assessment of water eutrophication. *Journal of Zhejiang University SCIENCE B*, 9, 3 2008.

[14] I Gusti Agung Putu Eryani, Made Widya Jayantari, and I Kadek Merta Wijaya. Sensitivity analysis in parameter calibration of the weap model for integrated water resources management in unda watershed. *Civil Engineering and Architecture*, 10(2):455–469, March 2022.

[15] Dapeng Feng, Kuai Fang, and Chaopeng Shen. Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. *Water Resources Research*, 56(9), September 2020.

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2015.

[17] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016.

[18] RK Garg, RJ Rao, and DN Saksena. Water quality and conservation management of ramsagar reservoir, datia, madhya pradesh. *Journal of Environmental Biology*, 30(5):909, 2009.

[19] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. *Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation*, page 597–613. Springer International Publishing, 2016.

[20] 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.

[21] Hoshin V. Gupta, Harald Kling, Koray K. Yilmaz, and Guillermo F. Martinez. Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, 377(1–2):80–91, October 2009.

[22] Hoshin V. Gupta, Harald Kling, Koray K. Yilmaz, and Guillermo F. Martinez. Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, 377(1):80–91, 2009.

[23] Hoshin Vijai Gupta and Harald Kling. On typical range, sensitivity, and normalization of mean squared error and nash-sutcliffe efficiency type metrics. *Water Resources Research*, 47(10), October 2011.

[24] Robert M. Hirsch, Douglas L. Moyer, and Stacey A. Archfield. Weighted regressions on time, discharge, and season (wrtds), with an application to chesapeake bay river inputs1. *JAWRA Journal of the American Water Resources Association*, 46(5):857–880, September 2010.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[26] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation, 2017.

[27] Vasileios C. Kapsalis and Ioannis K. Kalavrouziotis. *Eutrophication—A Worldwide Water Quality Issue*, pages 1–21. Springer International Publishing, Cham, 2021.

[28] Kul Khand and Gabriel B. Senay. Evaluation of streamflow predictions from lstm models in water- and energy-limited regions in the united states. *Machine Learning with Applications*, 16:100551, June 2024.

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[30] Harald Kling, Martin Fuchs, and Maria Paulin. Runoff conditions in the upper danube basin under an ensemble of climate change scenarios. *Journal of Hydrology*, 424-425:264–277, 2012.

[31] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy, 2017.

[32] A Dennis Lemly, Richard T Kingsford, and Julian R Thompson. Irrigated agriculture and wildlife conservation: conflict on a global scale. *Environmental management*, 25(5):485–512, 2000.

[33] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

[34] Peiyue Li and Jianhua Wu. Drinking water quality and public health. *Exposure and Health*, 11(2):73–79, 2019.

[35] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representation, 2018.

[36] Zijian Li, Ruichu Cai, Jiawei Chen, Yuguan Yan, Wei Chen, Keli Zhang, and Junjian Ye. Time-series domain adaptation via sparse associative structure alignment: Learning invariance and variance, 2022.

[37] M. F. McCabe, M. Rodell, D. E. Alsdorf, D. G. Miralles, R. Uijlenhoet, W. Wagner, A. Lucieer, R. Houborg, N. E. C. Verhoest, T. E. Franz, J. Shi, H. Gao, and E. F. Wood. The future of earth observation in hydrology. *Hydrology and Earth System Sciences*, 21(7):3879–3914, 2017.

[38] Matthew F. McCabe, Matthew Rodell, Douglas E. Alsdorf, Diego G. Miralles, Remko Uijlenhoet, Wolfgang Wagner, Arko Lucieer, Rasmus Houborg, Niko E. C. Verhoest, Trenton E. Franz, Jiancheng Shi, Huilin Gao, and Eric F. Wood. The future of earth observation in hydrology. *Hydrology and Earth System Sciences*, 21(7):3879–3914, July 2017.

[39] Angelika M. Meyer, Christina Klein, Elisabeth Fünfrocken, Ralf Kautenburger, and Horst P. Beck. Real-time monitoring of water quality to identify pollution pathways

in small and middle scale rivers. *Science of The Total Environment*, 651:2323–2333, February 2019.

[40] Yuri G Motovilov, Lars Gottschalk, Kolbjørn Engeland, and Allan Rodhe. Validation of a distributed hydrological model against spatial observations. *Agric. For. Meteorol.*, 98-99:257–277, December 1999.

[41] J.E. Nash and J.V. Sutcliffe. River flow forecasting through conceptual models part i — a discussion of principles. *Journal of Hydrology*, 10(3):282–290, April 1970.

[42] NCWQR. Heidelberg tributary loading program (htlp) dataset, 2022.

[43] Silvia E. Newell, Timothy W. Davis, Thomas H. Johengen, Duane Gossiaux, Ashley Burtner, Danna Palladino, and Mark J. McCarthy. Reduced forms of nitrogen are a driver of non-nitrogen-fixing harmful cyanobacterial blooms and toxicity in lake erie. *Harmful Algae*, 81, 1 2019.

[44] Nicholas A. Odoni and Stuart N. Lane. Knowledge-theoretic models in hydrology. *Progress in Physical Geography: Earth and Environment*, 34(2):151–171, March 2010.

[45] Government of Ontario. Provincial (stream) water quality monitoring network, 2024.

[46] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation, 2019.

[47] Trung Phung, Trung Le, Long Vuong, Toan Tran, Anh Tran, Hung Bui, and Dinh Phung. On learning domain-invariant representations for transfer learning with multiple sources, 2021.

[48] S. Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations*, 2016.

[49] JongCheol Pyo, Yakov Pachepsky, Soobin Kim, Ather Abbas, Minjeong Kim, Yong Sung Kwon, Mayzonee Ligaray, and Kyung Hwa Cho. Long short-term memory models of water quality in inland water environments. *Water Research X*, 21:100207, December 2023.

[50] Jiuhui Qu and Maohong Fan. The current state of water quality and technology development for water pollution control in china. *Critical reviews in environmental science and technology*, 40(6):519–560, 2010.

[51] Chuan-Xian Ren, Yong-Hui Liu, Xi-Wen Zhang, and Ke-Kun Huang. Multi-source unsupervised domain adaptation via pseudo target domain. *IEEE Transactions on Image Processing*, 31:2122–2135, 2022.

[52] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[53] Jerry C. Ritchie and Frank R. Schiebe. *Water Quality*, pages 287–303. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[54] Dale M. Robertson, Laura E. Hubbard, David L. Lorenz, and Daniel J. Sullivan. A surrogate regression approach for computing continuous loads for the tributary nutrient and sediment monitoring program on the great lakes. *Journal of Great Lakes Research*, 44(1):26–42, February 2018.

[55] Robert L. Runkel, Charles G. Crawford, and Timothy A. Cohn. *Load estimator (LOADEST): a FORTRAN program for estimating constituent loads in streams and rivers*. USGS, 2004.

[56] Gourab Kumer Saha, Farshid Rahmani, Chaopeng Shen, Li Li, and Raj Cibin. A deep learning-based novel approach to generate continuous daily stream nitrate concentration for nitrate data-sparse watersheds. *Science of The Total Environment*, 878:162930, June 2023.

[57] Gourab Kumer Saha, Farshid Rahmani, Chaopeng Shen, Li Li, and Raj Cibin. A deep learning-based novel approach to generate continuous daily stream nitrate concentration for nitrate data-sparse watersheds. *Science of The Total Environment*, 878:162930, 2023.

[58] Muhammad Ayaz Shaikh. Water conservation in textile industry. *Pakistan Textile Journal*, 58(11):48–51, 2009.

[59] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation, 2015.

[60] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation, 2016.

[61] Ryuichi Tadano, Adithya Kumar Pediredla, Kaushik Mitra, and Ashok Veeraraghavan. Spatial phase-sweep: Increasing temporal resolution of transient imaging using a light source array. *arXiv*, 2015.

[62] Nima Tajbakhsh, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016.

[63] Nana Terangna Bukit. Water quality conservation for the citarum river in west java. *Water Science and Technology*, 31(9):1–10, 1995. Water Conservation and Pollution Control.

[64] Jamie Towner, Hannah L. Cloke, Ervin Zsoter, Zachary Flamig, Jannis M. Hoch, Juan Bazo, Erin Coughlan de Perez, and Elisabeth M. Stephens. Assessing the performance of global hydrological models for capturing peak river flows in the amazon basin. *Hydrology and Earth System Sciences*, 23(7):3057–3080, July 2019.

[65] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks, 2015.

[66] Marija Vella and Joao F. C. Mota. Robust single-image super-resolution via cnns and tv-tv minimization. *IEEE Transactions on Image Processing*, 30, 2021.

[67] Shunnan Wang, Min Gao, Huan Wu, Fengji Luo, Feng jiang, and Liang Tao. Many-to-many: Domain adaptation for water quality prediction. *Applied Soft Computing*, page 112381, October 2024.

[68] Yuanyuan Wang, Jian Zhou, Kejia Chen, Yunyun Wang, and Linfeng Liu. Water quality prediction method based on lstm neural network. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. IEEE, November 2017.

[69] Garrett Wilson, Janardhan Rao Doppa, and Diane J. Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data, 2020.

[70] Wayne A. Wurtsbaugh, Hans W. Paerl, and Walter K. Dodds. Nutrients, eutrophication and harmful algal blooms along the freshwater to marine continuum. *WIREs Water*, 6(5), August 2019.

[71] Rui Xiong, Yi Zheng, Nengwang Chen, Qing Tian, Wei Liu, Feng Han, Shijie Jiang, Mengqian Lu, and Yan Zheng. Predicting dynamic riverine nitrogen export in unmonitored watersheds: Leveraging insights of ai from data-rich regions. *Environmental Science & Technology*, 56(14):10530–10542, 2022. PMID: 35772808.

[72] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift, 2018.

[73] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2016.

[74] Zhen Zhang, Mianzhi Wang, Yan Huang, and Arye Nehorai. Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3437–3445, 2018.

[75] Wei Zhi, Alison P Appling, Heather E Golden, Joel Podgorski, and Li Li. Deep learning for water quality. *Nat Water*, 2:228–241, March 2024.

[76] Wei Zhi, Dapeng Feng, Wen-Ping Tsai, Gary Sterle, Adrian Harpold, Chaopeng Shen, and Li Li. From hydrometeorology to river water quality: Can a deep learning model predict dissolved oxygen at the continental scale? *Environmental Science & Technology*, 55(4):2357–2368, 2021. PMID: 33533608.

[77] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5989–5996, July 2019.

[78] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources, July 2019.

APPENDICES

Appendix A

Extra Information on Water Quality Data

A.1 Pearson Correlation Coefficient between Flow and the Other Water Quality Paramters

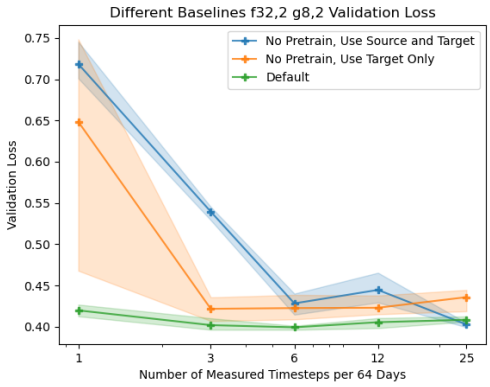
Name	TSS	TP	SRP	NO_3	TKN	CL	SO_4	SI	COND
Wolf	0.39117	0.24796	-0.08938	0.39594	0.46935	-0.27362	-0.36201	-0.17782	-0.17728
Huron River	0.52227	0.54965	0.34850	0.20966	0.52635	-0.34166	-0.47020	0.18059	-0.45660
Muskingum	0.58581	0.48233	-0.25210	0.33035	0.21439	-0.65358	-0.60710	0.37152	-0.47593
Shallow Run	0.60105	0.23568	0.00314	0.19206	0.27084	-0.16161	-0.32997	0.09073	-0.28685
Scioto River @ Chillicothe	0.43877	-0.03747	-0.36138	0.03639	0.29013	-0.43056	-0.64773	0.08802	-0.13755
Chickasaw	0.51306	0.38363	0.14365	0.20698	0.18022	-0.27313	-0.31176	0.22275	-0.19287
Raisin	0.26739	0.41304	0.22906	0.47834	0.34808	-0.45955	-0.35711	-0.06316	0.00378
Maumee	0.66014	0.69481	0.22693	0.29335	0.36372	-0.44401	-0.21972	0.23048	0.01762
Potato Run	0.41530	0.54620	0.21260	0.22669	0.37055	-0.34965	-0.30312	0.02497	-0.37887
Grand River	0.45909	0.45714	-0.09162	0.21208	0.38027	-0.20271	-0.27310	0.04202	-0.01071
Portage	0.64852	0.55463	0.16772	0.16718	0.53593	-0.43972	-0.58229	0.29240	-0.35826
Tiffin River	0.21669	0.43254	0.33568	0.40676	0.40193	-0.55479	-0.63304	-0.05346	-0.30148
Blanchard River	0.57425	0.40870	-0.17479	-0.23879	0.52111	-0.43391	-0.54445	0.15626	-0.14776
Honey Creek	0.42748	0.60226	0.28125	0.12997	0.36946	-0.37588	-0.21067	0.02524	0.02572
Beaver	-0.03884	-0.14193	-0.06072	0.28012	-0.06323	-0.15906	0.04908	-0.09029	-0.02758
Coldwater	0.19714	0.27036	0.12532	0.10994	0.10033	-0.10632	-0.16933	0.07164	-0.12140
S. Turkeyfoot	0.68189	0.79251	0.41926	0.30564	0.70385	-0.46594	-0.50219	0.24321	-0.51924
Rock Creek	0.48765	0.51367	0.24472	0.13260	0.44175	-0.29996	-0.28241	-0.07529	0.02590
Vermillion River	0.42438	0.47811	0.39582	0.11778	0.42473	-0.20009	-0.34814	0.12392	-0.17443
Sandusky River	0.43415	0.55370	0.38475	0.22595	0.38828	-0.42343	-0.35839	0.07570	0.03976
Great Miami	0.56749	0.19839	-0.34718	0.18979	0.40179	-0.54353	-0.59709	0.06366	-0.20042
Cuyahoga	0.42544	0.26117	-0.31374	-0.59254	0.27878	-0.12990	-0.34414	-0.02415	0.00356
West	0.65506	0.53235	0.09981	0.27386	0.50761	-0.21618	-0.32614	0.11198	-0.37438
Scioto River @ Piketon	0.41568	0.07292	-0.37984	0.05152	0.13548	-0.58873	-0.69234	0.13189	-0.49159
Lost	0.54579	0.55216	0.30773	0.05641	0.24528	-0.23190	-0.33079	-0.09969	-0.15416

Table A.1: PCC between flow and the other 9 WQ parameters for all 25 stations. Note the values differ among stations

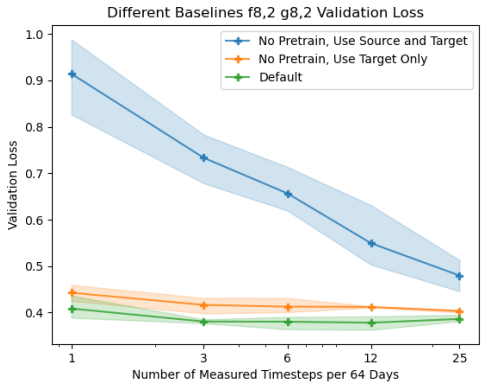
Appendix B

Extra Plots

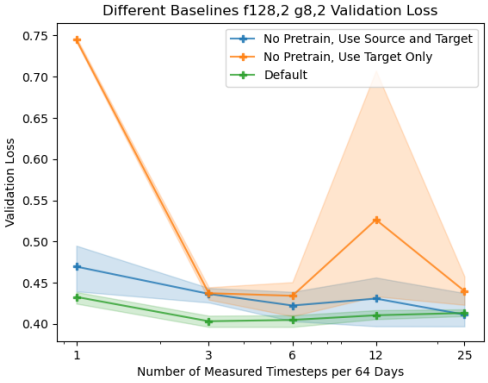
B.1 Validity of the DA Method



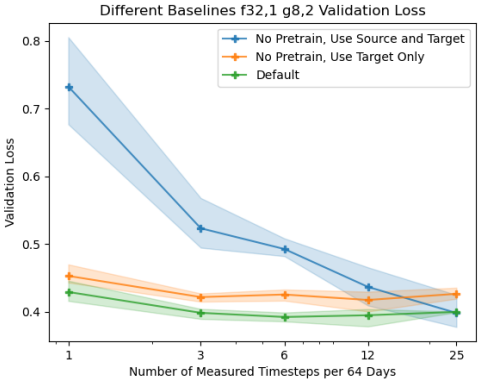
(a) f32,2 g8,2



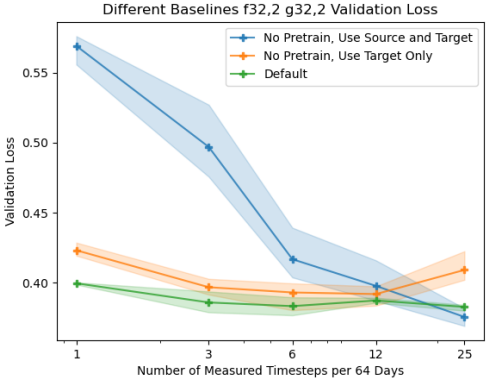
(b) f8,2 g8,2



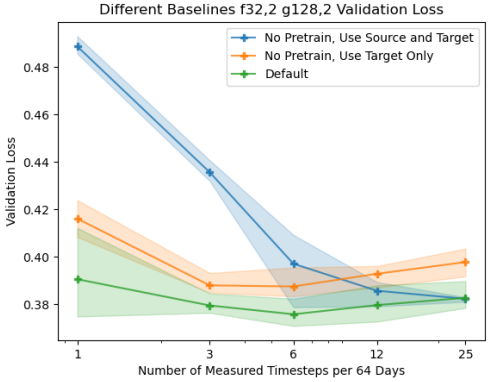
(c) f128,2 g8,2



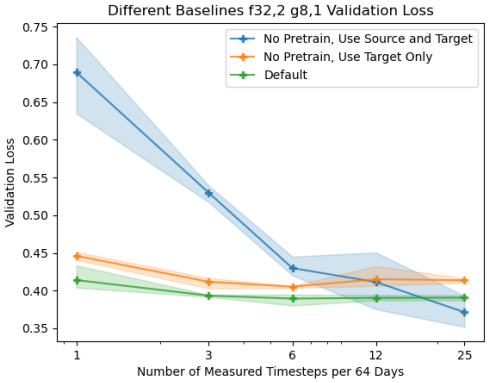
(d) f32,1 g8,2



(e) f32,2 g32,2



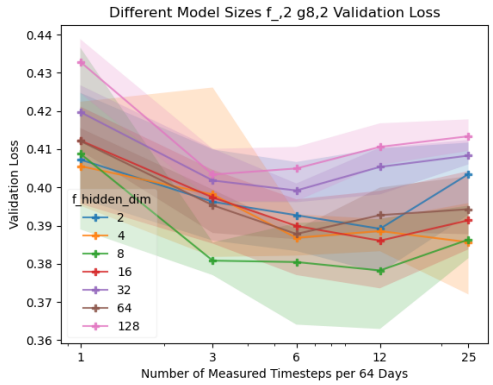
(f) f32,2 g128,2



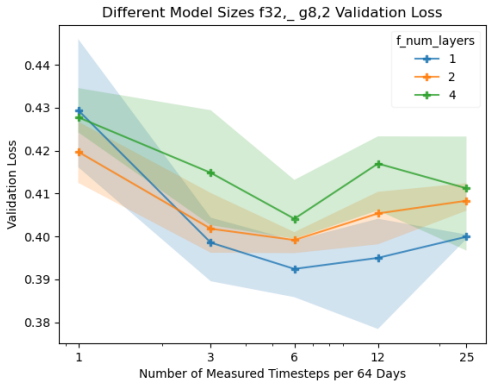
(g) f32,2 g8,1

Figure B.1: Comparing the validation loss of the default DA method versus no pretraining (Baseline), at various sparsities and model size.

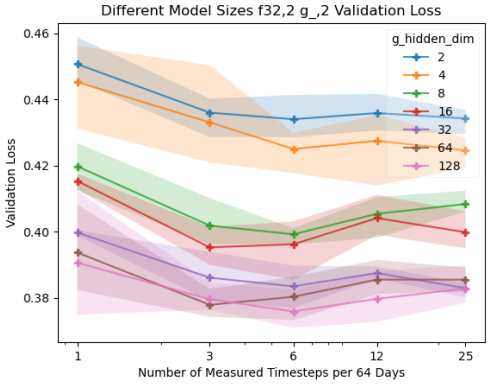
B.2 Model Size



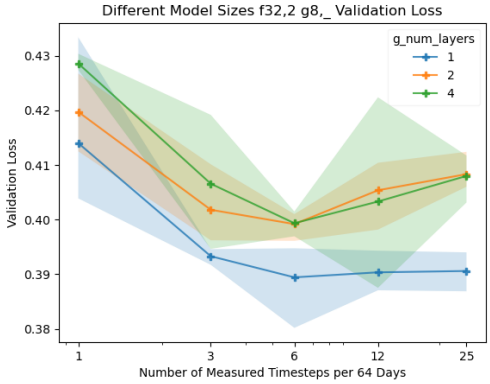
(a) f_, 2 g8,2



(b) f32, _ g8,2

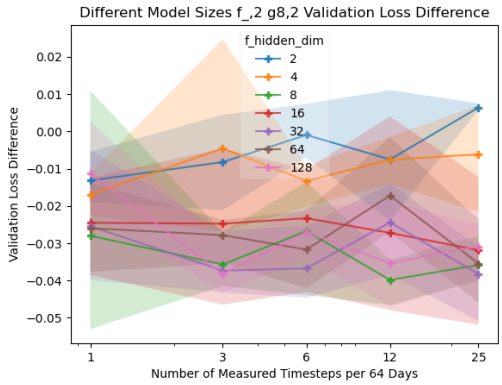


(c) f32, 2 g_,2

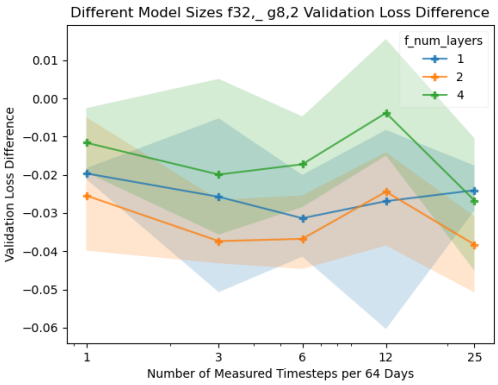


(d) f32, 2 g8,_

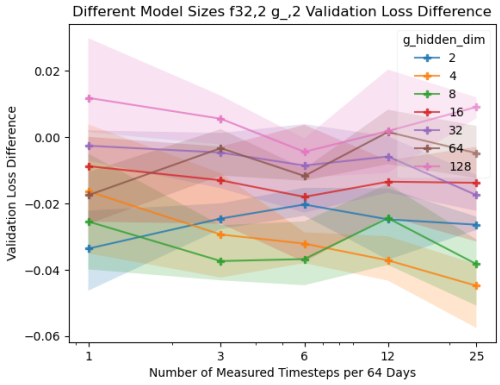
Figure B.2: Validation loss of various sizes of models at different resolutions. The result of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.



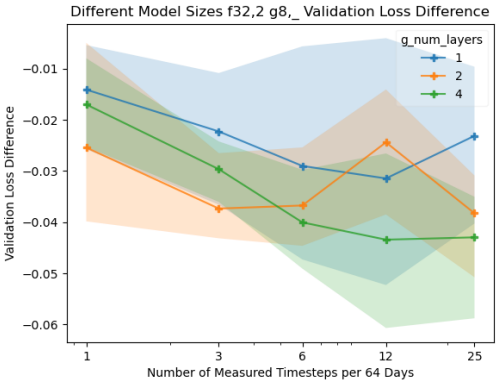
(a) f_, 2 g8,2



(b) f32, _ g8,2



(c) f32, 2 g_,2

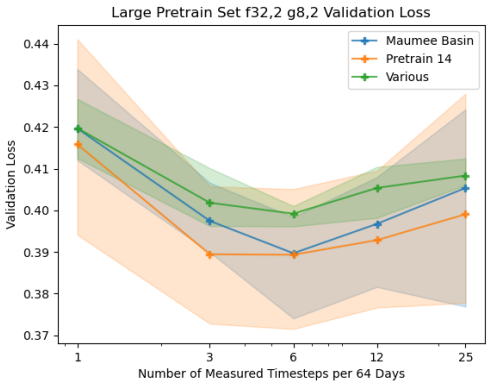


(d) f32, 2 g8,_

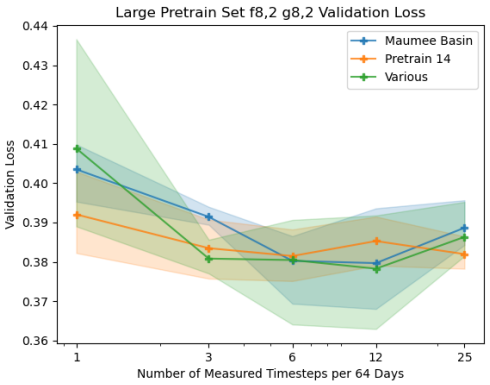
Figure B.3: Validation loss difference between freezing and not freezing f at different resolutions. The result of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

B.3 Choice of Source Domains

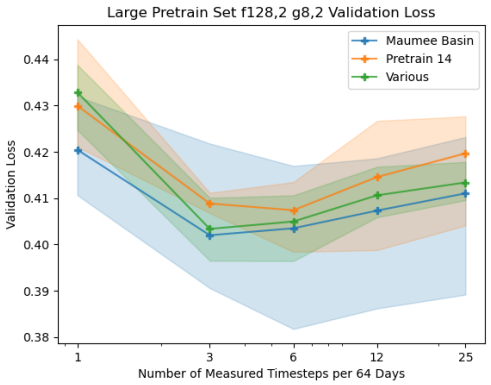
B.3.1 Investigating Sets With Multiple Domains



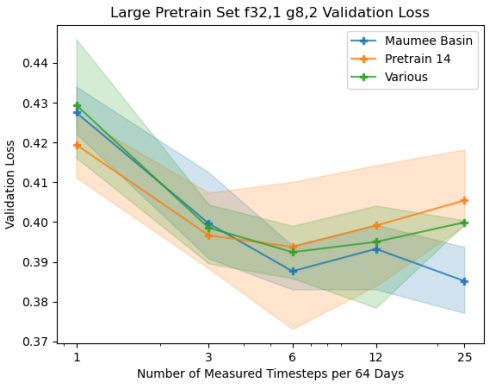
(a) f32,2 g8,2



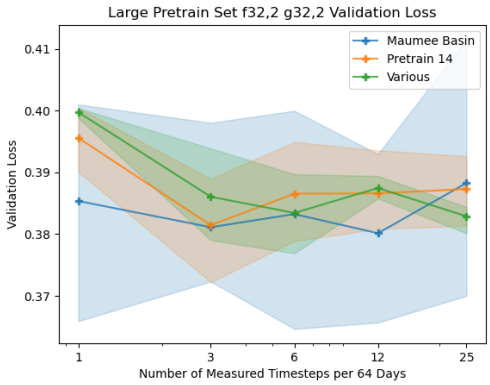
(b) f8,2 g8,2



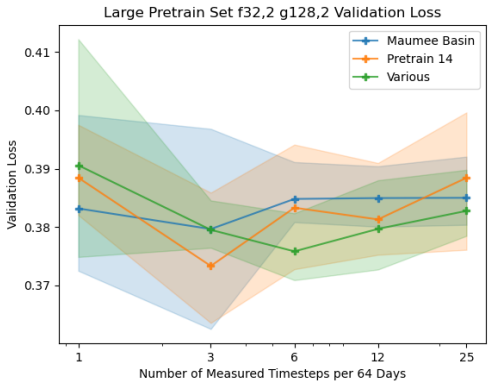
(c) f128,2 g8,2



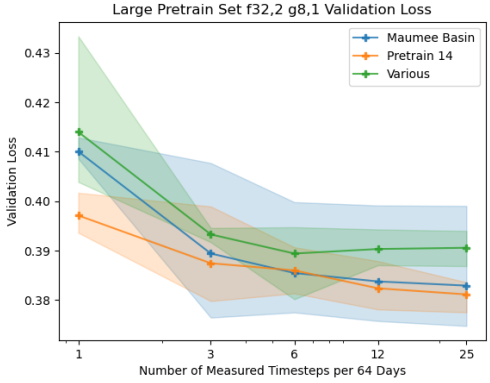
(d) f32,1 g8,2



(e) f32,2 g32,2

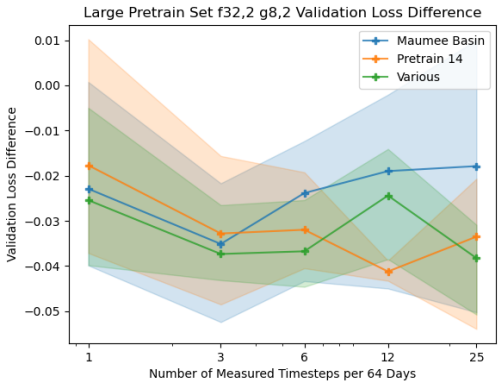


(f) f32,2 g128,2

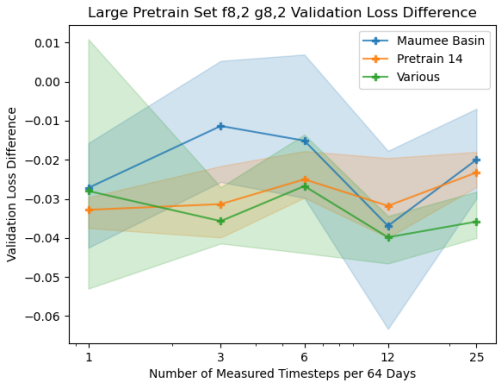


(g) f32,2 g8,1

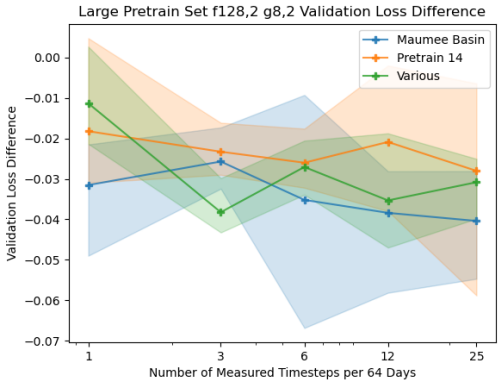
Figure B.4: Validation loss of sets with multiple source domains, for all the various sizes of models, and at different resolutions. The result of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.



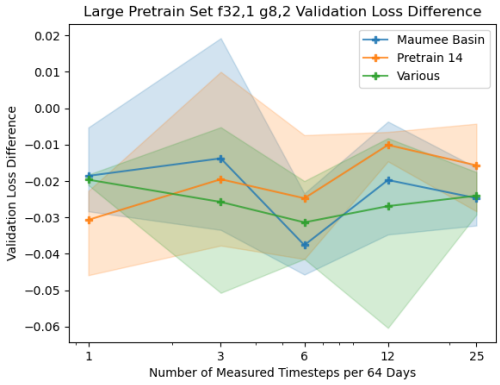
(a) f32,2 g8,2



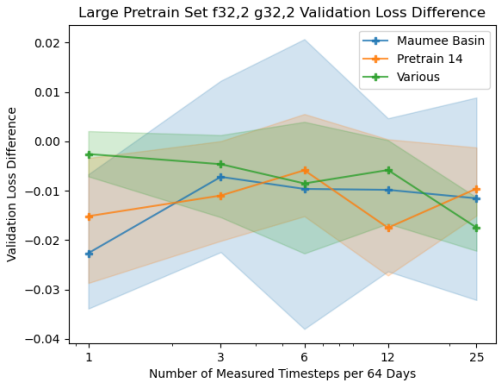
(b) f8,2 g8,2



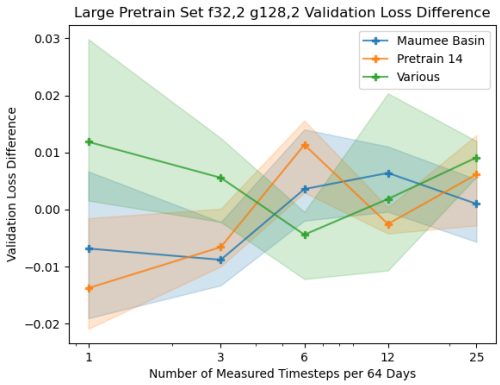
(c) f128,2 g8,2



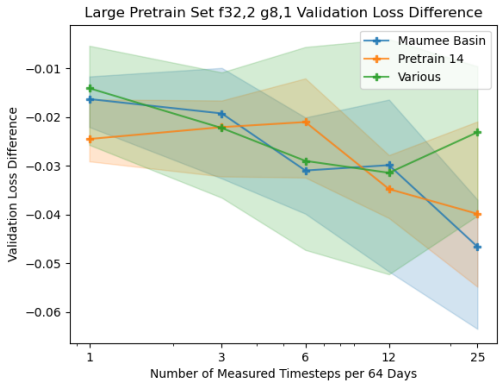
(d) f32,1 g8,2



(e) f32,2 g32,2



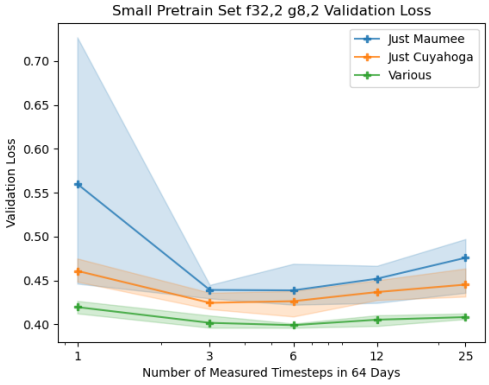
(f) f32,2 g128,2



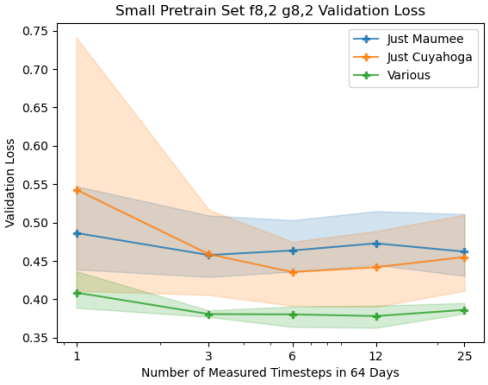
(g) f32,2 g8,1

Figure B.5: Validation loss difference of sets with multiple source domains, for all the tested sizes of models, and at different resolutions. The result of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

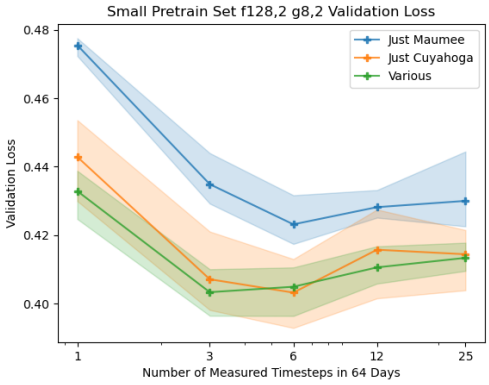
B.3.2 Investigating Sets With a Single Domain



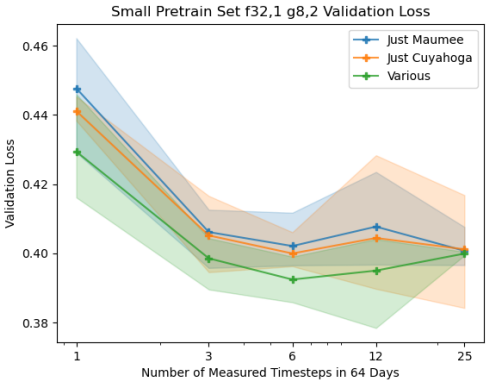
(a) f32,2 g8,2



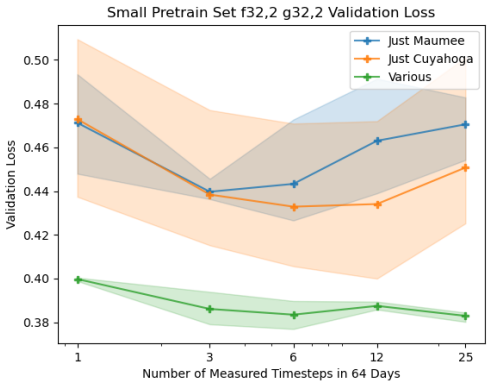
(b) f8,2 g8,2



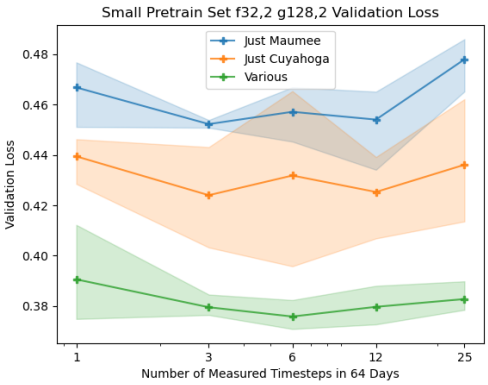
(c) f128,2 g8,2



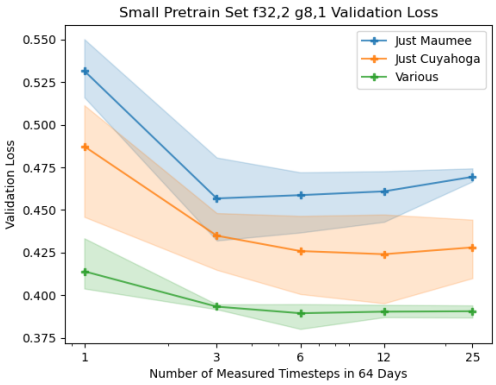
(d) f32,1 g8,2



(e) f32,2 g32,2

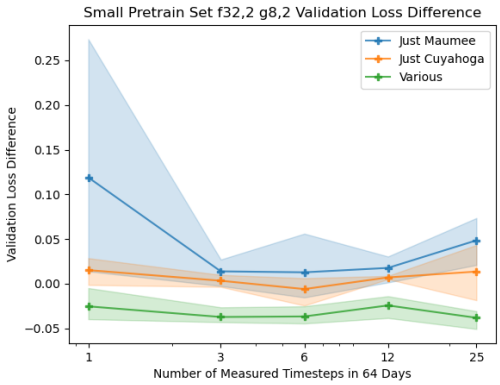


(f) f32,2 g128,2

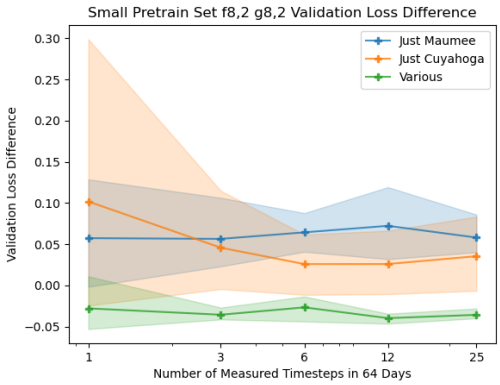


(g) f32,2 g8,1

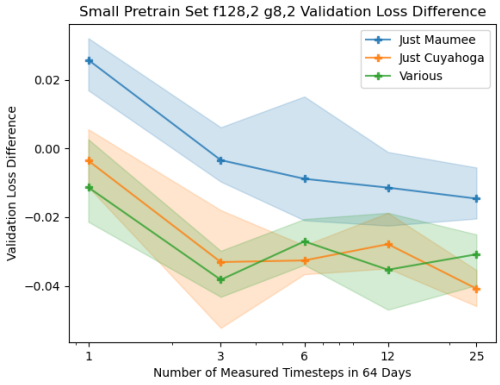
Figure B.6: Validation loss of sets with multiple source domains, for all the various sizes of models, and at different resolutions. The result of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.



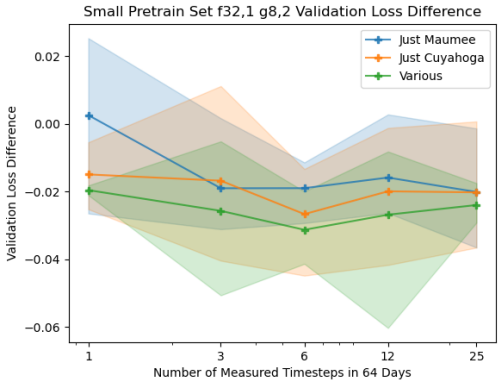
(a) f32,2 g8,2



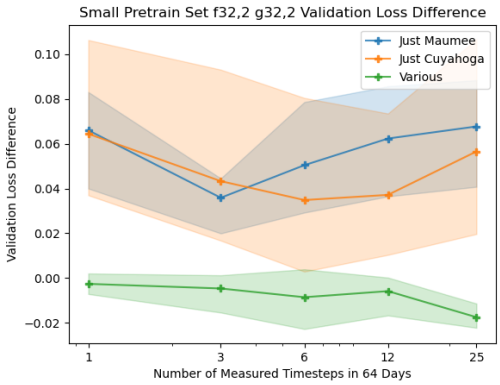
(b) f8,2 g8,2



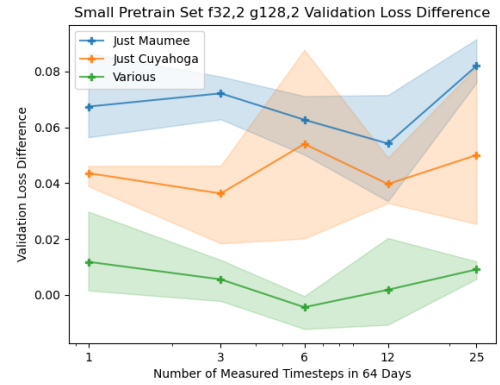
(c) f128,2 g8,2



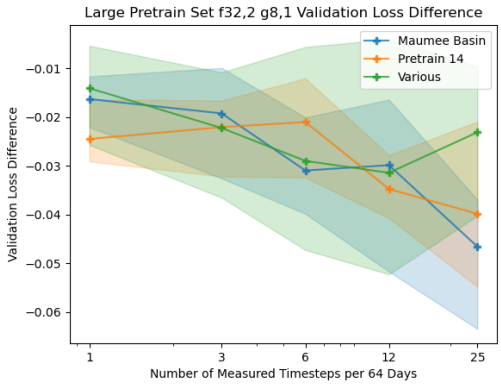
(d) f32,1 g8,2



(e) f32,2 g32,2



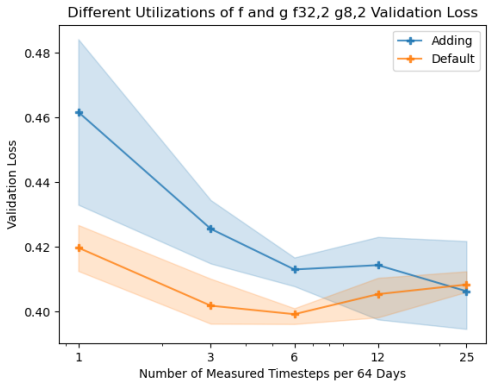
(f) f32,2 g128,2



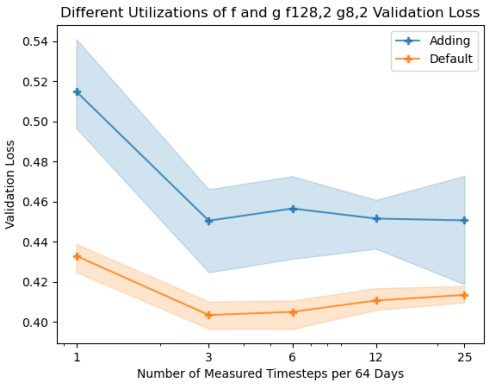
(g) f32,2 g8,1

Figure B.7: Validation loss difference of sets with multiple source domains, for all the tested sizes of models, and at different resolutions. The result of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.

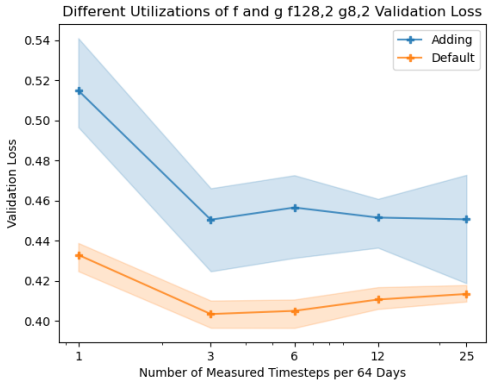
B.4 Method to Utilize f and g



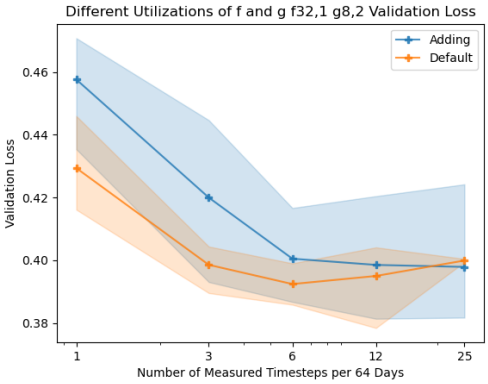
(a) f32,2 g8,2



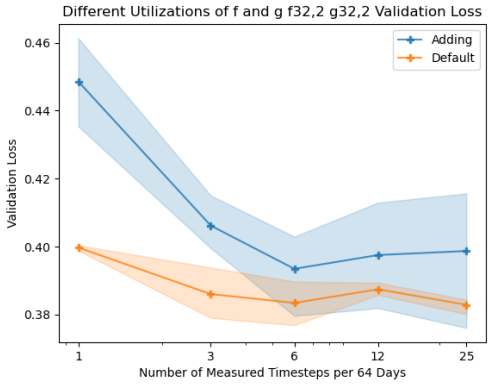
(b) f8,2 g8,2



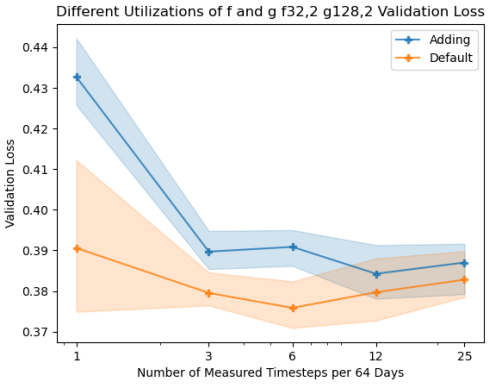
(c) f128,2 g8,2



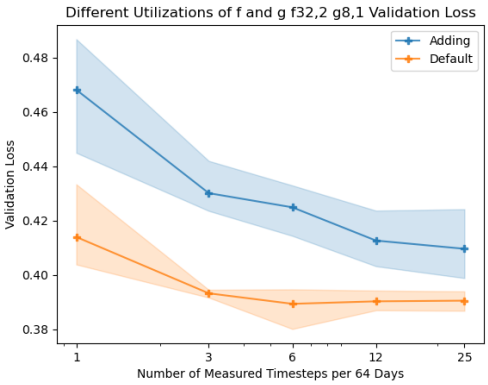
(d) f32,1 g8,2



(e) f32,2 g32,2

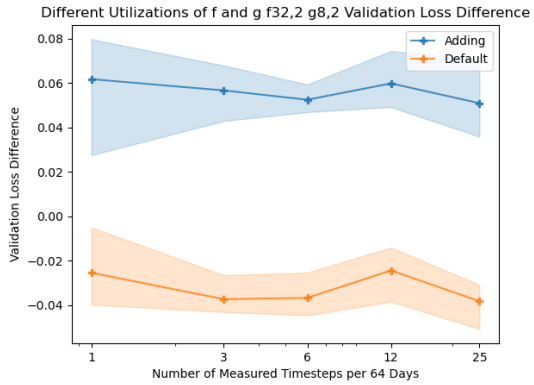


(f) f32,2 g128,2

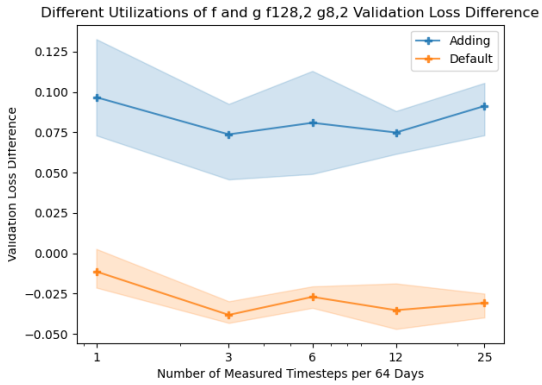


(g) f32,2 g8,1

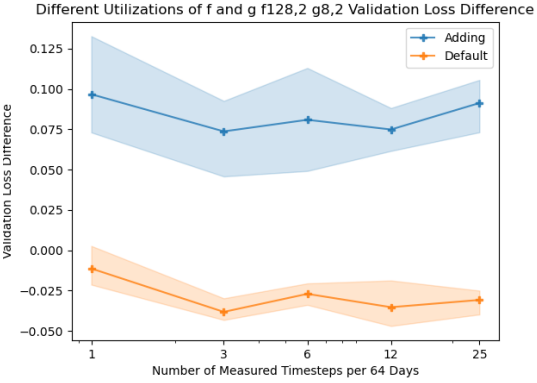
Figure B.8: Validation loss between the two methods of utilizing f and g, for all the tested sizes of models, and at different numbers of measured timestamps. The result of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.



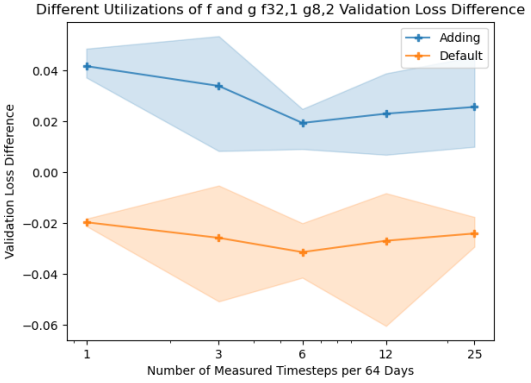
(a) f32,2 g8,2



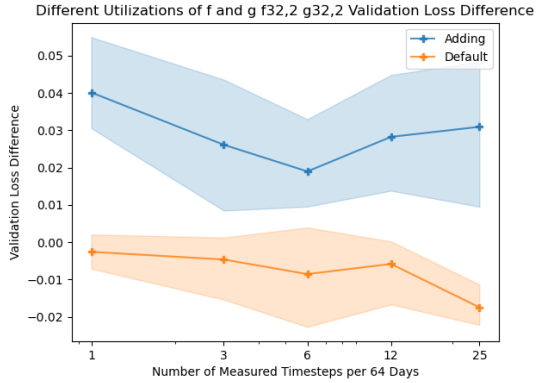
(b) f8,2 g8,2



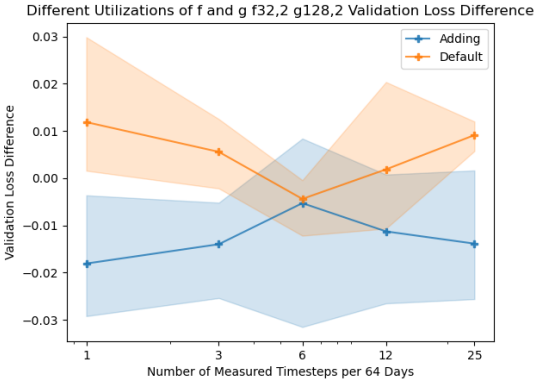
(c) f128,2 g8,2



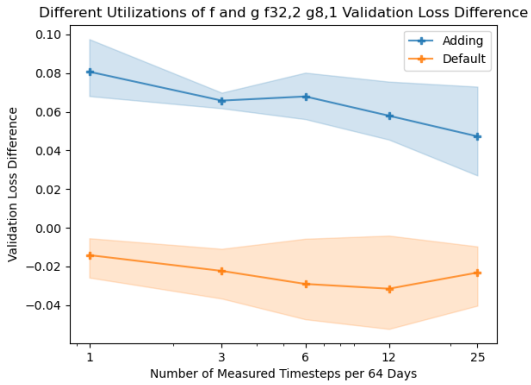
(d) f32,1 g8,2



(e) f32,2 g32,2



(f) f32,2 g128,2



(g) f32,2 g8,1

Figure B.9: Validation loss difference between the two methods of utilizing f and g, or all the tested sizes of models, and at different numbers of measured timestamps. The result of 3 runs with different seeds were averaged and the envelope represents the maximum and minimum validation loss.