Generative Flow Models in Weight Space for Detecting Covariate Shifts

Daniel Saragih[1 2], **Deyu Cao**[3*], **Tejas Balaji**[4*]

[1]Queen's University, [2]Vector Institute, [3]University of Tokyo, [4]University of Toronto
daniel.saragih@queensu.ca, deyu.cao@mail.utoronto.ca, tejas.balaji@mail.utoronto.ca

Abstract

Flow-based generative modeling provides a powerful framework for reasoning about uncertainty in weight space. In this work, we explore model uncertainty and distributional anomalies through weight space learning, where a generative meta-model learns a distribution over neural network parameters that achieve comparable performance. Leveraging flow matching, we capture the geometry of weight space to enable conditional generation and reward-guided adaptation, allowing the weight distribution to evolve in response to shifts in the data. Experiments demonstrate that this approach not only captures in-distribution models but also adapts effectively under distribution shift. Finally, we show that this adaptation provides a practical tool for detecting harmful covariate shifts on limited data, outperforming comparable methods.

1 Introduction

In machine learning, multiple distinct models can achieve comparable performance on a given task, despite differing substantially in their internal representations or parameterizations. This *multiplicity* underscores a key source of model uncertainty: if many models fit the data equally well, which of them will remain reliable when encountering anomalous or shifted inputs? Addressing this question is central to building dependable systems for two reasons. First, it guides model selection under uncertainty. Second, it shapes how we approach anomaly detection since the impact of anomalous data depends not only on the data itself but also on the inductive biases of the model architecture.

In this work, we approach model multiplicity through the lens of probability distributions in weight space. We introduce a generative framework that learns distributions over neural network parameters using flow matching (FM) (Albergo and Vanden-Eijnden 2023; Lipman et al. 2023; Liu, Gong, and Liu 2023). This formulation provides a geometric and probabilistic foundation for reasoning about model uncertainty and adaptation. When the data distribution changes, instead of retraining from scratch, we can tilt or fine-tune the learned weight distribution in response to a reward or cost signal—an approach closely related to stochastic optimal control.

In this work, we make preliminary steps toward understanding model uncertainty and distribution shift by using generative flow models. Our contributions include:

1. We develop a flow-based framework for generating neural network weights that incorporates theoretical insights from continuous dynamics. The resulting models match or exceed conventionally trained networks on in-distribution tasks and can be conditioned on context data to retrieve pre-trained weights.
2. We demonstrate a fine-tuning mechanism which enables efficient reward-guided adaptation of weight-space distributions in response to test-time shifts.
3. We further demonstrate that these reward-tilted distributions can serve as effective signals for detecting harmful covariate shifts on limited data.

2 Background & Related Works

Conditional flow models. Chen et al. (2019) first introduced continuous normalizing flows as an effective data generation process through modeling dynamics. Simulation-free methods improve on this concept by simplifying the training objective (Albergo and Vanden-Eijnden 2023; Lipman et al. 2023; Liu, Gong, and Liu 2023). Following the formulation of Lipman et al. (2023), given random variables $\bar{\mathbf{x}}_0 \sim p_0$ and $\bar{\mathbf{x}}_1 \sim p_1$ a data distribution, define a reference flow $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_t)_{t\in[0,1]}$ where $\bar{\mathbf{x}}_t = \beta_t \bar{\mathbf{x}}_0 + \alpha_t \bar{\mathbf{x}}_1$ with the constraint that $\alpha_0 = \beta_1 = 0$ and $\alpha_1 = \beta_0 = 1$. The aim of flow modeling is to learn a path $\mathbf{x} = (\mathbf{x}_t)_{t\in[0,1]}$ which has the same marginal distribution as $\bar{\mathbf{x}}$. To make this a feasible task, we describe this process as an ODE: $d\mathbf{x}_t = v(\mathbf{x}_t, t)dt$ where $\mathbf{x}_0 \sim \mathcal{N}(0, \boldsymbol{I})$. Training proceeds by first parameterizing $v(\mathbf{x}_t, t)$ by a neural network θ and matching the reference flow velocity, i.e. $u(\mathbf{x}_t, t) := \frac{d}{dt}\bar{\mathbf{x}}_t$. This would, however, be an unfeasible training objective, therefore, we condition on samples from the distribution $\mathbf{x}_1 \sim p_1$ and train

$$\mathcal{L}_{\text{cfm}}(\theta) = \mathbb{E}_{\text{t},\mathbf{x}_1,\mathbf{x}_t} ||v_\theta(\mathbf{x}_t, t) - u(\mathbf{x}_t, t \mid \mathbf{x}_1)||. \quad (1)$$

Lipman et al. (2023) proved that this loss produces the same gradients as the marginal loss, thus optimizing it will result in convergence to the reference $u(\mathbf{x}_t, t)$. Moreover, we can always marginalize an independent conditioning variable $\boldsymbol{y}$ on v_θ, u—this will be our context conditioning vector.

[*]These authors contributed equally.

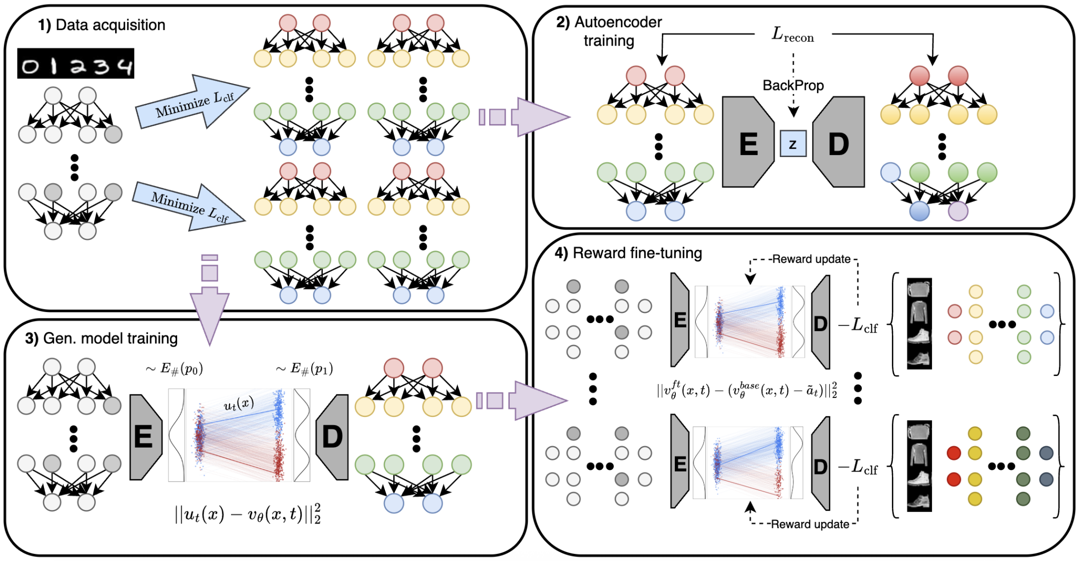

Figure 1: **Example unconditional pipeline. (1)** Base model pre-training, shown here on MNIST, producing checkpoints across epochs. **(2) Optional:** variational autoencoder training with a weight-space reconstruction objective. **(3)** Generative meta-model training; here we illustrate *unconditional* N$\mathcal{M}$-CFM w/ (trained) VAE (our default N$\mathcal{M}$-CFM is on weight space directly) using the weight initialization from **(1)** as p_0. **(4) Optional:** reward fine-tuning via adjoint matching where $r(\cdot) = -L_{\text{clf}}(\boldsymbol{X}_{\text{FashionMNIST}}; \cdot)$, steering the *trained* meta-model towards generating FashionMNIST classifiers.

Harmful covariate shift detection. Covariate shifts refers to changes in the test data distribution $p_{\text{test}}(x)$ as compared to the training distribution $p_{\text{train}}(x)$ while the relation between inputs and outputs remain fixed, i.e. $p_{\text{test}}(y|x) = p_{\text{train}}(y|x)$. Importantly, we do not require labels to determine this shift, thus it is practical to do so in a standard deployment setting. Prior work in this domain include deep kernel MMD (Liu et al. 2020), H-divergences (Zhao et al. 2022), and Detectron (Ginsberg, Liang, and Krishnan 2023). As Detectron requires minimal tuning and is most performant in low-data regimes ($N < 100$ samples), we emphasize the use of this approach. In particular, Detectron (Ginsberg, Liang, and Krishnan 2023) builds off of *selective classification*—building classifiers that accept/reject test data depending on closeness to the training distribution—and *PQ-learning* (Goldwasser et al. 2020) that extends the conventional theory of PAC learning to arbitrary test distributions by employing selective classification. The main idea considers the generalization set $\mathcal{R}$ of a classifier f_θ and samples $\mathcal{Q}$ from an unknown distribution. The strategy is to fine-tune constrained disagreement classifiers (CDCs) to agree with f_θ on $\mathcal{R}$ but disagree on $\mathcal{Q}$. If $\mathcal{Q} \subset \mathcal{R}$, then it will be difficult to disagree on $\mathcal{Q}$, but if the CDCs behave inconsistently on $\mathcal{Q}$, that suggests a covariate shift. Notably, this method is sample-efficient, agnostic to classifier architecture, and may be used in tests of statistical significance.

3 Methods

We describe the components of our approach below and leave more details to Appendices A, C, E. Throughout, we use the N$\mathcal{M}$- prefix[1] to denote our methods, e.g. N$\mathcal{M}$-CFM to denote conditional flow matching. Our framework is designed to be modular, with different components that can be instantiated in various ways. In this work, we prioritize simplicity in order to highlight the generative framework and the proposed reward fine-tuning mechanism. Figure 1 provides an example of how these components connect.

Modeling drift on weight space

The continuity equation on neural network parameters. For the purpose of our analysis, let us restrict our view to neural networks that are optimized by gradient descent (GD) algorithms to minimize a loss $\mathcal{L}(\theta_k) := d(\mathcal{M}_\theta(\boldsymbol{X}) - \boldsymbol{Y})$, where $\mathcal{M}_\theta$ is a neural network parameterized by trainable weights $\theta \in \mathbb{R}^p$, $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ are inputs, $\boldsymbol{Y} \in \mathbb{R}^{N \times c}$ are labels, and d is some differentiable distance function, such as cross-entropy. To minimize via GD, parameter updates are done by $\theta_{k+1} = \theta_k - \alpha \nabla \mathcal{L}(\theta_k)$ given some learning rate $\alpha > 0$. Taking the learning rate to zero, we can view parameter evolution as a *gradient flow*. For simplicity, we assume that the updates are deterministic (contrary to stochastic gradient descent which randomly selects training batches), and defer to App. B an approach incorporating stochasticity via Schrödinger bridges. For now, the sole source of randomness is the initialization $\theta_0 \sim p_0$. Within this setting, we can write down a continuity equation. In later sections, we show how this result underpins the choice of modeling framework.

Theorem 1 (Informal; follows Ch. 8.3 of Santambrogio (2015))**.** *Let $\theta_0 \sim p_0$ be initialized network parameters and*

[1] Short for *neural manifold*, a reference to data manifolds (Bengio, Courville, and Vincent 2013).

the loss $\mathcal{L}$ is C^1 in θ. If $(\theta_t)_{t\geq 0}$ is the gradient descent curve, we have $p_t = \mathrm{Law}(\theta_t)$ with

$$\partial_t p_t - \nabla \cdot (p_t \nabla \mathcal{L}) = 0. \tag{2}$$

Weight encoder. Due to the often intractable size of weight space, it is sometimes necessary for modeling to happen in latent space (see App. C for scaling remarks). We justify this design by appealing to work on the Lottery Ticket Hypothesis (Frankle and Carbin 2019) as well as the body of work on pruning (Cheng, Zhang, and Shi 2024) which suggests that, like natural data, neural networks live on a low-dimensional manifold within its ambient space. There are a variety of encoders to choose from, such as the variational autoencoder (VAE) (Kingma and Welling 2022), and specialized encoders for neural network parameters. As we are mostly focused on the generative aspect, we use the simple VAE following Soro et al. (2025) detailed in App. C.

Generative meta-model. The problem of learning the continuous dynamics of a system governed by a continuity equation has been studied in many forms in existing literature. In our setting, Theorem 1 establishes a link between the practical dynamics of SGD and the continuity equation (Eqn. 2), which provides a more tractable theoretical framework. Building on this connection, we study methods that realize parameterized solutions to Eqn. 2: $\partial_t p_t + \nabla \cdot (p_t v_t^{\Theta}) = 0$, thereby providing a common lens of interpretation. Lipman et al. (2023, Theorem 1) showed that CFM may be viewed through this lens, hence motivating its use. We follow the implementation of Tong et al. (2024). Exploiting the flexibility of FM to use a non-Gaussian prior, we use the Kaiming uniform or normal initializations (He et al. 2015) as the source p_0; see App. D for an ablation.

Reward fine-tuning. FM models lend themselves to the recently proposed reward fine-tuning method, based on the adjoint ODE (Domingo-Enrich et al. 2025), which casts stochastic optimal control as a regression problem. This allows us to tune pre-trained flow meta-models for downstream applications, exemplified in this work by detecting harmful covariate shifts, and improved generative performance (see App. E). Specifically, this method modifies the base generative distribution $\hat{p}_1^{\text{base}}$ to generate the reward-tilted distribution $p_1^*(x) \propto p_1^{\text{base}}(x)\exp(r(x))$ via the *Adjoint Matching* (AM) algorithm. Naturally, in our setting, we suppose p_1^{base} is obtained from meta-training and governs classifiers that predict on $\mathcal{D}_1$, but we wish to modify the meta-model to generate base models that predict on $\mathcal{D}_2$. Therefore, we set the reward $r(X_1) := -\mathcal{L}_2(X_1)$ where $\mathcal{L}_2$ is a loss on $\mathcal{D}_2$ such as cross-entropy and proceed with AM. See App. C for further details.

	CIFAR10	STL10	MNIST	FMNIST
Original	63.38	53.88	98.93	89.77
N$\mathcal{M}$-CFM w/ VAE	62.79	53.41	98.54	90.59
N$\mathcal{M}$-CFM	62.89	53.47	98.69	90.24

Table 1: Mean validation accuracy of top-5 N$\mathcal{M}$ model retrievals. A single meta-model is used for all base datasets, with a conditioning signal obtained from image samples used to distinguish between each set.

Detecting harmful covariate shifts

Training CDCs. Continuing the exposition in Section 2, we specify the training regiment of CDCs. Let $g(\cdot)$ represent one CDC and $f(\cdot)$ the base classifier; further, let $\mathbf{P} = \{(x_i, y_i)\}_{i=1}^n$ be samples from the generalization set $\mathcal{R}$ and $\mathbf{Q} = \{\tilde{x}_i\}_{i=1}^m$ from the unknown distribution. Our objective is for g to maintain performance on $\mathbf{P}$, but to disagree with f on $\mathbf{Q}$. Naturally, we use the cross-entropy loss ℓ_{ce} on $\mathbf{P}$, but on $\mathbf{Q}$, Ginsberg, Liang, and Krishnan (2023) introduces the *disagreement-cross-entropy*: $\ell_{dce}(\hat{y}, f(x_i)) = \frac{1}{1-N}\sum_{c=1}^N \mathbf{1}_{f(x_i)\neq c}\log(\hat{y}_c)$, where N denotes the total number of classes. We combine these objectives by minimizing the CDC loss:

$$\ell_{cdc}(\mathbf{P}, \mathbf{Q}) := \frac{1}{|\mathbf{P} \cup \mathbf{Q}|}\Bigg(\sum_{(x_i, y_i)\in\mathbf{P}} \ell_{ce}(g(x_i), y_i) + \lambda \sum_{\tilde{x}_i \in \mathbf{Q}} \ell_{dce}(g(\tilde{x}_i), f(\tilde{x}_i))\Bigg) \tag{3}$$

To test for shift, Detectron compares g trained with $\mathbf{Q}$ sampled from the unknown distribution ($g_{\mathbf{Q}}$) against $\mathbf{Q} = \mathbf{P}^*$ ($g_{\mathbf{P}}$), i.e. samples from the generalization set. In particular, the disagreement rate or the class entropy for each case is obtained and hypothesis tested. In both cases, the disagreement and entropy are higher if $\mathbf{Q}$ represents a significant shift.

Motivation. The problem of detecting covariate shift is not just about the data—a lot of modern neural networks are robust to such changes. The essence of the problem is whether or not *the classifier weights* required to predict on $\mathbf{Q}$ differs from the current classifier, motivating a method that is sensitive to changes in the weights required to predict on a new set. Building on the finding that the support of CNN3 classifiers is narrow (see Sec. 4) and the fact that the reward-tilted distribution (obtained from reward fine-tuning) has the same support, if the ideal classifier required to predict on a new dataset lies far outside of the original support, then we would expect a noticeable performance difference after reward fine-tuning than if it were close to the original support (see corruption experiments in Sec. 4, App. E).

Meta-detectron. Our approach, termed *meta-detectron*, builds on reward fine-tuning by adjoint matching. We start by meta-training *separate* N$\mathcal{M}$-CFM meta-models to learn classifier distributions on each of the datasets. Next, we reward fine-tune, maintaining the procedure of sampling from the meta-model at each iteration to compute the reward, but now the reward function is $r(X_1) = -\ell_{cdc}(\mathbf{P}, \mathbf{Q}; X_1)$, where X_1 serves the role of g, and the original (not fine-tuned) meta-model generates the base classifier f in Eqn. 3. As the method requires training $g_{\mathbf{P}}$ and $g_{\mathbf{Q}}$, we fine-tune two different meta-models depending on the disagreement set, and compare the disagreement rate and entropy of the generated $g_{\mathbf{P}}$ and $g_{\mathbf{Q}}$. Returning to our motivation, it is more

	CIFAR10		
Corruption Level	0	1	2
SGD fine-tuning	63.38 → 63.38	59.93 → 60.91	24.18 → 49.90
N$\mathcal{M}$-CFM	62.53 ± 0.02 → 63.33 ± 0.08	58.65 ± 0.22 → 60.34 ± 0.76	24.84 ± 0.93 → 34.15 ± 0.74
	MNIST		
Corruption Level	0	1	2
SGD fine-tuning	98.93 → 98.93	96.58 → 97.78	18.8 → 97.55
N$\mathcal{M}$-CFM	98.52 ± 0.01 → 98.79 ± 0.04	95.87 ± 0.01 → 97.01 ± 2.27	15.68 ± 0.17 → 91.21 ± 3.05

Table 2: Mean validation accuracy over five generated CNN3 classifiers after reward fine-tuning on increasingly corrupted datasets. The arrow '→' indicates the accuracy before (left) and after (right) reward fine-tuning. Further results in Table 8.

likely for the support to lie closer to classifiers that disagree on out-of-distribution $\mathbf{Q}$ than those disagreeing on $\mathbf{P}^*$.

4 Experiments

First, we confirm that CFM is able to do the basic task of model retrieval. Next, we explore reward fine-tuning applications. Further experiments and details may be found in Apps. D and E respectively.

Model retrieval

We first evaluate the conditional modeling capacity of the flow meta-model. The target distribution p_1 is defined by training base models on CIFAR10, CIFAR100, STL10, and MNIST, and saving weight checkpoints across 50 epochs. Due to its modest parameter count, we select the medium-CNN (CNN3) (Schürholt et al. 2022). We train just **one** meta-model conditioned on context samples from their respective training sets passed through a CLIP (Radford et al. 2021) encoder. At validation, we pass in a random support sample and generate the full CNN3. Table 1 shows that mean top-5 validation accuracies largely match base models, confirming capacity for conditional generation (see App. E for computational details).

Adapting to distribution shifts

In this section, we investigate the use of reward fine-tuning for shifting a distribution of classifier weights. Detail of our modifications can be seen in App. C.

Support of classifier weights Notably, the method of reward fine-tuning *cannot* be applied for arbitrary meta-model fine-tuning since supp $p_1^{\text{ft}} =$ supp p_1^{base}. For some loss $\mathcal{L}$, a soft (due to discretization and random sampling) loss lower bound for weights p_1^{ft} is $\arg\inf_\alpha\{\alpha > 0 : \text{supp}\, p_1^{\text{base}} \cap \{x : \mathcal{L}(x) \leq \alpha\} \neq \emptyset\}$. We stress that this property of the support is a function of both the downstream data *and* the model architecture. Indeed, due to the small size of the CNN3, the parameters that predict on different datasets e.g. CIFAR10 and STL10, will differ considerably, but this may not be the case for larger neural networks which possess a larger generalization set. We hypothesize: *the support set of CNN3 weights trained for different datasets are narrow and mostly disjoint, thus, small changes in the training data will noticeably affect the support w.r.t. validation accuracy.* Note how this ties back to the motivation of Meta-Detectron (Sec. 3).

Results We found experimental evidence to support this hypothesis, but also to suggest that reward fine-tuning goes a long way towards improving validation accuracy on out-of-distribution data. Table 2 shows an experiment where we reward fine-tuned the N$\mathcal{M}$-CFM meta-model on increasingly corrupted versions of the base training dataset (see App. E for further details and results). The effect of the corruption is noticeable on the support as reward fine-tuning, which is constrained within the support set, fails to reach the accuracy of SGD fine-tuning. Indeed, we find accuracies to be bounded above, often far below the validation accuracy obtained from SGD fine-tuning for the most corrupted data. This holds true even for mild corruption schemes, suggesting the ideal classifier support on the corrupted set has little intersection with the original support, indicating narrowness of the set. Given this finding, we use it to approach the problem of harmful covariate shifts.

Detecting harmful covariate shifts

Shift Detection via Two-Sample Testing We adopt a standard two-sample test, following Ginsberg, Liang, and Krishnan (2023), to detect distributional shifts between an in-distribution set $\mathbf{P}^*$ and an unknown set $\mathbf{Q}$. The null hypothesis $\mathbf{P}^* \sim \mathbf{Q}$ is rejected at the 5% significance level based on two statistics: *entropy* and *disagreement rate*.

Entropy. We measure predictive uncertainty over class logits as

$$\text{Entropy}(x) = -\sum_{c=1}^{N} \hat{p}_c \log \hat{p}_c \text{ where } \hat{p} = \tfrac{1}{2}\big(f(x) + g(x)\big), \tag{4}$$

with f denoting the base classifier and g the generated classifier. Unlike Detectron, our approach does not employ CDC ensembles.

Disagreement rate. We define the disagreement rate on a sample set $\mathbf{S} \in \{\mathbf{P}^*, \mathbf{Q}\}$ as

$$\text{Disagree}(\mathbf{S}) = \frac{1}{|\mathbf{S}|}\big|\{x \in \mathbf{S} : f(x) \neq g(x)\}\big|. \tag{5}$$

TPR@5	CIFAR10			Camelyon		
$\|\mathbf{Q}\|$	10	20	50	10	20	50
Detectron (DAR)	0	0	$.10 \pm .10$	$.10 \pm .10$	$.20 \pm .13$	$.50 \pm .17$
Meta-Detectron (DAR)	$.53 \pm .13$	$.47 \pm .13$	$.53 \pm .13$	$.73 \pm .12$	$\mathbf{.40 \pm .13}$	$\mathbf{.68 \pm .10}$
Detectron (Ent)	$.60 \pm .16$	$.10 \pm .10$	$.10 \pm .10$	0	0	0
Meta-Detectron (Ent)	$.47 \pm .13$	$\mathbf{.93 \pm .07}$	**1.00**	**1.00**	0	$.24 \pm .09$
AUROC	CIFAR10			Camelyon		
$\|\mathbf{Q}\|$	10	20	50	10	20	50
Detectron (DAR)	0.480	0.495	0.665	0.665	0.750	0.875
Meta-Detectron (DAR)	**0.876**	0.838	0.900	0.867	0.760	**0.930**
Detectron (Ent)	0.775	0.740	0.785	0.490	0.445	0.660
Meta-Detectron (Ent)	0.809	**0.987**	**1.000**	**1.000**	**0.836**	0.755

Table 3: TPR@5 and AUROC for detection of harmful covariate shift on CIFAR10.1 and Camelyon17. We test on both the disagreement rate (DAR) and the entropy (Ent), setting $\lambda = \kappa/(|\mathbf{Q}|+1)$. See App. E for details on choosing κ and extra results; the runs here vary κ between $|\mathbf{Q}|$. The best results are **bolded**.

Statistical testing. We apply a Kolmogorov–Smirnov test to the entropy distributions obtained from $g_{\mathbf{P}^*}$ and $g_{\mathbf{Q}}$. Intuitively, when $\mathbf{Q}$ is out-of-distribution, the generated classifier exhibits higher entropy and disagreement on $\mathbf{Q}$ compared to $\mathbf{P}^*$, reflecting greater uncertainty due to the base model f being less confident on $\mathbf{Q}$. Formally, we test the null hypothesis $\mathbb{E}[\phi_{\mathbf{Q}}] \leq \mathbb{E}[\phi_{\mathbf{P}^*}]$, where ϕ denotes either the entropy or disagreement statistic and the expectation is taken over random seeds. The result is deemed significant at level α if $\phi_{\mathbf{Q}}$ exceeds the $(1-\alpha)$ quantile of $\phi_{\mathbf{P}^*}$, with $\alpha = 0.05$ in all experiments.

Results We evaluate Meta-detectron on CNN3 with experiments following Ginsberg, Liang, and Krishnan (2023) on CIFAR10.1 (Recht et al. 2019), where shift comes from the dataset pipeline, and Camelyon17 (Veeling et al. 2018), which consists of histopathological slides from multiple hospitals. Table 3 shows the *True Positive Rate at 5% Significance Level (TPR@5)* and *AUROC* aggregated over 10 randomly chosen seeds for sampling $\mathbf{P}^*$ and $\mathbf{Q}$ of varying sample sizes. In addition, we ablated over the weight λ; see App. E for details and further results. Compared to the original tests (Ginsberg, Liang, and Krishnan 2023, Table 1) on ResNet-18, we observe that covariate shift is highly architecture dependent. This is expected as CNN3 underfits CIFAR10 ($\sim$ 63% validation accuracy). Our approach accounts for this as the base classifiers are generated directly by the fine-tuned meta-models. We also observe–though not shown–lower disagreement rates overall, which pays off in the TPR@5 as the $\mathbf{P}^*$ disagreement rates are close to zero in all cases, and showcases the conservative nature of our method. Indeed, we expect the disagreement rate to be more conservateive overall due to the tightness of the support. Importantly, we also observe in Table 10 (App. E) that the validation accuracy on $\mathbf{P}$ is mostly unchanged. Regarding meta-training behavior, Figure 2 shows that the AUROC increases sharply early in the reward fine-tuning phase, requiring only about 50 batch iterations to reach its peak. This coincides with a marked decrease in ℓ_{cdc}. However, we also note some instability in the AUROC, particularly in the Camelyon experiments, where fluctuations are more pronounced.

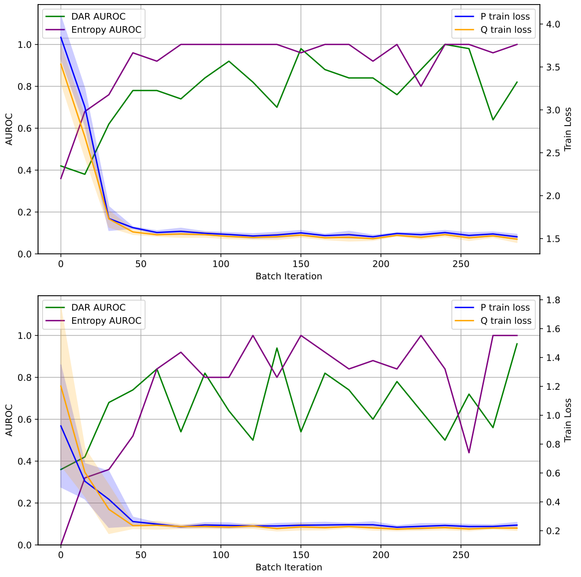

Figure 2: Plots illustrating how AUROC and ℓ_{cdc} evolves over meta-detectron training iterations for CIFAR10 and Camelyon17 when $|\mathbf{Q}| = 20$. See App. E for more figures.

5 Conclusion

In this work, we presented a preliminary investigation of flow matching for generative modeling in weight space, with a focus on detecting harmful covariate shifts. While our experiments demonstrate the promise of this approach, the scope of our study is limited by computational constraints, restricting our evaluation to architectures with fewer than 10^6 parameters. Future work will address these limitations by exploring equivariant architectures to reduce the effective dimensionality of weight space, increasing the diversity of training datasets, and incorporating stochastic weight evolution through Schrödinger bridge matching. Beyond shift detection, the proposed framework also opens several exciting avenues for research, including model merging via superposition of the inference ODE/SDE (Skreta et al. 2025) and extensions to network-constrained settings such as the generation of binary neural networks.

Acknowledgments

The authors would like to thank Lazar Atanackovic, Kirill Neklyudov, and Kacper Kapusniak for valuable discussions on methodology and implementation. We are also grateful to Soro Bedionita for assistance in resolving issues related to the D2NWG implementation. Special thanks go to Edwin Chacko, Vedant Swamy, and Ashwin Santhosh—developers in the UTMIST project group where this work was originally conceived—for their early collaboration.

References

Ainsworth, S. K.; Hayase, J.; and Srinivasa, S. 2023. Git Re-Basin: Merging Models modulo Permutation Symmetries. arXiv:2209.04836.

Albergo, M. S.; and Vanden-Eijnden, E. 2023. Building Normalizing Flows with Stochastic Interpolants. In *The Eleventh International Conference on Learning Representations*.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8): 1798–1828.

Berlinghieri, R.; Shen, Y.; and Broderick, T. 2025. Beyond Schrödinger Bridges: A Least-Squares Approach for Learning Stochastic Dynamics with Unknown Volatility. In *7th Symposium on Advances in Approximate Bayesian Inference – Workshop Track*.

Chen, A. M.; Lu, H.-m.; and Hecht-Nielsen, R. 1993. On the Geometry of Feedforward Neural Network Error Surfaces. *Neural Computation*, 5(6): 910–927.

Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2019. Neural Ordinary Differential Equations. arXiv:1806.07366.

Chen, T.; Liu, G.-H.; Tao, M.; and Theodorou, E. 2023. Deep Momentum Multi-Marginal Schrödinger Bridge. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Cheng, H.; Zhang, M.; and Shi, J. Q. 2024. A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 10558–10578.

Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*.

Domingo-Enrich, C.; Drozdzal, M.; Karrer, B.; and Chen, R. T. Q. 2025. Adjoint Matching: Fine-tuning Flow and Diffusion Generative Models with Memoryless Stochastic Optimal Control. In *The Thirteenth International Conference on Learning Representations*.

Falk, D.; Meynent, L.; Pfammatter, F.; Schürholt, K.; and Borth, D. 2025. A Model Zoo of Vision Transformers. arXiv:2504.10231.

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

Ginsberg, T.; Liang, Z.; and Krishnan, R. G. 2023. A Learning Based Hypothesis Test for Harmful Covariate Shift. In *The Eleventh International Conference on Learning Representations*.

Goldwasser, S.; Kalai, A. T.; Kalai, Y. T.; and Montasser, O. 2020. Beyond perturbations: learning guarantees with arbitrary adversarial test examples. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026–1034.

Hecht-Nielsen, R. 1990. On the Algebraic Structure of Feedforward Network Weight Spaces. In Eckmiller, R., ed., *Advanced Neural Computers*, 129–135. Amsterdam: North-Holland. ISBN 978-0-444-88400-8.

Jordan, R.; Kinderlehrer, D.; and Otto, F. 1998. The Variational Formulation of the Fokker–Planck Equation. *SIAM Journal on Mathematical Analysis*, 29(1): 1–17.

Kingma, D. P.; and Welling, M. 2022. Auto-Encoding Variational Bayes. arXiv:1312.6114.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Master's thesis*.

Lanzetti, N.; Terpin, A.; and Dörfler, F. 2024. Variational Analysis in the Wasserstein Space. arXiv:2406.10676.

Lavenant, H.; Zhang, S.; Kim, Y.-H.; and Schiebinger, G. 2024. Toward a mathematical theory of trajectory inference. *The Annals of Applied Probability*, 34(1A): 428 – 500.

Li, Z.; Malladi, S.; and Arora, S. 2021. On the Validity of Modeling SGD with Stochastic Differential Equations (SDEs). In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2023. Flow Matching for Generative Modeling. arXiv:2210.02747.

Liu, F.; Xu, W.; Lu, J.; Zhang, G.; Gretton, A.; and Sutherland, D. J. 2020. Learning deep kernels for non-parametric two-sample tests. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Liu, X.; Gong, C.; and Liu, Q. 2023. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *The Eleventh International Conference on Learning Representations*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable

Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, 5389–5400. PMLR.

Rohbeck, M.; Bunne, C.; Brouwer, E. D.; Huetter, J.-C.; Biton, A.; Chen, K. Y.; Regev, A.; and Lopez, R. 2025. Modeling Complex System Dynamics with Flow Matching Across Time and Conditions. In *The Thirteenth International Conference on Learning Representations*.

Santambrogio, F. 2015. *Optimal Transport for Applied Mathematicians: Calculus of Variations, Pdes, and Modeling*. Progress in Nonlinear Differential Equations and Their Applications. Springer International Publishing. ISBN 978-3-319-20828-2.

Schürholt, K.; Mahoney, M. W.; and Borth, D. 2024. Towards Scalable and Versatile Weight Space Learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR.

Schürholt, K.; Taskiran, D.; Knyazev, B.; Giró-i Nieto, X.; and Borth, D. 2022. Model Zoos: A Dataset of Diverse Populations of Neural Network Models. In *Thirty-Sixth Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.

Schürholt, K.; Meynent, L.; Zhou, Y.; Lu, H.; Yang, Y.; and Borth, D. 2025. A Model Zoo on Phase Transitions in Neural Networks. arXiv:2504.18072.

Shen, Y.; Berlinghieri, R.; and Broderick, T. 2025. Multi-marginal Schrödinger Bridges with Iterative Reference Refinement. In *The 28th International Conference on Artificial Intelligence and Statistics*.

Skreta, M.; Atanackovic, L.; Bose, J.; Tong, A.; and Neklyudov, K. 2025. The Superposition of Diffusion Models Using the Itô Density Estimator. In *The Thirteenth International Conference on Learning Representations*.

Soro, B.; Andreis, B.; Lee, H.; Jeong, W.; Chong, S.; Hutter, F.; and Hwang, S. J. 2025. Diffusion-based Neural Network Weights Generation. In *The Thirteenth International Conference on Learning Representations*.

Terpin, A.; Lanzetti, N.; Gadea, M.; and Dorfler, F. 2024. Learning diffusion at lightspeed. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Tong, A.; Fatras, K.; Malkin, N.; Huguet, G.; Zhang, Y.; Rector-Brooks, J.; Wolf, G.; and Bengio, Y. 2024. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*.

Veeling, B. S.; Linmans, J.; Winkens, J.; Cohen, T.; and Welling, M. 2018. Rotation equivariant CNNs for digital pathology. In *Medical image computing and computer assisted intervention–mICCAI 2018: 21st international conference, granada, Spain, September 16-20, 2018, proceedings, part II 11*, 210–218. Springer.

Wang, K.; Tang, D.; Zeng, B.; Yin, Y.; Xu, Z.; Zhou, Y.; Zang, Z.; Darrell, T.; Liu, Z.; and You, Y. 2024. Neural Network Diffusion. arXiv:2402.13144.

Wang, K.; Tang, D.; Zhao, W.; Schürholt, K.; Wang, Z.; and You, Y. 2025. Recurrent Diffusion for Large-Scale Parameter Generation. arXiv:2501.11587.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zhao, S.; Sinha, A.; He, Y.; Perreault, A.; Song, J.; and Ermon, S. 2022. Comparing Distributions by Measuring Differences that Affect Decision Making. In *International Conference on Learning Representations*.