Enabling Efficient Domain Adaptation via Noise-Enhanced Flow Matching

Aitian Ma, Dongsheng Luo, Mo Sha
Florida International University, Miami, Florida, USA
{*ama003, dluo, msha*}*@fiu.edu*

Keywords: Domain Adaptation, Simulation to Reality Transfer, Flow Matching, Noise Aware Learning, Few Shot Learning, Distribution Shift, Knowledge Distillation, Network Configuration

Abstract: Domain adaptation remains a significant challenge in deploying data-driven models under distribution shifts, particularly when transferring from simulated to real-world environments. Existing approaches often rely on large labeled target datasets, suffer negative transfer, and provide limited interpretability. In this paper, we present NoiseFlow, a data-efficient domain adaptation framework that leverages noise-aware modeling and flow matching to enable robust cross-domain generalization. Our key insight is that feature dimensions exhibit heterogeneous sensitivity to noise, which can be amplified under domain shift. NoiseFlow introduces a feature-aware teacher student architecture that combines knowledge distillation, distribution alignment, and continuous flow matching to learn smooth transformations between source and target domains. Experimentation on wireless network configuration tasks demonstrates that NoiseFlow achieves good performance in low-data regimes, reaching 69.8% accuracy with a single target sample and improving zero-shot transfer performance by up to 40% over existing methods.

1 Introduction

Deploying machine learning models in real-world environments often requires transferring knowledge across domains with different data distributions. This challenge is particularly pronounced when models trained on simulated or synthetic data are applied to real-world settings, where discrepancies in noise, environmental conditions, and measurement processes can lead to substantial performance degradation. Bridging this simulation-to-reality gap is a long-standing problem in data science, with applications spanning robotics, networked systems, and cyber-physical systems (Tzeng et al., 2017; Ma et al., 2025b; Ma et al., 2026b; Ma et al., 2026a).

Recent advances in domain adaptation have improved cross-domain generalization, but several key limitations remain. Many methods depend on large amounts of labeled target data, which are expensive or impractical to obtain in real-world settings (Pan and Yang, 2010). Others are sensitive to domain shifts and may suffer from negative transfer, where adaptation degrades rather than improves performance. In addition, many approaches operate as black-box models, limiting interpretability and making them less suitable for applications where transparency and reliability are required (Lipton, 2018).

In this work, we investigate the role of noise in domain adaptation and observe that different feature dimensions exhibit significantly different sensitivities to noise, and these disparities are amplified under domain shift. This observation motivates the need for feature-aware adaptation strategies rather than uniform treatment of all features.

We develop NoiseFlow, a domain adaptation framework that explicitly accounts for heterogeneous noise sensitivity and models domain shift as a continuous transformation process. NoiseFlow integrates three components within a unified framework. First, a teacher-student architecture preserves source domain knowledge while enabling adaptation to the target domain. Second, a feature-aware mechanism selectively emphasizes robust features and stabilizes sensitive ones during training. Third, a flow matching formulation models domain adaptation as learning continuous transformations between distributions, providing improved stability and interpretability compared to adversarial approaches (Lipman et al., 2023).

We evaluate NoiseFlow on cross-domain generalization tasks under low-data and zero-shot settings. Experimental results demonstrate that our approach achieves strong performance with minimal target supervision while maintaining robustness under significant domain shifts. More broadly, this work highlights the importance of noise-aware modeling and continuous transformation learning as general principles for scalable and reliable domain adaptation.

Experimental results show that flow matching provides a fundamental advancement for wireless domain adaptation, offering the stability, interpretability, and data efficiency necessary for practical industrial deployment. The framework's ability to maintain competitive performance (77.8% at 5-shot) while excelling in few-shot scenarios addresses critical deployment barriers where data collection is expensive and time-consuming.

2 Empirical Study

This section presents a systematic empirical analysis of feature-specific sensitivity under distribution shift. The objective is to understand how different performance-related features respond to noise perturbations and cross-domain transfer, and to derive insights that motivate noise-aware adaptation strategies.

2.1 Experimental Methodology

We conduct controlled experiments using publicly available simulation traces (Ma, 2026), consisting of 8,847 simulated samples and 2,156 physical measurements collected from a WirelessHART testbed. The analysis focuses on a set of performance features $\mathbf{x} = \{L, B, R\}$ representing latency, battery lifetime, and reliability, which are used to infer configuration parameters $\mathbf{y} = \{R_t, C, A\}$ corresponding to reception threshold, channel selection, and transmission attempts.

To evaluate robustness, we apply structured noise perturbations drawn from Gaussian, Laplace, uniform, and salt-and-pepper distributions with intensity $\sigma \in [0, 0.8]$. Each setting is evaluated over 500 independent trials, and statistical significance is verified using paired t-tests with $p < 0.05$. In addition, we evaluate cross-domain generalization by training models on simulation data and testing on physical measurements, thereby capturing realistic simulation-to-reality shift effects.

2.2 Feature Sensitivity under Noise

As Figure 1 shows, feature robustness varies significantly under noise perturbations within the same domain. Latency demonstrates strong stability across all noise levels, maintaining high predictive accuracy

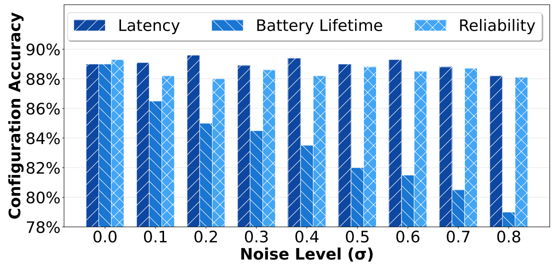

Figure 1: Feature sensitivity under noise perturbations in the same domain.

even at $\sigma = 0.8$ with less than 2% degradation. This indicates that latency is primarily governed by structural properties of the underlying system, such as routing topology and protocol behavior, which remain relatively invariant to measurement noise.

In contrast, battery lifetime exhibits substantially higher sensitivity. Its predictive accuracy decreases from 89% to 79% as noise increases, reflecting strong vulnerability of energy-related metrics to small perturbations. Such sensitivity is particularly critical because battery lifetime directly influences configuration decisions in resource-constrained settings, where small estimation errors can lead to significantly suboptimal outcomes.

Reliability lies between these two extremes, showing relatively stable but slightly fluctuating behavior across noise levels. This suggests that reliability depends on a combination of multiple interacting features, which provides partial robustness against perturbations while still being affected by noise.

Overall, the results reveal up to a fivefold difference in sensitivity across features, highlighting the inadequacy of uniform noise modeling assumptions commonly used in standard domain adaptation methods.

2.3 Cross-Domain Sensitivity Analysis

We next evaluate the effect of cross-domain transfer by training models on simulation data and testing them on physical measurements. As illustrated in Figure 2, domain shift significantly degrades overall performance and further amplifies feature-specific sensitivity.

Even in the absence of additional noise, performance drops from over 89% in simulation to approximately 50% in the real-world setting, indicating a substantial distribution mismatch between the two domains. This baseline gap already reflects the challenges of transferring models trained under idealized conditions to real-world environments.

When noise perturbations are introduced under

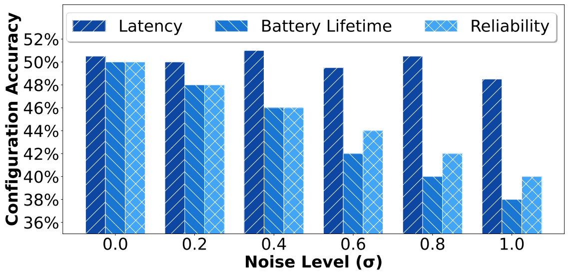

Figure 2: Feature sensitivity under cross-domain transfer with additional noise perturbations.

this shifted setting, latency remains comparatively stable, with only marginal additional degradation. This suggests that latency preserves structural consistency across domains and is less affected by unmodeled environmental variation.

Battery lifetime, however, experiences the most severe degradation, dropping from the 50% baseline to approximately 38% under noise. This reflects the difficulty of accurately modeling energy consumption in real environments, where factors such as hardware variability, environmental fluctuations, and operational irregularities are not fully captured in simulation.

Reliability again exhibits intermediate behavior, degrading steadily under noise but remaining more stable than battery-related metrics. This indicates partial transferability of reliability-related patterns across domains.

Overall, the results show that cross-domain transfer amplifies feature sensitivity by approximately two to three times compared to the single-domain setting. Rather than uniformly shifting performance, domain mismatch disproportionately affects specific feature dimensions, leading to structured degradation patterns across metrics.

2.4 Key Insights

The empirical results provide several important insights. Feature sensitivity is highly heterogeneous, with energy-related metrics significantly more vulnerable than latency-based features. Domain shift does not simply reduce performance uniformly but instead amplifies existing disparities across feature dimensions. This interaction between noise sensitivity and distribution shift suggests that uniform adaptation strategies are insufficient in settings with heterogeneous feature robustness. Instead, effective domain adaptation methods should explicitly account for feature-level sensitivity differences when designing robust cross-domain learning frameworks.

3 Design of NoiseFlow

This section presents the design of NoiseFlow, a feature-aware domain adaptation framework that reformulates simulation-to-reality transfer as a continuous flow matching problem. The key motivation stems from our empirical finding that wireless network features exhibit highly heterogeneous noise sensitivity, with energy-related metrics being significantly more vulnerable than latency. This observation motivates a departure from uniform adaptation strategies toward structured, feature-aware transformation.

3.1 System Architecture

NoiseFlow adopts a dual-network architecture designed to balance knowledge preservation and domain adaptation. This separation is critical in preventing catastrophic forgetting when transferring from simulation to real-world environments.

As Figure 3 shows, NoiseFlow consists of four interacting components. The **Teacher Network** is a fixed model trained on simulation data that encodes structured knowledge about wireless system behavior under idealized conditions. It remains frozen during adaptation to provide a stable reference for preserving useful simulation priors. The **Student Network** shares the same architecture but is updated during training to learn domain-invariant representations. It is optimized through a combination of knowledge distillation, distribution alignment, and flow matching objectives, enabling adaptation to real-world data while retaining simulation-derived structure. The **Flow Matching Module** models domain shift as a continuous transformation process by learning a velocity field that transports simulation distributions toward real-world distributions. This formulation provides a smooth and stable alternative to adversarial alignment methods, particularly in high-noise wireless environments. The **Adaptive Noise Scheduler** incorporates feature-level sensitivity information into the learning process. Instead of applying uniform perturbations, it selectively adjusts noise injection according to empirical robustness profiles, ensuring that vulnerable features such as energy-related metrics are protected while more stable features such as latency can be more aggressively augmented.

3.2 Information Processing Pipeline

Input data flows through a structured pipeline designed for efficient and stable domain adaptation. The system first performs feature-aware preprocessing guided by sensitivity-aware noise control, fol-

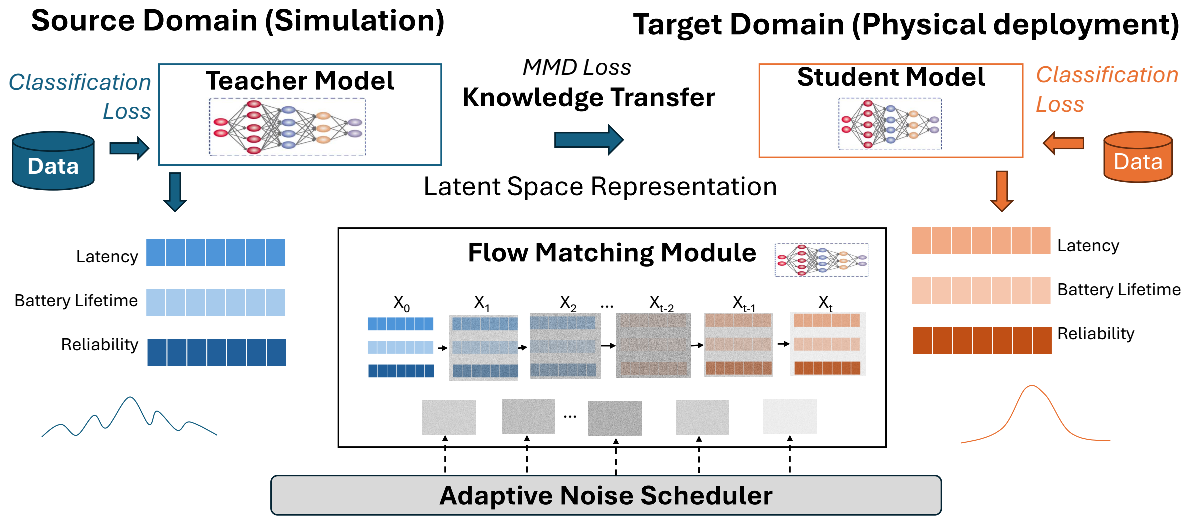

Figure 3: Overview of the NoiseFlow framework for feature-aware domain adaptation.

lowed by representation learning in a shared latent space. The flow matching module then gradually transforms source-domain representations toward the target domain through continuous dynamics. Finally, a task-specific prediction head produces configuration decisions suitable for deployment in dynamic environments.

This design supports deployment in resource-constrained settings by avoiding iterative adversarial optimization and enabling deterministic forward inference through learned transformations.

3.3 Flow Matching Framework

Traditional domain adaptation methods often assume uniform feature sensitivity, which is inconsistent with our empirical findings that battery lifetime exhibits substantially higher noise sensitivity than latency. This mismatch leads either to under-protection of critical energy-related features or unnecessary distortion of stable features.

NoiseFlow addresses this limitation by reformulating domain adaptation as a continuous flow matching problem with feature-aware transformation dynamics. The goal is to learn a transport process that maps simulation data to real-world data while respecting feature-specific robustness constraints.

3.3.1 Mathematical Formulation

We model domain adaptation as a continuous transformation defined by a time-dependent flow $\phi(t, \mathbf{x})$ that evolves simulation data into real-world distributions over the interval $t \in [0, 1]$.

The dynamics are governed by a neural ordinary differential equation:

$$\frac{d\mathbf{x}(t)}{dt} = v_\theta(\mathbf{x}(t), t), \tag{1}$$

where v_θ is a learnable velocity field parameterized by the student network.

At $t = 0$, samples originate from the simulation domain, while at $t = 1$, they align with the real-world distribution. The flow, therefore, defines a continuous interpolation between idealized and real-world wireless conditions.

This formulation provides three key advantages: it ensures smooth transitions that preserve physical consistency, enables invertible mappings between domains, and improves optimization stability compared to adversarial training.

3.3.2 Velocity Field Learning

The objective of flow matching is to learn a velocity field that approximates the true transport dynamics between domains. This is achieved by minimizing the discrepancy between the predicted velocity and the target transport direction:

$$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1}\left[\|v_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1 - \mathbf{x}_0)\|^2\right], \tag{2}$$

where $\mathbf{x}_t$ represents an interpolated state between source $\mathbf{x}_0$ and target $\mathbf{x}_1$.

Unlike standard interpolation strategies, NoiseFlow introduces feature-aware path construction that incorporates sensitivity information into the transport process. This ensures that sensitive features such as energy consumption follow conservative transformation trajectories, while robust features such as latency undergo more flexible adaptation.

3.4 Time-Conditioned Adaptation

To capture non-stationary transformation dynamics, NoiseFlow incorporates time conditioning into the velocity field. A learned embedding function maps temporal information into a representation that modulates the transformation process:

$$\mathbf{e}(t) = \mathrm{TimeEmbed}(t), \tag{3}$$

which is integrated into the velocity network as

$$v_\theta(\mathbf{x}, t) = \mathrm{VelocityNet}(\mathbf{x}, \mathbf{e}(t)). \tag{4}$$

This design allows the model to learn different adaptation behaviors at different stages of the transformation. Early stages focus on preserving simulation structure, intermediate stages perform most of the domain alignment, and later stages refine alignment with real-world characteristics.

3.5 Optimization Algorithm

Algorithm 1 summarizes the training procedure of NoiseFlow, which jointly learns a feature-aware flow matching model and a noise-sensitive adaptation strategy. The key idea is to couple continuous transport learning with empirically derived feature sensitivity masks, ensuring that the learned transformation respects heterogeneous robustness properties across wireless metrics.

At a high level, each training iteration constructs paired samples from simulation and real-world domains, injects controlled feature-aware perturbations into the simulation samples, and then learns a velocity field that aligns the two distributions through a continuous interpolation process. Unlike standard domain adaptation methods that optimize static alignment objectives, this procedure explicitly models domain shift as a dynamic transport process.

The training procedure can be interpreted as learning a continuous vector field that transports simulation samples toward real-world distributions. The interpolation step $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$ defines a linear coupling between domains, while the neural velocity field v_θ learns to approximate the instantaneous direction of this transformation.

A key component of the algorithm is the feature-aware noise injection controlled by the sensitivity mask $\mathbf{M}$. This mask encodes empirically measured robustness differences across features, allowing NoiseFlow to selectively perturb stable features while preserving the integrity of sensitive ones. This design directly reflects our empirical observation that not all wireless metrics respond equally to noise, and therefore should not be treated uniformly during adaptation.

Algorithm 1 Wireless-Aware Flow Matching Training

Require: Simulation batch $\mathcal{X}_s$, real batch $\mathcal{X}_r$, sensitivity mask $\mathbf{M}$

Ensure: Learned velocity field v_θ

Sample time index $t \sim \mathcal{U}(0,1)$ for interpolation along the flow trajectory

for each paired sample $(\mathbf{x}_s, \mathbf{x}_r)$ **do**

  Draw stochastic perturbation $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I)$

  Apply feature-aware noise injection:

  $\mathbf{x}_0 \leftarrow \mathbf{x}_s + \boldsymbol{\varepsilon} \odot \mathbf{M}$

  Set target sample:

  $\mathbf{x}_1 \leftarrow \mathbf{x}_r$

  Construct intermediate state along a continuous path:

  $\mathbf{x}_t \leftarrow (1-t)\mathbf{x}_0 + t\mathbf{x}_1$

  Define ground-truth transport direction:

  $\mathbf{v}_{\mathrm{target}} \leftarrow \mathbf{x}_1 - \mathbf{x}_0$

  Predict velocity using neural flow model:

  $\mathbf{v}_{\mathrm{pred}} \leftarrow v_\theta(\mathbf{x}_t, t)$

  Compute flow matching loss:

  $\mathcal{L}_{\mathrm{flow}} \leftarrow \|\mathbf{v}_{\mathrm{pred}} - \mathbf{v}_{\mathrm{target}}\|_2^2$

end for

Update parameters using gradient descent:

$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{flow}}$

return v_θ

The objective function $\|\mathbf{v}_{\mathrm{pred}} - \mathbf{v}_{\mathrm{target}}\|_2^2$ enforces consistency between the learned velocity field and the true transport direction induced by paired samples. Unlike adversarial objectives, this formulation provides stable optimization dynamics and avoids the instability associated with min-max training.

Overall, this algorithm operationalizes NoiseFlow's central principle: domain adaptation should be modeled as a structured, continuous transport problem with explicit awareness of feature-level sensitivity rather than a uniform alignment task.

3.6 Inference via ODE Integration

At inference time, NoiseFlow applies the learned velocity field to progressively transform simulation inputs into real-world-aligned representations using numerical ODE integration. Starting from $\mathbf{x}_{\mathrm{sim}}$, the system iteratively updates the state according to the learned dynamics:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot v_\theta(\mathbf{x}(t), t). \tag{5}$$

This iterative process yields a continuous transformation trajectory that can be controlled in terms of step size and accuracy, enabling flexible deployment in real-world systems.

3.7 Design Summary

NoiseFlow integrates feature-aware modeling, continuous transformation learning, and knowledge distillation into a unified framework for domain adaptation. By explicitly incorporating heterogeneous feature sensitivity into the adaptation process, the model avoids the limitations of uniform perturbation strategies and provides a stable, interpretable, and data-efficient solution for simulation-to-reality transfer in complex data-driven systems.

4 Evaluation

We evaluate NoiseFlow using data traces collected from a real-world WirelessHART testbed and a matched NS-3 simulation environment, and study its effectiveness in bridging the simulation-to-reality gap in both few-shot and hyperparameter sensitivity settings. We compare against two representative baselines: a teacher-student framework (Shi et al., 2021; Shi et al., 2024) and WMN-CDA (Ma and Sha, 2025), implemented in PyTorch (Paszke et al., 2019). All methods are trained using Adam (Diederik, 2015) on NVIDIA A100 GPUs.

Our testbed consists of 50 TelosB motes (Polastre et al., 2005), producing 6,600 traces across 88 configurations. Each configuration is defined by the packet reception ratio threshold $R \in \{0.7, \ldots, 0.9\}$, the channel count $C \in \{1, \ldots, 8\}$, and the transmission attempts $A \in \{1, 2, 3\}$. We evaluate three metrics: latency (L), battery lifetime (B), and reliability (E). A matched NS-3 (NS-3 Model, 2011) environment is used to replicate identical configurations under idealized simulation conditions.

4.1 Main Performance

We first evaluate the performance of the few-shot adaptation under limited target-domain supervision, where only a small number of labeled physical samples are available. We vary the number of shots from 1 to 5, with each shot corresponding to 88 real-world samples.

Figure 4 summarizes the results. NoiseFlow consistently achieves superior performance in the low-data regime, reaching 55.6% accuracy with a single shot, significantly outperforming both baselines. This

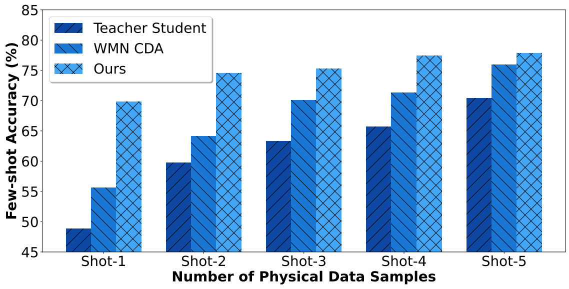

Figure 4: Few-shot performance comparison on the WirelessHART testbed. Each shot corresponds to 88 additional real-world samples used for adaptation.

indicates that NoiseFlow is able to effectively transfer knowledge from simulation to real-world conditions even under severe data scarcity.

As the number of shots increases, all methods improve but exhibit different learning dynamics. The Teacher-Student baseline shows steady and predictable gains, reaching 79.8% at 5-shot, reflecting its reliance on incremental supervision. WMN-CDA exhibits a more pronounced improvement curve, starting from 48.8% at 1-shot and reaching 75.7% at 5-shot, suggesting that sufficient target-domain coverage is required before effective alignment emerges. In contrast, NoiseFlow maintains strong performance across all settings and achieves competitive accuracy at higher shot levels while preserving its advantage in low-shot scenarios.

4.2 Hyperparameter Analysis

We analyze the sensitivity of NoiseFlow to two key hyperparameters: the flow weight λ_f, which balances flow matching and classification objectives, and the noise scale σ, which controls the strength of perturbations during training. These parameters directly influence the trade-off between domain alignment and task preservation.

Figure 5 shows the impact of these hyperparameters on mean performance. The flow weight exhibits a clear optimal value at $\lambda_f = 0.3$, achieving the highest mean accuracy of 74.97%. This indicates that moderate flow matching strength is sufficient to enable effective domain alignment while preserving discriminative structure. Smaller values underutilize the flow objective, while larger values over-constrain representation learning and reduce task performance.

The noise scale shows a more stable trend across different values, with $\sigma = 0.8$ achieving the best mean performance of 74.74%. This suggests that stronger perturbations improve generalization by exposing the model to a broader range of intermediate domain states.

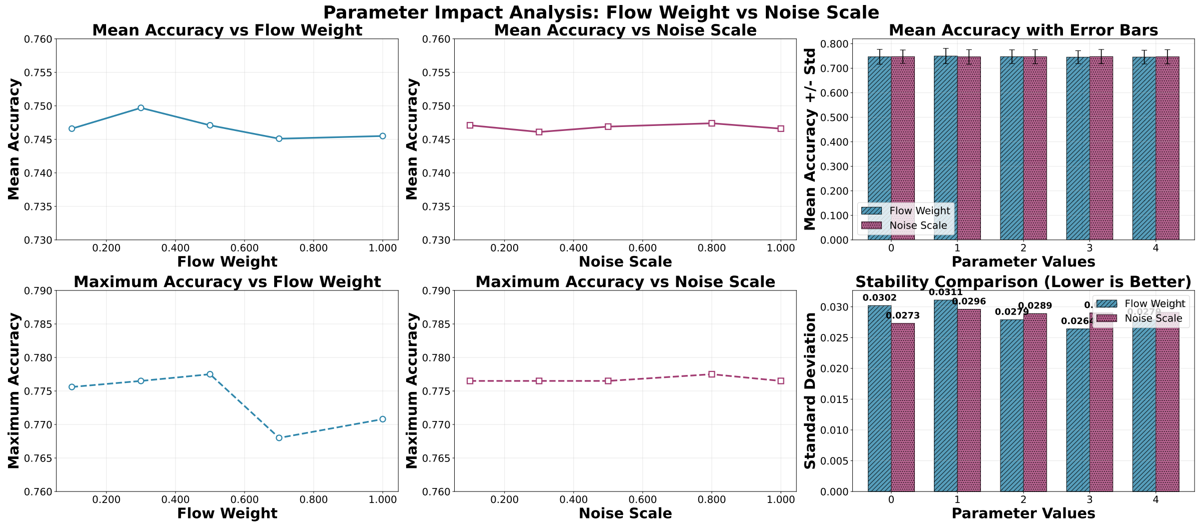

Figure 5: Effect of flow weight λ_f and noise scale σ on model performance.

We further examine the performance–stability trade-off in Figure 6. The results show that higher flow weights, particularly $\lambda_f = 0.7$, yield the most stable training behavior with lower variance, while $\lambda_f = 0.3$ achieves higher accuracy at the cost of increased variability. For noise scale, smaller values such as $\sigma = 0.1$ provide the most stable behavior while maintaining competitive performance, making them preferable for deployment scenarios where consistency is critical.

Overall, these results indicate that NOISEFLOW is robust to moderate hyperparameter variations while maintaining a clear optimal operating region. This robustness is important for practical deployment in industrial wireless systems where tuning opportunities are limited.

5 Related Work

5.1 Domain Adaptation for Cross-Domain Transfer

Domain adaptation aims to mitigate distribution shift between training (simulation) and deployment (real-world) environments through representation alignment (Goodfellow et al., 2014) and adversarial training methods. Early approaches focus on feature alignment and adversarial training strategies (Tzeng et al., 2017; Ganin et al., 2016), demonstrating consistent but moderate gains over source-only models in standard benchmarks.

In networked systems, several methods have been explored for domain adaptation for wireless optimization. Shi et al. (Shi et al., 2021) propose a teacher-student framework that improves performance under limited labeled target data, while Ma et al. (Ma and Sha, 2025) introduce contrastive learning-based approaches for improved generalization. Graph-based approaches (Zhao et al., 2020) have enabled topology-aware adaptation. Despite these advances, such methods often depend on substantial labeled target data, which can be costly or impractical to obtain in real deployments. They may also degrade under large domain shifts and provide limited interpretability for operational decision-making (Lipton, 2018).

5.2 Flow-Based Generative and Transport Models

Flow-based generative models learn invertible mappings between probability distributions through either discrete transformations, such as normalizing flows (Dinh et al., 2017), or continuous formulations based on neural ordinary differential equations (Chen et al., 2018). These methods provide tractable likelihood estimation and stable training dynamics compared to adversarial approaches. Recent advances in flow matching and velocity field learning (Tong et al., 2024; Liu et al., 2023) further simplify training by directly learning transport dynamics without requiring explicit likelihood computation or architectural constraints. These approaches have shown strong performance in generative modeling and density estimation tasks.

However, most existing work focuses on image, audio, or general continuous data distributions and does not explicitly account for feature-level hetero-

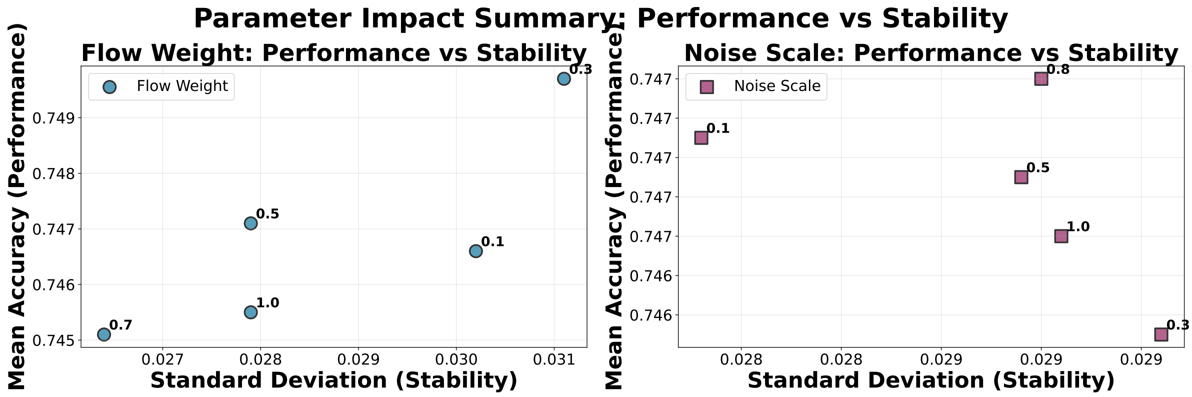

Figure 6: Performance–stability trade-off analysis for flow weight and noise scale.

geneity or physically meaningful constraints. In contrast, wireless network data exhibits structured and semantically heterogeneous features, where different metrics (e.g., latency and energy consumption) have fundamentally different noise sensitivities and operational roles. Our work builds on these ideas by adapting flow matching to cross-domain network configuration, introducing feature-aware transport dynamics that respect sensitivity heterogeneity and preserve physically meaningful relationships in wireless systems.

5.3 Wireless Network Configuration

Network configuration in wireless systems involves optimizing parameters such as transmission power, channel allocation, and routing policies under dynamic and uncertain conditions. Traditional approaches rely on rule-based heuristics or classical optimization techniques, which often assume stationary environments and therefore exhibit limited adaptability in practice. More recent data-driven methods leverage deep learning to enable adaptive configuration in complex environments (Cheng et al., 2024; Ma et al., 2025a). While these approaches improve responsiveness and performance, they typically require large-scale labeled data from target deployments and offer limited interpretability, which remains a key challenge in safety-critical settings where decision transparency and reliability are required.

6 Conclusion

We develop NoiseFlow, a feature-aware flow matching framework for bridging the simulation-to-reality gap. Motivated by our empirical finding that energy-related metrics exhibit substantially higher sensitivity than latency, NoiseFlow departs from uniform domain adaptation and introduces a continuous flow formulation that explicitly models feature-dependent vulnerability. By integrating flow matching with a feature-aware adaptation mechanism, NoiseFlow selectively preserves fragile battery lifetime predictions while enabling smooth and stable transformation of robust features across simulation and physical domains. This design improves both interpretability and training stability, which are critical requirements for safety-critical industrial deployments. Experiments on wireless network configuration tasks show that NoiseFlow achieves good performance under limited supervision, reaching 69.8% accuracy in few-shot settings while significantly reducing the need for expensive real-world data collection. These results highlight the effectiveness of incorporating feature-level sensitivity into domain adaptation.

Acknowledgment

This work was supported in part by the National Science Foundation under grants CNS-2150010, ECCS-2242700, and IIS-2529283.

REFERENCES

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Cheng, X., Sha, M., and Chen, D. (2024). Configuring Industrial Wireless Mesh Networks via Multi-Source Domain Adaptation. In *ACM International Conference on Embedded Wireless Systems and Networks (EWSN)*.

Diederik, P. K. (2015). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density Estimation Using Real NVP. In *International Conference on Learning Representations (ICLR)*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial Training of Neural Networks. In *Journal of machine learning research*, volume 17, pages 2096–2030.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*.

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

Lipton, Z. C. (2018). The Mythos of Model Interpretability. *Queue*, 16(3).

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

Ma, A. (2026). WMN-CDA Data traces, github.com/aitianma/WSNConfDomainAdaptation. [Online; accessed 12-April-2026].

Ma, A., Cruz, J. M., Luo, D., and Sha, M. (2026a). Enabling Large Language Model based Data Synthesis in Wireless Mesh Network Configuration for Internet of Things. In *IEEE Computers, Software, and Applications Conference (COMPSAC)*.

Ma, A., Luo, D., Maatouk, A., Ying, R., and Sha, M. (2026b). Configuring industrial wireless mesh network via dual-mind reasoning. In *IEEE International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*.

Ma, A., Rodriguez, J. T., Sha, M., and Luo, D. (2025a). Sensorless Air Temperature Sensing Using LoRa Link Characteristics. In *IEEE International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*.

Ma, A. and Sha, M. (2025). WMN-CDA: Contrastive Domain Adaptation for Wireless Mesh Network Configuration. In *ACM/SIGAPP Symposium On Applied Computing (SAC)*.

Ma, D., Ma, A., Luo, S., de Jode, M., Hudson-Smith, A., and Sha, M. (2025b). LORADOCTOR: LLM-Driven Diagnosis and Adaptive Policy Optimization for Reducing Packet Error Rate in LoRaWAN Networks. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*.

NS-3 Model (2011). NS-3 Shadowing Model.

Pan, S. J. and Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An Imperative Style, High-performance Deep Learning Library. *Advances in Neural Information Processing Systems (NeurIPS)*.

Polastre, J., Szewczyk, R., and Culler, D. (2005). Telos: Enabling Ultra-low Power Wireless Research. In *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*.

Shi, J., Ma, A., Cheng, X., Sha, M., and Peng, X. (2024). Adapting Wireless Network Configuration from Simulation to Reality via Deep Learning based Domain Adaptation. *IEEE/ACM Transactions on Networking*, 32(3):1983–1998.

Shi, J., Sha, M., and Peng, X. (2021). Adapting Wireless Mesh Network Configuration from Simulation to Deployment. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*.

Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., and Bengio, Y. (2024). Improving and Generalizing Flow-based Generative Models with Minibatch Optimal Transport. *Transactions on Machine Learning Research*, pages 1–34.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2017). Adversarial Discriminative Domain Adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, W., Queralta, J. P., and Westerlund, T. (2020). Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In *IEEE Symposium Series on Computational Intelligence (SSCI)*.