# LOFT: LATENT OPTIMAL TRANSPORT-GUIDED FEATURE TRANSFORMATION FOR ROBUST AUTOMATIC MODULATION RECOGNITION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Automatic Modulation Recognition (AMR) seeks to infer the modulation schemes of received signals without prior knowledge. While deep learning has greatly advanced AMR, severe noise in low signal-to-noise ratio (SNR) regimes corrupts learned feature representations, thereby degrading recognition performance. Unlike previous methods that focus on data augmentation or signal denoising, this paper investigates low-SNR AMR failures through the lens of shortcut learning. As SNR decreases, models tend to take shortcuts by collapsing their predictions onto a few dominant classes, leading to class confusion. To tackle this issue, we propose **L**atent **O**ptimal Transport-guided **F**eature **T**ransformation (**LOFT**), a plug-and-play framework for reversing feature collapse. Motivated by the semantic invariance of modulation classes across SNRs, LOFT constructs clean class prototypes as semantic anchors and formulates latent recovery through an optimal transport-guided alignment objective. A lightweight transport adapter is then trained to predict feature displacements that move degraded representations toward clean semantic anchors in a single step. Furthermore, LOFT utilizes confidence-aware routing to selectively activate latent correction without requiring SNR information during inference. Extensive experiments on benchmark datasets demonstrate that LOFT consistently improves AMR performance, yielding an average accuracy gain of 1.63% across all SNR conditions and a 3.04% improvement in severe noise conditions. Code will be released upon acceptance.

## 1 INTRODUCTION

Automatic Modulation Recognition (AMR) is a critical task in civil and military communication applications (Nandi & Azzouz, 1998), aiming to identify the modulation schemes of received signals without prior knowledge (Wang et al., 2019). In recent years, deep learning has become the dominant paradigm for AMR (Zhang et al., 2022), replacing hand-crafted feature engineering (Dobre et al., 2007) with deep representations learned from raw signals (Chen et al., 2026). Despite this progress, deep models still struggle with accurate modulation recognition when signal observations are corrupted by noise, particularly in low signal-to-noise ratio (SNR) regimes (Zhang et al., 2024b).

To address this challenge, existing methods mainly focus on data augmentation (Zheng et al., 2021), noise-aware learning (Park et al., 2025), or self-supervised learning (Liu et al., 2026). Although these methods improve robustness to noisy observations, they generally require training from scratch and provide limited flexibility for adapting off-the-shelf pretrained models (Wang et al., 2025a). More fundamentally, they address noise robustness without asking why the model itself degrades under low SNR (Zhang et al., 2024b). This raises an interesting question: how does a well-trained AMR model fail, and can this degradation be corrected while keeping the original model frozen?

It is well established that deep models tend to take shortcuts during learning and rely on simplistic features (Geirhos et al., 2020). Under low SNR, where discriminative information is scarce (Zheng et al., 2025), we find this tendency becomes particularly pronounced. As shown in Figure 1, the model does not simply yield uncertain predictions. Instead, its predictions concentrate on a dominant class, while representations from different classes become entangled and collapse toward a dominant



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

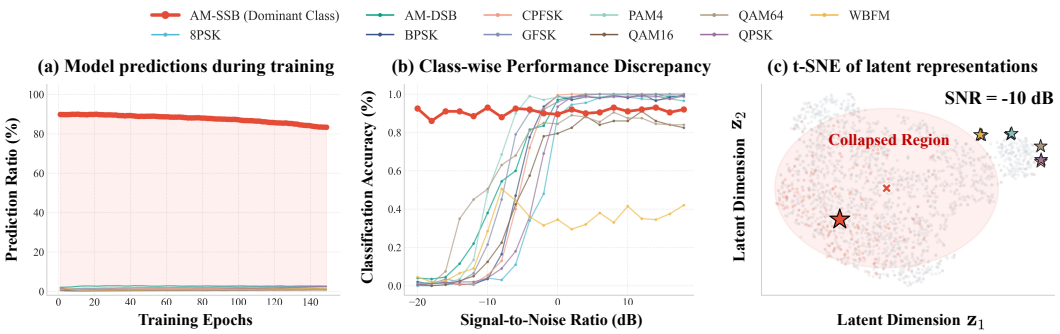



Figure 1: **The evidence of shortcut learning and semantic representation collapse.** (a) During training at low SNRs ($\leq 0$ dB), the model fails to learn robust features and instead takes a shortcut, collapsing its prediction mass onto a dominant class. (b) This strategy leads to extreme class-wise performance discrepancy during testing, yielding high accuracy for the dominant class while causing total recognition failure for all others. (c) At $-10$ dB, corrupted representations drift from their clean prototypes and become heavily entangled, forming a dense collapsed region.

region in the latent space. We term this phenomenon **semantic representation collapse**, where input corruption triggers a learning shortcut that ultimately collapses the representation space.

Importantly, this observation also reveals a key property that makes recovery possible: the semantic identity of a modulation class should remain invariant to SNR. A clean signal and its corrupted counterpart may occupy very different regions of the learned feature space, yet they should represent the same underlying modulation scheme. This semantic invariance provides a natural reference for recovering degraded representations. Thus, we bypass the reconstruction of the corrupted signal and instead seek to restore degraded features to class-consistent states directly in the latent space.

In this paper, we propose **L**atent **O**ptimal Transport-guided **F**eature **T**ransformation (**LOFT**), a plug-and-play framework for reversing shortcut-induced feature collapse. LOFT first constructs clean class prototypes in the latent space from high-SNR observations, providing stable semantic anchors for each modulation scheme. Since class entanglement in the collapsed region renders direct sample-wise matching unreliable, we recast latent feature recovery as a distribution transport problem. Under a squared Euclidean cost, optimal transport (OT) establishes class-consistent couplings between degraded features and clean class prototypes. Continuous trajectory integration may risk perturbing decision boundaries in discriminative tasks. LOFT circumvents this by learning a Monge map parameterized by a lightweight transport adapter. This adapter predicts a deterministic displacement field, directly transporting each degraded representation toward its corresponding clean state in a single step. Finally, applying the transformation indiscriminately may distort already reliable representations at inference time. LOFT addresses this by using predictive confidence to selectively activate latent correction, enabling robust recognition without explicit SNR estimation.

In summary, our contributions are as follows:

- **Revisiting Low-SNR Failure.** We identify shortcut learning as a key failure mode of low-SNR AMR, manifesting as semantic representation collapse. This finding shifts the focus from reconstructing corrupted signals to rectifying the collapsed representations.
- **Turning Collapse into Correction.** To the best of our knowledge, **LOFT** is the first framework to formulate latent feature recovery in low-SNR AMR as an optimal transport problem. LOFT learns a Monge map parameterized by a lightweight transport adapter, which predicts a deterministic displacement field for single-step latent correction. A confidence-guided adaptive strategy then enables selective latent correction for robust inference.
- **Establishing Robustness and Efficiency.** Extensive experiments on two widely used benchmark datasets demonstrate that LOFT consistently improves existing AMR models across a broad range of SNR conditions, achieving an average performance improvement of 1.63%, with larger improvements of 3.04% at SNR $\leq 0$ dB. Moreover, the adapter introduces only 0.05M trainable parameters and requires merely 3 training epochs, achieving parameter-efficient latent correction and rapid adaptation with minimal training overhead.

## 2 RELATED WORK

**Robust Automatic Modulation Recognition.** As a longstanding task, AMR has evolved from handcrafted feature engineering (Dobre et al., 2007) to data-driven deep representation learning (Zhang et al., 2022). End-to-end models, including CNN- and RNN-based architectures such as IC-AMCNet (Tekbıyık et al., 2020) and GRU2 (Hong et al., 2017), automatically learn spatiotemporal representations from raw signals, while recent hybrid architectures such as TransGroupNet (Zeng et al., 2024) and IQFormer (Shao et al., 2024) have further advanced recognition performance. However, existing deep models still exhibit inaccurate classification in harsh and low-SNR environments (Zhang et al., 2024b). To improve robustness, recent studies can be broadly grouped into three streams. **(i) Data augmentation** enriches the training distribution with synthetic noise or signal perturbations (Zheng et al., 2021; Huang et al., 2026). **(ii) Noise-aware learning** incorporates noise characteristics or SNR information into the learning process (Park et al., 2025; Yang et al., 2025). **(iii) Self-supervised learning** learns robust signal representations through auxiliary objectives such as signal reconstruction and contrastive learning (Du et al., 2025; Liu et al., 2026). Despite their effectiveness, existing methods typically rely on retraining or architectural modification to adapt the recognition model. In contrast, LOFT directly corrects degraded latent representations while keeping the pretrained recognition backbone frozen.

**Modular Adaptation and Representation Alignment.** Due to the additional cost of retraining, parameter-efficient adapters have emerged as a practical paradigm for modifying pretrained representations with few trainable parameters (Rebuffi et al., 2017; Houlsby et al., 2019). Recent studies have extended modular adaptation to domain-specific settings, enabling pretrained models to incorporate task knowledge without full fine-tuning (Zhang et al., 2024b; Wu et al., 2025). Alongside modular adaptation, representation alignment has been widely explored in AMR to mitigate feature shifts across varying domains or environmental conditions (Zhang et al., 2024a; Xing et al., 2025). Existing methods typically align source and target representations by minimizing distribution discrepancies, often through adversarial learning or divergence-based objectives (Wang et al., 2025b). Conversely, LOFT treats low-SNR degradation as a latent recovery problem, transporting collapsed representations from degraded states toward the class-consistent representation distribution.

## 3 METHODOLOGY

As illustrated in Figure 2, **LOFT** is a plug-and-play framework for recovering collapsed latent representations while keeping the pretrained AMR backbone frozen. The fundamental insight of LOFT is that SNR degradation can be compressed into an endpoint recovery problem.

### 3.1 PROBLEM FORMULATION AND ANALYSIS

Let a pretrained AMR model be formulated as $f(x) = h(g(x))$, where $g : \mathcal{X} \to \mathbb{R}^d$ denotes the feature extractor that maps an input signal $x \in \mathcal{X}$ to a $d$-dimensional latent representation, and $h : \mathbb{R}^d \to \mathbb{R}^K$ denotes the classifier over $K$ modulation classes. For an input $x_s$ with modulation class $y$ under SNR condition $s$, its latent representation is defined as $z_s = g(x_s)$.

We denote $p(z \mid y, s)$ as the class-conditional latent distribution. Under ideal conditions, latent representations of the same modulation class are expected to remain concentrated around a stable semantic region. As the SNR decreases, the latent representation deviates from this class-specific semantic structure. To quantify this deviation, we define the semantic drift of class $y$ at SNR $s$ as

$$\Delta^{y}_{\text{drift}}(s) = \left\| \mathbb{E}_{z \sim p(z|y,s)}[z] - \mu_y \right\|_2 , \tag{1}$$

where $\mu_y$ denotes the clean semantic prototype of class $y$, which will be formally constructed in Section 3.2. Beyond intra-class drift, noise can reduce the separability between different modulation classes in the latent space. We characterize this phenomenon through the Wasserstein-2 distance between class-conditional latent distributions (Villani et al., 2009):

$$\mathcal{D}_{ij}(s) = \mathcal{W}_2 \left( p(z \mid y_i, s), p(z \mid y_j, s) \right), \qquad y_i \neq y_j. \tag{2}$$

The SNR-induced semantic representation collapse is jointly characterized by a larger $\Delta^{y}_{\text{drift}}(s)$ and a smaller $\mathcal{D}_{ij}(s)$, which capture the intra-class drift from clean anchors and the shrinking inter-class separation. Therefore, the low-SNR AMR problem can be viewed as a latent-space recovery problem, motivating us to seek a transformation that reverses the feature collapse.

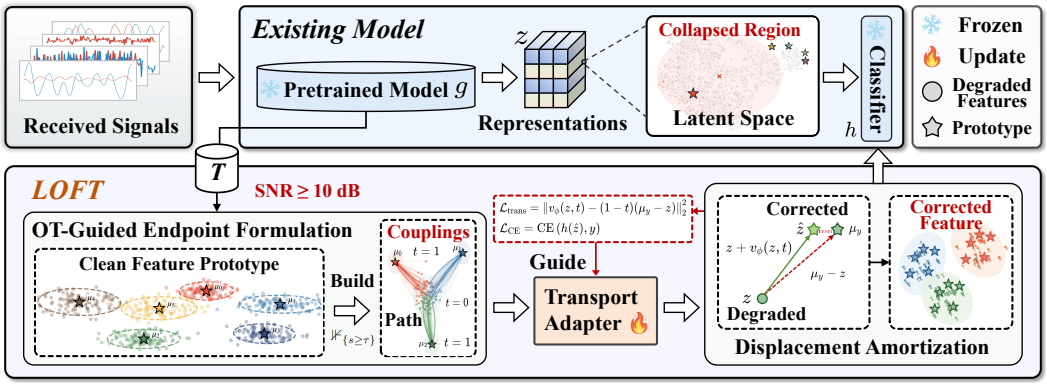



Figure 2: **Overview of the proposed LOFT framework. Upper**: Representations $z$ extracted by a frozen backbone $g$ suffer from SNR-induced drift, significantly degrading the frozen classifier's performance. **Lower**: A plug-and-play module avoiding backbone fine-tuning, comprising three stages: 1) OT-Guided Endpoint Formulation: Constructs clean semantic prototypes from high-SNR samples as transport endpoints; 2) Transport Adapter: A lightweight adapter jointly optimized via $\mathcal{L}_{\text{trans}}$ and $\mathcal{L}_{\text{CE}}$ to learn feature dynamics; 3) Displacement Amortization: Predicts an amortized displacement $v_\phi(z,t)$ approximating the prototype-directed target $\mu_y - z$, effectively steering degraded features toward clean semantic anchors to restore discriminative capability.

### 3.2 OT-Guided Endpoint Formulation

Latent recovery aims to align degraded representations with a clean semantic endpoint distribution. To construct such an endpoint, we first define the clean feature bank $\mathcal{Z}_y^{\text{clean}}$ for class $y$ collected from high-SNR observations, and represent the class-conditional endpoint by its empirical distribution $\nu_y$:

$$\mathcal{Z}_y^{\text{clean}} = \{g(x_i) \mid y_i = y,\ s_i \geq 10\,\text{dB}\}, \quad \nu_y = \frac{1}{|\mathcal{Z}_y^{\text{clean}}|} \sum_{z_i \in \mathcal{Z}_y^{\text{clean}}} \delta_{z_i}, \tag{3}$$

where $g(\cdot)$ denotes the frozen feature extractor, $s_i$ is the SNR associated with sample $x_i$, and $\delta_{z_i}$ denotes the Dirac measure centered at $z_i$. This empirical endpoint retains the class-specific structure exhibited by high-SNR latent representations without imposing an assumption on their distribution.

Given a degraded class-conditional latent distribution $p(z \mid y, s)$, we formulate latent recovery as transporting it toward the corresponding clean endpoint distribution $\nu_y$. Under the quadratic transport cost, the ideal Monge map is defined as:

$$\mathcal{T}_y^* = \arg \min_{\mathcal{T}_y : (\mathcal{T}_y)_{\#} p(z|y,s) = \nu_y} \mathbb{E}_{z \sim p(z|y,s)} \left[ \|\mathcal{T}_y(z) - z\|_2^2 \right], \tag{4}$$

where $(\mathcal{T}_y)_{\#} p(z \mid y, s)$ denotes the push-forward distribution induced by $\mathcal{T}_y$. Equation 4 characterizes the desired recovery by identifying the optimal map that minimizes the expected quadratic displacement when transporting the degraded distribution into its clean class-conditional endpoint. However, directly estimating $\mathcal{T}_y^*$ from empirical latent distributions requires resolving sample-wise correspondences between degraded and clean representations. Performing such transport estimation repeatedly during adapter training introduces unnecessary computational overhead and makes the correction dependent on finite-sample transport plans. We therefore use the endpoint formulation in Equation 4 to motivate a computationally efficient prototype-directed surrogate.

Specifically, we summarize the clean endpoint distribution by its first moment, which defines the clean semantic prototype of class $y$:

$$\mu_y = \mathbb{E}_{z' \sim \nu_y}[z'] = \frac{1}{|\mathcal{Z}_y^{\text{clean}}|} \sum_{z_i \in \mathcal{Z}_y^{\text{clean}}} z_i. \tag{5}$$

The prototype $\mu_y$ provides a class-specific semantic anchor representing the central tendency of the clean latent region. Based on this anchor, we define the prototype-directed displacement target as

$$u_y^{\text{proto}}(z) = \mu_y - z. \tag{6}$$

Rather than explicitly computing the exact Monge map $\mathcal{T}_y^*$, $u_y^{\text{proto}}(z)$ acts as a tractable surrogate that captures the dominant semantic shift. By preserving the endpoint geometry without costly empirical transport estimation, this displacement is then amortized via a lightweight adapter for efficient, single-step latent correction.

### 3.3 Transport Displacement Amortization

To realize the amortization motivated above, we introduce a lightweight conditional neural transformation $v_\phi(z, t)$, where $z$ is the observed latent feature and $t$ denotes a discrete transport state. Guided by the endpoint OT formulation, LOFT abstracts the degradation trajectory into a binary transport state space. We map the continuous SNR spectrum $s$ to the transport coordinate $t \in \{0, 1\}$ via an indicator function:

$$t(s) = \mathbb{1}_{\{s \geq \tau\}}, \tag{7}$$

where $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function and $\tau$ is a threshold. This formalization partitions the input space into a degraded source domain ($t = 0$) and a corrected target domain ($t = 1$), rendering the specific threshold $\tau$ functionally independent of the definable boundary in Equation 5 used to construct the semantic prototypes.

To parameterize this transformation, we encode the coordinate $t$ via sinusoidal embeddings and predict the conditional displacement field $v_\phi(z, t)$ using lightweight MLPs (Hornik et al., 1989):

$$e(t) = \text{MLP}_t\big([\cos(\omega t), \sin(\omega t)]\big), \quad v_\phi(z, t) = \text{MLP}_z\big(z + e(t)\big). \tag{8}$$

The latent correction is formulated as a residual transformation modulated by the transport horizon:

$$\hat{z} = z + v_\phi(z, t). \tag{9}$$

Given a training sample with class $y$, we optimize the transport adapter to amortize $u_y^{\text{proto}}(z)$ in Equation 6, while preserving compatibility with the decision geometry of the frozen classifier. We formulate the transport displacement loss $\mathcal{L}_{\text{trans}}$ and a cross-entropy regularization term $\mathcal{L}_{\text{CE}}$ as:

$$\mathcal{L}_{\text{trans}} = \|v_\phi(z, t) - (1 - t)(\mu_y - z)\|_2^2, \quad \mathcal{L}_{\text{CE}} = \text{CE}\left(h(\hat{z}), y\right). \tag{10}$$

Minimizing $\mathcal{L}_{\text{trans}}$ drives the adapter to explicitly reverse the intra-class semantic drift $\Delta_{\text{drift}}^y(s)$. Simultaneously, $\mathcal{L}_{\text{CE}}$ ensures that the restored representations maintain inter-class separability $\mathcal{D}_{ij}(s)$ while remaining geometrically compatible with the non-linear decision boundaries of the pretrained model. The joint training objective of the LOFT framework is therefore:

$$\mathcal{L}_{\text{LOFT}} = \mathcal{L}_{\text{trans}} + \lambda \mathcal{L}_{\text{CE}}, \tag{11}$$

where $\lambda$ controls the strength of the classification objective. LOFT learns an external mapping that transports degraded features back to the semantic manifolds already established by the backbone.

### 3.4 Confidence-Aware Latent Correction

During deployment, the true SNR condition $s$ is unknown, rendering the oracle transport state $t(s)$ in Equation 7 unavailable. An intuitive approach would rely on sample-wise prediction confidence to trigger the correction. However, as identified in Section 3.1, the model succumbs to shortcut learning under low SNR, often yielding spuriously overconfident predictions for collapsed representations. Relying on individual samples is thus highly vulnerable to this spurious overconfidence. Wireless channels typically exhibit temporal coherence over short time intervals, causing consecutively received signals to experience similar channel conditions (Clarke, 1968). We exploit this property to estimate the transport state at the batch level. Specifically, for an inference batch $\mathcal{B}$ of size $B$, we quantify its semantic integrity by computing the entropy of its marginal prediction distribution:

$$\bar{p} = \frac{1}{B} \sum_{b=1}^{B} \text{softmax}(h(z_b)), \quad \mathcal{H}(\mathcal{B}) = -\sum_{k=1}^{K} \bar{p}_k \log \bar{p}_k, \tag{12}$$

where $\text{softmax}(\cdot)$ is applied to the classifier logits over $K$ classes. We thus establish a confidence-aware routing mechanism to infer the transport state $\hat{t}$:

$$\hat{t} = \mathbb{1}_{\{\mathcal{H}(\mathcal{B}) \geq \log(K)/2\}}. \tag{13}$$

The representations are then corrected accordingly and passed to the frozen classifier:

$$z_{\text{corr}} = z + v_\phi(z, \hat{t}), \quad \hat{y} = h(z_{\text{corr}}). \tag{14}$$

Unlike continuous neural ordinary differential equations that require iterative latent updates, LOFT amortizes the target transport displacement into a single feed-forward latent transformation.

Table 1: AMR accuracy (%) under varying SNR levels (-12 dB to 2 dB) on the RML2016.10A and RML2016.10B. We report the performance of pretrained backbones alongside their LOFT-equipped counterparts (mean ± std over 3 runs). **Bold red** indicates the performance equipped with LOFT.

| Method | SNR (dB) | | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| | -12 | -10 | -8 | -6 | -4 | -2 | 0 | 2 | |
| **RML2016.10A** | | | | | | | | | |
| FEA-T | 16.64 | 24.73 | 36.50 | 48.23 | 59.50 | 69.00 | 75.55 | 78.36 | 55.49 |
| + **LOFT** | **21.59±0.60** | **29.71±0.70** | **36.67±0.46** | **48.67±0.20** | **60.48±0.23** | **69.27±0.20** | **76.41±0.08** | **79.42±0.05** | **56.69±0.05** |
| TransGroupNet | 15.82 | 25.45 | 35.50 | 51.50 | 65.00 | 73.14 | 79.23 | 83.59 | 57.35 |
| + **LOFT** | **19.79±0.61** | **28.73±1.13** | **41.02±0.31** | **52.05±0.25** | **65.36±0.12** | **73.23±0.09** | **79.58±0.05** | **83.83±0.11** | **58.48±0.08** |
| PETCGDNN | 16.18 | 25.82 | 37.32 | 52.32 | 65.55 | 73.77 | 80.23 | 84.32 | 58.01 |
| + **LOFT** | **19.08±1.01** | **30.88±0.82** | **43.62±0.78** | **53.02±0.19** | **67.88±0.23** | **75.39±0.14** | **81.50±0.05** | **84.48±0.05** | **59.48±0.04** |
| GRU2 | 16.45 | 25.14 | 40.50 | 55.36 | 66.36 | 74.77 | 80.50 | 82.45 | 58.28 |
| + **LOFT** | **21.79±0.76** | **33.89±1.01** | **41.34±0.40** | **56.29±0.03** | **68.30±0.05** | **77.17±0.13** | **83.14±0.16** | **85.29±0.03** | **60.77±0.02** |
| AWN | 17.91 | 27.95 | 40.00 | 55.32 | 67.41 | 77.09 | 86.68 | 88.77 | 61.28 |
| + **LOFT** | **20.08±1.27** | **32.80±2.08** | **40.80±0.82** | **56.21±0.30** | **71.33±0.46** | **79.42±0.33** | **88.17±0.09** | **89.61±0.17** | **62.98±0.07** |
| IQFormer | 16.23 | 28.32 | 43.32 | 57.91 | 69.59 | 80.91 | 88.73 | 90.36 | 62.40 |
| + **LOFT** | **21.55±1.03** | **34.17±0.45** | **44.27±0.18** | **58.19±0.24** | **71.23±0.21** | **82.55±0.08** | **90.20±0.05** | **91.33±0.07** | **64.12±0.06** |
| **RML2016.10B** | | | | | | | | | |
| FEA-T | 20.14 | 29.66 | 43.11 | 58.17 | 72.24 | 83.67 | 89.66 | 91.24 | 63.73 |
| + **LOFT** | **24.93±0.77** | **38.39±0.51** | **45.34±0.49** | **63.82±0.19** | **75.15±0.07** | **84.29±0.04** | **89.79±0.19** | **91.36±0.23** | **65.26±0.03** |
| TransGroupNet | 20.13 | 29.19 | 41.32 | 56.95 | 73.04 | 86.13 | 90.93 | 91.90 | 64.01 |
| + **LOFT** | **22.81±0.09** | **35.17±0.91** | **44.37±0.71** | **61.10±0.12** | **75.63±0.09** | **87.28±0.04** | **91.66±0.04** | **92.43±0.02** | **65.31±0.05** |
| PETCGDNN | 21.13 | 30.32 | 39.62 | 54.06 | 70.68 | 83.54 | 90.00 | 92.09 | 63.89 |
| + **LOFT** | **24.06±0.33** | **37.23±0.91** | **42.68±1.82** | **60.76±0.52** | **75.71±0.36** | **84.72±0.28** | **90.39±0.06** | **92.33±0.09** | **65.40±0.13** |
| GRU2 | 21.11 | 30.26 | 43.48 | 56.91 | 71.90 | 84.33 | 89.97 | 91.18 | 63.97 |
| + **LOFT** | **25.06±0.99** | **37.95±1.49** | **44.89±1.28** | **65.08±0.39** | **78.76±0.31** | **86.47±0.18** | **91.16±0.07** | **92.24±0.02** | **65.99±0.05** |
| AWN | 21.33 | 30.84 | 41.93 | 55.79 | 70.81 | 82.73 | 89.93 | 91.60 | 63.98 |
| + **LOFT** | **26.13±1.47** | **35.98±2.32** | **42.49±0.48** | **60.24±0.26** | **73.87±0.25** | **82.99±0.17** | **90.26±0.32** | **91.82±0.25** | **65.44±0.06** |
| IQFormer | 20.56 | 33.66 | 46.41 | 57.91 | 70.43 | 82.03 | 89.34 | 91.99 | 64.28 |
| + **LOFT** | **25.96±0.32** | **38.63±0.28** | **47.82±0.17** | **64.71±0.07** | **77.85±0.27** | **86.09±0.20** | **91.46±0.09** | **92.16±0.43** | **66.35±0.02** |

# 4 EXPERIMENTS

In this section, we evaluate LOFT for robust automatic modulation recognition under low-SNR conditions. LOFT is integrated with six commonly used AMR backbones as a plug-and-play module, with all pretrained backbones kept frozen during single-pass latent correction.

## 4.1 EXPERIMENTAL SETTINGS

**Datasets and Baselines.** We evaluate our method on two widely adopted AMR benchmarks, including RML2016.10A and RML2016.10B (O'shea & West, 2016). Both datasets comprise $2 \times 128$ I/Q sequences across 20 SNR levels (-20 to 18 dB). Specifically, RML2016.10A contains 220,000 samples across 11 modulation schemes under standard channel impairments, such as Rayleigh fading, multipath, and AWGN. In contrast, RML2016.10B is a larger, more rigorous variant with 1,200,000 samples over 10 schemes. Notably, it introduces severe real-world hardware imperfections by incorporating carrier frequency offset and sample rate clock offset. To validate the plug-and-play capability, we select six representative backbones spanning diverse architectures: GRU2 (Hong et al., 2017), PETCGDNN (Zhang et al., 2021), FEA-T (Chen et al., 2022), AWN (Zhang et al., 2023), TransGroupNet (Zeng et al., 2024), and IQFormer (Shao et al., 2024). This diverse selection ensures a rigorous evaluation against various deep representation learning strategies.

**Implementation Details.** All methods are implemented in PyTorch and executed on a single NVIDIA GeForce RTX 4090 GPU. For baseline pretraining, we employ the Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of $10^{-3}$, a weight decay of $10^{-5}$, and a fixed batch size of 512. The learning rate decays by a factor of 0.9 every 5 epochs over a maximum of 200 epochs. For the proposed **LOFT**, the pretrained backbones are frozen. The lightweight transport adapter is optimized for merely 3 epochs using a learning rate of $10^{-3}$, demonstrating extreme parameter and training efficiency. The clean feature bank $\mathcal{Z}_y^{\text{clean}}$ is constructed from high-SNR

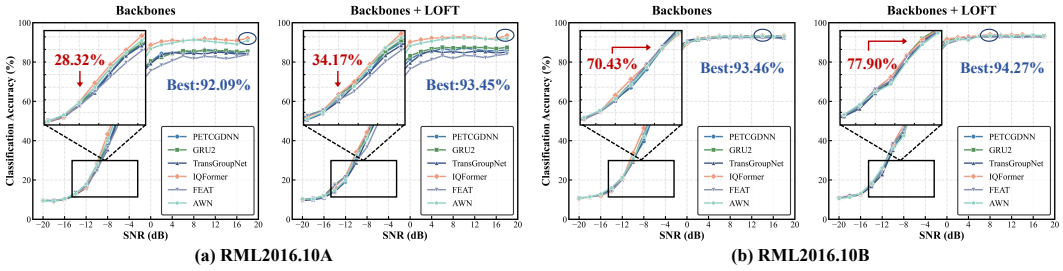



Figure 3: Visualization of performance across the full SNR spectrum. Red arrows and blue circles highlight the best improvement at low SNR and the peak accuracy at high SNR, respectively.

training samples ($\geq$ 10 dB). The semantic alignment coefficient is set to $\lambda = 1.0$, and the batch-level confidence routing threshold is fixed at $0.5$. To ensure statistical rigor, all LOFT improvements are reported as the mean and standard deviation across three independent runs.

### 4.2 Main Results

Table 1 shows that LOFT consistently improves recognition robustness across all evaluated backbones and benchmarks. Notably, these gains are obtained by optimizing only the lightweight adapter for three epochs while keeping the pretrained backbones fully frozen, demonstrating the effectiveness of LOFT as a parameter-efficient and plug-and-play adaptation strategy. On RML2016.10A, LOFT achieves average accuracy improvements of 1.13%–2.49% across architectures with different topologies. Similar improvements are observed on the more challenging RML2016.10B benchmark, where LOFT provides gains of up to 2.07% (e.g., IQFormer), indicating its robustness under realistic hardware-induced distortions. The benefits of LOFT are particularly evident in low-SNR conditions ($-12$ dB to $-4$ dB), where recognition performance is highly vulnerable to noisy and distorted representations. For example, on RML2016.10B at $-6$ dB, LOFT improves GRU2 from 56.91% to 65.08%, yielding an absolute gain of 8.17%, and increases IQFormer from 57.91% to 64.71%. On RML2016.10A at $-12$ dB, IQFormer further benefits from LOFT, improving from 16.23% to 21.55%. These results support our motivation that transport-guided representation adjustment can effectively restore more reliable class-discriminative structures.

Additionally, Figure 3 shows that LOFT maintains baseline performance under favorable channel conditions, with IQFormer achieving the highest accuracy improvements of 1.36% and 0.81% on RML2016.10A and RML2016.10B, respectively. This behavior verifies the effectiveness of the confidence-aware routing mechanism: when representations already exhibit reliable semantic structures, LOFT avoids unnecessary transformations and preserves stable predictions. As a result, LOFT provides robust adaptation across varying SNR conditions without requiring backbone retraining.

### 4.3 Ablation Studies

To verify the contribution of each component in LOFT, we conduct comprehensive ablation studies on RML2016.10A with the GRU2 backbone, as shown in Table 2. **1) Effect of the Joint Optimization Objective.** We examine the role of optimal transport by removing $\mathcal{L}_{\text{trans}}$. Without the transport objective, the adapter degenerates into a conventional feature transformation module, overfitting discriminative shortcuts under severe noise and resulting in only 27.30% accuracy at -10 dB. In contrast, removing $\mathcal{L}_{\text{CE}}$ preserves geometric alignment through OT flow, improving low-SNR robustness (31.00% at -10 dB), but lacks explicit semantic supervision to align transported features with the frozen classifier. These results demonstrate the joint objective is essential for balancing distribution alignment and task discrimination. **2) Effect of Confidence-aware Correction.** We remove the confidence-aware correction mechanism and apply uniform transformation to all samples. Although this improves recovery under extreme noise (33.50% at -10 dB), it introduces unnecessary modifications to already reliable high-SNR features, reducing the 10 dB accuracy from 85.77% to 79.10% and lowering the overall performance below the baseline. **3) Effect of Clean Target Prototypes.** We replace the clean high-SNR prototypes with global class centroids computed from all SNR levels. Since these prototypes are contaminated by noisy representations, they provide less

Table 2: Ablation study of each component on RML2016.10A. We report the average accuracy alongside performance in extreme low-SNR (-10 dB) and high-SNR (10 dB) regimes to demonstrate the necessity of each mechanism. **Bold** indicates the optimal configuration.

| Model Configuration | Components | | | | Accuracy (%) | | |
|---|---|---|---|---|---|---|---|
| | $\mathcal{L}_{\text{trans}}$ | $\mathcal{L}_{\text{CE}}$ | Routing ($\hat{t}$) | Clean Proto. | **Avg.** | **-10 dB** | **10 dB** |
| Baseline (GRU2) | × | × | × | × | 58.28 | 25.14 | 85.77 |
| (a) w/o OT | × | ✓ | ✓ | ✓ | 58.90 | 27.30 | 85.80 |
| (b) w/o CE | ✓ | × | ✓ | ✓ | 59.80 | 31.00 | 85.90 |
| (c) w/o Confidence-aware Correction | ✓ | ✓ | × | ✓ | 57.14 | 33.50 | 79.10 ↓ |
| (d) w/ Global Prototypes | ✓ | ✓ | ✓ | × | 60.20 | 28.50 | 87.00 |
| **LOFT (Ours)** | ✓ | ✓ | ✓ | ✓ | **60.77** | **33.89** | **87.48** |

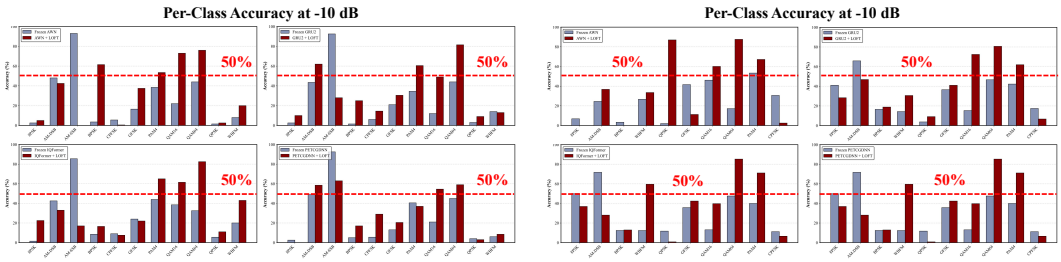
reliable transport targets and weaken semantic recovery under severe degradation. As a result, the -10 dB accuracy decreases from 33.89% to 28.50%. This confirms that clean prototypes provide precise semantic anchors for robust feature transport. Overall, removing any individual component consistently degrades performance, validating that LOFT relies on the synergistic interaction among geometric transport, semantic supervision, adaptive routing, and clean prototypes.

## 5 Analysis of the Mechanism Behind LOFT

In this section, we further scrutinize the efficacy of LOFT by addressing the following questions:

**Q1**: Does LOFT mitigate the shortcut learning phenomenon that occurred in the frozen model?

To answer this question, we visualize the per-class prediction distributions under severe noise degradation ($-10$ dB). As shown in Figure 4, the frozen baselines (gray bars) inevitably succumb to shortcut learning, minimizing empirical risk by collapsing their predictions into a biased subset of categories. Specifically, they exhibit catastrophic semantic collapse, heavily overfitting to a single dominant class (e.g., AM-SSB) while completely failing on the remaining modulation schemes. In contrast, integrating LOFT (dark red bars) explicitly dismantles this spurious correlation and effectively alleviates the shortcut learning bottleneck. By guiding the collapsed features toward uniformly distributed clean prototypes through optimal transport, LOFT successfully restores a balanced semantic structure. As a result, a substantial portion of previously degraded classes are propelled above the 50% accuracy threshold, demonstrating that LOFT reconstructs a more geometrically aligned and semantically consistent feature space even under heavily corrupted environments.




(a) RML2016.10A (b) RML2016.10B

Figure 4: Per-class accuracy breakdown at $-10$ dB across four backbones on two datasets.

**Q2**: How does LOFT geometrically reshape the corrupted latent space?

To further understand how LOFT reshapes the latent space, we project the high-dimensional latent features extracted at $-10$ dB into a 2D subspace. As shown in Figure 5, noise destroys the topological structure of the frozen backbone. Features from distinct modulation schemes are heavily entangled, resulting in a collapsed space with low inter-class separability ($D_{\text{inter}} = 26.93$) and high intra-class variance ($D_{\text{intra}} = 15.73$). This geometric entanglement directly triggers the shortcut learning behavior diagnosed in Q1.

In contrast, Figure 5 (right) visualizes the feature space after applying LOFT. Driven by the optimal transport mechanism, the corrupted representations are deterministically pulled toward their corresponding clean semantic anchors. This structural reshaping significantly expands the decision margins, increasing $D_{\text{inter}}$ to 36.23, while simultaneously restoring intra-class compactness by reducing $D_{\text{intra}}$ to 10.56. This quantitative and geometric evidence confirms that LOFT functions as a precise discrete flow transformation, fundamentally reconstructing the boundaries necessary for robust classification in severely degraded environments.

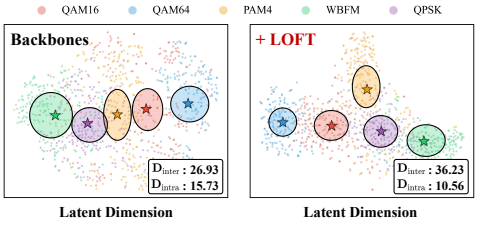



Figure 5: t-SNE visualization of latent representations at -10 dB (averaged across all backbones on RML2016.10B).

**Q3**: How lightweight is LOFT in terms of computational and storage overhead compared to the frozen backbones?

To quantify the computational and storage footprint of the LOFT module, we analyze its architectural complexity and training efficiency. Because LOFT operates on latent spaces rather than high-dimensional raw inputs, its structural overhead is minimal and depends solely on the backbone's latent feature dimensions. The transport adapter consists merely of a lightweight Timestep Embedder and a bottleneck transport network. This minimalist design translates to training efficiency; optimizing the adapter requires only 3 epochs, taking approximately 7.14 seconds in total to converge.

Table 3: Computational and storage overhead. LOFT introduces an exceptionally marginal absolute footprint (only ∼0.05M) across all backbones.

| Backbone | Parameters (M) | | FLOPs (M) | |
|---|---|---|---|---|
| | Base | w/ LOFT | Base | w/ LOFT |
| GRU2 | 0.15 | 0.20 (+0.05) | 19.40 | 19.45 (+0.05) |
| FEA-T | 0.34 | 0.39 (+0.05) | 2.69 | 2.74 (+0.05) |
| PETCGDNN | 0.07 | 0.12 (+0.05) | 8.31 | 8.36 (+0.05) |
| TransGroupNet | 3.01 | 3.04 (+0.03) | 24.32 | 24.35 (+0.03) |
| IQFormer | 0.33 | 0.35 (+0.02) | 40.86 | 40.88 (+0.02) |

As summarized in Table 3, integrating LOFT introduces exactly $49,728$ parameters ($\sim 0.05$ M) and approximately $0.05$ M FLOPs per inference step. Regardless of the chosen pretrained backbone, the absolute footprint increment remains fixed and negligible. Furthermore, because LOFT routes high-confidence signals directly to the classifier, the actual expected inference overhead during deployment is even lower. This remarkable parameter efficiency verifies that LOFT achieves profound geometric reshaping (Q2) and restores blind inference robustness (Q1) without compromising the deployment viability of the original baselines on resource-constrained communication devices.

## 6 Conclusion

In this paper, we propose **LOFT** for robust AMR. By analyzing low-SNR degradation through the lens of shortcut learning, LOFT identifies shortcut-induced semantic representation collapse and recasts latent feature recovery as an optimal transport problem. With clean class prototypes as semantically consistent targets, LOFT learns a lightweight transport adapter to amortize the optimal displacement into an efficient single-step latent correction, which is further combined with confidence-aware routing for adaptive inference. Extensive experiments demonstrate that LOFT substantially improves recognition robustness, particularly under severe degradation, while keeping pretrained backbones frozen. These findings highlight the geometric reversal of representation collapse as a promising paradigm for enhancing pretrained models against severe distribution shifts.

### AI USE STATEMENT

In this work, we used generative AI tools for polishing the language of the manuscript. We have not used generative AI tools for generating research ideas or writing original content, and tasks such as data synthesis or code generation are not applicable to this work. Additionally, we used generative AI tools for improving the readability and grammatical structure of the text. We have reviewed all AI-assisted work. Specifically, we utilized Gemini and GPT exclusively to refine the English writing of our drafted manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Yantao Chen, Binhong Dong, Cuiting Liu, Wenhui Xiong, and Shaoqian Li. Abandon locality: Frame-wise embedding aided transformer for automatic modulation recognition. *IEEE Communications Letters*, 27(1):327–331, 2022.

Yifan Chen, Xiangrong Zhang, Guanchun Wang, Ziyi Liu, and Licheng Jiao. Msn: Rethinking cross-domain automatic modulation recognition via decoupled modulation signal normalization. *IEEE Transactions on Cognitive Communications and Networking*, 2026.

Richard Hedley Clarke. A statistical theory of mobile-radio reception. *Bell system technical journal*, 47(6):957–1000, 1968.

Octavia A Dobre, Ali Abdi, Yeheskel Bar-Ness, and Wei Su. Survey of automatic modulation classification techniques: classical approaches and new trends. *IET communications*, 1(2):137–156, 2007.

Mingyang Du, Jifei Pan, and Daping Bi. A contrastive learner for automatic modulation classification. *IEEE Transactions on Wireless Communications*, 24(4):3575–3589, 2025.

Peyré Gabriel and Cuturi Marco. Computational optimal transport with applications to data sciences. *Foundations and trends® in machine learning*, 11(5-6):355–607, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Dehua Hong, Zilong Zhang, and Xiaodong Xu. Automatic modulation classification using recurrent neural networks. In *2017 3rd IEEE international conference on computer and communications (ICCC)*, pp. 695–700. IEEE, 2017.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Wenjie Huang, Rong Fan, and Yishan Su. Diffusion model-based data augmentation with mel spectrogram for automatic modulation recognition in underwater acoustic communication. *IEEE Internet of Things Journal*, 2026.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yaoqi Liu, Jin Wang, Hui Wang, and Chuan Shi. TS-DDAE: A novel temporal-spectral denoising diffusion autoencoder for wireless signal recognition model pre-training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RKDkqkkZ5m.

Asoke K Nandi and Elsayed Elsayed Azzouz. Algorithms for automatic modulation recognition of communication signals. *IEEE Transactions on communications*, 46(4):431–436, 1998.

Timothy J O'shea and Nathan West. Radio machine learning dataset generation with gnu radio. In *Proceedings of the GNU radio conference*, volume 1, 2016.

Do-Hyun Park, Min-Wook Jeon, Jinwoo Jeong, Isaac Sim, Sangbom Yun, Junghyun Seo, and Hyoung-Nam Kim. Noise-aware ensemble learning for efficient radar modulation recognition. *IEEE Internet of Things Journal*, 12(14):28937–28949, 2025.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017.

Mingyuan Shao, Dingzhao Li, Shaohua Hong, Jie Qi, and Haixin Sun. Iqformer: A novel transformer-based model with multi-modality fusion for automatic modulation recognition. *IEEE Transactions on Cognitive Communications and Networking*, 11(3):1623–1634, 2024.

Kürşat Tekbıyık, Ali Rıza Ekti, Ali Görçin, Güneş Karabulut Kurt, and Cihat Keçeci. Robust and fast automatic modulation classification with cnn under multipath fading channels. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6. IEEE, 2020.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Guanchun Wang, Ziyi Liu, Xiangrong Zhang, Yifan Chen, Yifang Zhang, Jin Zhu, and Licheng Jiao. Pid: A parameter-efficient isolation domain-incremental learning framework for signal modulation classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2025a.

Xiang Wang, Yurui Zhao, and Zhitao Huang. A survey of deep transfer learning in automatic modulation classification. *IEEE Transactions on Cognitive Communications and Networking*, 11(3):1357–1381, 2025b.

Yu Wang, Miao Liu, Jie Yang, and Guan Gui. Data-driven deep learning for automatic modulation recognition in cognitive radios. *IEEE Transactions on Vehicular Technology*, 68(4):4074–4077, 2019.

Junde Wu, Ziyue Wang, Mingxuan Hong, Wei Ji, Huazhu Fu, Yanwu Xu, Min Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation. *Medical image analysis*, 102:103547, 2025.

Hantong Xing, Shuang Wang, Chenxu Wang, Dou Quan, Pengtao Li, Huaji Zhou, and Licheng Jiao. C2f-net: Coarse-to-fine feature alignment for cross-channel automatic modulation classification. *IEEE Transactions on Cognitive Communications and Networking*, 12:4610–4623, 2025.

Zheng Yang, Weiwei Jiang, Sai Huang, Shuo Chang, Jiashuo He, Yifan Zhang, and Zhiyong Feng. Snr-enhanced automatic modulation classification in artificial intelligence of things for consumer electronics. *IEEE Transactions on Consumer Electronics*, 71(1):2051–2060, 2025.

Rui Zeng, Zhilin Lu, Xudong Zhang, Jintao Wang, and Jian Wang. Convolutional neural network assisted transformer for automatic modulation recognition under large cfos and sros. *IEEE Signal Processing Letters*, 31:741–745, 2024.

Fuxin Zhang, Chunbo Luo, Jialang Xu, and Yang Luo. An efficient deep learning model for automatic modulation recognition based on parameter estimation and transformation. *IEEE Communications Letters*, 25(10):3287–3290, 2021.

Fuxin Zhang, Chunbo Luo, Jialang Xu, Yang Luo, and Fu-Chun Zheng. Deep learning based automatic modulation recognition: Models, datasets, and challenges. *Digital Signal Processing*, 129: 103650, 2022.

Jiawei Zhang, Tiantian Wang, Zhixi Feng, and Shuyuan Yang. Toward the automatic modulation classification with adaptive wavelet network. *IEEE Transactions on Cognitive Communications and Networking*, 9(3):549–563, 2023.

Maomao Zhang, Guofeng Wei, Peng Tang, Xue Ni, Guoru Ding, and Huali Wang. Semi-supervised domain adaptation for automatic modulation recognition in unseen scenarios. *IEEE Transactions on Cognitive Communications and Networking*, 11(3):1609–1622, 2024a.

Xiangrong Zhang, Yifan Chen, Guanchun Wang, Yifang Zhang, and Licheng Jiao. Edda: an efficient divide-and-conquer domain adapter for automatics modulation recognition. *IEEE Journal of Selected Topics in Signal Processing*, 19(1):140–153, 2024b.

Qinghe Zheng, Penghui Zhao, Yang Li, Hongjun Wang, and Yang Yang. Spectrum interference-based two-level data augmentation method in deep learning for automatic modulation classification. *Neural Computing and Applications*, 33(13):7723–7745, 2021.

Qinghe Zheng, Xinyu Tian, Lisu Yu, Abdussalam Elhanashi, and Sergio Saponara. Recent advances in automatic modulation classification technology: Methods, results, and prospects. *International Journal of Intelligent Systems*, 2025(1):4067323, 2025.

# A Theoretical Justification

In Section 3.2, we formulate latent recovery as transporting a degraded class-conditional distribution toward its corresponding clean empirical endpoint. Directly estimating the sample-wise Monge map to this empirical endpoint, however, requires resolving transport correspondences between degraded and clean representations. LOFT therefore adopts the prototype-directed displacement

$$u_y^{\mathrm{proto}}(z) = \mu_y - z \tag{15}$$

as a tractable surrogate target.

In this section, we provide a formal justification for this surrogate. Specifically, we show that the population counterpart of the prototype regression objective controls the only adapter-dependent term in an upper bound on the Wasserstein-2 distance between the corrected latent distribution and the clean empirical endpoint. The resulting bound also makes explicit the approximation introduced by replacing sample-wise empirical transport with prototype-directed recovery.

## A.1 Preliminaries

Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the space of probability measures on $\mathbb{R}^d$ with finite second moments. For any $\rho, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, the squared Wasserstein-2 distance is defined as

$$\mathcal{W}_2^2(\rho, \nu) = \inf_{\pi \in \Pi(\rho,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|z - z'\|_2^2 \, \mathrm{d}\pi(z, z'), \tag{16}$$

where $\Pi(\rho, \nu)$ denotes the set of couplings whose marginals are $\rho$ and $\nu$ (Villani et al., 2009).

Recall from Section 3.2 that the clean feature bank for class $y$ is

$$\mathcal{Z}_y^{\mathrm{clean}} = \{g(x_i) \mid y_i = y,\ s_i \geq 10\,\mathrm{dB}\}, \tag{17}$$

and its empirical endpoint distribution is

$$\nu_y = \frac{1}{|\mathcal{Z}_y^{\mathrm{clean}}|} \sum_{z_i \in \mathcal{Z}_y^{\mathrm{clean}}} \delta_{z_i}. \tag{18}$$

The corresponding clean semantic prototype is the empirical first moment

$$\mu_y = \mathbb{E}_{z' \sim \nu_y}[z'] = \frac{1}{|\mathcal{Z}_y^{\mathrm{clean}}|} \sum_{z_i \in \mathcal{Z}_y^{\mathrm{clean}}} z_i. \tag{19}$$

Equivalently, $\mu_y$ is the Euclidean Fréchet mean of $\nu_y$ under the squared Euclidean distance (Gabriel & Marco, 2019):

$$\mu_y = \arg\min_{m \in \mathbb{R}^d} \mathbb{E}_{z' \sim \nu_y} \left[\|z' - m\|_2^2\right]. \tag{20}$$

For notational clarity, we define the empirical covariance using the same normalization as the empirical measure $\nu_y$:

$$\Sigma_y = \frac{1}{|\mathcal{Z}_y^{\mathrm{clean}}|} \sum_{z_i \in \mathcal{Z}_y^{\mathrm{clean}}} (z_i - \mu_y)(z_i - \mu_y)^\top. \tag{21}$$

### A.2 Adapter-Induced Corrected Distribution

We next formalize the distribution induced by the transport adapter. For a degraded sample, the correction state is $t = 0$, and the adapter produces the residual displacement $v_\phi(z, 0)$. We therefore define the correction map as follows.

**Definition 1** (Adapter-Induced Correction Map)**.** *For a degraded class-conditional latent distribution $p(z \mid y, s)$, define the adapter-induced correction map*

$$F_\phi(z) = z + v_\phi(z, 0). \tag{22}$$

*The corresponding corrected class-conditional distribution is the push-forward measure*

$$q_{\phi,y,s} = (F_\phi)_\# p(\cdot \mid y, s). \tag{23}$$

The distribution $q_{\phi,y,s}$ therefore represents the latent distribution after applying the learned correction to degraded representations of class $y$ at SNR $s$.

### A.3 Wasserstein Upper Bound

We establish the connection between prototype-directed regression and distribution-level recovery.

**Theorem 1** (Prototype-Directed Wasserstein Upper Bound)**.** *Let $p(\cdot \mid y, s) \in \mathcal{P}_2(\mathbb{R}^d)$ be a degraded class-conditional latent distribution, and let $q_{\phi,y,s} = (F_\phi)_\# p(\cdot \mid y, s)$ be the corrected distribution induced by Definition 1. Let $\nu_y$ denote the clean empirical endpoint, with prototype $\mu_y$ and covariance $\Sigma_y$ defined in Equations 19 and 21, respectively. Then*

$$\mathcal{W}_2(q_{\phi,y,s}, \nu_y) \le \sqrt{\mathbb{E}_{z\sim p(\cdot|y,s)}\left[\left\|v_\phi(z,0) - u_y^{\mathrm{proto}}(z)\right\|_2^2\right]} + \sqrt{\mathrm{Tr}(\Sigma_y)}, \tag{24}$$

*where*

$$u_y^{\mathrm{proto}}(z) = \mu_y - z. \tag{25}$$

*Proof.* Introduce the Dirac measure centered at the clean semantic prototype,

$$\widetilde{\nu}_y = \delta_{\mu_y}. \tag{26}$$

Since $\mathcal{W}_2$ is a metric on $\mathcal{P}_2(\mathbb{R}^d)$, the triangle inequality gives

$$\mathcal{W}_2(q_{\phi,y,s}, \nu_y) \le \mathcal{W}_2(q_{\phi,y,s}, \delta_{\mu_y}) + \mathcal{W}_2(\delta_{\mu_y}, \nu_y). \tag{27}$$

We analyze the two terms on the right-hand side separately.

**Adapter-dependent term.** Because the target measure is the Dirac mass $\delta_{\mu_y}$, every point in $q_{\phi,y,s}$ must be transported to $\mu_y$. Hence,

$$\begin{aligned}\mathcal{W}_2^2(q_{\phi,y,s}, \delta_{\mu_y}) &= \mathbb{E}_{\hat{z}\sim q_{\phi,y,s}}\left[\|\hat{z} - \mu_y\|_2^2\right] \\ &= \mathbb{E}_{z\sim p(\cdot|y,s)}\left[\|F_\phi(z) - \mu_y\|_2^2\right],\end{aligned} \tag{28}$$

where the second equality follows from $q_{\phi,y,s} = (F_\phi)_\# p(\cdot \mid y, s)$.

Substituting the correction map $F_\phi(z) = z + v_\phi(z, 0)$ yields

$$\begin{aligned}\mathcal{W}_2^2(q_{\phi,y,s}, \delta_{\mu_y}) &= \mathbb{E}_{z\sim p(\cdot|y,s)}\left[\|z + v_\phi(z,0) - \mu_y\|_2^2\right] \\ &= \mathbb{E}_{z\sim p(\cdot|y,s)}\left[\|v_\phi(z,0) - (\mu_y - z)\|_2^2\right] \\ &= \mathbb{E}_{z\sim p(\cdot|y,s)}\left[\left\|v_\phi(z,0) - u_y^{\mathrm{proto}}(z)\right\|_2^2\right].\end{aligned} \tag{29}$$

Thus, the population prototype-regression objective is exactly the squared Wasserstein-2 distance between the adapter-corrected distribution and the prototype Dirac measure.

**Clean-endpoint dispersion term.** Similarly, because one of the measures is a Dirac mass,

$$\begin{aligned}\mathcal{W}_2^2(\delta_{\mu_y}, \nu_y) &= \mathbb{E}_{z' \sim \nu_y}\left[\|z' - \mu_y\|_2^2\right] \\ &= \frac{1}{|\mathcal{Z}_y^{\text{clean}}|} \sum_{z_i \in \mathcal{Z}_y^{\text{clean}}} \|z_i - \mu_y\|_2^2 \\ &= \operatorname{Tr}(\Sigma_y),\end{aligned} \tag{30}$$

where the last equality follows directly from the definition of $\Sigma_y$ in Equation 21. Therefore,

$$\mathcal{W}_2(\delta_{\mu_y}, \nu_y) = \sqrt{\operatorname{Tr}(\Sigma_y)}. \tag{31}$$

Substituting Equations 29 and 31 into the triangle inequality in Equation 27 gives

$$\mathcal{W}_2(q_{\phi,y,s}, \nu_y) \le \sqrt{\mathbb{E}_{z \sim p(\cdot|y,s)}\left[\left\|v_\phi(z,0) - u_y^{\text{proto}}(z)\right\|_2^2\right]} + \sqrt{\operatorname{Tr}(\Sigma_y)}, \tag{32}$$

which proves the result. □

### A.4 Implications for Prototype-Directed Training

Theorem 1 connects the prototype-directed training objective to the distribution-level recovery formulation introduced in Section 3.2. In particular, the first term of Equation 24 depends on the adapter parameters $\phi$, whereas the second term depends only on the fixed clean feature bank.

This observation leads to the following corollary.

**Corollary 1** (Control of the Adapter-Dependent Bound)**.** *Define the population prototype-regression loss for degraded samples of class $y$ at SNR $s$ as*

$$\mathcal{L}_{\text{proto}}(\phi; y, s) = \mathbb{E}_{z \sim p(\cdot|y,s)}\left[\left\|v_\phi(z,0) - u_y^{\text{proto}}(z)\right\|_2^2\right]. \tag{33}$$

*Then*

$$\mathcal{W}_2(q_{\phi,y,s}, \nu_y) \le \sqrt{\mathcal{L}_{\text{proto}}(\phi; y, s)} + \sqrt{\operatorname{Tr}(\Sigma_y)}. \tag{34}$$

*Consequently, minimizing $\mathcal{L}_{\text{proto}}(\phi; y, s)$ minimizes the adapter-dependent term in this upper bound, while $\sqrt{\operatorname{Tr}(\Sigma_y)}$ remains fixed with respect to $\phi$.*

*Proof.* Equation 34 follows immediately from Theorem 1 and the definition of $\mathcal{L}_{\text{proto}}(\phi; y, s)$ in Equation 33. Since the clean feature bank is fixed during adapter training, $\mu_y$ and $\Sigma_y$ are independent of $\phi$. Therefore, only the prototype-regression term is affected by optimization of the adapter. □

**Corollary 2** (Exact Prototype Recovery)**.** *Suppose the adapter exactly realizes the prototype-directed displacement for $p(\cdot \mid y, s)$-almost every $z$, i.e.,*

$$v_\phi(z, 0) = u_y^{\text{proto}}(z) = \mu_y - z. \tag{35}$$

*Then*

$$q_{\phi,y,s} = \delta_{\mu_y}, \tag{36}$$

*and consequently*

$$\mathcal{W}_2(q_{\phi,y,s}, \nu_y) = \sqrt{\operatorname{Tr}(\Sigma_y)}. \tag{37}$$

*Proof.* Under Equation 35, the correction map satisfies

$$F_\phi(z) = z + v_\phi(z, 0) = z + (\mu_y - z) = \mu_y \tag{38}$$

for $p(\cdot \mid y, s)$-almost every $z$. Therefore, the push-forward of $p(\cdot \mid y, s)$ under $F_\phi$ is the Dirac measure $\delta_{\mu_y}$. Equation 37 then follows from Equation 31. □

**Remark 1** (Scope of the Prototype-Directed Approximation)**.** *The above results clarify both the justification and the limitation of the prototype-directed surrogate. The surrogate does not claim to recover the exact sample-wise Monge map from $p(\cdot \mid y, s)$ to the endpoint $\nu_y$. Instead, it replaces sample-wise transport estimation with regression toward the first moment of the clean endpoint.*

*Theorem 1 shows that this regression has a direct distributional interpretation: its population objective is exactly $\mathcal{W}_2^2(q_{\phi,y,s}, \delta_{\mu_y})$, which controls the adapter-dependent component of an upper bound on $\mathcal{W}_2(q_{\phi,y,s}, \nu_y)$. The remaining term $\sqrt{\mathrm{Tr}(\Sigma_y)}$ quantifies the dispersion of the clean empirical endpoint around its prototype and is independent of the adapter.*

*Accordingly, prototype-directed recovery explicitly controls class-conditional displacement toward the clean semantic center, but it does not explicitly reconstruct higher-order intra-class structure of $\nu_y$. This trade-off enables LOFT to avoid solving an explicit discrete transport problem between degraded and clean empirical samples during adapter training, while retaining a principled connection to the clean endpoint geometry.*

Table 4: Summary of the benchmark datasets used in our experiments.

| Dataset | Classes | Shape | Total Samples | SNR (dB) | Signal Impairments |
|---|---|---|---|---|---|
| RML2016.10A | 11 | $2 \times 128$ | 220,000 | $-20$ to 18 | Standard (Rayleigh fading, multipath, AWGN). |
| RML2016.10B | 10 | $2 \times 128$ | 1,200,000 | $-20$ to 18 | Severe realistic (standard plus CFO and SRO). |

# B DETAILS OF THE EXPERIMENTS

## B.1 DETAILS OF THE DATASETS

In this section, we provide a detailed description of the two benchmark datasets utilized in our experiments to evaluate the proposed LOFT framework. Both datasets are widely adopted in AMR research and feature complex-valued I/Q sequences (O'shea & West, 2016).

**RML2016.10A** is a standard benchmark containing $220,000$ complex-valued signal samples. The underlying signals are generated using ASCII-encoded text sourced from "Serial Episode" and Shakespeare's Gutenberg works. It encompasses $11$ distinct modulation schemes, including three analog modulations (WB-FM, AM-SSB, and AM-DSB) and eight digital modulations (BPSK, QPSK, 8PSK, 16QAM, 64QAM, GFSK, CPFSK, and PAM4). The simulation environment spans $20$ distinct Signal-to-Noise Ratio (SNR) levels, ranging from $-20\,\mathrm{dB}$ to $18\,\mathrm{dB}$ in $2\,\mathrm{dB}$ increments. For each modulation scheme at each SNR level, the dataset provides $1,000$ independent samples. Each sample is formatted as a temporal sequence of length 128 with two channels representing the in-phase (I) and quadrature (Q) components, yielding an input tensor shape of $2 \times 128$.

**RML2016.10B** is a rigorously scaled-up variant of RML2016.10A, designed to evaluate AMR robustness under more severe and realistic channel impairments. It consists of $1,200,000$ samples, dramatically scaling the data volume to $6,000$ samples per modulation scheme at each of the $20$ SNR settings. The modulation pool is refined to $10$ schemes by explicitly removing the AM-SSB class, retaining the remaining two analog and eight digital schemes. Crucially, the simulated wireless channel in RML2016.10B introduces compounded physical-layer distortions. In addition to the standard Rayleigh fading, multipath effects, and Additive White Gaussian Noise (AWGN) present in RML2016.10A, it incorporates rigorous simulations of Center Frequency Offset (CFO) and Sample Rate Clock Offset (SRO). All samples retain the $2 \times 128$ I/Q sequence format. This combination of increased data scale and heightened channel complexity makes RML2016.10B a highly challenging evaluation environment for robustness under representation collapse.

Across all experiments, each dataset is randomly divided into training, validation, and testing subsets with a fixed ratio of 6:2:2. For all baseline methods, the model checkpoint achieving the highest validation accuracy during training is selected for final evaluation on the test set. To ensure a fair comparison and eliminate potential data leakage, the optimization of the LOFT transport adapter is conducted using only the training subset. Specifically, both the construction of clean semantic

prototypes and adapter optimization are performed without access to the validation or testing subsets, ensuring a fully blind evaluation protocol during inference.

### B.2 Details of the Adapter Architecture

As shown in Figure 6, the Transport Adapter is a lightweight module designed to estimate the transport displacement field with minimal additional parameters. It consists of a Time Step Embedder and a bottleneck Transport Network.

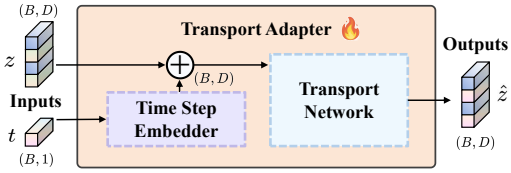



Figure 6: **The detailed architecture of the lightweight Transport Adapter.** The discrete transport state $t \in \mathbb{R}^{B\times 1}$ is encoded by a Time Step Embedder and injected into the latent representation $z \in \mathbb{R}^{B\times D}$ via element-wise addition. A Transport Network subsequently processes this condition-injected feature to generate the corrected output $\hat{z}$.

Given the transport state $t$, we first encode it using sinusoidal positional embeddings with a base frequency of $10,000$, producing a 128-dimensional representation. A two-layer MLP with SiLU activation then projects this embedding to the latent dimension $D$ of the frozen backbone, yielding the condition embedding $e(t)$. The condition is injected into the latent feature through element-wise addition $h = z + e(t)$.

The Transport Network learns the displacement dynamics $v_\phi(z, t)$ through a compact bottleneck design. Specifically, the conditioned feature $h$ is projected to a 64-dimensional hidden space, followed by Layer Normalization and GELU activation, and subsequently mapped back to the original dimension $D$ through a linear layer. This design introduces only approximately $0.05$M additional parameters across all backbones, providing an efficient adaptation mechanism with negligible inference overhead.

### B.3 Details of the Implementation details

All methods are implemented in PyTorch and executed on a single NVIDIA GeForce RTX 4090 GPU. For baseline pretraining, we employ the Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of $10^{-3}$, a weight decay of $10^{-5}$, and a fixed batch size of 512. The learning rate decays by a factor of $0.9$ every $5$ epochs over a maximum of $200$ epochs. For the optimization of the proposed **LOFT** framework, the pretrained backbones are kept strictly frozen. The lightweight transport adapter is trained using the Adam optimizer with a learning rate of $10^{-3}$ and a batch size of $512$. To ensure training stability, gradient clipping is applied with a maximum norm of $1.0$. The clean feature bank $\mathcal{Z}_y^{\text{clean}}$ is constructed using high-SNR training samples ($\geq 10$ dB). The semantic alignment coefficient is set to $\lambda = 1.0$. Notably, the adapter optimization converges extremely fast, requiring merely $3$ epochs to complete, which demonstrates its remarkable parameter and training efficiency. To ensure statistical rigor, all metrics are reported across three independent runs.

**Simulating Real-World Streaming Inference.** In practical wireless communication systems, received signals arrive as continuous data streams. Environmental factors and channel impairments (e.g., weather conditions, fading) typically exhibit temporal coherence, meaning they remain relatively stable over short observation windows (Clarke, 1968). To rigorously simulate this physical reality during evaluation, we group signal samples within specific SNR intervals to form inference batches. This deliberate arrangement preserves the temporal correlation inherent in real-world wireless channels, ensuring that each batch accurately reflects a coherent environmental window. Consequently, this continuous streaming setup fundamentally justifies our batch-level entropy-aware routing mechanism, seamlessly aligning the evaluation protocol with realistic deployment scenarios where consecutive received signals naturally share a similar degradation regime.

## C More Experimental Results

### C.1 Convergence Speed and Training Efficiency

In Section 5, we highlighted that the proposed LOFT adapter is optimized for merely $3$ epochs. To empirically justify this early-stopping hyperparameter and demonstrate the extreme training effi-

ciency of our plug-and-play framework, we track the classification accuracy across the first 5 epochs of adapter training on both the RML2016.10A and RML2016.10B datasets.

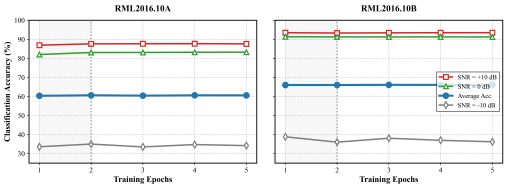



Figure 7: **Evolution of classification accuracy across training epochs.** The visualization tracks the average accuracy alongside representative low ($-10$ dB), critical (0 dB), and high (10 dB) SNR conditions. The transport adapter converges near-optimally within the first two epochs (highlighted in gray) and remains strictly stable across all SNR spectra, rigorously justifying our efficient 3-epoch optimization strategy without risking overfitting.

As illustrated in Figure 7, the transport adapter exhibits an ultra-fast convergence behavior on both benchmarks. Astonishingly, the framework reaches near-optimal performance within the first two epochs. For instance, on the large-scale RML2016.10B dataset, the average recognition accuracy hits $65.99\%$ after a single epoch and remains strictly stable at $65.99\%$ at Epoch 2, with negligible fluctuations thereafter ($66.08\%$ at Epoch 3). Similar rapid stabilization is observed on RML2016.10A, where the average accuracy converges to $60.61\%$ by Epoch 2. Notably, this instantaneous stabilization is consistently maintained across severe noise regimes ($-10$ dB), critical thresholds (0 dB), and favorable channel conditions (10 dB) uniformly. This rapid convergence phenomenon fundamentally validates our geometric premise: rather than learning complex discriminative representations from scratch, the lightweight adapter is simply amortizing a deterministic displacement field (the optimal transport flow) toward already established semantic anchors. The mapping from a degraded distribution to its prototype surrogate is highly structured, allowing the bottleneck network to optimize its vector field almost immediately. Consequently, extending the training beyond 2 or 3 epochs offers strictly diminishing returns. By fixing the optimization strictly to 3 epochs, LOFT achieves profound robustness reconstruction with virtually negligible training overhead, completely eliminating any risk of overfitting the source latent representations.

Table 5: Sensitivity analysis of the coefficient $\lambda$. The results demonstrate that LOFT exhibits extreme stability across a broad range of values, proving its insensitivity to hyperparameter tuning.

| **Coefficient ($\lambda$)** | **$-10$ dB** | **0 dB** | **10 dB** | **Average Acc (%)** |
|---|---|---|---|---|
| $\lambda = 0.1$ | 30.36 | 82.55 | 87.32 | 60.06 |
| $\lambda = 0.5$ | 33.18 | 83.09 | 87.55 | 60.20 |
| $\lambda = 1.0$ | 34.18 | 83.15 | 87.74 | 60.77 |
| $\lambda = 1.5$ | 33.14 | 83.09 | 87.64 | 60.56 |
| $\lambda = 2.0$ | 33.32 | 83.00 | 87.68 | 60.52 |

### C.2 Hyperparameter Sensitivity Analysis

A robust representation alignment framework should not rely on intensive, dataset-specific hyperparameter tuning. In the LOFT objective, the semantic alignment coefficient $\lambda$ balances the optimal transport displacement loss and the cross-entropy regularization. To demonstrate the extreme insensitivity of LOFT to this hyperparameter, we evaluate its performance across a wide spectrum of $\lambda \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$. As presented in Table 5, the performance of LOFT remains remarkably stable across the entire evaluated range. The average recognition accuracy fluctuates within an extremely narrow margin of merely $0.7\%$ (ranging from $60.06\%$ to $60.77\%$). Even under severe noise conditions (e.g., $-10$ dB) and pristine environments (10 dB), the variance across different $\lambda$ values remains negligible. This empirical evidence definitively proves that the geometric constraints enforced by the optimal transport flow are the fundamental drivers of representation recovery, rather than a finely tuned classification loss weight. Consequently, setting $\lambda$ within $[1.0, 2.0]$ yields consistently optimal behavior. We safely fix $\lambda = 1.0$ across all architectures and datasets without any need for per-scenario hyperparameter searches, thereby eliminating the risk of overfitting to specific validation sets.