MIRA: Mutual Information guided calibration for Reliable Test-Time Adaptation

Hyeongyu Kim[1,4*] **Mingyeong Jang**[1*] **Youngjun Song**[1] **GeonHui Han**[1] **Dosik Hwang**[1,2,3†]
[1]School of Electrical and Electronic Engineering, Yonsei University
[2]Department of Radiology and Center for Clinical Imaging Data Science,
College of Medicine, Yonsei University
[3]Artificial Intelligence and Robotics Institute, Korea Institute of Science and Technology
[4]Hyundai Motor Company
{lion4309, mini10211, dosik.hwang}@yonsei.ac.kr
https://hyeongyu-kim.github.io/mira/

Abstract

Test-time adaptation (TTA) updates a deployed model using only unlabeled target data, but standard entropy-based methods often rely on deterministic confidence and can reinforce overconfident errors under distribution shift. We propose MIRA, **M**utual **I**nformation guided calibration for **R**eliable Test-Time **A**daptation, a source-free online adaptation method that uses lightweight last-layer Gaussian perturbations to probe prediction reliability. For each target batch, MIRA reuses a single feature extraction pass and samples perturbed classifiers around the source classifier, forming an efficient local predictive distribution. Rather than fitting a Bayesian posterior, MIRA treats the perturbation scale as a local probing radius and selects a batch-specific scale using mutual information (MI) and prediction agreement. The resulting perturbation-sensitivity signal identifies predictions that are confident and locally stable under classifier perturbations, while suppressing brittle predictions with high stochastic disagreement. MIRA then performs adaptation with an MI-confidence weighted entropy loss, which down-weights perturbation-sensitive or low-confidence samples before they drive entropy minimization. This weighted loss reduces the influence of brittle predictions that can otherwise cause harmful adaptation under distribution shift. The method requires no source data, retraining, model ensembles, or offline posterior fitting, and can be applied as a plug-and-play modification to entropy-minimization TTA. Experiments on distribution-shift benchmarks show improvements in accuracy, calibration, and negative log-likelihood, suggesting that mutual-information-guided perturbation probing and weighted adaptation yield more robust online TTA than deterministic confidence.

1 Introduction

Deep neural networks perform well under i.i.d. training and test sampling from $p(x, y)$ [6, 11, 18]. Deployment often violates this assumption [13]. Under covariate shift, $p_{\mathrm{src}}(x) \neq p_{\mathrm{tgt}}(x)$ with unchanged label space and task semantics [32]. Such shifts, including corruptions from weather, sensor noise, styles, or acquisition conditions, degrade both accuracy and calibration [30]. Confident errors are especially problematic. Training-time domain generalization also seeks robustness to unseen domains, including through content–style disentanglement and style mixing [16].

*These authors contributed equally to this work.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Complementing these training-time strategies, test-time adaptation (TTA) adapts a pre-trained model on unlabeled target data during inference, without access to source data or target labels [36, 38, 41, 26, 14, 15, 35]. Existing methods improve robustness through normalization updates, entropy minimization, sample filtering, or consistency regularization [38, 28, 40, 43].

Despite this progress, most entropy-based TTA methods remain deterministic. They adapt from a single prediction of a single maximum a posteriori (MAP) network and often treat confidence as reliability. Under distribution shift, however, high confidence may reflect overconfidence rather than correctness [10, 30]. Entropy minimization explicitly sharpens target predictions [2]; while useful when predictions are correct, it can also reinforce erroneous predictions and worsen overconfidence under severe shift [2, 44]. Thus, reliable TTA requires more than confidence maximization.

Recent uncertainty-aware TTA methods estimate target reliability, filter updates, or incorporate stochasticity into adaptation [46, 23, 37, 4, 47]. However, many obtain reliability through mechanisms that complicate online deployment, such as ensembles [46], repeated stochastic forward passes [23], source-dependent posterior fitting [46, 4], uncertainty-specific regularizers [37], or oracle feedback with reinforcement-learning-based adaptation [24]. PETAL fits a SWAG-diagonal posterior during source training and uses its log density to regularize lifelong teacher–student adaptation without revisiting source data [4]. These costs conflict with the practical requirements of TTA: source-freeness, low overhead, immediacy, and plug-and-play deployment.

A related line of work studies efficient uncertainty estimation for neural networks [17, 8]. Last-layer Laplace approximations place a lightweight distribution around the final classifier of a deterministic MAP network [17, 5], avoiding full Bayesian neural networks, ensembles, and full-network curvature estimation. Post-hoc Bayesianization methods such as ABNN instead attach Bayesian adaptation layers to normalization layers and fine-tune pretrained models [8]. Yet these methods are typically offline post-hoc uncertainty estimators that rely on source data, validation data, or posterior fitting, making them unsuitable for online source-free TTA. At the same time, recent results such as ICLA show that useful uncertainty behavior can arise from simple isotropic approximations when the perturbation scale is chosen appropriately [45]. This suggests a more pragmatic question: can a lightweight perturbation scale be selected directly from the current unlabeled target batch, and can the resulting stochastic predictions be used to weight adaptation?

We propose **MIRA**, a lightweight source-free online TTA framework that perturbs only the last-layer classifier of a deployed MAP network and uses stochastic predictions to identify reliable target samples. MIRA selects an informative perturbation scale per batch and applies an MI-confidence weighted entropy loss, promoting stable high-confidence predictions while down-weighting perturbation-sensitive samples. Figure 1 summarizes the probe and reliability-weighted update. Because backbone features are computed once and reused across classifier samples, MIRA is efficient and plug-and-play. Across covariate-shift benchmarks, it improves accuracy, calibration, and NLL without source data, posterior fitting, ensembles, full-network stochastic inference, Hessian estimation, or retraining.

Contributions.

• We propose **MIRA**, a source-free online TTA framework that performs lightweight Gaussian perturbation probing around the last-layer classifier and selects a target-batch perturbation scale as an online reliability parameter, producing stochastic predictions useful for reliable adaptation and calibration without source data, validation data, or offline fitting.

• We present an MI-confidence weighted loss that treats MI as perturbation-induced disagreement and down-weights perturbation-sensitive or low-confidence samples before entropy minimization.

• We demonstrate consistent gains in accuracy and calibration across distribution-shift benchmarks, while keeping computation low through cached last-layer Monte Carlo sampling.

2 Background

2.1 Test-Time Adaptation and Uncertainty

Test-time adaptation (TTA) updates a source-trained model during inference using only unlabeled target samples [36, 38, 26, 41]. We consider the standard source-free covariate-shift setting, where $p_{\mathrm{src}}(x) \neq p_{\mathrm{tgt}}(x)$ while the label space and task semantics remain unchanged.

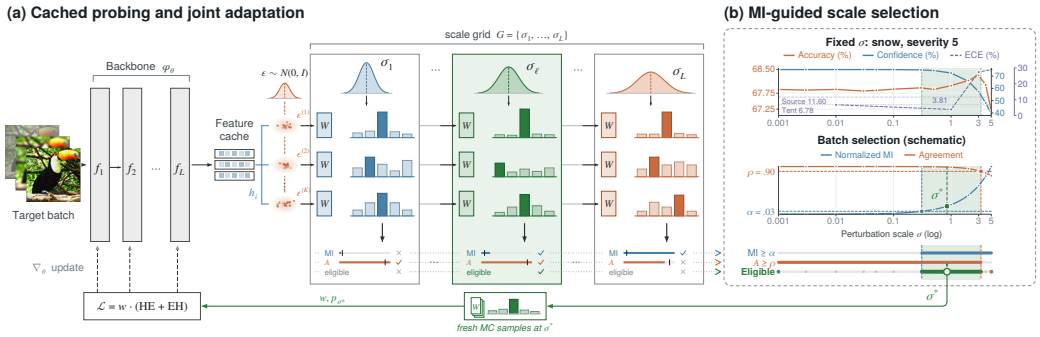

Figure 1: **MIRA overview.** (a) Cached features support Gaussian classifier probes with shared noise across candidate radii. At the selected σ^*, fresh Monte Carlo samples supply detached MI–confidence weights and a stochastic entropy update through one backward pass. (b) Fixed-scale annotations report measured results; MI/agreement curves and eligibility bars schematically illustrate radius selection. HE and EH denote entropy of the mean prediction and mean predictive entropy.

Given an incoming target batch $\{x_i\}_{i=1}^{B}$, most TTA methods adapt the model by updating normalization statistics, minimizing prediction entropy, enforcing consistency, or selecting reliable samples for update [25, 38, 28, 29, 43]. A representative objective is entropy minimization,

$$\mathcal{L}_{\mathrm{ent}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{C} p_\theta(y = c \mid x_i)\log p_\theta(y = c \mid x_i), \tag{1}$$

which encourages confident predictions on the target batch. While effective, this deterministic objective can treat confidence as reliability. Under severe shift, incorrect but low-entropy predictions may be reinforced, leading to overconfident or poorly calibrated adaptation.

Recent methods address this issue by incorporating uncertainty into TTA. EATA and SAR filter unreliable samples or unstable updates [28, 29]; EATA-C models data and model uncertainty for calibration [37]; AETTA uses dropout-based disagreement to estimate adaptation quality without labels [23]; SSA stabilizes online adaptation through time-varying covariance [21]; and BACS and PETAL introduce Bayesian or probabilistic formulations of adaptation [46, 4]. Uncertainty-constrained pseudo-label refinement also supports volumetric self-training in cross-modality medical domain adaptation [33]. Our goal is to obtain a lightweight predictive distribution directly from each target batch and use it to guide adaptation, **without source data**, offline posterior fitting, or repeated full-network sampling.

2.2 Perturbation-Based Reliability Probing

Predictive reliability can be probed through disagreement under stochastic predictors or controlled perturbations. Monte Carlo dropout, deep ensembles, MEMO, and weight-space Bayesian methods capture uncertainty through stochastic subnetworks, independently trained models, input augmentations, or parameter distributions [9, 20, 43, 3, 30]. These approaches may require repeated full-network inference, multiple models, source-side training, or additional fitting. Self-supervised representation learning can structure features through disentanglement [22], and pretrained feature spaces can support auxiliary self-supervision and consistency objectives [34]. MIRA uses the deployed representation for reliability probing: it perturbs only the last-layer classifier and reuses cached backbone features within each batch, following cached-feature uncertainty estimators [5] and evidence that simplified perturbation scales can yield useful uncertainty behavior [45]. From the current unlabeled target batch, MIRA selects a Gaussian perturbation scale. MI measures perturbation-induced predictive disagreement, while prediction agreement checks class-identity stability. MIRA uses these perturbation responses to regulate sample-wise adaptation, directly linking predictive reliability to test-time optimization.

3 Motivation & Observation

Deterministic entropy-based TTA sharpens target predictions by minimizing the entropy of the model's own outputs. This is effective when predictions are correct, but it treats low entropy

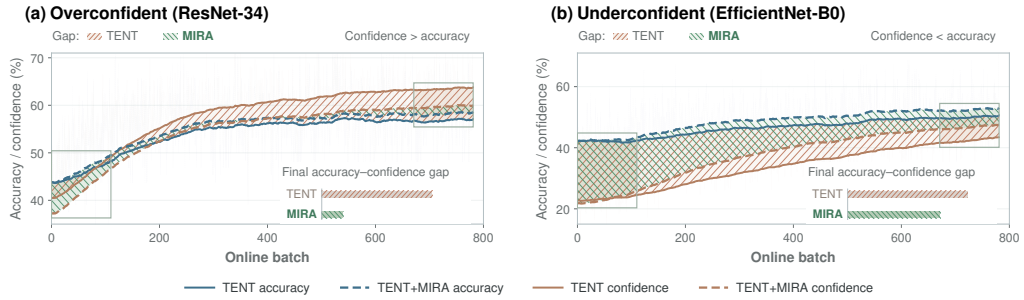

Figure 2: Online accuracy and confidence during TENT and TENT+MIRA on ImageNet-C fog (severity 5, EMA coefficient 0.01). MIRA narrows the gap in both (a) the overconfident ResNet-34 and (b) the underconfident EfficientNet-B0. Shading and bars show the absolute accuracy–confidence gap, rather than ECE.

as evidence of reliability without testing whether the resulting confidence is actually supported. Consequently, confidence is not anchored to accuracy during adaptation. Figure 2 illustrates this on ImageNet-C fog: under TENT, confidence stays above accuracy for ResNet-34, whereas it stays below accuracy for EfficientNet-B0. In both cases, the gap persists throughout the stream rather than closing as adaptation proceeds.

These observations motivate an additional reliability signal for adaptation. Entropy minimization can reinforce the model's current prediction even when it is incorrect. A single deterministic confidence score does not reveal whether that prediction is sensitive to small changes in the classifier. We therefore ask whether prediction stability under controlled classifier perturbations can provide an additional signal for regulating adaptation.

To examine this, we sweep fixed perturbation scales on CIFAR-100-C snow at severity 5 with ResNeXt (Figure 1(b)). Starting from a deployed MAP classifier, we perturb only the last-layer weights,

$$W_\sigma^{(k)} = W_{\mathrm{MAP}} + \sigma S \odot \epsilon_W^{(k)}, \qquad \epsilon_W^{(k)} \sim \mathcal{N}(0, I). \tag{2}$$

Here S has the same shape as W_{MAP} and specifies the base perturbation scale for each weight; $\odot$ denotes element-wise multiplication. Section 4 specifies the corresponding weight and bias scale factors $A_{\mathcal{B}}$ and $a_{\mathcal{B}}$, with construction details in Appendix D.4. We sweep the perturbation scale σ on the incoming target stream. Larger σ induces stronger classifier stochasticity. This sweep is not intended to recover a Bayesian posterior; it probes how the target model responds as the classifier is gradually perturbed.

The sweep reveals two key observations.

1. Perturbation scale can control confidence and calibration. When σ is very small, perturbed classifiers are nearly identical to the deterministic MAP model, so overconfident predictions remain unchanged. As σ increases, perturbation averaging lowers confidence for predictions unstable under classifier perturbations. In an intermediate regime, this reduces unsupported confidence while preserving most predictions, improving ECE and NLL. If σ becomes too large, perturbed classifiers cross decision regions, causing confidence collapse and degraded probabilistic metrics.

2. Perturbation search can also improve adaptation accuracy. The useful regime is not merely a confidence-smoothing region. During TTA, the perturbation scale also changes the adaptation signal: brittle MAP predictions contribute less, while perturbation-stable predictions have greater influence. This can suppress erroneous high-confidence updates and improve accuracy. Thus, even without posterior fitting, searching a lightweight perturbation family can reveal scales that improve accuracy, calibration, or both.

These observations motivate MIRA. They suggest viewing σ not as a posterior variance but as a local probing radius around the deployed classifier. A useful radius should be large enough to expose locally sensitive predictions, yet small enough to preserve class identity for stable samples; Appendix A formalizes this through local margin crossings. Rather than fixing or pre-fitting the perturbation scale, MIRA searches each unlabeled target batch for this stable perturbation-response

regime using MI and perturbation agreement, and uses the resulting stochastic predictions to compute MI–confidence weights for adaptation (Section 4).

4 Method

MIRA performs source-free online test-time adaptation using lightweight Gaussian perturbations of the final classifier. Given a source-trained model, we decompose the prediction into a feature extractor and a linear classifier,

$$z(x) = W\phi_\theta(x) + b, \qquad p(y \mid x) = \mathrm{softmax}(z(x)), \tag{3}$$

where ϕ_θ denotes the backbone representation and W, b denote the last-layer classifier. MIRA updates only the parameters adapted by the underlying TTA method, such as normalization affine parameters, while W, b remain fixed. The perturbations serve two roles: they define a stochastic adaptation objective and provide a reliability signal for weighting it.

4.1 Last-Layer Gaussian Reliability Probe

For an incoming target batch $\mathcal{B} = \{x_i\}_{i=1}^B$, MIRA first computes the backbone features once, $h_i = \phi_\theta(x_i)$. It then samples K perturbed last-layer classifiers for each perturbation scale σ:

$$W_\sigma^{(k)} = W + \sigma A_\mathcal{B} \odot \epsilon_W^{(k)}, \qquad b_\sigma^{(k)} = b + \sigma a_\mathcal{B} \odot \epsilon_b^{(k)}, \qquad \epsilon_W^{(k)}, \epsilon_b^{(k)} \sim \mathcal{N}(0, I). \tag{4}$$

Here $A_\mathcal{B}$ and $a_\mathcal{B}$ are fixed-rule scale factors (see Appendix D.4), so only the scalar radius σ is selected online. Within each batch, the same noise $\epsilon^{(k)}$ is shared across all scales, so candidate classifiers at different σ lie along the same perturbation directions and differ only in radius. This coupling helps reduce Monte Carlo variability in comparisons across candidate scales.

For each sampled classifier, MIRA computes

$$z_{i,\sigma}^{(k)} = W_\sigma^{(k)} h_i + b_\sigma^{(k)}, \qquad p_{i,\sigma}^{(k)} = \mathrm{softmax}(z_{i,\sigma}^{(k)}), \qquad \bar{p}_{i,\sigma} = \frac{1}{K}\sum_{k=1}^K p_{i,\sigma}^{(k)}. \tag{5}$$

From these stochastic predictions, MIRA computes predictive entropy, expected entropy, and perturbation-induced mutual information:

$$\mathrm{HE}_{i,\sigma} = H(\bar{p}_{i,\sigma}), \qquad \mathrm{EH}_{i,\sigma} = \frac{1}{K}\sum_{k=1}^K H(p_{i,\sigma}^{(k)}), \qquad \mathrm{MI}_{i,\sigma} = \mathrm{HE}_{i,\sigma} - \mathrm{EH}_{i,\sigma}. \tag{6}$$

Here HE measures uncertainty of the averaged stochastic prediction, EH measures the average uncertainty of individual sampled classifiers, and MI measures disagreement induced by perturbation. MIRA also computes prediction agreement,

$$a_{i,\sigma} = \frac{1}{K}\sum_{k=1}^K \mathbf{1}\left[\arg\max_y p_{i,\sigma}^{(k)}(y) = \arg\max_y \bar{p}_{i,\sigma}(y)\right], \tag{7}$$

which checks whether sampled classifiers preserve the same class identity as the Monte Carlo mean prediction.

4.2 Mutual-Information-Guided Scale Selection

The perturbation scale controls how strongly the local classifier neighborhood is probed. As σ increases, sampled classifiers deviate more from the deployed classifier and their predictions begin to disagree. A moderate rise in mutual information can reveal local prediction sensitivity that deterministic confidence misses. However, excessive disagreement is undesirable: when σ is too large, sampled classifiers frequently cross decision regions, and the resulting disagreement reflects predictive collapse rather than useful sensitivity. MIRA therefore selects a batch-specific scale from $\mathcal{G} = \{\sigma_1, \ldots, \sigma_L\}$ by seeking disagreement that remains class-consistent.

For each $\sigma_\ell \in \mathcal{G}$, MIRA computes the batch-level mutual information and prediction agreement,

$$M(\sigma_\ell) = \frac{1}{B}\sum_{i=1}^B \mathrm{MI}_{i,\sigma_\ell}, \qquad A(\sigma_\ell) = \frac{1}{B}\sum_{i=1}^B a_{i,\sigma_\ell}. \tag{8}$$

Here $M(\sigma_\ell)$ measures perturbation-induced disagreement and $A(\sigma_\ell)$ compares sampled top-1 classes with the Monte Carlo mean. Let $M_{\min} = \min_j M(\sigma_j)$ and $T_M = M_{\min} + \alpha[\max_j M(\sigma_j) - M_{\min}]$. MIRA selects the earliest informative, high-agreement candidate:

$$\sigma^* = \min_{\sigma_\ell \in \mathcal{G}} \{\sigma_\ell : M(\sigma_\ell) \geq T_M,\ A(\sigma_\ell) \geq \rho\}. \tag{9}$$

The MI condition seeks an informative response relative to the observed MI range, while the agreement condition checks whether the candidate remains class-consistent. The complete selection procedure, including interpolation between grid points and the case in which no candidate satisfies both conditions, is given in Appendix D.6. When both conditions hold, $1 - A(\sigma^*) \leq 1 - \rho$ bounds the batch-average empirical disagreement on the selection samples; it does not guarantee correctness, preservation of the unperturbed MAP class, or agreement after fresh sampling. The selected σ^* is a probing radius rather than a posterior variance. Fresh Monte Carlo samples at this radius provide the adaptation loss below.

4.3 Reliability-Gated Adaptation

After selecting σ^*, MIRA uses the stochastic predictions at this scale to decide which target samples should drive adaptation. Standard entropy minimization treats low entropy as reliable evidence and therefore reinforces the model's current predictions. MIRA instead asks whether this confidence is stable under controlled classifier perturbations. A sample with low entropy but high perturbation-induced MI is locally brittle under the chosen perturbation geometry: its confidence depends strongly on the particular classifier sample. Such brittleness does not prove that the prediction is incorrect, but it indicates that the sample should not dominate an entropy-minimization update.

Let

$$\mathrm{HE}_i = H(\bar{p}_{i,\sigma^*}), \qquad \mathrm{EH}_i = \frac{1}{K}\sum_{k=1}^{K} H(p_{i,\sigma^*}^{(k)}), \qquad m_i = \mathrm{MI}_{i,\sigma^*}. \tag{10}$$

MIRA first assigns each sample an MI-based weight that decreases with perturbation-induced disagreement,

$$w_i^{\mathrm{MI}} = \frac{\exp[-(m_i - \bar{m})/\tau]}{B^{-1}\sum_{j=1}^{B}\exp[-(m_j - \bar{m})/\tau]}, \qquad \bar{m} = \frac{1}{B}\sum_{j=1}^{B} m_j. \tag{11}$$

This normalization gives the weights unit batch mean. Thus, predictions that remain stable across perturbed classifiers receive larger weights, while high-MI predictions are suppressed. The centering by $\bar{m}$ does not change the relative weights, but stabilizes the batch-wise weighting scale. Unit-mean normalization redistributes average sample weight, but does not by itself match gradient magnitudes; Appendix B.1 provides that control.

MIRA combines this MI gate with a confidence gate computed from the Monte Carlo mean prediction,

$$c_i = \max_y \bar{p}_{i,\sigma^*}(y), \qquad w_i^{\mathrm{conf}} = \mathrm{Norm}(c_i), \tag{12}$$

where $\mathrm{Norm}(\cdot)$ normalizes the batch mean weight to one. The final sample weight is

$$w_i = \mathrm{sg}\left[\mathrm{Norm}\left(w_i^{\mathrm{MI}} w_i^{\mathrm{conf}}\right)\right], \tag{13}$$

where $\mathrm{sg}[\cdot]$ stops gradients through the reliability estimate. The MI term down-weights perturbation-sensitive samples, while the confidence term down-weights low-evidence samples. Together, these factors favor samples that are confident and low-disagreement under the selected probing radius.

The adaptation loss is

$$\mathcal{L}_{\mathrm{MIRA}} = \frac{1}{B}\sum_{i=1}^{B} w_i \left(\mathrm{HE}_i + \gamma \mathrm{EH}_i\right). \tag{14}$$

We use the common default $\gamma = 1$; coefficient sensitivity is reported in Appendix B.6. The HE term performs entropy minimization on the stochastic predictive mean, and the EH term encourages low entropy across the sampled predictions. If $w_i = 1$, this reduces to the unweighted stochastic objective HE + EH, which provides the main adaptation signal; the MI–confidence weight complements it by biasing the update away from predictions that are sensitive to last-layer perturbations or have low confidence. This does not guarantee pseudo-label correctness; it assumes that perturbation-stable predictions are more likely to be correct, and reduces the influence of brittle predictions that pose a higher risk of harmful entropy-minimization updates under distribution shift.

4.4 Efficient Cached Last-Layer Sampling

MIRA is efficient because all Monte Carlo sampling is confined to the last-layer reliability probe. Unlike dropout-based or full-network stochastic methods, MIRA does not resample the backbone or perform repeated full forward passes. For each target batch, it computes features once, $h_i = \phi_\theta(x_i)$, and reuses them for all classifier samples and all candidate scales. The stochastic logits are computed only by final-layer operations:

$$z_{i,\sigma}^{(k)} = Wh_i + b + \sigma\big[(A_{\mathcal{B}} \odot \epsilon_W^{(k)})h_i + a_{\mathcal{B}} \odot \epsilon_b^{(k)}\big]. \tag{15}$$

MIRA selects a probing scale using the MI/agreement criteria (Appendix D.6), then adapts with a reliability-weighted stochastic entropy objective using fresh Monte Carlo samples. All sampling uses cached features and the final classifier, without source/validation data or posterior fitting.

5 Experiments

We evaluate source-free online MIRA along three axes: accuracy and calibration, MI-based reliability signals, and the efficiency of cached last-layer sampling.

5.1 Experimental Setup

Datasets and models. We evaluate MIRA on CIFAR-100-C and ImageNet-C, each with 15 corruption types and five severity levels. We use WideResNet-40-2 and ResNeXt on CIFAR-100-C, and ResNet-34 and ResNet-50 on ImageNet-C. To assess generalization beyond conventional CNN classifiers, we additionally evaluate ViT-B/16 [7] on ImageNet-C and CLIP ViT-B/16 [31] on CIFAR-10-C and CIFAR-100-C. Unless otherwise specified, adaptation is performed without access to source samples or target labels.

Baselines. We compare against the source model without adaptation, TENT, EATA, SAR, and ROID [38, 28, 29, 27], and uncertainty-aware TTA methods in the extended comparisons. We also include perturbation- or uncertainty-estimation baselines when applicable to separate the effect of stochastic prediction from online adaptation.

Metrics. We report classification accuracy for predictive performance and ECE and NLL for calibration and probabilistic prediction quality. We also report mean maximum confidence (MMC) to distinguish genuine calibration improvement from simple confidence sharpening. For reliability analysis, we compare batch accuracy at matched predictive entropy.

Implementation. MIRA perturbs only the final classifier, using K Monte Carlo samples per batch. Cached backbone features are reused across samples and candidate scales, so probing requires only last-layer operations. Unless otherwise specified, batch size, optimizer, update parameters, and online protocol match each TTA baseline. Appendix D provides baseline configurations, metric definitions, perturbation construction, and hyperparameters.

5.2 Calibration with MI & Agreement-Guided Scale Selection

Figure 3 places adapted models in the confidence–accuracy plane. (a) Across architectures, MIRA reaches higher accuracy than TENT while lying closer to the diagonal. (b) On ImageNet-C, every base method is overconfident; MIRA raises accuracy and narrows the gap for all four. (c) All γ variants stay within 2 points of the diagonal and above TENT's accuracy.

We use MI and prediction agreement to select an informative probing scale and to check its class consistency. The earliest candidate satisfying both criteria is selected. This scale-selection role is separate from MI-based loss weighting, which we assess below. Additional reliability analyses are provided in Appendix E.

Joint (α, ρ), entropy-coefficient, and scale-grid sensitivity are reported in Appendices B.5–B.7. For CIFAR-100-C, accuracy spans only 0.16 percentage points in the threshold sweep, while ECE ranges from 0.0294 to 0.0675. Coarsening the scale grid reduces ImageNet-C accuracy by 0.985 points and raises ECE by 0.0429. Thus, the default is a common operating choice rather than a universally optimal setting. Appendix B.8 reports an HE-only cross-architecture snow diagnostic.

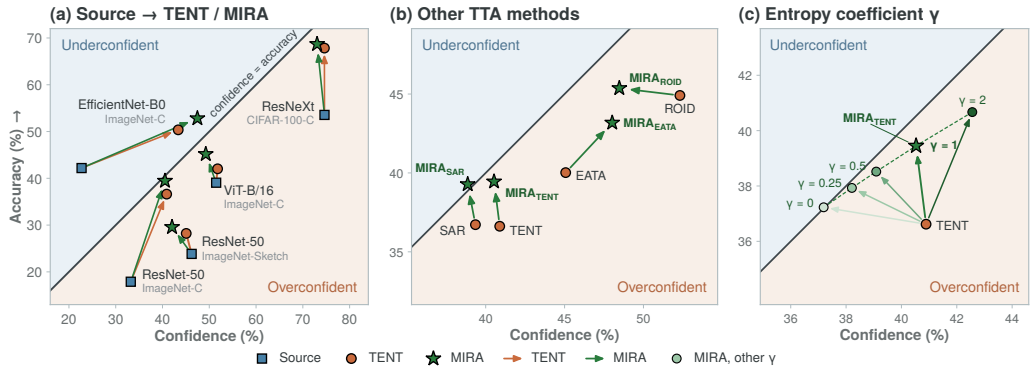

Figure 3: Confidence–accuracy plane; the diagonal marks confidence = accuracy. (a) Architectures, (b) MIRA on base TTA methods, (c) γ variants of $\text{MIRA}_{\text{TENT}}$ on ImageNet-C / ResNet-50.

5.3 Loss Design: MI as a Reliability Gate

MIRA provides the entropy of the Monte Carlo mean prediction HE, the expected entropy of sampled classifiers EH, and their difference $MI = \text{HE} - \text{EH}$. The entropy terms provide adaptation objectives: HE sharpens the averaged stochastic prediction, while EH encourages individual perturbed classifiers to remain confident. MI instead measures perturbation-induced disagreement. Table 1 shows why it should not be the sole optimization target: direct MI minimization lowers MMC and ECE but substantially reduces accuracy and worsens NLL. A low ECE alone therefore does not imply a better predictor. Neither HE nor EH alone improves accuracy over TENT, whereas their combination increases accuracy to 36.37% and reduces NLL to 3.2806. This highlights the benefit of jointly optimizing the mean prediction and individual stochastic predictions.

Table 1: Loss ablation on ImageNet-C / ResNet-34: three-run mean ± sample SD (seeds 1, 2, and 3); accuracy is a fraction. Stochastic variants retain MI-based scale selection. Green: Full MIRA; bold: best means.

Method	Loss	Acc. ↑	MMC	ECE ↓	NLL ↓
Source	–	$0.1840_{\pm 0.0015}$	$0.3742_{\pm 0.0007}$	$0.1903_{\pm 0.0009}$	$5.4870_{\pm 0.0057}$
TENT	H	$0.3421_{\pm 0.0019}$	$0.4053_{\pm 0.0009}$	$0.0642_{\pm 0.0013}$	$3.4602_{\pm 0.0022}$
TENT	$2H$	$0.3660_{\pm 0.0021}$	$0.4388_{\pm 0.0012}$	$0.0734_{\pm 0.0012}$	$3.2892_{\pm 0.0026}$
$\text{MIRA}_{\text{TENT}}$	HE	$0.3408_{\pm 0.0020}$	$0.3661_{\pm 0.0010}$	$0.0412_{\pm 0.0008}$	$3.4498_{\pm 0.0031}$
$\text{MIRA}_{\text{TENT}}$	EH	$0.3392_{\pm 0.0020}$	$0.3644_{\pm 0.0009}$	$0.0412_{\pm 0.0008}$	$3.4637_{\pm 0.0030}$
$\text{MIRA}_{\text{TENT}}$	MI	$0.2977_{\pm 0.0016}$	$0.3094_{\pm 0.0002}$	$\mathbf{0.0373}_{\pm 0.0003}$	$3.8173_{\pm 0.0075}$
$\text{MIRA}_{\text{TENT}}$	HE + EH	$0.3637_{\pm 0.0024}$	$0.3961_{\pm 0.0013}$	$0.0442_{\pm 0.0011}$	$3.2806_{\pm 0.0055}$
$\text{MIRA}_{\text{TENT}}$	$w_{\text{MI}}\text{HE}$	$0.3425_{\pm 0.0021}$	$0.3675_{\pm 0.0010}$	$0.0415_{\pm 0.0007}$	$3.4387_{\pm 0.0021}$
$\text{MIRA}_{\text{TENT}}$	$w_{\text{MI}}(\text{HE} + \text{EH})$	$0.3658_{\pm 0.0023}$	$0.3971_{\pm 0.0014}$	$0.0441_{\pm 0.0010}$	$3.2681_{\pm 0.0007}$
$\text{MIRA}_{\text{TENT}}$	$w_{\text{MI}} w_{\text{conf}}(\text{HE} + \text{EH})$	$\mathbf{0.3699}_{\pm 0.0024}$	$0.3992_{\pm 0.0011}$	$0.0435_{\pm 0.0020}$	$\mathbf{3.2421}_{\pm 0.0208}$

The joint stochastic loss HE + EH accounts for a substantial part of the improvement over standard entropy minimization. MI weighting down-weights perturbation-sensitive predictions, while the confidence gate suppresses low-evidence samples. Applying w_{MI} to HE+EH changes mean accuracy from 36.37% to 36.58%, with little change in ECE; Full MIRA reaches 36.99% and lower NLL. These modest differences should be interpreted relative to seed variation and support gating as a complementary mechanism. This ablation concerns loss weighting; MI-based scale selection remains active in all stochastic variants. Gradient-matched controls are reported in Appendix B.1. Even under gradient-matched updates, Full MIRA improves accuracy over unweighted HE + EH by 0.28 and 0.04 percentage points on CIFAR-10-C and CIFAR-100-C, respectively (Appendix B.1).

The deterministic $2H$ control shows why loss strength and calibration must be assessed together. Doubling TENT's entropy objective raises accuracy to 36.60%, but MMC increases to 0.4388 and ECE to 0.0734. Full MIRA instead reaches 36.99% with MMC 0.3992 and ECE 0.0435, so its accuracy gain accompanies a smaller confidence–accuracy gap. This comparison supports evaluating adaptation strength and calibration jointly. It does not isolate the gate by itself, because stochastic prediction, scale selection, and weighting change together; the gradient-matched comparisons in Appendix B.1 address the separate question of sample weighting at a common update norm. NLL

complements this comparison by penalizing confident mistakes, whereas MMC records the direction of confidence change and must be interpreted relative to accuracy.

5.4 Calibration Plug-and-Play MI Gating

Beyond using MIRA as a standalone TTA method, we evaluate whether its mutual-information signal can be attached to existing adaptation rules. Many TTA methods differ in their sample-selection strategy or adaptation loss. MIRA provides an orthogonal reliability signal: low perturbation-induced MI indicates stable predictions, whereas high MI indicates brittle predictions that should be down-weighted. Thus, MIRA can be used as a lightweight plug-and-play gate without modifying the core update rule of the base method.

Table 2: Plug-and-play evaluation: three-run means with accuracy ± SD. SDs for all metrics of $\text{MIRA}_{\text{TENT}}$ are reported in Table 12. Bold: best accuracy, ECE, and NLL per dataset. Protocols: Appendix D.1–D.2.

Method	CIFAR-100-C / ResNeXt				ImageNet-C / ResNet-50			
	Acc. (%) ↑	MMC	ECE ↓	NLL ↓	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
MAP	53.55	0.7463	0.2109	2.4944	17.89	0.3322	0.1535	5.2161
TENT	$67.84_{\pm 0.05}$	0.7461	0.0693	1.2547	$36.62_{\pm 0.12}$	0.4089	0.0478	3.2549
$\text{MIRA}_{\text{TENT}}$	$68.69_{\pm 0.15}$	0.7305	0.0452	1.1940	$39.45_{\pm 0.14}$	0.4053	0.0361	3.0205
EATA	$65.37_{\pm 0.14}$	0.7039	0.0559	1.3515	$40.02_{\pm 0.29}$	0.4507	0.0549	2.9796
$\text{MIRA}_{\text{EATA}}$	$68.47_{\pm 0.04}$	0.7217	**0.0399**	1.1921	$43.18_{\pm 0.12}$	0.4802	0.0533	2.7547
SAR	$67.55_{\pm 0.11}$	0.7357	0.0630	1.2574	$36.72_{\pm 0.15}$	0.3935	0.0355	3.2487
MIRA_{SAR}	$68.61_{\pm 0.10}$	0.7348	0.0500	1.2001	$39.28_{\pm 0.56}$	0.3886	**0.0290**	3.0494
ROID	$69.16_{\pm 0.01}$	0.7644	0.0735	1.1931	$44.91_{\pm 0.20}$	0.5231	0.0758	2.6526
$\text{MIRA}_{\text{ROID}}$	$\mathbf{69.21}_{\pm 0.12}$	0.7407	0.0497	**1.1674**	$\mathbf{45.37}_{\pm 0.25}$	0.4847	0.0383	**2.5916**

Table 2 shows lower ECE and NLL with method-dependent accuracy gains. Small accuracy differences, particularly ROID on CIFAR-100-C, should be read relative to seed variation. MIRA improves SAR's ImageNet-C accuracy from 36.72% to 39.28% while reducing ECE and NLL.

The effect on confidence is not unidirectional, which is expected because the MI gate is not designed to monotonically sharpen or soften predictions. MMC decreases for TENT, SAR, and ROID but increases for EATA on both benchmarks. What is consistent is the absolute accuracy–confidence gap: every base method in Table 2 is overconfident, and attaching MIRA narrows this gap in all eight settings. For EATA, MMC rises, but accuracy rises even more, so the gap still shrinks. This distinguishes MIRA from confidence-driven TTA, which typically pushes predictions to become sharper regardless of calibration. By controlling which target samples influence adaptation, MIRA acts as a general reliability module for online TTA.

5.5 Generalization and Comparison with Uncertainty Estimation Methods

Generalization beyond conventional classifiers. Table 3 extends MIRA to ViT-B/16 on ImageNet-C and CLIP ViT-B/16 on CIFAR-10-C and CIFAR-100-C. Attaching MIRA to TENT improves accuracy, ECE, and NLL in all three settings. For CLIP, class text embeddings serve as the effective classifier and are perturbed only for reliability probing. The released CLIPTTA's TENT implementation [19] uses ten updates per batch; the modified one-step comparison and implementation details are given in Appendices C.5 and D.7.

Table 3: ViT/CLIP generalization results.

Dataset	Method	Acc. (%) ↑	ECE ↓	NLL ↓
ImageNet-C	TENT	42.03	0.1141	3.0312
	$\text{MIRA}_{\text{TENT}}$	**45.17**	**0.0529**	**2.8159**
CIFAR-10-C	TENT	56.38	0.3629	3.5747
	$\text{MIRA}_{\text{TENT}}$	**58.65**	**0.3362**	**3.3509**
CIFAR-100-C	TENT	31.28	0.4814	6.5500
	$\text{MIRA}_{\text{TENT}}$	**35.27**	**0.4066**	**5.5736**

Continual, dynamic, and computationally realistic adaptation. MIRA also improves continual CoTTA on CIFAR-100-C from 64.16% to 65.20% accuracy, with ECE decreasing from 0.0529 to 0.0342. Under a dynamic PTTA stream [42], RoTTA improves from 74.89% to 75.67%, with ECE decreasing from 0.0842 to 0.0674. Under the computationally realistic protocol, TENT accuracy improves from 38.97% to 39.92% on ImageNet-C and from 40.74% to 41.13% on ImageNet-R, with lower ECE and NLL on both. These results and their full comparisons are given in Appendix C, including the alternative PTTA protocol, where accuracy and ECE improve but NLL worsens.

These extensions test accumulated adaptation, changing target distributions, and time-limited update opportunities. Their accuracy–calibration trade-offs complement the conventional corruption-stream comparison. The alternative PTTA result illustrates why lower ECE alone does not establish better

probabilistic prediction: NLL worsens despite higher accuracy. We consequently assess each protocol using accuracy, ECE, and NLL together.

Comparison with uncertainty-estimation methods. We compare against deep ensembles, last-layer Laplace approximation (LLLA), a simplified Laplace-style variant (ICLA), and Bayesian neural-network baselines (BBB, ABNN) in Table 4.

These baselines use uncertainty to form predictions, whereas MIRA uses local perturbation responses to guide online adaptation. MIRA achieves the highest accuracy and lowest NLL on both benchmarks, and the lowest ECE on CIFAR-100-C. On ImageNet-C, Laplace-style methods attain slightly lower ECE, but substantially lower accuracy and higher NLL. Together with the loss ablation, these results support stochastic adaptation complemented by reliability weighting while delimiting the gate's incremental contribution.

Table 4: Uncertainty-estimation comparison. Accuracy is a fraction; values are retained single-run results.

Method	CIFAR-100-C WRN-40-2				ImageNet-C ResNet-50			
	Acc ↑	MMC	ECE ↓	NLL ↓	Acc ↑	MMC	ECE ↓	NLL ↓
MAP	0.5389	0.7041	0.1653	2.0589	0.2084	0.2246	0.0584	4.6490
Deep Ens.	0.5811	0.6269	0.0504	1.6998	0.2532	0.1874	0.0750	4.1725
LLLA	0.5304	0.6282	0.0978	1.9300	0.2140	0.1948	**0.0431**	4.4985
ICLA	0.5389	0.4449	0.0956	1.8801	0.2143	0.1879	0.0468	4.5034
BBB	0.3445	0.5445	0.2001	3.7816	0.2057	0.3079	0.1035	4.8407
ABNN	0.5179	0.6568	0.1389	2.2254	0.2326	0.2968	0.0634	4.2755
$\text{MIRA}_{\text{TENT}}$	**0.6391**	0.6701	**0.0390**	**1.3505**	**0.3726**	0.3303	0.0512	**3.2092**

5.6 Efficiency of Last-Layer Probing

MIRA computes backbone features once per target batch and reuses them for all Gaussian classifier samples and candidate scales. Thus, probing adds classifier-level matrix multiplications rather than repeated backbone inference. Table 5 reports moderate runtime overhead over TENT, with runtime in the same range as SAR and ROID in the evaluated settings. Memory can increase more: the ImageNet ratio is $3.72\times$ the source model, compared with $2.21\times$ for TENT. Ablations on perturbation scope and the Monte Carlo sampling budget are provided in Appendix B.

Table 5: Efficiency comparison. Runtime and memory are relative to the source model.

Method	CIFAR100 WRN		ImageNet ResNet34	
	Runtime	Memory	Runtime	Memory
Source	$1.00\times$	$1.00\times$	$1.00\times$	$1.00\times$
TENT	$1.48\times$	$1.68\times$	$1.21\times$	$2.21\times$
EATA	$1.67\times$	$1.68\times$	$1.45\times$	$2.21\times$
SAR	$1.90\times$	$1.45\times$	$1.73\times$	$2.24\times$
ROID	$2.81\times$	$2.06\times$	$2.21\times$	$2.99\times$
$\textbf{MIRA}_{\textbf{TENT}}$	$2.05\times$	$2.05\times$	$1.79\times$	$3.72\times$

6 Conclusion

We proposed MIRA, a source-free online TTA method that estimates reliability with last-layer Gaussian perturbations. MIRA uses mutual information and prediction agreement, rather than a Bayesian posterior or deterministic confidence, to identify stable predictions. By suppressing brittle high-confidence updates while preserving entropy-based adaptation, MIRA improves the accuracy–calibration trade-off; confining Monte Carlo sampling to the final classifier keeps it an efficient plug-and-play reliability module.

Across conventional classifiers, ViT, and CLIP, a local stochastic probe supported adaptation without repeated backbone evaluations. Loss controls separate stochastic entropy from reliability weighting, and continual, dynamic, and time-constrained evaluations test the signal beyond conventional streams. These findings suggest a role for MI alongside confidence: MI sets the probing scale, agreement checks class consistency, and sensitivity helps determine which predictions influence the update. Future work should simplify scale selection and extend the probe to richer adaptation modules.

Limitations. MIRA adds a scale grid, thresholds α, ρ, entropy coefficient γ, and sampling budget K. Calibration remains sensitive to these choices and to grid density. The agreement bound holds only when both selection conditions are met; under the default thresholds, the MI criterion alone often determines the scale. Stochastic entropy accounts for much of the accuracy gain, and the gate's additional benefit is setting-dependent. The probe is not calibrated Bayesian posterior uncertainty and cannot fully correct severe feature-level misalignment. Adapting to unrepresentative unlabeled streams may also reinforce existing bias, so confidence improvements do not establish fairness or safe deployment across subpopulations.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-16070382, RS-2025-02215070, RS-2025-02217919, RS-2024-00436680), Artificial Intelligence Graduate School Program at Yonsei University (RS 2020-II201361), the Korea Institute of Science and Technology (KIST) Institutional Program under Grant 26E0170.

Competing Interests. The authors declare no competing interests.

References

[1] Motasem Alfarra, Hani Itani, Alejandro Pardo, Shyma Yaser Alhuwaider, Merey Ramazanova, Juan Camilo Perez, Zhipeng Cai, Matthias Müller, and Bernard Ghanem. Evaluation of test-time adaptation under computational time constraints. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 976–991, 2024.

[2] Fin Amin and Jung-Eun Kim. The over-certainty phenomenon in modern test-time adaptation algorithms. *arXiv preprint arXiv:2404.16168*, 2024.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

[4] Dhanajit Brahma and Piyush Rai. A probabilistic framework for lifelong test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3582–3591, 2023.

[5] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux—effortless Bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20103, 2021.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[8] Gianni Franchi, Olivier Laurent, Maxence Leguery, Andrei Bursuc, Andrea Pilzer, and Angela Yao. Make me a BNN: A simple strategy for estimating Bayesian uncertainty from pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12194–12204, 2024.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[13] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[14] Hyeongyu Kim, GeonHui Han, and Dosik Hwang. Buffer layers for test-time adaptation. In *Advances in Neural Information Processing Systems*, volume 38, pages 156540–156562, 2025.

[15] Hyeongyu Kim, Geonhui Han, and Dosik Hwang. AcTTA: Rethinking test-time adaptation via dynamic activation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22217–22226, 2026.

[16] Hyeongyu Kim, Yejee Shin, and Dosik Hwang. DiMix: Disentangle-and-mix based domain generalizable medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pages 242–251. Springer, 2023.

[17] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In *International Conference on Machine Learning*, pages 5436–5446. PMLR, 2020.

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[19] Marc Lafon, Gustavo Vargas Hakim, Clément Rambour, Christian Desrosiers, and Nicolas Thome. CLIPTTA: Robust contrastive vision-language test-time adaptation. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[21] Jae-Hong Lee. Bayesian weight enhancement with steady-state adaptation for test-time adaptation in dynamic environments. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 32889–32906, 2025.

[22] Jeong Ryong Lee, Geonhui Son, and Dosik Hwang. FeDi: Feature disentanglement for self-supervised learning. *Pattern Recognition*, 172:112619, 2026.

[23] Taeckyung Lee, Sorn Chottananurak, Taesik Gong, and Sung-Ju Lee. AETTA: Label-free accuracy estimation for test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28643–28652, 2024.

[24] Taeckyung Lee, Sorn Chottananurak, Junsu Kim, Jinwoo Shin, Taesik Gong, and Sung-Ju Lee. Test-time adaptation with binary feedback. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 33005–33024, 2025.

[25] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

[26] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

[27] Robert A Marsden, Mario Döbler, and Bin Yang. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2555–2565, 2024.

[28] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*, pages 16888–16905. PMLR, 2022.

[29] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

[30] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763, 2021.

[32] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[33] Hyungseob Shin, Hyeongyu Kim, Sewon Kim, Yohan Jun, Taejoon Eo, and Dosik Hwang. SDC-UDA: Volumetric unsupervised domain adaptation framework for slice-direction continuous cross-modality medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7412–7421, 2023.

[34] Geonhui Son, Jeong Ryong Lee, and Dosik Hwang. HP-GAN: Harnessing pretrained networks for GAN improvement with FakeTwins and discriminator consistency. *Neural Networks*, 199:108666, 2026.

[35] Youngjun Song, Hyeongyu Kim, and Dosik Hwang. CD-Buffer: Complementary dual-buffer framework for test-time adaptation in adverse weather object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15050–15059, 2026.

[36] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.

[37] Mingkui Tan, Guohao Chen, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Peilin Zhao, and Shuaicheng Niu. Uncertainty-calibrated test-time model adaptation without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[38] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[39] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P. Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[40] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[41] Zixin Wang, Yadan Luo, Liang Zheng, Zhuoxiao Chen, Sen Wang, and Zi Huang. In search of lost online test-time adaptation: A survey. *International Journal of Computer Vision*, 133(3):1106–1139, 2025.

[42] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932, 2023.

[43] Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35:38629–38642, 2022.

[44] Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. COME: Test-time adaptation by conservatively minimizing entropy. In *International Conference on Learning Representations*, 2025.

[45] Maksim Zhdanov, Stanislav Dereka, and Sergey Kolesnikov. Identity curvature Laplace approximation for improved out-of-distribution detection. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7019–7028. IEEE, 2025.

[46] Aurick Zhou and Sergey Levine. Bayesian adaptation for covariate shift. *Advances in Neural Information Processing Systems*, 34:914–927, 2021.

[47] Lihua Zhou, Mao Ye, Shuaifeng Li, Nianxin Li, Xiatian Zhu, Lei Deng, Hongbin Liu, and Zhen Lei. Bayesian test-time adaptation for vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29999–30009, 2025.

A Rationale for MIRA

MIRA is based on the principle that predictions stable under controlled local classifier perturbations provide safer signals for entropy-based adaptation than locally brittle predictions. We do not interpret the resulting mutual information as calibrated Bayesian posterior uncertainty; rather, the perturbation family is used as an operational probe of local prediction stability.

To motivate the agreement criterion, consider the margin between the predicted class $\hat{y}_i$ and a competing class j,

$$m_{ij} = z_{i,\hat{y}_i} - z_{i,j}. \tag{16}$$

Under Gaussian perturbation of the final classifier, the corresponding pairwise boundary-crossing probability is

$$\Pr[m_{ij} + \Delta m_{ij} < 0] = \Phi\left(-\frac{m_{ij}}{\sigma s_{ij}}\right). \tag{17}$$

Here $\hat{y}_i = \arg\max_c z_{i,c}$, Φ is the standard-normal CDF, and $\Delta m_{ij} \sim \mathcal{N}(0, \sigma^2 s_{ij}^2)$ conditional on the cached features and perturbation factors. This motivates top-1 stability, but MIRA measures empirical agreement with the Monte Carlo mean class; it does not guarantee preservation of the unperturbed MAP class.

Mutual information provides complementary information. Agreement only changes when the top-1 prediction crosses a decision boundary, whereas MI captures changes in the full predictive distribution before such a crossing occurs. Under a small-perturbation approximation,

$$\mathrm{MI}_{i,\sigma} \approx \frac{1}{2} \operatorname{tr}(F(p_i)\Sigma_{z,i}). \tag{18}$$

Here $p_i = \mathrm{softmax}(z_i)$, $F(p_i) = \mathrm{diag}(p_i) - p_i p_i^\top$, and $\Sigma_{z,i} = \mathrm{Cov}(\Delta z_i)$ includes the σ^2 scale factor. This gives perturbation-induced MI an interpretation as local predictive sensitivity.

MIRA selects the earliest scale with non-trivial MI and high agreement, balancing detectable predictive sensitivity with top-1 stability. The entropy-matched analysis in Appendix E evaluates whether MI provides reliability information beyond predictive entropy.

The exponential MI weight also follows entropy-regularized reliability reweighting. For batch probabilities $q_i \geq 0$ with $\sum_i q_i = 1$,

$$\min_q \sum_i q_i \mathrm{MI}_i + \tau \sum_i q_i \log q_i \quad \Longrightarrow \quad q_i \propto \exp(-\mathrm{MI}_i/\tau). \tag{19}$$

The unit-mean weight is $w_i^{\mathrm{MI}} = B q_i$; confidence weighting and detachment then determine the final loss. Agreement provides its empirical guard only when selection is feasible.

B Additional Ablation Studies

New loss, gradient-control, construction, threshold, coefficient, and grid analyses report mean $\pm$ sample SD across paired seeds 1, 2, and 3. Accuracy SD is in percentage points; MMC, ECE, and NLL use their native units. Each comparison retains its own backbone, optimizer, and stream protocol. The existing scope and Monte Carlo-budget studies remain single-run. Green highlights MIRA variants or the default configuration; bold marks the best values in directional metrics within each comparison.

B.1 Gradient-Matched Reliability Gating

Unit-mean sample weights do not ensure equal gradient norms. To control uniform loss strengthening, we normalize the complete adaptation gradient before the optimizer step. We multiply each objective by a detached reciprocal of the complete adaptation gradient's global L2 norm, targeting norm 1 before the optimizer step. The original optimizer and learning rate are retained; the maximum recorded norm error is below 2.49×10^{-4}. CIFAR-10-C uses WRN AugMix, Adam at 10^{-3}, $K = 100$, and Gaussian noise, snow, fog, and pixelation. CIFAR-100-C uses ResNeXt AugMix, SGD at 10^{-3}, $K = 300$, and 15 corruptions. Both protocols reset between corruptions.

Table 6: Gradient-matched loss and gating controls on CIFAR-10-C and CIFAR-100-C: global adaptation-gradient L2 norm 1, mean ± sample SD over three seeds. Panels use distinct protocols and are separate from ordinary adaptation. Bold marks best accuracy, ECE, and NLL within each panel.

Objective	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
CIFAR-10-C / WRN; Adam, four corruptions, $K = 100$				
TENT	$87.79_{\pm 0.05}$	$0.9412_{\pm 0.0005}$	$0.0635_{\pm 0.0006}$	$0.4462_{\pm 0.0025}$
HE + EH	$88.14_{\pm 0.05}$	$0.9221_{\pm 0.0008}$	$0.0409_{\pm 0.0011}$	$0.3905_{\pm 0.0032}$
+ MI gate	$88.34_{\pm 0.10}$	$0.9236_{\pm 0.0007}$	$0.0403_{\pm 0.0015}$	$0.3838_{\pm 0.0027}$
+ confidence gate	$88.29_{\pm 0.09}$	$0.9229_{\pm 0.0008}$	$0.0402_{\pm 0.0016}$	$0.3848_{\pm 0.0025}$
Full MIRA$_{\text{TENT}}$	$\mathbf{88.42}_{\pm 0.07}$	$0.9235_{\pm 0.0007}$	$\mathbf{0.0396}_{\pm 0.0011}$	$\mathbf{0.3808}_{\pm 0.0023}$
CIFAR-100-C / ResNeXt; SGD, $K = 300$				
TENT	$65.97_{\pm 0.11}$	$0.7150_{\pm 0.0004}$	$0.0586_{\pm 0.0005}$	$1.3250_{\pm 0.0029}$
HE + EH	$66.14_{\pm 0.10}$	$0.6895_{\pm 0.0005}$	$0.0333_{\pm 0.0005}$	$1.2980_{\pm 0.0028}$
+ MI gate	$\mathbf{66.27}_{\pm 0.12}$	$0.6913_{\pm 0.0005}$	$0.0333_{\pm 0.0005}$	$\mathbf{1.2925}_{\pm 0.0033}$
+ confidence gate	$66.11_{\pm 0.10}$	$0.6872_{\pm 0.0005}$	$0.0325_{\pm 0.0007}$	$1.2979_{\pm 0.0023}$
Full MIRA$_{\text{TENT}}$	$66.18_{\pm 0.11}$	$0.6880_{\pm 0.0005}$	$\mathbf{0.0322}_{\pm 0.0009}$	$1.2937_{\pm 0.0025}$

Relative to unweighted stochastic entropy, Full MIRA changes accuracy by $+0.278$ and $+0.045$ percentage points on CIFAR-10-C and CIFAR-100-C, respectively, with the same direction in all paired seeds. These results support a setting-dependent contribution from reliability weighting. Gradient normalization changes the optimizer step, so the absolute accuracies in Table 6 are distinct from the main protocol.

Table 7: Unnormalized CIFAR-10-C objective and gating controls, mean ± sample SD over three seeds. The Adam/four-corruption/$K = 100$ protocol matches the corresponding normalized panel. Bold marks best accuracy, ECE, and NLL.

Objective	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
TENT	$87.77_{\pm 0.06}$	$0.9407_{\pm 0.0005}$	$0.0634_{\pm 0.0008}$	$0.4449_{\pm 0.0029}$
HE + EH	$88.15_{\pm 0.06}$	$0.9218_{\pm 0.0007}$	$0.0405_{\pm 0.0011}$	$0.3902_{\pm 0.0038}$
+ MI gate	$88.32_{\pm 0.04}$	$0.9231_{\pm 0.0008}$	$0.0401_{\pm 0.0011}$	$0.3834_{\pm 0.0026}$
+ confidence gate	$88.30_{\pm 0.10}$	$0.9225_{\pm 0.0008}$	$0.0398_{\pm 0.0016}$	$0.3843_{\pm 0.0023}$
Full MIRA$_{\text{TENT}}$	$\mathbf{88.39}_{\pm 0.02}$	$0.9232_{\pm 0.0007}$	$\mathbf{0.0396}_{\pm 0.0004}$	$\mathbf{0.3800}_{\pm 0.0029}$

B.2 Perturbation Design Sensitivity

We additionally examine whether MIRA is sensitive to the construction of the last-layer perturbation scale. We separately vary the scale parameterization and the base perturbation rule on CIFAR-100-C at severity level 5. The WRN AugMix construction protocol uses all examples shuffled once, $K = 100$, SGD at 10^{-3}, and reset between corruptions; it differs from the ResNeXt headline protocol.

Scale parameterization. We compare feature-wise, scalar, and class-wise parameterizations while keeping the remaining MIRA configuration fixed.

Table 8: Ablation on the perturbation-scale parameterization on CIFAR-100-C. Mean ± sample SD over seeds 1, 2, and 3; WRN AugMix, $K = 100$, SGD 10^{-3}. Bold marks best accuracy, ECE, and NLL.

Construction	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
TENT	$63.63_{\pm 0.08}$	$0.7184_{\pm 0.0005}$	$0.0824_{\pm 0.0008}$	$1.4078_{\pm 0.0044}$
Feature-wise	$\mathbf{64.33}_{\pm 0.10}$	$0.6787_{\pm 0.0001}$	$0.0387_{\pm 0.0009}$	$1.3426_{\pm 0.0061}$
Scalar	$64.29_{\pm 0.15}$	$0.6589_{\pm 0.0006}$	$\mathbf{0.0296}_{\pm 0.0007}$	$1.3414_{\pm 0.0062}$
Class-wise	$64.28_{\pm 0.15}$	$0.6590_{\pm 0.0007}$	$0.0297_{\pm 0.0008}$	$\mathbf{1.3410}_{\pm 0.0062}$

The mean accuracy differences between feature-wise and scalar/class-wise parameterizations are only $0.038/0.046$ percentage points; scalar and class-wise variants have lower ECE. This indicates that MIRA is not highly sensitive to the exact parameterization in accuracy, while calibration still depends on it.

Base perturbation scale. We next compare He-in, fan-in, weight-magnitude, and absolute base-scale rules.

Table 9: Ablation on the base perturbation scale rule on CIFAR-100-C. Mean ± sample SD over seeds 1, 2, and 3; WRN AugMix, $K = 100$, SGD 10^{-3}. Bold marks best accuracy, ECE, and NLL.

Construction	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
TENT	$63.63_{\pm 0.08}$	$0.7184_{\pm 0.0005}$	$0.0824_{\pm 0.0008}$	$1.4078_{\pm 0.0044}$
He-in	$\mathbf{64.33}_{\pm 0.10}$	$0.6787_{\pm 0.0001}$	$0.0387_{\pm 0.0009}$	$1.3426_{\pm 0.0061}$
Fan-in	$64.27_{\pm 0.12}$	$0.6599_{\pm 0.0001}$	$0.0206_{\pm 0.0015}$	$\mathbf{1.3326}_{\pm 0.0052}$
Absolute	$64.19_{\pm 0.15}$	$0.6440_{\pm 0.0001}$	$\mathbf{0.0152}_{\pm 0.0006}$	$1.3332_{\pm 0.0050}$
Weight magnitude	$64.23_{\pm 0.13}$	$0.6559_{\pm 0.0003}$	$0.0246_{\pm 0.0009}$	$1.3443_{\pm 0.0047}$

The different base-scale rules yield similar accuracy while all substantially improve calibration over deterministic TENT. We therefore use the feature-wise He-in construction as the fixed default rather than tuning the perturbation rule for each dataset or model.

B.3 Perturbation Scope

We compare the default classifier-only perturbation with perturbing the last two or three layers on ImageNet-C / ResNet-34 with batch size 64.

Table 10: Effect of perturbation scope on ImageNet-C / ResNet-34 (batch size 64), retained single-run results. Bold marks the best values in directional columns.

Scope	Acc. (%) ↑	MMC	ECE ↓	NLL ↓	Time (s) ↓	VRAM (MiB) ↓	Params
Classifier only	37.04	0.4013	**0.0458**	3.2690	**379.9**	**8180.4**	513,000
Last 2 layers	37.15	0.3576	0.0519	3.2372	14,818.9	8212.3	2,872,296
Last 3 layers	**37.31**	0.3592	0.0649	**3.2332**	21,572.5	8212.3	5,231,592

Deeper perturbations give only small accuracy/NLL gains while worsening ECE and increasing runtime by roughly $39\times$–$57\times$. We therefore use the final classifier as the best accuracy–calibration–efficiency trade-off.

B.4 Monte Carlo Sampling Budget

Table 11: Effect of the Monte Carlo sampling budget K on ImageNet-C / ResNet-50, retained single-run results. Bold marks the best values in directional columns.

K	Acc. (%) ↑	ECE ↓	NLL ↓	Time (s) ↓	Peak VRAM (MiB) ↓
100	29.55	0.0410	3.6454	**60**	**5,489**
200	29.55	0.0408	3.6407	66	7,175
300	29.39	0.0420	**3.6374**	74	8,671
500	29.29	0.0443	3.6511	88	11,939
1000	**29.76**	**0.0381**	3.6387	116	22,953

The scope and sampling-budget studies use different backbones, so their absolute metrics should not be compared across tables. These retained single-run metrics vary non-monotonically across K, while runtime and memory increase with the sampling budget. We therefore use $K = 300$ as a conservative accuracy–efficiency trade-off.

B.5 Joint MI and Agreement Threshold Sensitivity

We evaluate $\alpha \in \{0.01, 0.03, 0.10\}$ and $\rho \in \{0.85, 0.90, 0.95\}$ jointly, retaining the Full objective, original 16-point grid, and $K = 300$. The panels use the headline TENT protocols: CIFAR-100-C / ResNeXt with SGD 10^{-3} and ImageNet-C / ResNet-50 with SGD 2.5×10^{-4}. Default controls shared with other sweeps are counted once.

Table 12: Joint α–ρ sensitivity, mean $\pm$ sample SD over three seeds. The common default is $(0.03, 0.90)$; test labels are used only for evaluation. Bold marks best accuracy, ECE, and NLL within each dataset; displayed ties are included.

Setting	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
CIFAR-100-C / ResNeXt				
$\alpha = 0.01,\ \rho = 0.85$	$68.58_{\pm 0.11}$	$0.7524_{\pm 0.0009}$	$0.0675_{\pm 0.0010}$	$1.2197_{\pm 0.0073}$
$\alpha = 0.01,\ \rho = 0.90$	$68.58_{\pm 0.12}$	$0.7522_{\pm 0.0009}$	$0.0675_{\pm 0.0012}$	$1.2196_{\pm 0.0074}$
$\alpha = 0.01,\ \rho = 0.95$	$68.59_{\pm 0.13}$	$0.7499_{\pm 0.0009}$	$0.0651_{\pm 0.0010}$	$1.2159_{\pm 0.0084}$
$\alpha = 0.03,\ \rho = 0.85$	$68.64_{\pm 0.13}$	$0.7396_{\pm 0.0011}$	$0.0542_{\pm 0.0009}$	$1.2032_{\pm 0.0083}$
$\alpha = 0.03,\ \rho = 0.90$	$68.69_{\pm 0.15}$	$0.7305_{\pm 0.0012}$	$0.0452_{\pm 0.0007}$	$1.1940_{\pm 0.0111}$
$\alpha = 0.03,\ \rho = 0.95$	$\mathbf{68.70}_{\pm 0.16}$	$0.7279_{\pm 0.0011}$	$0.0426_{\pm 0.0009}$	$1.1906_{\pm 0.0113}$
$\alpha = 0.10,\ \rho = 0.85$	$68.55_{\pm 0.15}$	$0.6576_{\pm 0.0011}$	$\mathbf{0.0294}_{\pm 0.0003}$	$\mathbf{1.1853}_{\pm 0.0094}$
$\alpha = 0.10,\ \rho = 0.90$	$68.54_{\pm 0.15}$	$0.6570_{\pm 0.0011}$	$0.0299_{\pm 0.0003}$	$1.1859_{\pm 0.0093}$
$\alpha = 0.10,\ \rho = 0.95$	$68.54_{\pm 0.14}$	$0.6570_{\pm 0.0011}$	$0.0300_{\pm 0.0003}$	$1.1859_{\pm 0.0093}$
ImageNet-C / ResNet-50				
$\alpha = 0.01,\ \rho = 0.85$	$\mathbf{39.64}_{\pm 0.19}$	$0.4435_{\pm 0.0018}$	$0.0523_{\pm 0.0025}$	$3.0261_{\pm 0.0118}$
$\alpha = 0.01,\ \rho = 0.90$	$\mathbf{39.64}_{\pm 0.19}$	$0.4422_{\pm 0.0016}$	$0.0514_{\pm 0.0028}$	$3.0257_{\pm 0.0101}$
$\alpha = 0.01,\ \rho = 0.95$	$\mathbf{39.64}_{\pm 0.20}$	$0.4419_{\pm 0.0016}$	$0.0512_{\pm 0.0028}$	$3.0253_{\pm 0.0111}$
$\alpha = 0.03,\ \rho = 0.85$	$39.44_{\pm 0.16}$	$0.4053_{\pm 0.0014}$	$0.0360_{\pm 0.0020}$	$3.0216_{\pm 0.0022}$
$\alpha = 0.03,\ \rho = 0.90$	$39.45_{\pm 0.14}$	$0.4053_{\pm 0.0014}$	$0.0361_{\pm 0.0022}$	$\mathbf{3.0205}_{\pm 0.0020}$
$\alpha = 0.03,\ \rho = 0.95$	$39.44_{\pm 0.16}$	$0.4053_{\pm 0.0014}$	$\mathbf{0.0359}_{\pm 0.0020}$	$3.0206_{\pm 0.0022}$
$\alpha = 0.10,\ \rho = 0.85$	$38.47_{\pm 0.10}$	$0.3058_{\pm 0.0008}$	$0.0791_{\pm 0.0008}$	$3.1142_{\pm 0.0029}$
$\alpha = 0.10,\ \rho = 0.90$	$38.43_{\pm 0.10}$	$0.3058_{\pm 0.0008}$	$0.0787_{\pm 0.0009}$	$3.1150_{\pm 0.0036}$
$\alpha = 0.10,\ \rho = 0.95$	$38.44_{\pm 0.09}$	$0.3058_{\pm 0.0007}$	$0.0788_{\pm 0.0008}$	$3.1147_{\pm 0.0023}$

At $\alpha = 0.03$, ImageNet-C accuracy is nearly identical across ρ. In this setting, the two conditions are jointly satisfied in at most 0.30% of batches, so the MI criterion determines σ^* and ρ has little effect; the flat response reflects this rather than robustness to ρ. On CIFAR-100-C, where joint satisfaction is more frequent, ρ mainly affects calibration (Table 12). Across both benchmarks, accuracy stays within 0.2 percentage points of the default for $\alpha \in \{0.01, 0.03\}$ and all tested ρ, whereas calibration is more sensitive to α: $\alpha = 0.10$ lowers CIFAR-100-C ECE but raises ImageNet-C ECE to 0.079, and ImageNet-C ECE is lowest at $\alpha = 0.03$. We therefore recommend $\alpha \approx 0.03$ as the operating point and treat ρ as a secondary setting.

B.6 Entropy-Coefficient Sensitivity

We vary only the EH coefficient in $w(\mathrm{HE} + \gamma\mathrm{EH})$ over $\gamma \in \{0, 0.25, 0.5, 1, 2\}$, keeping the confidence exponent fixed at 1. The selector, optimizer, learning rate, and ordinary gradient update remain unchanged; $\gamma = 1$ shares the Full baseline.

Table 13: Entropy-coefficient sensitivity, mean $\pm$ sample SD over three seeds. The coefficient of EH changes independently of confidence weighting. Bold marks best accuracy, ECE, and NLL within each dataset.

Setting	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
CIFAR-100-C / ResNeXt				
$\gamma = 0$	$67.91_{\pm 0.11}$	$0.7144_{\pm 0.0006}$	$\mathbf{0.0384}_{\pm 0.0003}$	$1.2209_{\pm 0.0064}$
$\gamma = 0.25$	$68.22_{\pm 0.11}$	$0.7198_{\pm 0.0008}$	$0.0402_{\pm 0.0007}$	$1.2094_{\pm 0.0069}$
$\gamma = 0.5$	$68.45_{\pm 0.12}$	$0.7241_{\pm 0.0010}$	$0.0420_{\pm 0.0006}$	$1.2022_{\pm 0.0083}$
$\gamma = 1$	$68.69_{\pm 0.15}$	$0.7305_{\pm 0.0012}$	$0.0452_{\pm 0.0007}$	$\mathbf{1.1940}_{\pm 0.0111}$
$\gamma = 2$	$\mathbf{68.87}_{\pm 0.11}$	$0.7391_{\pm 0.0011}$	$0.0517_{\pm 0.0009}$	$1.1984_{\pm 0.0064}$
ImageNet-C / ResNet-50				
$\gamma = 0$	$37.23_{\pm 0.10}$	$0.3719_{\pm 0.0012}$	$\mathbf{0.0329}_{\pm 0.0013}$	$3.1832_{\pm 0.0038}$
$\gamma = 0.25$	$37.93_{\pm 0.14}$	$0.3821_{\pm 0.0012}$	$0.0331_{\pm 0.0015}$	$3.1275_{\pm 0.0031}$
$\gamma = 0.5$	$38.52_{\pm 0.17}$	$0.3909_{\pm 0.0013}$	$0.0343_{\pm 0.0017}$	$3.0838_{\pm 0.0031}$
$\gamma = 1$	$39.45_{\pm 0.14}$	$0.4053_{\pm 0.0014}$	$0.0361_{\pm 0.0022}$	$3.0205_{\pm 0.0020}$
$\gamma = 2$	$\mathbf{40.67}_{\pm 0.19}$	$0.4256_{\pm 0.0016}$	$0.0396_{\pm 0.0026}$	$\mathbf{2.9442}_{\pm 0.0023}$

Increasing γ can improve accuracy while worsening ECE. Equal coefficients provide a common trade-off, rather than a unique metric-wise optimum.

B.7 Scale-Grid Sensitivity

We keep $K = 300$ fixed and compare the original grid with a coarser grid, a denser grid, and an extended range (Appendix D.6). This tests scale coverage independently of the Monte Carlo-budget ablation.

Table 14: Scale-grid sensitivity, mean ± sample SD over three seeds. Full MIRA and the headline dataset-specific TENT protocols are fixed. Bold marks best accuracy, ECE, and NLL within each dataset.

Grid	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
CIFAR-100-C / ResNeXt				
Default (16)	$\mathbf{68.69}_{\pm 0.15}$	$0.7305_{\pm 0.0012}$	$0.0452_{\pm 0.0007}$	$1.1940_{\pm 0.0111}$
Coarse (9)	$68.54_{\pm 0.15}$	$0.6570_{\pm 0.0011}$	$0.0299_{\pm 0.0004}$	$1.1859_{\pm 0.0091}$
Dense (31)	$68.68_{\pm 0.16}$	$0.7317_{\pm 0.0011}$	$0.0463_{\pm 0.0011}$	$1.1938_{\pm 0.0115}$
Extended (20)	$68.56_{\pm 0.17}$	$0.6603_{\pm 0.0014}$	$\mathbf{0.0273}_{\pm 0.0010}$	$\mathbf{1.1838}_{\pm 0.0093}$
ImageNet-C / ResNet-50				
Default (16)	$39.45_{\pm 0.14}$	$0.4053_{\pm 0.0014}$	$\mathbf{0.0361}_{\pm 0.0022}$	$3.0205_{\pm 0.0020}$
Coarse (9)	$38.46_{\pm 0.10}$	$0.3059_{\pm 0.0008}$	$0.0789_{\pm 0.0010}$	$3.1134_{\pm 0.0032}$
Dense (31)	$\mathbf{39.59}_{\pm 0.15}$	$0.4281_{\pm 0.0015}$	$0.0429_{\pm 0.0030}$	$3.0206_{\pm 0.0088}$
Extended (20)	$39.46_{\pm 0.13}$	$0.4045_{\pm 0.0014}$	$0.0366_{\pm 0.0018}$	$\mathbf{3.0196}_{\pm 0.0003}$

The default uses 16 candidate radii. Coarse9 reduces their density to 9; Dense31 inserts geometric midpoints to obtain 31; Extended20 adds two smaller and two larger radii (Appendix D.6). These variants test scale coverage and computational cost. Coarse9 reduces ImageNet-C accuracy by 0.985 percentage points and increases ECE by 0.0429; Dense31 increases accuracy by 0.142 points, wall time from 502.0 to 583.7 seconds, and steady allocated memory from 11.23 to 16.59 GiB. Timing includes batch loading; steady memory excludes the first corruption.

No tested grid selects its boundary. Under the default thresholds, both conditions are jointly satisfied in 22.63±0.95% of CIFAR-100-C batches and in no ImageNet-C batch; otherwise, σ^* is the first candidate satisfying the MI condition (Appendix D.6). Because the MI threshold is defined relative to the observed MI range, changing the grid also shifts the threshold.

B.8 Cross-Architecture Numerical Values

Table 15 gives the values for the cross-architecture diagnostic: ImageNet-C snow at severity 5, seed 1, 10,000 examples, $K = 100$, and an HE objective. The (α, ρ) settings are listed for each backbone.

Table 15: Numerical values and selected settings for the cross-architecture diagnostic; single-run ImageNet-C snow diagnostics. Bold marks best accuracy, ECE, and NLL per backbone.

Backbone	Method	α	ρ	Acc. (%)	MMC	ECE	NLL
VGG-19-BN	Source	–	–	13.24	0.3653	0.2329	5.5392
VGG-19-BN	TENT	–	–	34.60	0.4578	0.1118	3.4309
VGG-19-BN	MIRA	0.10	0.90	**34.69**	0.3359	**0.0281**	**3.3540**
ResNet-50	Source	–	–	16.95	0.3945	0.2250	5.2586
ResNet-50	TENT	–	–	39.55	0.4350	0.0395	2.9735
ResNet-50	MIRA	0.01	0.90	**39.67**	0.4231	**0.0266**	**2.9709**
ResNet-34	Source	–	–	15.47	0.4018	0.2471	5.5277
ResNet-34	TENT	–	–	37.47	0.4299	0.0553	3.1236
ResNet-34	MIRA	0.01	0.80	**37.50**	0.4171	**0.0430**	**3.1211**
WRN-101-2	Source	–	–	22.24	0.4501	0.2277	4.7710
WRN-101-2	TENT	–	–	44.78	0.4817	0.0423	2.7460
WRN-101-2	MIRA	0.01	0.90	**44.91**	0.4709	**0.0330**	**2.7411**

C Additional Experiments

We further evaluate MIRA under continual adaptation, alternative TTA protocols, stronger distribution shifts, dynamic target streams, and computationally realistic settings. Green marks MIRA variants; bold follows each table's comparison scope.

C.1 Continual and Dynamic Adaptation

The continual comparison uses CIFAR-100-C / ResNeXt at severity 5. The dynamic comparisons follow the PTTA protocol [42], which models changing distributions and correlated target samples.

Table 16: Continual and dynamic adaptation. Bold compares each base/MIRA pair.

Protocol	Method	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
Continual	CoTTA	64.16	0.6170	0.0529	1.4173
	$\text{MIRA}_{\text{CoTTA}}$	**65.20**	0.6787	**0.0342**	**1.3459**
Dynamic	TENT	14.75	0.9675	0.8201	17.6335
	$\text{MIRA}_{\text{TENT}}$	**15.16**	0.9650	**0.8134**	**17.0462**
Dynamic	RoTTA	74.89	0.8331	0.0842	0.8933
	$\text{MIRA}_{\text{RoTTA}}$	**75.67**	0.8241	**0.0674**	**0.8622**

MIRA improves accuracy, ECE, and NLL for all three base-method pairs. A separate reset-based ImageNet-C / ResNet-50 comparison under the PTTA protocol (batch size 64, 10,000 examples) increases accuracy from 37.50% to 40.67% and decreases ECE from 0.0284 to 0.0278, while NLL increases from 2.7080 to 2.9286.

C.2 Computationally Realistic Adaptation

We additionally evaluate MIRA on ImageNet-C and ImageNet-R [12] using ResNet-50 and batch size 64 under the time-constrained protocol of Alfarra et al. [1]. The stream arrival rate is ηr, where r is the forward-pass rate of the non-adapted model. The recorded conventional and realistic settings use $\eta = 0$ and $\eta = 1$, respectively. In the conventional setting, every batch can be used for adaptation regardless of processing time. In the realistic setting, batches arrive at a constant rate, so a slower method receives fewer adaptation opportunities.

Table 17: Evaluation under conventional ($\eta = 0$) and computationally realistic ($\eta = 1$) TTA protocols. Bold compares each dataset and realistic regime.

Dataset	**Method**	**Realistic**	**Acc. (%) ↑**	**MMC**	**ECE ↓**	**NLL ↓**
ImageNet-C	TENT	No	42.71	0.5203	0.0935	2.9071
	$\text{MIRA}_{\text{TENT}}$	No	**45.02**	0.5007	**0.0573**	**2.7948**
	TENT	Yes	38.97	0.4444	0.0572	3.0884
	$\text{MIRA}_{\text{TENT}}$	Yes	**39.92**	0.4412	**0.0484**	**3.0072**
ImageNet-R	TENT	No	41.84	0.5834	0.1649	3.2015
	$\text{MIRA}_{\text{TENT}}$	No	**44.10**	0.5771	**0.1361**	**2.9875**
	TENT	Yes	40.74	0.5404	0.1330	3.2062
	$\text{MIRA}_{\text{TENT}}$	Yes	**41.13**	0.5405	**0.1292**	**3.1631**

MIRA improves accuracy, ECE, and NLL in both conventional and computationally realistic settings. Although the gains become smaller under the realistic protocol, they remain consistent across the evaluated metrics.

C.3 Accuracy–Cost Trade-off

Finally, we compare MIRA with TENT and computationally heavier TTA methods under the same hardware setting. Table 18 reports classification performance, runtime, and peak memory for the single-run CIFAR-100-C / ResNeXt comparison.

Table 18: Single-run accuracy–cost comparison on CIFAR-100-C / ResNeXt (batch size 64). Bold marks the best values in directional columns.

Method	**Acc. (%) ↑**	**MMC**	**ECE ↓**	**NLL ↓**	**Time (s) ↓**	**VRAM (MiB) ↓**
TENT	67.10	0.7474	0.0779	1.2873	**125**	**2338**
CoTTA	64.16	0.6170	**0.0529**	1.4173	1142	3062
PETAL	62.30	0.5816	0.0582	1.5059	1586	3930
$\text{MIRA}_{\text{TENT}}$	**67.75**	0.7310	0.0556	**1.2401**	157	2494

Compared with TENT, MIRA introduces moderate additional runtime and memory overhead while improving accuracy, ECE, and NLL. It remains substantially more efficient than CoTTA and PETAL, supporting the practicality of cached last-layer Monte Carlo sampling.

C.4 Repeated Evaluation under Representation Shift

ImageNet-Sketch [39] tests a stronger representation shift. Table 19 uses ResNet-50, learning rate 2.5×10^{-4}, and seeds 0, 1, and 2. MIRA improves mean accuracy by 1.35 percentage points and reduces ECE and NLL.

Table 19: ImageNet-Sketch / ResNet-50, mean $\pm$ sample SD over seeds 0, 1, and 2. Bold marks best accuracy, ECE, and NLL.

Method	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
Source	$23.84_{\pm 0.35}$	$0.4623_{\pm 0.0013}$	$0.2240_{\pm 0.0029}$	$5.0889_{\pm 0.0089}$
TENT	$28.22_{\pm 0.42}$	$0.4507_{\pm 0.0031}$	$0.1685_{\pm 0.0014}$	$4.1317_{\pm 0.0163}$
$\text{MIRA}_{\text{TENT}}$	$\mathbf{29.57}_{\pm 0.29}$	$0.4204_{\pm 0.0025}$	$\mathbf{0.1247}_{\pm 0.0007}$	$\mathbf{3.9175}_{\pm 0.0264}$

C.5 Modified One-Step CLIP Evaluation

Table 20: CLIP ViT-B/16 TENT-component comparison: one update per batch with a shared $10\times$ released learning rate. Bold compares each dataset pair.

Method	Acc. (%) ↑	MMC	ECE ↓	NLL ↓
CIFAR-10-C				
TENT	52.74	0.9443	0.4170	4.6682
$\text{MIRA}_{\text{TENT}}$	**54.13**	0.9387	**0.3977**	**4.5456**
CIFAR-100-C				
TENT	27.83	0.8382	0.5603	8.9705
$\text{MIRA}_{\text{TENT}}$	**32.71**	0.8047	**0.4784**	**7.5339**

D Experimental and Implementation Details

D.1 Experimental Setup

We use corruption severity 5 unless otherwise stated. Source training, preprocessing, and target streams are matched within each base/MIRA comparison. All experiments are conducted on NVIDIA RTX A6000 GPUs.

For the TENT pairs in Table 2, both benchmarks use 15 severity-5 corruptions, batch size 64, and 10,000 shuffled examples per corruption. Model and optimizer are reset between corruptions, and each base/MIRA pair receives the same ordered streams. MIRA uses the feature-wise He-in construction in Appendix D.4 and the defaults in Appendix D.5.

The loss, construction, and sensitivity comparisons use paired seeds 1, 2, and 3. Scope, budget, architecture diagnostics, and most stream/ViT/CLIP extensions are single-run.

D.2 Evaluation Metrics

We report classification accuracy, mean maximum confidence (MMC), expected calibration error (ECE), and negative log-likelihood (NLL).

Let p_i denote the predictive probability for sample i, $\hat{y}_i = \arg\max_c p_{i,c}$, and $c_i = \max_c p_{i,c}$. Accuracy is the fraction of correct top-1 predictions, while $\text{MMC} = \frac{1}{N}\sum_i c_i$ measures average predictive confidence.

ECE uses 10 equal-width confidence bins [10], and NLL uses the probability assigned to the correct class:

$$\begin{aligned} \mathrm{ECE} &= \sum_{m=1}^{10} \frac{|\mathcal{B}_m|}{N} \left|\mathrm{acc}(\mathcal{B}_m) - \mathrm{conf}(\mathcal{B}_m)\right|, \\ \mathrm{NLL} &= -\frac{1}{N} \sum_{i=1}^{N} \log p_{i,y_i}, \end{aligned} \tag{20}$$

with probabilities lower-bounded by 10^{-12} for numerical stability. Each metric is computed over the evaluated examples of each corruption and then macro-averaged across corruptions.

For three-run results, we report sample standard deviation (SD), $s = \sqrt{\sum_{r=1}^{3}(v_r - \bar{v})^2/2}$, rather than the standard error $s/\sqrt{3}$. Table 1 reports accuracy and its SD as fractions; accuracy columns labeled (%) use percentages, with SD in percentage points. MMC, ECE, and NLL use their native units. Run variability includes target-stream order, batch composition, Gaussian sampling, and cuDNN nondeterminism. SD describes variability rather than establishing statistical significance.

D.3 Baseline Implementation Details

We retain each base method's optimizer settings. For the TENT pairs in Table 2, SGD uses a learning rate of 10^{-3} on CIFAR-100-C / ResNeXt and 2.5×10^{-4} on ImageNet-C / ResNet-50, with seeds 1, 2, and 3 on both benchmarks. CIFAR-100-C EATA and ROID use a learning rate of 2.5×10^{-4} and seeds 1, 2, and 3; their ImageNet-C counterparts use seeds 2, 4, and 6. We use momentum 0.9 and zero weight decay unless otherwise specified. When MIRA is integrated with a base TTA method, we retain the optimizer and learning rate of that method rather than retuning them.

The ImageNet-C / ResNet-34 loss ablation in Table 1 uses SGD at 2.5×10^{-4}, $K = 300$, 15 severity-5 corruptions, and reset between corruptions.

For EATA, we additionally use Fisher regularization of 1.0, with the Fisher information matrix estimated from 2000 source samples. Fisher estimation is a source-dependent component of the EATA baseline; attaching MIRA introduces no additional source-data requirement.

We also compare against uncertainty-estimation baselines including Deep Ensemble (DE), LLLA, ICLA, BBB, and ABNN. DE averages predictions from independently trained source models. LLLA follows the official Laplace implementation, while the remaining methods follow their released implementations or published settings using the same backbone and evaluation protocol whenever possible.

D.4 Perturbation Scale Construction

Let $W \in \mathbb{R}^{C \times D}$ denote the final classifier weight and $h_i \in \mathbb{R}^D$ the cached feature of target sample i. All headline experiments use a He-style base scale with a feature-wise, batch-dependent factor:

$$\begin{aligned} s_0 &= \sqrt{2/D}, & r_{\mathcal{B},d} &= \frac{1}{B} \sum_i |h_{i,d}|, \\ g_{\mathcal{B},d} &= \mathrm{clip}\left(\frac{r_{\mathcal{B},d}}{D^{-1} \sum_j r_{\mathcal{B},j}}, 0.25, 4.0 \right), \\ A_{\mathcal{B}}[c,d] &= s_0 g_{\mathcal{B},d}, & a_{\mathcal{B}}[c] &= s_0, \end{aligned} \tag{21}$$

where $g_{\mathcal{B},d}$ is shared across output classes. The He-style factor accounts for classifier input dimension, while the feature-wise factor adapts the perturbation geometry to the current target representation. Neither factor is tuned per dataset or model; only the global perturbation radius σ is selected online.

D.5 Hyperparameter Settings

Full MIRA uses the common defaults

$$\alpha = 0.03, \quad \rho = 0.90, \quad \tau = 0.25, \quad \gamma = 1, \quad K = 300, \quad B = 64. \tag{22}$$

For all headline experiments, we use the fixed candidate scale set

$$\mathcal{G} = \{10^{-5}, 10^{-4}, 3{\times}10^{-4}, 10^{-3}, 3{\times}10^{-3}, 10^{-2}, 3{\times}10^{-2}, 5{\times}10^{-2}, \\ 10^{-1}, 3{\times}10^{-1}, 5{\times}10^{-1}, 1, 2, 3, 4, 5\}. \tag{23}$$

The same Monte Carlo noise samples are reused across candidate scales within each target batch. Since the backbone features are cached, scale evaluation requires only repeated final-layer operations.

No corruption-specific or dataset-specific tuning is performed.

D.6 Scale Selection

At each grid candidate we compute batch-average MI and agreement using shared Gaussian draws. The implementation forms $T_M = \min_j M(\sigma_j) + \alpha[\max_j M(\sigma_j) - \min_j M(\sigma_j)]$ and selects the first candidate satisfying $M \geq T_M$ and $A \geq \rho$. When the preceding candidate lies below T_M and both interval endpoints meet the agreement condition, MI is interpolated linearly along $\log \sigma$. If no candidate satisfies both conditions, σ^* is the first candidate satisfying the MI condition, which always exists for $\alpha \leq 1$; the disagreement bound does not apply to such selections. The MI-range minimum need not equal the first candidate's MI in a numerically non-monotone curve. After selection, independent Gaussian draws are used for the loss and prediction at σ^*.

Default16 is the candidate set specified above. Coarse9 is

$$\{10^{-5}, 3 \times 10^{-4}, 3 \times 10^{-3}, 3 \times 10^{-2}, 10^{-1}, 5 \times 10^{-1}, 2, 4, 5\}. \tag{24}$$

Dense31 inserts the geometric midpoint between each pair of default candidates; Extended20 adds $\{10^{-6}, 3 \times 10^{-6}, 10, 20\}$ to Default16. All grid comparisons keep $K = 300$ and identical common-noise shape.

For deployment, increase α when probes repeatedly remain weak, and reduce α when perturbations frequently change class predictions. The agreement bound applies only to selections that satisfy both conditions. Appendix B.5 reports the joint sweep.

D.7 CLIP Implementation and Configuration

CLIP ViT-B/16 uses the released CLIPTTA [19] architecture, data pipeline, and TENT component. Logits are similarities between normalized image features and class-specific text embeddings. The text-embedding matrix is treated as the effective classifier and receives Gaussian perturbations only for reliability probing; the unperturbed embeddings remain fixed. Adaptation updates visual encoder normalization parameters, not the text embeddings.

Table 3 uses the released ten-update-per-batch TENT-component configuration, with identical optimization settings for TENT and MIRA and no MIRA-specific tuning. The modified comparison in Table 20 uses one update and a shared $10\times$ learning-rate multiplier. The effective similarity classifier is not a learned linear head, and these comparisons assess the TENT component rather than the complete CLIPTTA method.

E Reliability Analysis

Across CIFAR-10-C, CIFAR-100-C, and ImageNet-C, Figure 4 relates predictive entropy and perturbation-induced MI to batch accuracy. These empirical diagnostics complement the fixed-scale example in Figure 1(b).

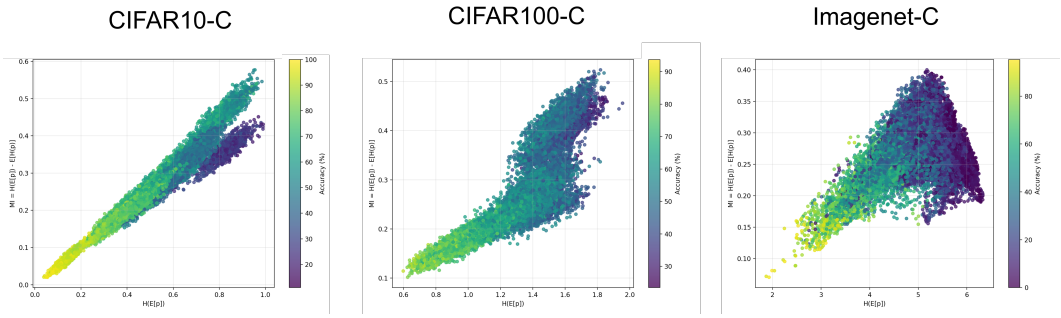

Figure 4: Predictive entropy and perturbation-induced MI across CIFAR-10-C, CIFAR-100-C, and ImageNet-C (left to right); color denotes batch accuracy. MI measures local classifier disagreement, rather than calibrated Bayesian uncertainty.

To test whether MI provides information beyond predictive entropy, we compare batches with similar $H(\mathbb{E}[p])$. Figure 5 shows that high-MI batches tend to have lower accuracy than low-MI batches at similar entropy. Thus, local perturbation sensitivity supplies a complementary reliability signal. This observational comparison supports the interpretation of MI and does not establish a causal benefit from weighting.

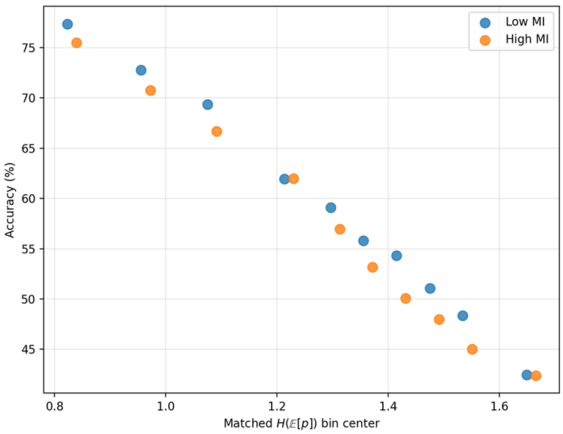

Figure 5: Batch accuracy at matched predictive-entropy bin centers, pooling corruption types. Low-MI batches tend to be more accurate than high-MI batches.

Matching predictive entropy makes the two groups comparable in the overall sharpness of their mean predictions. At a similar bin center, the remaining accuracy gap therefore reflects an association with perturbation sensitivity beyond the mean entropy alone. The groups approach one another in some bins, so MI is a relative reliability cue within the probed classifier neighborhood, rather than a calibrated error probability. This is consistent with MIRA's detached weighting: low-MI samples receive more influence, while the confidence term separately limits updates from low-evidence predictions. The analysis therefore motivates combining the two signals instead of treating either as a complete description of reliability.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's main contributions and scope. They clearly state the proposed method, the problem setting, and the intended contributions, while the claims are supported by the theoretical analysis and empirical results presented in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation is included in the last section of the paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix A states the Gaussian-perturbation assumptions and the scope of the local approximation; the agreement bound is explicitly conditional on feasible selection.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and Appendix D specify the core method and repeated Full MIRA protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have included a main part of the code at `https://hyeongyu-kim.github.io/mira/`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All information is included in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Main accuracy and the repeated loss, construction, and sensitivity results report sample SD; Appendix D.2 defines SD and its sources of variability; Appendix D.1 specifies the repeated comparisons. Single-run extensions are identified separately.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have included compute regarding results in the main paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 1 motivates more reliable deployment under distribution shift; the Limitations paragraph discusses bias reinforcement under unrepresentative unlabeled streams and limits of confidence-based deployment claims.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cited the original sources, and each license and its terms of use can be found in the corresponding original paper or official repository. All assets are used in accordance with these licenses and terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.