

AdMiT: Adaptive Multi-Source Tuning in Dynamic Environments

Xiangyu Chang¹, Fahim Faisal Niloy¹, Sk Miraj Ahmed^{1,2*}, Srikanth V. Krishnamurthy¹, Basak Guler¹, Ananthram Swami⁴, Samet Oymak³, Amit Roy-Chowdhury¹

¹University of California, Riverside ²Brookhaven National Laboratory

³University of Michigan, Ann Arbor ⁴DEVCOM Army Research Laboratory

{cxian008@, fnilo001@, sahme047@, krish@cs, basakg@, amitrc@ece.}ucr.edu
ananthram.swami.civ@army.mil, oymak@umich.edu

Abstract

Incorporating transformer models into edge devices poses a significant challenge due to the computational demands of adapting these large models across diverse applications. Parameter-efficient tuning (PET) methods (e.g. LoRA, Adapter, Visual Prompt Tuning, etc.) allow for targeted adaptation by modifying only small parts of the transformer model. However, adapting to dynamic unlabeled target distributions at the test time remains complex. To address this, we introduce AdMiT: Adaptive Multi-Source Tuning in Dynamic Environments. AdMiT innovates by pre-training a set of PET modules, each optimized for different source distributions or tasks, and dynamically selecting and integrating a sparse subset of relevant modules when encountering a new, few-shot, unlabeled target distribution. This integration leverages Kernel Mean Embedding (KME)-based matching to align the target distribution with relevant source knowledge efficiently, without requiring additional routing networks or hyperparameter tuning. AdMiT achieves adaptation with a single inference step, making it particularly suitable for resource-constrained edge deployments. Furthermore, AdMiT preserves privacy by performing an adaptation locally on each edge device, without the need for data exchange. Our theoretical analysis establishes guarantees for AdMiT’s generalization, while extensive benchmarks demonstrate that AdMiT consistently outperforms other PET methods across a range of tasks, achieving robust and efficient adaptation.

1. Introduction

Pretrained transformers [1–5] have achieved remarkable success across diverse tasks, but their large parameter counts—often reaching billions [2, 5]—present challenges for deployment, especially on edge devices with limited computational resources. To address these limitations, parameter-

efficient tuning (PET) methods, such as prefix/prompt tuning [6–9], adapters [10], and LoRA [11], have been introduced. These methods allow the pretrained model to remain fixed while only adjusting a small set of additional parameters tailored to specific target distributions, significantly reducing memory and computation needs while often achieving performance comparable to that of full fine-tuning.

Most existing PET methods are either single-source—focusing on a single PET module trained for one distribution—or, when incorporating multiple PET modules, require additional computational resources such as routing networks [12] or extensive hyperparameter tuning [13, 14]. These approaches lack the ability to directly integrate knowledge from multiple PET modules, each trained on different source distributions. In *dynamically evolving environments*, adaptation methods benefit from leveraging multiple sources of pre-trained knowledge. Instead of relying on a single PET module trained on a single source, integrating multiple PET modules enables more robust adaptation to shifting distributions by drawing from a diverse set of source-specific knowledge[15]. This multi-source approach is particularly advantageous when access to the original source data used for training each module is restricted due to privacy, storage, or other constraints. In such scenarios, training a unified PET module across combined sources is infeasible, making it both practical and effective to adaptively employ and integrate an array of pre-trained PET modules during test time, resulting in performance improvements often unattainable with single-source PET adaptation.

In this work, we introduce **AdMiT** (Adaptive Multi-Source Tuning in Dynamic Environments), a novel framework designed to efficiently adapt pre-trained PET modules across multiple dynamic distributions. AdMiT pre-trains a structured set of PET modules, each specifically tuned to a different source distribution, providing a versatile foundation for multi-source adaptation. During test time, when faced with new, small-batch target data from a new distri-

*Work done while the author was at University of California, Riverside.

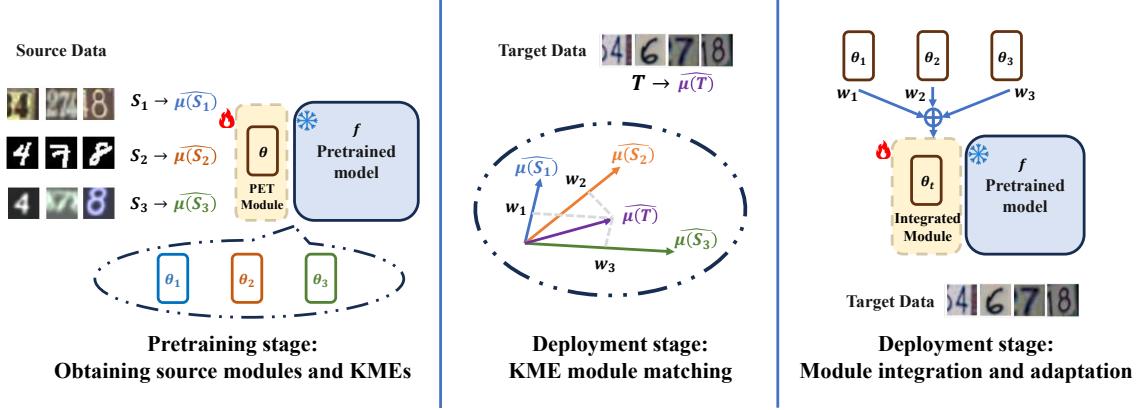


Figure 1. The diagram illustrates the AdMiT workflow, which includes pretraining source modules, matching modules during deployment, and updating the integrated module. In the **pretraining stage**, given a loss function \mathcal{L} and source distributions $\{\mathcal{D}_{S_j}\}_{j=1}^N$, we freeze the base model f and fine-tune each module θ_j for its respective distribution. We also map the source data to an embedding space \mathcal{H} as empirical Kernel Mean Embeddings (KMEs) $\{\widehat{\mu(S_j)}\}_{j=1}^N$. During the **KME module matching stage (Sec. 3.1)**, AdMiT maps the target data T to an empirical KME $\widehat{\mu(T)}$ and approximates it as a weighted combination of source KMEs, determining the weight coefficients \hat{w}_j . In the **module integration and adaptation stage (Sec. 3.2)**, AdMiT integrates the source modules with the highest weights to create an adaptive module $\theta(t)$ for the current target batch. This module $\theta(t)$ can then be directly applied to the target distribution or further adapted using sharpness-aware pseudo-label minimization for enhanced alignment with the target data.

bution, AdMiT (1) selects a relevant subset of PET modules based on their alignment with the target distribution, and (2) integrates these selected modules into a newly composed module optimized for the current target batch. This integrated module can then be directly applied to the target distribution, achieving efficient adaptation with lower computational overhead compared to existing PET methods [12, 14, 16]. Additionally, this integrated module enables further fine-tuning if desired, allowing AdMiT to dynamically enhance its alignment with the target distribution. This dual capability—of zero-shot applicability and optional on-the-fly adaptation—enables AdMiT to adapt robustly to evolving target distributions with low computational cost. Our experiments demonstrate AdMiT’s effectiveness in both zero-shot and test-time adaptation settings, highlighting its adaptability and strong performance across dynamic distributions.

AdMiT offers two key advantages. First, by dynamically matching and adapting multiple pre-trained PET modules to the target distribution using small batches of target data, AdMiT achieves superior performance over traditional single-source PET adaptation methods. The ability to integrate multiple PET modules enables AdMiT to more effectively capture complex target distributions by leveraging a diverse set of source-specific knowledge through Kernel Mean Embedding (KME)-based matching (Sec. 3.1). Second, AdMiT bypasses the need for additional hyperparameter tuning or routing network training during deployment by using KME to align the target with relevant source distributions. This approach eliminates the computational burden of full model inference, allowing for fast multi-source PET adaptation.

As a result, AdMiT is particularly well-suited for resource-constrained edge deployments, where both efficiency and flexibility are essential. Moreover, this KME-based distribution matching ensures data privacy, as no raw data exchange is required during adaptation.

Main Contributions. We present a new multi-source PET approach, AdMiT, that enables edge devices to selectively integrate a minimal subset of PET modules from a pre-trained collection, adapting in real-time to new, unlabeled data in an unsupervised, few-shot setting. Our contributions include the following:

- **Adaptive Multi-source Module Selection and Integration.** AdMiT efficiently selects and integrates a subset of pre-trained PET modules from a structured collection, based on the distributional characteristics of incoming target data. This adaptive integration avoids the computational burden associated with training routing networks or hyperparameter tuning for each new target distribution. By selectively combining relevant modules, AdMiT achieves performance comparable to that obtained by using all modules simultaneously but with significantly reduced storage and computational requirements, making it practical for edge deployment.
- **Efficient, Privacy-preserving Adaptation.** Unlike existing PET methods that utilize additional routing networks or data-alignment steps for multi-source adaptation [12], AdMiT achieves adaptation by transforming empirical data distributions in a kernel embedding space [17, 18]. This approach avoids the need to exchange raw data, preserving data privacy and reducing computational costs. Moreover,

AdMiT performs efficient module selection and combination without additional inference steps, enabling real-time adaptation on edge devices.

- **Theoretical Guarantees.** We provide theoretical guarantees on AdMiT’s generalization performance, showing how it can effectively balance the sample sizes of source modules and the target batch size to ensure reliable multi-source adaptation. This guarantee highlights that well-trained source modules can provide robust adaptation even with limited target data.
- **Comprehensive Empirical Evaluation and Insights.** Extensive evaluations on challenging datasets, including Digit-Five, CIFAR-100C, and ImageNet-C, demonstrate that AdMiT consistently surpasses existing PET methods across various adaptation distributions. AdMiT shows a notable improvement in accuracy when adapting to new distributions (Table 1) and effectively preserves knowledge from source distributions (Table 2), showcasing strong performance in adaptation and retention. Additionally, AdMiT effectively identifies and integrates the most relevant modules (Figure 3) consistently achieving optimal results with minimal computational overhead. Additional results on the segmentation task, using Cityscapes [19] and ACDC [20] datasets, are provided in the Appendix, demonstrating AdMiT’s effectiveness in handling dynamic distributions across different tasks.

2. Related Works

Parameter Efficient Tuning (PET). Large-scale pre-trained models have greatly enhanced performance in natural language processing [1] and computer vision [21] by transferring learned knowledge to downstream tasks. PET methods, such as prompt tuning [21] and adapters [10], allow efficient adaptation by fine-tuning a small subset of parameters. Techniques like CoOp [22] and CoCoOp [23] leverage prompt optimization for out-of-distribution generalization, while CLIP-Adapter [24] and Tip-Adapter [25] fine-tune CLIP using adapters or key-value cache models, improving adaptability to target distributions. However, most PET methods are not designed to handle continuously shifting small-batch target distributions effectively; they adapt independently to each distribution, creating distribution-specific PET modules and often forgetting previously learned information. In contrast, our method enables consistent adaptation to dynamic target distributions while preserving knowledge of the pre-trained source distributions, addressing a gap in existing PET approaches. Our approach is compatible with various PET modules, including LoRA [11], VPT [21], and adapters [10].

Test Time Adaptation (TTA). Unsupervised Domain Adaptation (UDA) requires extensive target distribution data for offline adaptation, whereas TTA operates continuously on incoming test batches [26–28]. Initial TTA approaches [29] used test-batch statistics rather than training data, with meth-

ods like TENT [30] updating batch-normalization parameters to reduce entropy on target data. DUA [31] further refines alignment with target distributions by persistently updating batch-norm statistics across test batches. While these single-source TTA methods are effective, they often struggle with forgetting source knowledge over time, particularly in dynamic settings. Approaches like CoTTA and BeCoTTA [32, 33] use stochastic source restoration to mitigate drift, and EATA [34] employs regularization to preserve critical parameters, thus reducing forgetting. However, these methods often adapt to each batch separately, requiring substantial computational resources to balance adaptation and forgetting. In contrast, our multi-source approach dynamically matches relevant source modules to the target distribution, integrating them efficiently and minimizing forgetting with minimal computational overhead.

Ensemble Learning and Multi-source Adaptation. Ensemble learning, a well-known technique, enhances model robustness by combining outputs from various models [35]. Techniques like SESoM [12] and mixture models [14] aim to handle dynamic target distributions by combining multiple pre-trained models or PET modules. However, due to privacy constraints or storage limitations, direct access to source data from pre-trained models or PET modules is often unavailable. This limitation requires ensemble methods to rely on additional hyperparameter tuning [14, 16] or routing networks [12] to match the target and source distributions, leading to substantial computational overhead and frequent forward inference. For CNN-based applications, such adjustments are manageable [15, 36], but in the context of large pre-trained transformers, this tuning becomes prohibitively costly. Our approach bypasses these constraints by performing source-target matching through Kernel Mean Embeddings (KMEs), enabling efficient and privacy-preserving adaptation without requiring raw data or extensive computation, making it well-suited for large-scale pre-trained models in dynamic environments.

3. Proposed Method: AdMiT

Our method, AdMiT, leverages a pretrained transformer model f along with a collection of parameter-efficient tuning (PET) modules, each pretrained on distinct source tasks or domains. These PET modules, represented by parameters $\{\theta_j\}_{j=1}^N$, have significantly fewer parameters than the base model f and can be flexibly integrated into f as needed, thus forming a structured repository of source knowledge. We denote the transformer f combined with a module θ as f_θ . The core idea of AdMiT is to optimize adaptation to a target task or domain by selectively blending these pretrained modules based on their distributional representations in an embedding space and combining their weights to maximize relevance. The adaptation to target distribution is shown in figure 1.

Algorithm 1 AdMiT: Adaptive Multi-Source Tuning in Dynamic Environments.

- 1: **Input:** Pretrained transformer model f , Pretrained source modules $\{\theta_j\}_{j=1}^N$, empirical KME of source modules $\{\widehat{\mu}(S_j)\}_{j=1}^N$, number of modules to be selected M ($M < N$), streaming sequential unlabeled test data $T^{(1)} = \{x_i^{(1)}\}_{i=1}^{|T|} \rightarrow T^{(2)} = \{x_i^{(2)}\}_{i=1}^{|T|} \rightarrow \dots T^{(t)} = \{x_i^{(t)}\}_{i=1}^{|T|} \rightarrow \dots$
 - 2: **Output:** M scaled weights, finetuned new module $\theta(t)$, size of synthetic dataset Z
 - 3: Use Alg. A (in the Appendix) to generate synthetic data and estimate $\{\widehat{\mu}(S_j)\}_{j=1}^N$
 - 4: **while** $t \geq 1$ **do**
 - 5: **for** Each x_i in the t -th batch **do**
 - 6: Calculate the empirical KME of the target batch (Eqn. 3)
 - 7: **end for**
 - 8: Obtain mixture weights $\{\hat{w}(t)^j\}_{j=1}^N$ by solving Eqn. 4
 - 9: Find and select the top M in $\{\hat{w}(t)^j\}_{j=1}^N$
 - 10: Rescale selected weights to sum up to 1, thus obtain $\{\bar{w}(t)^j\}_{j=1}^M$.
 - 11: Create a new module $\theta(t)$ by a weighted averaging of the selected pretrained modules

$$\theta(t) = \sum_{j=1}^M \bar{w}(t)^j \theta_j$$
 - 12: Finetune $f_{\theta(t)}$ with Eqn. 8
 - 13: **end while**
-

- **Pretraining stage: Pretraining and KME calculation.**

Given a loss function \mathcal{L} and a set of source tasks or domains $\{\mathcal{D}_{Sj}\}_{j=1}^N$, we freeze the base transformer model f and only update the PET modules, rather than fully fine-tuning f on each domain. For each source \mathcal{D}_{Sj} , we optimize the parameters θ_j as $\theta_j = \operatorname{argmin}_\theta \mathcal{L}(f_\theta; \mathcal{D}_{Sj})$. To capture the source distributions, we map the source data to an embedding space \mathcal{H} and represent the empirical distributions using Kernel Mean Embeddings (KMEs) $\{\widehat{\mu}(S_j)\}_{j=1}^N$.

- **Deployment stage: KME Module matching. (Sec. 3.1)**

In this stage, AdMiT maps the target data T to an empirical Kernel Mean Embedding (KME) $\widehat{\mu}(T)$ and approximates it as a linear combination of source KMEs $\{\widehat{\mu}(S_j)\}_{j=1}^N$, expressed as $\widehat{\mu}(T) = \sum_{j=1}^N w_j \widehat{\mu}(S_j)$. This approach to multi-source distribution estimation, commonly applied in previous works [15, 16, 36], enables us to interpret the mixture weights $\{w_j\}$ as relevance scores for each source module. These weights guide the selection and integration of source modules, ensuring that the target is adapted efficiently and effectively. We also provide a theoretical bound on the estimation error for this approximation.

- **Deployment stage: Module integration and adaptation. (Sec. 3.2)**

Using the computed mixture weights \hat{w}_j , AdMiT selects the source modules with the highest weights and integrates them to create a combined module θ . This integrated module θ is subsequently fine-tuned on the target domain \mathcal{D}_T to refine its alignment with the target distribution. For further enhancement, we apply sharpness-aware pseudo-label minimization [37, 38] to adjust the ensemble module and improve its robustness on the target domain.

Throughout the deployment stage, only a small number of unlabeled target samples are required to identify and integrate the source modules most relevant to the current target distribution. A detailed pseudocode for AdMiT can be found in Algorithm 1. In the following sections, we provide an in-depth explanation of the principles guiding the design of AdMiT.

3.1. Module Matching using KME

In the pretraining stage, given the heterogeneous feature spaces across different models, we assume a unified feature space facilitated by a public feature extractor $G(\cdot)$, which maps the original data x' from both source and target distributions into a shared representation space $\mathbf{x} = G(x')$. This setup is practical, as publicly available pre-trained models can serve as feature extractors. In our experiments, we use a DenseNet201 model [39] pre-trained on ImageNet for this purpose. Since source data are inaccessible during the deployment stage, we require a metric to assess the similarity between source and target distributions without exchanging data or performing forward model inference.

Kernel Mean Embedding (KME) [40–43] provides a powerful tool for measuring distribution similarity. KME maps probability distributions into vectors in a high-dimensional Reproducing Kernel Hilbert Space (RKHS) \mathcal{H} using a positive semi-definite bounded kernel $0 \leq k(\cdot, \cdot) \leq K$, simplifying the similarity evaluation of two distributions to inner product calculations in RKHS. Given a distribution \mathcal{P} of an \mathcal{X} -valued random variable, its KME is defined as:

$$\mu_k(\mathcal{P}) := \int_{x \in \mathcal{X}} k(x, \cdot) d\mathcal{P}(x). \quad (1)$$

The norm of the KME in \mathcal{H} can be expressed by the inner product:

$$\|\mu_k(\mathcal{P})\|_{\mathcal{H}}^2 := \langle \mu_k(\mathcal{P}), \mu_k(\mathcal{P}) \rangle = \mathbf{E}_{x, y \sim \mathcal{P}} k(x, y). \quad (2)$$

Since true distributions \mathcal{P} are typically unknown, we estimate the KME and its norm using a finite batch $X = \{x_n\}_{n=1}^{|X|} \sim \mathcal{P}$:

$$\widehat{\mu}(X) := \frac{1}{|X|} \sum_{n=1}^{|X|} k(x_n, \cdot), \quad (3)$$

$$\|\widehat{\mu(X)}\|_{\mathcal{H}}^2 := \frac{1}{|X|^2} \sum_{x_i, x_j \in X} k(x_i, x_j).$$

These empirical KMEs are computed on source datasets $\{S_j\}_{j=1}^N \sim \mathcal{D}_{Sj}$ during the pretraining stage, and on the target batch $T \sim \mathcal{D}_{\mathcal{T}}$ in the deployment stage.

In the deployment stage, we assume the target distribution can be approximated as a linear combination of source distributions, such that $\mathcal{D}_{\mathcal{T}} \approx \sum_{j=1}^N w_j \mathcal{D}_{Sj}$ for some mixture weights $\{w_j\}_{j=1}^N$. Using the linearity of expectation, we can express the KME of the target as $\mu_{\mathcal{D}_{\mathcal{T}}} \approx \sum_{j=1}^N w_j \mu_{\mathcal{D}_{Sj}}$. Each source KME $\{\widehat{\mu(S_j)}\}_{j=1}^N$ serves as a basis in the Hilbert space \mathcal{H} , allowing us to decompose the target empirical KME $\widehat{\mu(T)}$ using these bases. By solving the following optimization, we obtain the mixture weights $\{\hat{w}_j\}_{j=1}^N$ to match source and target distributions:

$$\min_{\{w_j\}_{j=1}^N} \left\| \widehat{\mu(T)} - \sum_{j=1}^N w_j \widehat{\mu(S_j)} \right\|_{\mathcal{H}}. \quad (4)$$

This KME-based approach serves as a reliable metric for distribution similarity (See Section H), enabling efficient and privacy-preserving matching between source and target distributions.

Theorem 3.1. *For a bounded kernel $0 \leq k(\cdot, \cdot) \leq K$, with probability at least $1 - \delta$, the (biased) empirical MMD (obtained by drawing m samples from $p = \mathcal{D}_{\mathcal{T}}$ and n samples from $q = \sum_{j=1}^N w_j \mathcal{D}_{Sj}$, with $\sum_{j=1}^N w_j = 1$) is bounded by: (Proof in the Appendix Corollary H.3.)*

$$\frac{1}{2} \left\| \widehat{\mu(T)} - \sum_{j=1}^N w_j \widehat{\mu(S_j)} \right\|_{\mathcal{H}} < \sqrt{\frac{K}{m}} + \sqrt{\frac{K}{n}} + \sqrt{\frac{K(m+n) \log \frac{1}{\delta}}{2mn}}.$$

The mixture weight solution \hat{w}_j from optimization 4 also implies the similarity of the distribution of source domain \mathcal{D}_{Sj} and the target domain $\mathcal{D}_{\mathcal{T}}$, leading to the module selection strategy in Alg. 1.

Practical Considerations. In real-world applications, not all source modules are closely aligned with the target distribution, and calculating Kernel Mean Embeddings (KMEs) of source data in Eqn. 4 can be computationally intensive, as it involves summing up to $|S_j|$ kernel functions for each source. To enhance efficiency and reduce computational overhead, we employ the following strategies: (1) instead of using all source modules during adaptation, we select only the modules with the highest weights \hat{w}_j (as shown in Alg. 1), thus focusing on the most relevant sources, and (2) to approximate each source KME $\widehat{\mu(S_j)}$, we generate a smaller synthetic dataset $\{z_m\}_{m=1}^Z$ (where $Z \ll |S_j|$). This synthetic dataset enables efficient computation by reducing the number of kernel functions involved, and it preserves

privacy by eliminating the need for raw data exchange during KME decomposition [44]. Details of the synthetic data generation algorithm for source KMEs are provided in the Appendix, Alg. A.

Using synthetic datasets to approximate KMEs offers two major advantages. First, direct access to original source data is often restricted due to privacy or storage constraints, making it necessary to rely on the accessible information from source modules. Generating synthetic data allows us to create accurate KME approximations for each source module in a privacy-preserving way. Second, synthetic datasets provide a computationally efficient alternative to direct KME calculations using source data. By involving fewer data points, synthetic KMEs significantly reduce the computational load for matching the target distribution with source KMEs, making adaptation feasible even in resource-limited settings. We have theoretically demonstrated that (see Appendix Alg. A) the synthetic KMEs closely approximate the one calculated from the raw data. This fidelity ensures that the adaptation performance remains reliable and robust, as shown in our experiments, and enables efficient yet effective alignment of target and source distributions.

3.2. Module Integration and Adaptation

Drawing inspiration from the benefits of a good initialization in test-time adaptation and transfer learning [12, 16, 36, 45], we integrate PET modules trained on distributions related to the target distribution, as these modules are presumed to contain valuable knowledge relevant to these distributions. This integration is achieved through a weighted mixture of the selected modules, aiming to achieve a transfer gain:

$$\theta(t) = \sum_{i=1}^M \bar{w}_i \theta_i,$$

where \bar{w}_i is obtained from Alg. 1, and $\theta(t)$ represents the integrated module. Directly applying the integrated module on the target distribution results in a zero-shot adaptation, whose performance can be bounded by the following theorem.

Theorem 3.2 (Zero-shot adaptation loss bound). *Assume that the source training error is at most ϵ ; the loss $L(f_{\hat{\theta}_j}(x), f(x)) \in \mathcal{H}_k$; and the empirical MMD between $\sum_{j=1}^N w_j \mathcal{D}_{Sj}$ and $\mathcal{D}_{\mathcal{T}}$ is from Theorem 3.1. Then, the finite-sample loss is:*

$$\begin{aligned} \mathcal{L}(\mathcal{D}_{\mathcal{T}}, g, f_{\sum w_j \hat{\theta}_j}) &= \sum_{x_i \sim \mathcal{D}_{\mathcal{T}}} [L(f_{\sum w_j \hat{\theta}_j}(x_i), g(x_i))] \\ &\leq \epsilon + O\left(\sqrt{\frac{1}{m}} + \sqrt{\frac{1}{n}}\right) \end{aligned}$$

Proof can be found in the Appendix, Theorem H.5.

To further boost performance, we fine-tune $\theta(t)$ for the target distribution $\mathcal{D}_{\mathcal{T}}^{(t)}$ at time t :

$$\theta(t)^* = \arg \min_{\theta(t)} \mathcal{L}(f_{\theta(t)}; \mathcal{D}_{\mathcal{T}}^{(t)}).$$

Starting from the integrated PET module $\theta(t)$ offers an efficient initialization, as the fine-tuning does not incur additional forward inference costs even as the number of candidate modules N or selected modules M increases.

Practical Considerations. Although test-time adaptation (TTA) can stabilize the adapted models, it risks model collapse during the tuning process, where the model may incorrectly classify all inputs as belonging to a single category over time [38]. To mitigate this, we incorporate sharpness-aware techniques [37, 38] to make the model less sensitive to large gradients that may arise from test samples [34].

Once we obtain the new module $\theta(t)$ for the target batch $T^{(t)} = \{x_i^{(t)}\}_{i=1}^{|T|}$ at time step t , we compute the entropy of the pseudo-labels predicted by the model with this module. The entropy of the predictions for the t -th target batch from the model $f_{\theta(t)}$ is:

$$\mathcal{L}^{(t)} = -\mathbf{E}_{\mathcal{D}_T^{(t)}} \sum_{c=1}^K \hat{y}_c^{(t)} \log(\hat{y}_c^{(t)}), \quad (5)$$

To properly fine-tune the new module $\theta(t)$ with this pseudo-label entropy minimization, we aim to make the model insensitive to large gradients by encouraging convergence to a flat region of the entropy loss surface. This approach, which seeks a flat minimum, provides good generalization and robustness against large gradients [37, 38]:

$$\min_{\lambda} \mathcal{L}^{SA(t)}(\{x_i^{(t)}\}_{i=1}^B; \lambda), \quad (6)$$

$$\text{where } \mathcal{L}^{SA(t)} \triangleq \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}^{(t)}(\{x_i^{(t)}\}_{i=1}^B; \lambda + \epsilon) \quad (7)$$

"SA" denotes sharpness-aware. The gradient for this optimization can be approximated (see Appendix A for details):

$$\nabla_{\lambda} \mathcal{L}^{SA(t)} \approx \nabla_{\lambda} \mathcal{L}^{(t)}(\{x_i^{(t)}\}_{i=1}^B; \lambda)|_{\lambda+\epsilon^*(\lambda)}. \quad (8)$$

Applying Eqn. 8 instead of standard SGD to update the parameters of $\theta(t)$ based on Eqn. 5 results in a more robust solution for pseudo-label entropy minimization. The effect of sharpness-aware adaptation is discussed further in the ablation study.

4. Evaluations

In our experiments, we evaluate AdMiT's effectiveness by adapting PET modules pre-trained on source distributions to target data drawn from stationary or dynamically evolving

Table 1. Static Adaptation on ImageNet-C. Following a similar experiment setup in Fig. 2, we adapt to a target corruption domain by taking the rest $15 - 1 = 14$ domains as source domains, given varying target batch size. Due to space limitations, we report only the averaged accuracy across all target domains.

Source	Method	BS=256	BS=128	BS=64	BS=16	BS=1
Single	TENT-Best [30]	52.2	52.3	52.0	52.4	51.7
	TENT-Worst [30]	34.7	34.5	35.3	34.7	31.6
	BECoTTA-Best [33]	60.4	61.5	62.0	61.1	55.4
	BECoTTA-Worst [33]	35.3	36.4	37.9	37.3	30.4
	SAR-Best [38]	58.1	62.3	61.4	60.3	54.1
	SAR-Worst [38]	37.5	38.6	38.2	38.1	31.1
Multi	GT-Tuning	67.7	68.8	69.7	65.4	60.3
	π -tuning-PL [16]	61.4	62.4	62.7	61.5	57.1
	SESoM-PL [12]	62.3	62.6	62.5	62.0	56.3
	CONTRAST [15]	63.5	63.1	63.1	61.7	57.9
	Model soup [46]	52.3	53.4	52.2	51.7	49.5
	AdMiT	63.8	63.7	62.4	62.3	58.7
	AdMiT-ZeroShot	60.2	59.6	58.4	55.9	53.3
	AdMiT-Plain	63.5	62.2	62.1	61.0	56.6

distributions (See experiment setting details in Appendix Sec. F). The target distributions involve the same task as the source but differ due to distribution shifts relative to the source distributions on which the PET modules were trained.

We consider two adaptation scenarios: (1) a *static adaptation setting* where the target data are drawn from a stationary distribution, and (2) a *dynamic adaptation setting* where the target data are drawn sequentially from an evolving distribution. The former scenario demonstrates AdMiT's effectiveness in adapting to a stationary target distribution using pre-trained source modules, while the latter highlights AdMiT's robustness in adapting to evolving target distributions over time.

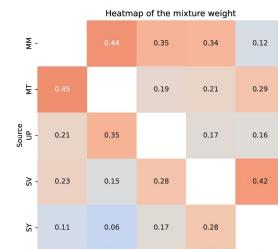
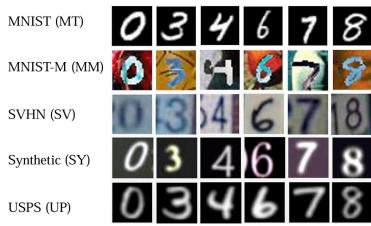
Datasets. For the static adaptation scenario, we evaluate AdMiT on the *Digits-Five* dataset [47], which includes five digit datasets—MNIST (MT), MNIST-M (MM), USPS (UP), SVHN (SV), and Synthetic Digits (SY)—each covering 10 classes (0-9). In these experiments, four distributions are used as sources, with the remaining one reserved for testing. We also use the *ImageNet-C* dataset [48], which applies 15 types of severe corruptions (details in Appendix Sec. G) to ImageNet images [49], following the setup in [38].

For the dynamic adaptation scenario, we utilize the *CIFAR-100C* benchmark [48], which extends the CIFAR-100 dataset [50] by introducing 15 types of noise at varying levels of severity (1 to 5). This setup results in up to 75 distinct distributions, allowing us to assess AdMiT's capacity for continuous adaptation as the target distribution evolves.

Finally, although our primary evaluation focuses on image classification tasks, our method is not limited to this setting. It can be extended to other tasks, such as semantic segmentation, with results for segmentation tasks provided in the Appendix E.

4.1. Baseline Methods

Our evaluation includes comparisons with state-of-the-art (SOTA) single-source test-time adaptation (TTA) methods,



Source	Method	MM	MT	UP	SV	SY	Avg
Single	TENT-Best [30]	56.1	98.4	84.9	87.0	95.2	84.3
	TENT-Worst [30]	17.6	54.2	59.6	11.4	15.5	31.7
	BECoTTA-Best [33]	52.3	97.6	85.7	86.4	95.6	83.5
	BECoTTA-Worst [33]	22.6	53.8	60.7	24.9	23.6	37.1
	SAR-Best [38]	57.8	97.3	86.2	86.4	92.0	83.9
	SAR-Worst [38]	36.9	57.3	62.9	40.4	35.8	46.6
	GT-Tuning	70.4	99.6	93.1	90.5	97.2	90.1
Multi	π -tuning-PL [16]	61.3	98.2	89.1	86.2	94.6	85.8
	SESoM-PL [12]	62.1	98.3	88.6	87.1	94.5	86.0
	CONTRAST [15]	63.1	97.6	88.5	<u>87.2</u>	94.3	86.1
	Model soup [46]	52.3	83.2	71.5	66.4	75.2	69.7
	AdMiT	<u>63.2</u>	98.9	<u>89.3</u>	86.5	<u>96.2</u>	<u>86.8</u>
	AdMiT-ZeroShot	54.5	97.2	86.3	80.5	92.3	82.2
	AdMiT-Plain	61.3	96.3*	87.4	85.3	93.1	84.7

Figure 2. **Static adaptation on Digits-Five.** (**Left**): Sample images from the source and target domains used in the adaptation task, which include MNIST (MT), MNIST-M (MM), SVHN (SV), Synthetic (SY), and USPS (UP). (**Center**): Heatmap depicting the mixture weights assigned to various source modules during adaptation to the target domain. Larger mixture weights (w_j) indicate a higher similarity between the target and source domains. The weights in each column sum to 1, as the four remaining domains are used to adapt to the target domain. (**Right**): We train the source modules using 4 digits datasets to perform adaptation on the remaining dataset. All the results are the average of 5 runs. **Best** performance is bolded, and second-best performance is underlined. The table clearly demonstrates that the average accuracy of AdMiT outperforms other baselines and is closest to the performance achieved by tuning with ground-truth labels. We also report the module integration (without tuning) results as AdMiT-ZeroShot, and the module adaption using plain SGD tuning results as AdMiT-Plain.

Source	Method	GN	SN	IN	DB	FGB	MB	ZB	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	Avg
Single	TENT-Best [30]	74.2	65.3	51.0	50.7	47.3	45.8	42.0	36.9	35.2	24.3	24.6	14.2	11.9	11.9	8.2	36.2
	SAR-Best [38]	71.2	71.7	69.2	64.4	56.0	54.2	59.8	59.3	56.2	52.7	49.4	46.8	48.2	42.0	44.9	56.4
	BECoTTA-Best [33]	69.7	71.4	65.8	68.7	54.3	53.4	55.7	52.4	55.7	47.2	52.4	46.3	49.6	42.4	41.9	55.1
Multi	π -tuning-PL [16]	79.8	76.3	81.1	75.7	66.2	62.5	68.1	61.6	61.9	58.7	63.4	52.9	50.7	53.1	56.0	64.5
	SESoM-PL [12]	75.0	<u>76.0</u>	<u>74.8</u>	69.0	65.6	60.4	60.3	54.5	55.8	54.5	55.3	49.7	50.8	51.4	53.9	60.5
	Model soup [46]	52.1	55.3	49.6	48.0	48.7	49.1	43.2	<u>73.0</u>	<u>76.5</u>	<u>74.1</u>	74.3	56.3	51.2	47.8	58.3	57.2
	CONTRAST [15]	78.2	75.3	74.4	76.1	72.2	70.8	71.7	72.1	73.9	71.4	<u>74.6</u>	71.5	<u>69.5</u>	<u>71.3</u>	65.4	72.5
	AdMiT	79.1	74.4	74.4	75.2	<u>69.2</u>	71.8	<u>73.7</u>	73.1	78.9	<u>75.2</u>	74.9	73.2	70.5	72.8	65.0	73.3

Table 2. **Dynamic adaptation forgetting evaluation on CIFAR-100C.** We take $N = 4$ source modules pretrained on *Snow*, *Frost*, *Fog*, and *Bright* for all the involved methods in the table. The table illustrates the average test accuracy (with all corruption domains of severity level 5) on the 4 source domains during a sequential adaptation across different target domains for various methods. AdMiT selects $M = 3$ modules to adapt to new domains. All the results are the average of 5 runs. We employ these models for adaptation on 15 sequential target domains. **Best** performance is bolded, and second-best performance is underlined.

such as TENT [30], BECoTTA [33], and SAR [38]. These methods serve as benchmarks for adapting individual source models to target distributions, providing insight into how well a single-source approach performs in adapting to unseen distributions. As our problem setting involves adapting pre-trained models to new, dynamic distributions during deployment, it is closely related to the objectives of TTA. Therefore, these widely recognized single-source TTA methods are natural baselines for evaluating AdMiT’s ability to adapt effectively. Following a setup similar to that in [36], we apply each source model independently to specific test distribution data, reporting **Best** and **Worst** results, corresponding to the highest and lowest performance achieved across individual source models.

We also compare against leading multi-source ensemble methods in both static adaptation and dynamic adaptation settings, as these approaches simultaneously leverage multiple sources and provide a baseline for assessing the benefits of multi-source adaptation. SESoM [12] trains an attention-based routing network for adaptive weighting

across source outputs, while π -tuning [16] fine-tunes hyperparameters based on a weighted mix of source modules, and CONTRAST [15] computes optimal weights for combining multiple source model outputs through gradient descent. These multi-source methods allow us to evaluate the effectiveness of combining knowledge from multiple distributions and highlight the computational trade-offs involved. For fair comparison, we implement pseudo-label (PL) entropy minimization for tuning mixture weights in SESoM and π -tuning, denoting these as SESoM-PL and π -tuning-PL, and apply a greedy model soup approach to minimize PL entropy by averaging module mixtures, following [46]. All baseline methods, including those originally based on CNN architectures, have been reproduced on a transformer architecture for consistency in our comparisons. Additional implementation details are provided in the Appendix.

Lastly, we provide an upper-bound baseline, GT-Tuning, which tunes a new PET module using ground-truth labels, offering insight into the best achievable performance with full label access on the target distribution. Together, these base-

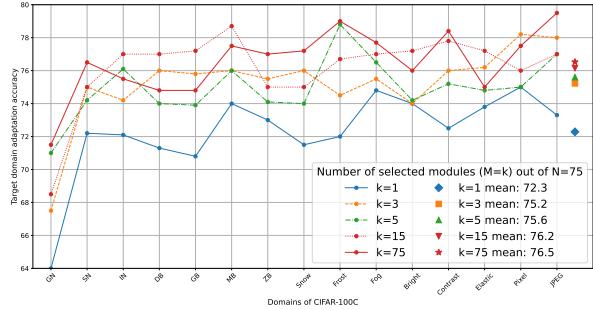


Figure 3. Module Selection on CIFAR-100C. Performance of AdMiT on various domains of CIFAR-100C with different numbers of selected source modules. We pretrain a set of 75 modules (each for a corruption domain and severity level) and select top- k modules based on empirical KME weights. $M = k$ indicates the number of selected source modules. Results show that with limited target data (batch size=128), selecting just a few modules ($k > 1$) maintains performance comparable to that from using all modules ($k = 75$). Mean performance across domains (shown as markers) improves with more modules but with diminishing returns, demonstrating our method’s efficiency even with significantly fewer modules.

lines capture both single-source and multi-source strategies, illustrating AdMiT’s effectiveness in adapting to dynamic target distributions efficiently and without label access.

Module adaptation. We evaluate AdMiT on the Digits-Five [47] dataset for digit classification, with $N = 4$ source modules and all $M = N$ modules used for inference on each target distribution. As shown in Figure 2, GT-Tuning achieves the best performance (serving as an upper bound with labeled data), while AdMiT achieves the second best results (underlined) in most target distributions. The accompanying heatmap illustrates the average weights assigned to each source module, with higher weights corresponding to greater similarity between target and source distributions. We also include results from *AdMiT-Plain*, which uses plain SGD instead of sharpness-aware tuning for adaptation. In some cases (star-marked cells), plain SGD leads to decreased performance, underscoring that sharpness-aware adaptation provides more stable tuning results.

Module integration. We also assess the performance of the integrated module without any tuning to gauge the efficiency of AdMiT in a zero-shot adaptation setting, as shown in Figure 2. *AdMiT-ZeroShot* denotes the accuracy achieved by directly applying the integrated module on target distributions without further adaptation, achieving higher average accuracy than most single-source TTA methods and demonstrating AdMiT’s efficiency and effectiveness in leveraging multi-source knowledge. We further evaluate AdMiT on 15 target distributions of the ImageNet-C dataset, varying the target batch sizes to assess stability. Due to space constraints, we report the average accuracy across target distributions in Table 1. The results show that AdMiT

is less sensitive to batch size variations compared to other TTA methods, providing stable performance across different batch sizes.

Module selection. In previous experiments, all source modules were used regardless of their relevance to the target. To investigate selective module integration, we conduct experiments on the CIFAR-100C dataset with a set of $N = 75$ pre-trained modules. For each target batch, AdMiT selects the top $M = k$ modules based on mixture weights. Results in Fig. 3 indicate that AdMiT effectively identifies and uses only the most relevant modules, achieving strong adaptation performance with fewer modules.

Forgetting of source knowledge. To evaluate the resistance of AdMiT to catastrophic forgetting in dynamic test distributions, we use the CIFAR-100C dataset with four source modules pretrained on *Snow*, *Frost*, *Fog*, and *Bright* distributions. AdMiT selects $M = 3$ modules for adaptation to each new target distribution, maintaining higher accuracy on the original source distributions after adaptation. Table 2 shows that AdMiT outperforms other methods, including multi-source approaches like π -tuning and SESoM, as well as anti-forgetting methods like BECoTTA and SAR. Methods like Model Soup and TENT, which adapt solely to the current target, show a faster rate of forgetting.

Computational efficiency. To demonstrate AdMiT’s computational advantages, we compare the overhead of our module matching approach against the additional costs incurred by hyperparameter tuning and routing network training. The results show that AdMiT achieves efficient source-target matching with lower computational cost and without any need for forward inference or retraining during deployment. Additionally, we evaluate the compatibility of AdMiT with various PET methods to confirm its minimal overhead and adaptability across different configurations. See the Appendix for the results.

5. Conclusion

We present AdMiT, a novel framework for transformers that dynamically integrates multiple source parameter-efficient tuning (PET) modules to address diverse and evolving target distributions. Unlike traditional PET methods that focus on single-source adaptation or require extra computation for multi-source integration, AdMiT selects and integrates relevant modules without the need for hyperparameter tuning, routing networks, or raw data sharing. This makes it well-suited for settings with privacy or resource constraints. AdMiT performs well in both zero-shot and dynamic adaptation scenarios, using a multi-source approach to handle distribution shifts while keeping source knowledge. Its theoretical guarantees ensure reliable adaptation, and its efficiency makes it practical for use in tasks with changing distributions.

Acknowledgments

This work was partially supported by NSF grants CCF-2046816, CCF-2403075, CCF-2008020, NSF CNS grant 210698, and Office of Naval Research grants N000142412289 and N000141812252. Additionally, research was sponsored by the OUSD (R&E)/RT&L and was accomplished under Cooperative Agreement Number W911NF-20-2-0267. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ONR, ARL and OUSD(R&E)/RT&L or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was also partially supported by gifts from Open Philanthropy, Amazon Research, and Google Research.

References

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018. [1](#) [3](#)
- [2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020. [1](#)
- [3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020. [1](#)
- [6] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” *arXiv preprint arXiv:2101.00190*, 2021. [1](#)
- [7] B. Lester, R. Al-Rfou, and N. Constant, “The power of scale for parameter-efficient prompt tuning,” *arXiv preprint arXiv:2104.08691*, 2021. [2](#)
- [8] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, “GPT understands, too,” *AI Open*, 2023.
- [9] T. Vu, B. Lester, N. Constant, R. Al-Rfou, and D. Cer, “Spot: Better frozen model adaptation through soft prompt transfer,” *arXiv preprint arXiv:2110.07904*, 2021. [1](#)
- [10] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-efficient transfer learning for NLP,” in *International Conference on Machine Learning*, pp. 2790–2799, PMLR, 2019. [1](#) [3](#)
- [11] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021. [1](#) [3](#)
- [12] X. Peng, C. Xing, P. K. Choubey, C.-S. Wu, and C. Xiong, “Model ensemble instead of prompt fusion: a sample-specific knowledge transfer method for few-shot prompt tuning,” *arXiv preprint arXiv:2210.12587*, 2022. [1](#) [2](#) [3](#) [5](#) [6](#) [7](#)
- [13] E. L. Buehler and M. J. Buehler, “X-LoRA: Mixture of low-rank adapter experts, a flexible framework for large language models with applications in protein mechanics and molecular design,” *APL Machine Learning*, vol. 2, no. 2, 2024. [1](#)
- [14] X. Wu, S. Huang, and F. Wei, “Mixture of LoRA experts,” in *The Twelfth International Conference on Learning Representations*, 2024. [1](#) [2](#) [3](#)
- [15] S. M. Ahmed, F. F. Niloy, X. Chang, D. S. Raychaudhuri, S. Oymak, and A. K. Roy-Chowdhury, “CONTRAST: Continual multi-source adaptation to dynamic distributions,” in *Advances in neural information processing systems*, 2024. [1](#) [3](#) [4](#) [6](#) [7](#) [5](#)
- [16] C. Wu, T. Wang, Y. Ge, Z. Lu, R. Zhou, Y. Shan, and P. Luo, “pi-tuning: Transferring multimodal foundation models with optimal multi-task interpolation,” in *International Conference on Machine Learning*, pp. 37713–37727, PMLR, 2023. [2](#) [3](#) [4](#) [5](#) [6](#) [7](#)
- [17] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, “A kernel method for the two-sample-problem,” *Advances in neural information processing systems*, vol. 19, 2006. [2](#)
- [18] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012. [2](#) [8](#) [9](#)
- [19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016. [3](#) [4](#)
- [20] C. Sakaridis, D. Dai, and L. Van Gool, “ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10765–10775, 2021. [3](#) [4](#)
- [21] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, “Visual prompt tuning,” in *European Conference on Computer Vision*, pp. 709–727, Springer, 2022. [3](#)
- [22] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, “Learning to prompt for vision-language models,” *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022. [3](#)
- [23] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, “Conditional prompt learning for vision-language models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16816–16825, 2022. [3](#)
- [24] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, “Clip-adapter: Better vision-language models with feature adapters,” *International Journal of Computer Vision*, vol. 132, no. 2, pp. 581–595, 2024. [3](#)

- [25] R. Zhang, R. Fang, W. Zhang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, “Tip-adapter: Training-free clip-adapter for better vision-language modeling,” *arXiv preprint arXiv:2111.03930*, 2021. 3
- [26] J. M. J. Valanarasu, P. Guo, V. VS, and V. M. Patel, “On-the-fly test-time adaptation for medical image segmentation,” *arXiv preprint arXiv:2203.05574*, 2022. 3
- [27] I. Shin, Y.-H. Tsai, B. Zhuang, S. Schulter, B. Liu, S. Garg, I. S. Kweon, and K.-J. Yoon, “MM-TTA: multi-modal test-time adaptation for 3d semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16928–16937, 2022.
- [28] M. Hu, T. Song, Y. Gu, X. Luo, J. Chen, Y. Chen, Y. Zhang, and S. Zhang, “Fully test-time adaptation for image segmentation,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24*, pp. 251–260, Springer, 2021. 3
- [29] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, “Revisiting batch normalization for practical domain adaptation,” *arXiv preprint arXiv:1603.04779*, 2016. 3
- [30] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, “Tent: Fully test-time adaptation by entropy minimization,” *arXiv preprint arXiv:2006.10726*, 2020. 3, 6, 7, 4
- [31] M. J. Mirza, J. Micorek, H. Possegger, and H. Bischof, “The norm must go on: Dynamic unsupervised domain adaptation by normalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14765–14775, 2022. 3
- [32] Q. Wang, O. Fink, L. Van Gool, and D. Dai, “Continual test-time domain adaptation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022. 3
- [33] D. Lee, J. Yoon, and S. J. Hwang, “BECoTTA: Input-dependent online blending of experts for continual test-time adaptation,” *arXiv preprint arXiv:2402.08712*, 2024. 3, 6, 7, 5
- [34] S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan, “Efficient test-time model adaptation without forgetting,” in *International conference on machine learning*, pp. 16888–16905, PMLR, 2022. 3, 6
- [35] L. K. Hansen and P. Salamon, “Neural network ensembles,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990. 3
- [36] S. M. Ahmed, D. S. Raychaudhuri, S. Paul, S. Oymak, and A. K. Roy-Chowdhury, “Unsupervised multi-source domain adaptation without access to source data,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10103–10112, 2021. 3, 4, 5, 7
- [37] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-aware minimization for efficiently improving generalization,” *arXiv preprint arXiv:2010.01412*, 2020. 4, 6, 1
- [38] S. Niu, J. Wu, Y. Zhang, Z. Wen, Y. Chen, P. Zhao, and M. Tan, “Towards stable test-time adaptation in dynamic wild world,” *arXiv preprint arXiv:2302.12400*, 2023. 4, 6, 7, 1, 5
- [39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017. 4
- [40] J. Mercer, “Functions of positive and negativetypeand their connection with theory ofintegral equations,” *Philosophical Trinsoctions of Royal Society*, pp. 4–415, 1909. 4, 6
- [41] H. Daumé III, “From Zero to Reproducing Kernel Hilbert Spaces in twelve pages or less,” *Online: http://pub. hal3. name/daume04rkh. ps*, 2004.
- [42] A. Smola, A. Gretton, L. Song, and B. Schölkopf, “A hilbert space embedding for distributions,” in *International conference on algorithmic learning theory*, pp. 13–31, Springer, 2007.
- [43] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002. 4
- [44] M. Balog, I. Tolstikhin, and B. Schölkopf, “Differentially private database release via kernel mean embeddings,” in *International Conference on Machine Learning*, pp. 414–422, PMLR, 2018. 5, 2
- [45] B. Neyshabur, H. Sedghi, and C. Zhang, “What is being transferred in transfer learning?,” *Advances in neural information processing systems*, vol. 33, pp. 512–523, 2020. 5
- [46] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, et al., “Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time,” in *International conference on machine learning*, pp. 23965–23998, PMLR, 2022. 6, 7, 5
- [47] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, “Moment matching for multi-source domain adaptation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019. 6, 8
- [48] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” *arXiv preprint arXiv:1903.12261*, 2019. 6
- [49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009. 6
- [50] A. Krizhevsky, G. Hinton, et al., “Learning multiple layers of features from tiny images,” *Scientific Research Publishing*, 2009. 6
- [51] A. Baevski and M. Auli, “Adaptive input representations for neural language modeling,” *arXiv preprint arXiv:1809.10853*, 2018. 3
- [52] C. Sakaridis, D. Dai, S. Hecker, and L. Van Gool, “Model adaptation with synthetic and real data for semantic dense foggy scene understanding,” in *Proceedings of the european conference on computer vision (ECCV)*, pp. 687–704, 2018. 4
- [53] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, “Depth-attentional features for single-image rain removal,” in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 8022–8031, 2019. 4
- [54] J. Gu, H. Kwon, D. Wang, W. Ye, M. Li, Y.-H. Chen, L. Lai, V. Chandra, and D. Z. Pan, “Multi-scale high-resolution vision transformer for semantic segmentation,” in *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pp. 12094–12103, 2022. [4](#)

- [55] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020. [4](#)
- [56] C. McDiarmid *et al.*, “On the method of bounded differences,” *Surveys in combinatorics*, vol. 141, no. 1, pp. 148–188, 1989. [8](#)
- [57] J. Wellner *et al.*, *Weak convergence and empirical processes: with applications to statistics*. Springer Science & Business Media, 2013. [8](#)
- [58] D. Lopez-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin, “Towards a learning theory of cause-effect inference,” in *International Conference on Machine Learning*, pp. 1452–1461, PMLR, 2015. [9](#)