Semi-Supervised Noise Adaptation: Transferring Knowledge from Noise Domain

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transfer learning aims to facilitate the learning of a target domain by transferring knowledge from a source domain. The source domain typically contains semantically meaningful samples (*e.g.*, images) to facilitate effective knowledge transfer. However, a recent study observes that the noise domain constructed from simple distributions (*e.g.*, Gaussian distributions) can serve as a surrogate source domain in the semi-supervised setting, where only a small portion of target samples are labeled while most remain unlabeled. Based on this surprising observation, we formulate a novel problem termed *Semi-Supervised Noise Adaptation* (SSNA), which aims to leverage a synthetic noise domain to improve the generalization performance of the target domain. To address this problem, we first establish a generalization bound characterizing the effect of the noise domain on generalization, based on which we propose a Noise Adaptation Framework (NAF). Extensive experiments conducted on five benchmark datasets demonstrate that NAF effectively utilizes the noise domain to tighten the generalization bound of the target domain, thereby achieving improved performance. The codes are available at `https://anonymous.4open.science/r/SSNA`.

1 Introduction

Transfer Learning (TL) [36, 48] aims to transfer knowledge from a label-rich source domain to a related but label-scarce target domain. Most TL approaches have been proposed [36, 11, 23, 48, 2], demonstrating substantial progress in various practical applications [19, 50, 34, 38]. While the source and target domains often exhibit distributional divergence, the source domain typically contains semantically meaningful samples (*e.g.*, images, text, or audio) that provide a crucial foundation for effective knowledge transfer. However, a recent study [49] has made a surprising finding: *Noise sampled from simple distributions (e.g., Gaussian distributions), can also serve as a viable source domain, provided that its discriminability and transferability are preserved*. Although noise is generally viewed as semantically meaningless and even detrimental, empirical evidence has demonstrated that knowledge can be transferred from the source domain to the target domain in the Semi-Supervised Learning (SSL) setting, where most target samples are unlabeled and only a small subset is labeled. This observation is particularly valuable, as concerns related to privacy, confidentiality, and copyright often hinder the acquisition of feasible source domains. However, this study has two key limitations: (i) it lacks a generalization bound analysis explaining why the noise domain improves generalization; and (ii) its experiments omit standard semi-supervised benchmark datasets such as CIFAR-10/100 [25] and ImageNet [12], limiting the generalizability of its findings.

Motivated by those limitations, we formalize a novel problem termed Semi-Supervised Noise Adaptation (SSNA), as illustrated in Figure 1. Under the SSNA setting, we define a *target* domain and a *noise* domain. The target domain comprises a small protion of labeled samples, with most remain-

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

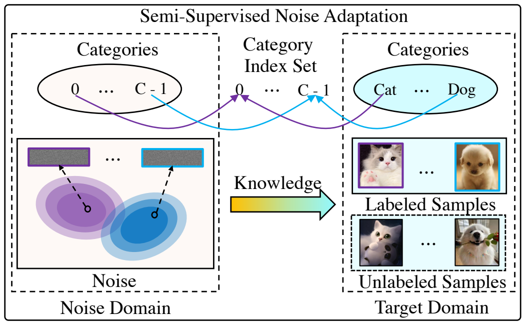


Figure 1: SSNA: The target domain includes a limited number of labeled samples, with most remaining unlabeled, while the noise domain is generated from random distributions. Noise categories, lacking semantic meaning, are mapped one-to-one to target categories (see solid arrows). The goal is to improve the generalization performance of the target domain by utilizing noise.

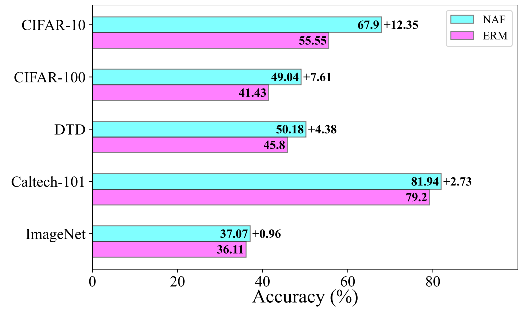


Figure 2: Accuracy (%) of ERM and NAF on five benchmark datasets, *i.e.*, CIFAR-10 [25], CIFAR-100 [25], DTD [8], Caltech-101 [14], and ImageNet [12], using ResNet-18 [21]. NAF consistently outperforms ERM across all the datasets, demonstrating the effectiveness of NAF in transferring knowledge from the noise domain to the target one.

ing unlabeled. In contrast, the noise domain is generated from random distributions and serves as a surrogate source domain. *Since noise inherently lacks semantic meanings, we follow [49] and randomly and uniquely assign the category indices from the target domain to each noise category in a one-to-one manner* (see solid arrow in Figure 1). Accordingly, the learning objectives in both domains are aligned. The objective of SSNA is to enhance the generalization performance of the target domain by leveraging both labeled and unlabeled target samples, as well as noise.

To address this problem, we first establish a generalization bound characterizing the effect of the noise domain on generalization. Based on this theoretical insight, we propose a Noise Adaptation Framework (NAF) that projects target samples and noise into a domain-invariant representation space by minimizing the empirical risks of both domains and reducing their distributional divergence. Optimizing NAF's objective effectively tightens the target domain's generalization bound, thereby improving its generalization performance. Experimental results on five benchmark datasets validate the effectiveness of NAF. As shown in Figure 2, NAF outperforms ERM by up to **12.35%**, **7.61%**, **4.38%**, and **2.73%** on CIFAR-10, CIFAR-100, DTD, and Caltech-101, respectively, with 4 labeled samples per category. Moreover, on the more challenging ImageNet dataset with 1000 categories and 100 labeled samples per category, NAF achieves an improvement of up to **0.96%** over ERM.

The main contributions of this paper are summarized as follows. (1) We introduce the SSNA problem, providing a fresh perspective on the utilization of noise. (2) We provide a generalization bound of SSNA that characterizes the impact of the noise domain on generalization, based on which we propose the NAF. (3) Extensive experiments on five benchmark datasets demonstrate that NAF can effectively tighten the generalization bound of the target domain.

2 Related Work

Our work is closely related to TL [36, 48] and semi-supervised learning (SSL) [43, 20], both of which aim to leverage unlabeled samples to improve the generalization performance of the target domain.

TL enhances generalization by leveraging abundant labeled source samples to guide the learning of unlabeled target samples. [5, 4] introduce the theoretical foundations for TL by establishing a generalization bound for the target domain. Based on this theoretical bound, a key objective in TL is to minimize the distributional discrepancy between the source and target domains. To this end, various distribution alignment methods have been proposed, primarily leveraging Maximum Mean Discrepancy (MMD) [18] and Adversarial Domain Alignment (ADA) [15]. For instance, several studies [32, 30, 29, 50, 7] propose MMD variants to quantify the distributional divergence between the source and target domains. Another line of research [15, 31, 28, 16, 39, 34] explores diverse forms of ADA, which mitigate this divergence via a min-max game between a feature extractor and a domain discriminator. Furthermore, several studies [19, 1, 27, 38] utilize other distributional alignment mechanisms to facilitate cross-domain knowledge transfer. Note that most of the above studies, the source domain consists of semantically meaningful samples (*e.g.*, images, text, or audio). Hence, research in this line has primarily focused on developing state-of-the-art TL approaches through increasingly sophisticated distribution alignment strategies.

SSL utilizes a few labeled target samples to guide the learning of unlabeled target samples. Many methods [46, 40, 51, 6, 45] utilize data augmentation and pseudo-label refinement mechanisms, where the former improves sample diversity and the latter mitigates pseudo-label bias. For instance, UDA [46] strengthens consistency training by replacing simple noise injection with strong data augmentation. FixMatch [40] generates pseudo-labels from weakly augmented samples and enforces consistency with their strongly augmented counterparts. FlexMatch [51] further refines this method by dynamically adjusting category-specific confidence thresholds. To alleviate pseudo-label bias, DST [6] decouples pseudo-label generation and utilization with two independent classifiers while adversarially optimizing the representation extractor. DebiasMatch [45] uses causal inference to adjust decision margins based on pseudo-label imbalance. Another line of research [17, 10, 52] focuses on directly guiding the learning of unlabeled samples. A recent example is LERM [52], which utilizes category-specific label-encodings to guide the learning of unlabeled samples.

Our work is primarily motivated by [49], which reveals that noise drawn from simple distributions (*e.g.*, Gaussian distributions) contains transferable knowledge, as long as its discriminability and transferability are preserved. This may initially appear counter-intuitive, as noise is typically viewed as semantically meaningless and potentially harmful. In practice, however, several studies [3, 26, 22, 44, 41, 33] have explored the potential of noise in addressing diverse machine learning tasks. For example, [3] leverages noise to pre-train a visual representation model using a contrastive loss, resulting in better downstream performance. Another line of research [22, 44] builds on the concept of *positive-incentive noise* introduced by [26], leveraging it to augment original samples or representations, aiming to enhance generalization performance. Moreover, [33, 41] propose utilizing noise to tackle the distribution heterogeneity issue across clients in federated learning. As an example, [41] randomly generates noise as source samples and reduces the distributional divergence between the noise and target samples on each client in a supervised setting.

In summary, unlike the aforementioned studies, our work explores how the noise domain can be leveraged to facilitate the learning of unlabeled target samples in SSL within a TL framework.

3 Problem Formulation

In this section, we formulate the SSNA problem. Let $\mathcal{C} = \{0, \dots, C-1\}$ be the category index set, where C denotes the total number of categories. Let $\mathcal{E}$ and $\mathcal{X}$ denote the noise space (*e.g.*, a d-dimensional feature space) and the sample space (*e.g.*, a pixel-level image space), respectively.

Definition 1. *(Target Domain). The target domain is defined as $\mathcal{D}_t = \mathcal{D}_l \cup \mathcal{D}_u \cup \mathcal{D}_e$, where all samples lie in the sample space $\mathcal{X}$. Specifically, $\mathcal{D}_l = \{(\mathbf{x}_i^l, y_i^l)\}_{i=1}^{n_l}$ consists of labeled target samples, where each sample $\mathbf{x}_i^l$ is associated with a semantic category (e.g., "dog") that is mapped to an integer label $y_i^l \in \mathcal{C}$. $\mathcal{D}_u = \{\mathbf{x}_i^u\}_{i=1}^{n_u}$ and $\mathcal{D}_e = \{\mathbf{x}_i^e\}_{i=1}^{n_e}$ include the unlabeled and test target samples, respectively. Furthermore, the number of labeled target samples is much smaller than that of the unlabeled target samples, i.e., $n_l \ll n_u$.*

Definition 2. *(Noise Domain). The noise domain is defined as $\mathcal{D}_n = \{(\mathbf{n}_i, y_i)\}_{i=1}^{n}$, where each noise $\mathbf{n}_i$ is drawn from a random distribution over $\mathcal{E}$. The corresponding label $y_i \in \mathcal{C}$ serves purely as an integer identifier without any semantic information.*

Definition 3. *(SSNA). Given a target domain $\mathcal{D}_t$, the objective of SSNA is to train a high-quality model $h_{\boldsymbol{\theta}^*}$ using samples from $\mathcal{D}_l$, $\mathcal{D}_u$, and noise from $\mathcal{D}_n$, and then apply $h_{\boldsymbol{\theta}^*}$ to classify the samples in $\mathcal{D}_e$ for evaluation.*

4 Generalization Bound Analysis and Empirical Verification

In this section, we first present a generalization bound analysis for SSNA and then design the NAF to empirically verify that utilizing the noise domain can tighten this bound.

4.1 Generalization Bound Analysis

Before presenting the generalization bound for SSNA, we first address two fundamental questions based on the findings in [49]:

(i) *What knowledge is contained in the noise domain that can benefit the target domain?*

(ii) *Is the semi-supervised setting in the target domain necessary?*

Regarding question (i), although the noise domain is constructed by randomly sampling from a noise space, the noise and target domains share a common category index set (see Figure 3), implying that they are aligned in terms of classification objectives. *Classifying noise into distinct category indices induces a discriminative structure, which encodes discriminative knowledge that can be leveraged in the target domain.* As for question (ii), without labeled target samples to align the category indices between the noise and target domains, a classifier trained solely on the noise domain cannot effectively classify target samples. This is because the noise is randomly generated and does not originate from the same sample space as the target domain, lacking any inherent relationship with the target samples. Consequently, *a few labeled target samples are needed to bridge the two domains by aligning their category indices, enabling the effective transfer of discriminative knowledge from the noise domain to the target one*. (see **Q4** in Section 5.3 for a detailed analysis).

Next, we apply the theoretical framework of semi-supervised TL in [4] to analyze the generalization bound of SSNA. Since the noise does not originate from the same sample space as the target domain, which makes it infeasible to directly measure the distributional discrepancy between them. To address this issue, we project both domains into a domain-shared representation space $\mathcal{Z}$ and derive the generalization bound for the target domain within this space. Specifically, let $\mathcal{F}$ be a hypothesis space over $\mathcal{Z}$ with VC dimension d. We denote by $\widetilde{\mathcal{P}}_t$ and $\widetilde{\mathcal{P}}_n$ the target and noise distributions over $\mathcal{Z}$, respectively. Given a data set $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_n$ of size m, where $\mathcal{D}_l$ consists of βm ($\beta \in [0,1]$) *i.i.d.* labeled samples from $\widetilde{\mathcal{P}}_t$ while $\mathcal{D}_n$ comprises $(1-\beta)m$ *i.i.d.* labeled samples from $\widetilde{\mathcal{P}}_n$. Define $\hat{\epsilon}_\alpha(f) = \alpha\hat{\epsilon}_t(f) + (1-\alpha)\hat{\epsilon}_n(f)$ ($\alpha \in [0,1]$) as the convex combination of the empirical target error $\hat{\epsilon}_t(f)$ and empirical noise error $\hat{\epsilon}_n(f)$, measured on $\mathcal{D}_l$ and $\mathcal{D}_n$, respectively. Based on those notations, we present the generalization bound of SSNA in Theorem 1.

Theorem 1. *(Generalization Bound of SSNA) Let $\hat{f} = \arg\min_{f\in\mathcal{F}} \hat{\epsilon}_\alpha(f)$ be the empirical minimizer of $\hat{\epsilon}_\alpha(f)$ on $\mathcal{D}$, and let $f_t^* = \arg\min_{f\in\mathcal{F}} \epsilon_t(f)$ be the target error minimizer. Then, for any $\delta \in (0,1)$, the expected error of $\hat{f}$ is bounded with probability at least $1-\delta$ by*

$$\epsilon_t(\hat{f}) \leq \epsilon_t(f_t^*) + 2(1-\alpha)\left(\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\widetilde{\mathcal{P}}_n, \widetilde{\mathcal{P}}_t) + \hat{\lambda}\right) + 4\sqrt{\frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{1-\beta}}\sqrt{\frac{2d\log(2(m+1)) + 2\log(\frac{8}{\delta})}{m}},$$

where $d_{\mathcal{H}\Delta\mathcal{H}}(\widetilde{\mathcal{P}}_n, \widetilde{\mathcal{P}}_t)$ is the $\mathcal{H}$-divergence between the noise and target domains, and $\hat{\lambda} := \min_{f\in\mathcal{F}} \hat{\epsilon}_n(f) + \hat{\epsilon}_t(f)$.

Proof sketch. This theorem is built upon Theorem 3 in [4], and the fact that $\lambda := \min_{f\in\mathcal{F}} \epsilon_n(f) + \epsilon_t(f) \leq \hat{\lambda} := \min_{f\in\mathcal{F}} \hat{\epsilon}_n(f) + \hat{\epsilon}_t(f)$.

Based on Theorem 1, the target error $\epsilon_t(\hat{f})$ is primarily upper-bounded by the empirical target error $\hat{\epsilon}_t(\hat{f})$, the empirical noise error $\hat{\epsilon}_n(\hat{f})$, and the distributional discrepancy $d_{\mathcal{H}\Delta\mathcal{H}}(\widetilde{\mathcal{P}}_n, \widetilde{\mathcal{P}}_t)$. This indicates that if a projected noise domain can effectively reduce both terms $\hat{\epsilon}_n(\hat{f})$ and $d_{\mathcal{H}\Delta\mathcal{H}}(\widetilde{\mathcal{P}}_n, \widetilde{\mathcal{P}}_t)$, it will lead to a tighter generalization bound for the target domain. Note that the term $d_{\mathcal{H}\Delta\mathcal{H}}(\widetilde{\mathcal{P}}_n, \widetilde{\mathcal{P}}_t)$ measures the distributional discrepancy between the projected noise and target domains in the representation space $\mathcal{Z}$ (see Figure 3). Therefore, we do not impose constraints on the original distribution of the noise domain, which can be arbitrarily constructed as long as its projection satisfies the required properties. Next, we empirically verify the above theoretical insight.

4.2 Empirical Verification of Theorem 1

To empirically verify Theorem 1, we design the NAF guided by this theorem and report several key empirical results.

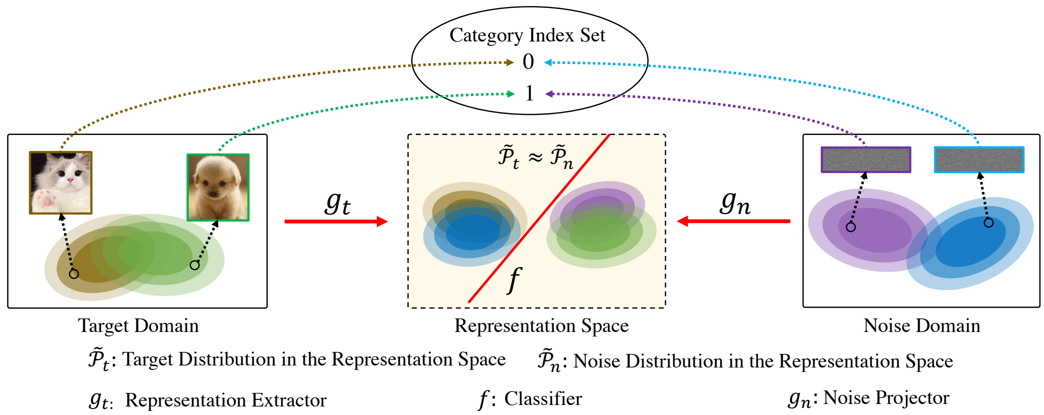


Figure 3: Under the SSNA setting, although the noise domain is generated from a random distribution, it shares a common set of category indices with the target domain. By assigning noise to distinct category indices, a discriminative structure is introduced, which encodes discriminative knowledge that can be transferred to the target domain. Furthermore, aligning the projected distributions of the noise and target domains in the representation space is vital for effective knowledge transfer.

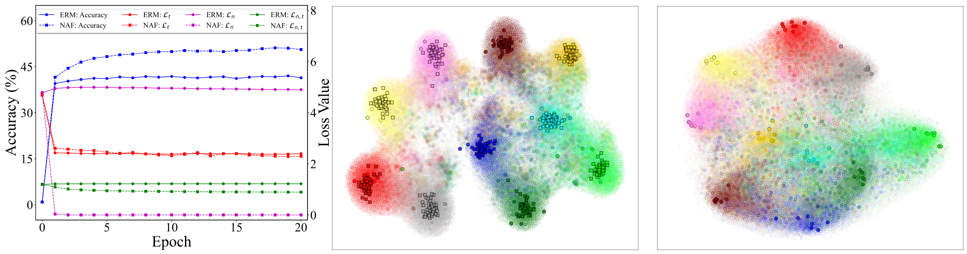
Building on Theorem 1, the generalization bound of the expected target error $\epsilon_t(\hat{f})$ can be minimized by jointly reducing the empirical target error $\hat{\epsilon}_t(\hat{f})$, the empirical noise error $\hat{\epsilon}_n(\hat{f})$, and the distributional discrepancy $d_{\mathcal{H}\Delta\mathcal{H}}(\widetilde{\mathcal{P}}_n, \widetilde{\mathcal{P}}_t)$ in $\mathcal{Z}$. Accordingly, we design the NAF to project target samples and noise into $\mathcal{Z}$ by minimizing three components: (i) $\mathcal{L}_t$: the empirical risk of labeled target samples, corresponding to $\hat{\epsilon}_t(\hat{f})$; (ii) $\mathcal{L}_n$: the empirical risk of noise, corresponding to $\hat{\epsilon}_n(\hat{f})$; and (iii) $\mathcal{L}_{n,t}$: the distributional discrepancy between projected domains, whose minimization implicitly reduces $d_{\mathcal{H}\Delta\mathcal{H}}(\widetilde{\mathcal{P}}_n, \widetilde{\mathcal{P}}_t)$. Thus, the optimization objective of the NAF is formulated as follows:

$$\min_{g_t, g_n, f} \mathcal{L}_t(\mathcal{D}_l; g_t, f) + \alpha \mathcal{L}_n(\mathcal{D}_n; g_n, f) + \beta \mathcal{L}_{n,t}(\mathcal{D}_l, \mathcal{D}_u, \mathcal{D}_n; g_t, g_n, f), \tag{1}$$

where $g_t(\cdot)$ is a representation extractor projecting target samples from $\mathcal{X}$ to $\mathcal{Z}$, $g_n(\cdot)$ is a noise projector mapping noise from $\mathcal{N}$ to $\mathcal{Z}$, $f(\cdot)$ is a classifier (see Figure 3), and α, β are two positive trade-off parameters to control the importance of $\mathcal{L}_n$ and $\mathcal{L}_{n,t}$, respectively. By optimizing the problem (1), the generalization bound of the target domain can be effectively tightened, thereby improving the generalization performance.

In the implementation, the *cross-entropy loss* is used to instantiate $\mathcal{L}_t$ and $\mathcal{L}_n$. To implement $\mathcal{L}_{n,t}$, we design a *Negative Domain Similarity* (NDS) mechanism, which quantifies the distributional discrepancy between the projected target and noise domains by calculating the cosine similarity between their global means and the sum of cosine similarities between their corresponding category-wise means, followed by negating the total similarity score to measure the degree of divergence. Also, we use the classifier $f(\cdot)$ to assign pseudo-labels to unlabeled target samples and iteratively update them to estimate category-wise means. *Involving those unlabeled samples in distribution alignment is crucial for guiding their learning and enhancing the generalization performance of the target domain* (see the subsequent representation visualization analysis). Alternative mechanisms for modeling distributional divergence are also analyzed in **Q6** of Section 5.3.

Next, we present empirical results showing that NAF tightens the target domain's generalization bound versus the ERM baseline using only $\mathcal{L}_t$. Figure 4a plots the loss trajectories of $\mathcal{L}_t$, $\mathcal{L}_n$, and $\mathcal{L}_{n,t}$, along with the test accuracy curves for both ERM and NAF on CIFAR-100 with ResNet-18, respectively. Several insightful observations can be drawn. (1) Both methods demonstrate notable reductions in $\mathcal{L}_t$, as it is explicitly minimized in their respective objective functions. (2) The values of $\mathcal{L}_n$ and $\mathcal{L}_{n,t}$ in ERM are consistently higher than those in NAF, which is reasonable since ERM does not explicitly minimize them. (3) When $\mathcal{L}_t$ is jointly minimized with $\mathcal{L}_n$ and $\mathcal{L}_{n,t}$ in NAF, the resulting accuracy curve shows a significant improvement over that of ERM. This indicates that simultaneously minimizing those losses within a unified framework can effectively tighten the generalization bound of the target domain, thereby improving its generalization performance. This observation is in line with the theoretical result in Theorem 1.



(a) NAF vs. ERM (b) NAF (c) ERM

Figure 4: Empirical Results of NAF. (a) Training loss and accuracy curves for ERM and NAF on CIFAR-100 with ResNet-18. $\mathcal{L}_t$ denotes labeled target risk, $\mathcal{L}_n$ is the noise risk, and $\mathcal{L}_{n,t}$ measures domain discrepancy. (b) Representations learned by ERM on CIFAR-10 with ResNet-18, where ■' indicates noise; •' and '○' represent labeled and unlabeled target samples, respectively. (c) Representations learned by NAF on CIFAR-10 with ResNet-18, with the same symbol scheme as in (b). Colors correspond to different categories.

Furthermore, we employ the t-SNE technique [42] to visualize the representations learned by ERM and NAF on CIFAR-10 with ResNet-18. As shown in Figure 4b, the noise representations of each category align closely with the target representations belonging to the same category, while noise representations from different categories remain well-separated. Compared to ERM, as plotted in Figure 4c, NAF effectively improves the discriminability of the target domain. The improvement mainly stems from jointly minimizing $\mathcal{L}_n$ and $\mathcal{L}_{n,t}$. *The former ensures the discriminability of the noise domain, while the latter* ***transfers this discriminability to guide the learning of numerous unlabeled target samples****, thereby improving the target domain's generalization performance.*

5 Experiments

In this section, we evaluate the proposed NAF on five benchmark datasets.

5.1 Experimental Setup

Datasets We use five benchmark datasets: CIFAR-10 [25], CIFAR-100 [25], DTD [8], Caltech-101 [14], and ImageNet [12]. CIFAR-10 and CIFAR-100 consist of 60,000 images across 10 and 100 categories, respectively. DTD includes 5,640 textural images from 47 categories, Caltech-101 contains images from 101 object categories plus a background, and ImageNet comprises nearly one million images spanning 1,000 categories. For the first four datasets, we randomly select 4 labeled samples from each category in the training set, with the remaining samples serving as unlabeled samples. As for the ImageNet dataset, we randomly choose 100 labeled samples per category in the training set, with the rest treated as unlabeled samples.

Noise Domain Construction. We randomly construct a noise domain in a 1024-dimensional space. Specifically, we first sample C category means from a standard Gaussian distribution, where C corresponds to the number of categories in the target domain. For each category, we assign an identity covariance matrix. Based on each class mean and its corresponding covariance matrix, we then sample 50 noise from the associated Gaussian distribution to form the noise domain.

Implementation Details. We implement the proposed NAF using the TLlib library [23] and apply weak and strong augmentation techniques [9] in the target domain. Also, we directly treat both weakly and strongly augmented target samples uniformly as target samples, omitting refined processing strategies like those in FlexMatch [51]. This allows us to focus on the impact of the noise domain with advanced mechanisms left for future work. In NAF, it is necessary to calculate the category mean for each category. To address the mini-batch issue, we follow [47] and employ an exponential moving average to update the category means as follows: $\mathbf{m}_n^c = \lambda \cdot \mathbf{m}_o^c + (1-\lambda) \cdot \mathbf{m}_b^c$, where $\mathbf{m}_o^c$ and $\mathbf{m}_n^c$ denote the previous and updated c-th category means, respectively, and $\mathbf{m}_b^c$ is the c-th category mean calculated from the current mini-batch. The hyperparameter λ is detailed in Table 6 in Appendix B. In addition, we implement the representation extractor g_t using ResNet [21] backbones pre-trained on ImageNet for all datasets, except ImageNet itself, where the backbone is trained from scratch. The noise projector g_n is implemented as a non-linear layer with ReLU activation [35], and the classifier f is a single linear layer. Furthermore, we utilize mini-batch SGD with a momentum of 0.9 as the optimizer, setting batch sizes to 32 for CIFAR-10 and CIFAR-100, DTD, and Caltech-101, and 128 for ImageNet. Additional settings are detailed in Appendix B. All experiments are conducted on NVIDIA V100 series GPUs.

Evaluation Metric. We evaluate performance using the classification accuracy in $\mathcal{D}_e$. For a fair comparison, we report the average accuracy of the last-epoch after 20 epochs for all baselines, calculated over three random experiments and its standard error [13].

5.2 Main Experiments

In this section, we conduct extensive main experiments to investigate the following research questions **Q1-Q3**.

Q1: Is NAF effectively improving the performance of the target domain? Table 1 presents the results on CIFAR-10, CIFAR-100, DTD, and Caltech-101 using ResNet-18 and ResNet-50. As shown, NAF consistently outperforms ERM across all datasets. Specifically, NAF achieves significant performance improvements of 12.35% and 15.15% over ERM using ResNet-18 and ResNet-50 on CIFAR-10, respectively. Those results demonstrate that NAF can effectively enhance the generalization performance of the target domain by leveraging the noise domain. This is because NAF tightens the generalization bound of the target domain by enhancing the discriminability of both domains and aligning their distributions within the domain-shared representation space. As a result, the discriminative knowledge of the source domain is effectively transferred to the target domain.

Table 1: Accuracy (%) comparison on CIFAR-10 and CIFAR-100, DTD, and Caltech-101 using ResNet-18 and ResNet-50, respectively. Here, **Δ** indicates the performance gain introduced by NAF.

Datasets	CIFAR-10		CIFAR-100		DTD		Caltech-101	
ResNet-18	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ERM	55.55±2.01	92.85±0.37	41.43±0.40	71.40±0.19	45.80±0.39	74.26±0.26	79.20±0.70	93.29±0.18
NAF	67.90±2.28	96.38±0.31	49.04±0.69	80.56±0.48	50.18±0.94	77.98±0.46	81.94±0.62	95.01±0.21
Δ	**+12.35**	**+3.52**	**+7.61**	**+9.16**	**+4.38**	**+3.72**	**+2.73**	**+1.72**
ResNet-50	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ERM	58.83±1.83	94.25±0.58	46.71±0.70	76.53±0.63	49.56±0.50	76.65±0.29	81.99±0.41	94.70±0.15
NAF	73.98±3.21	97.01±0.64	52.82±0.52	82.16±0.37	53.97±0.72	79.68±0.22	84.41±0.62	96.14±0.02
Δ	**+15.15**	**+2.76**	**+6.11**	**+5.63**	**+4.41**	**+3.03**	**+2.42**	**+1.44**

Q2: Is NAF still effective as a plug-in when combined with existing SSL methods? To investigate this question, we conduct experiments using six state-of-the-art (SOTA) SSL methods: UDA [46], FixMatch [40], FlexMatch [51], DebiasMatch [45], DST [6], and LERM [52]. NAF can be seamlessly integrated as a plugin into those SOTA SSL methods by incorporating $\mathcal{L}_n$ and $\mathcal{L}_{n,t}$ into their objective functions. Table 2 reports the results at the 5th, 10th, 15th, and 20th epochs on CIFAR-10 and CIFAR-100 using ResNet-18. We observe that incorporating NAF leads to consistent performance gains across all SSL methods. Specifically, NAF improves accuracy by 20.84% and 9.91% over UDA and FixMatch, respectively, at the 20th epoch on CIFAR-10. Those results indicate that NAF effectively enhances the generalization performance of SOTA methods by transferring knowledge from the noise domain. Additional results on DTD and Caltech-101 are offered in Appendix C.

Q3: Is NAF still effective on ImageNet? We evaluate NAF on ImageNet with 100 labeled samples per category using ResNet-18 to assess its performance on a more complex dataset. NAF achieves an accuracy of 37.07%, outperforming ERM (36.11%) by 0.96%. This result further highlights NAF's effectiveness, even on large-scale datasets with 1,000 categories, demonstrating its potential for addressing complex real-world challenges.

5.3 Analysis

In this section, we perform a series of analysis experiments to explore the research questions **Q4-Q9**.

Q4: How does the impact of NAF change as the number of labeled target samples varies? Table 3 reports the results on CIFAR-10 using ResNet-18 with different numbers of labeled samples per category. We can make several insightful observations. (1) When the number of labeled target

Table 2: Accuracy (%) comparison on CIFAR-10 and CIFAR-100 using ResNet-18. Here, $\boldsymbol{\Delta}$ indicates the performance gain introduced by NAF.

Datasets	CIFAR-10					CIFAR-100				
Epoch	5	10	15	20	Average	5	10	15	20	Average
UDA [46]	51.67±1.58	55.37±1.69	56.03±1.73	56.11±1.53	54.80±1.52	38.30±0.83	42.99±0.26	45.93±0.32	47.41±0.74	43.66±0.44
UDA + NAF	73.55±3.07	76.16±3.06	76.52±3.34	76.94±3.23	75.79±3.13	40.37±0.54	45.44±0.94	47.82±0.89	48.80±0.81	45.61±0.78
Δ	**+21.88**	**+20.78**	**+20.50**	**+20.84**	**+21.00**	**+2.07**	**+2.45**	**+1.90**	**+1.39**	**+1.95**
FixMatch [40]	66.41±1.68	68.41±1.91	69.01±1.97	69.40±2.22	68.31±1.94	39.38±0.28	40.78±0.33	41.98±0.16	42.45±0.21	41.15±0.13
FixMatch + NAF	75.51±3.14	77.89±3.50	79.00±3.42	79.31±3.62	77.93±3.41	40.97±1.55	43.28±0.88	44.06±1.14	44.93±1.47	43.31±1.22
Δ	**+9.09**	**+9.48**	**+9.99**	**+9.91**	**+9.62**	**+1.59**	**+2.50**	**+2.08**	**+2.48**	**+2.17**
FlexMatch [51]	73.61±1.06	79.85±0.54	83.46±0.65	84.53±0.57	80.36±0.60	45.41±1.49	50.28±1.99	51.91±1.15	54.30±1.20	50.48±1.45
FlexMatch + NAF	79.22±1.18	82.72±0.87	84.32±0.59	84.90±0.56	82.79±0.79	48.10±1.15	52.91±1.04	54.97±0.63	55.73±0.59	52.93±0.85
Δ	**+5.61**	**+2.87**	**+0.87**	**+0.38**	**+2.43**	**+2.68**	**+2.62**	**+3.06**	**+1.43**	**+2.45**
DebiasMatch [45]	68.71±1.47	77.68±0.58	79.86±1.80	82.04±1.85	77.07±1.39	46.71±0.92	51.97±0.26	54.73±0.39	56.30±0.32	52.43±0.42
DebiasMatch + NAF	76.12±1.17	80.89±0.90	82.54±0.68	83.05±0.70	80.65±0.85	49.57±0.53	54.02±0.29	56.36±0.41	57.45±0.50	54.35±0.39
Δ	**+7.40**	**+3.21**	**+2.68**	**+1.01**	**+3.58**	**+2.87**	**+2.05**	**+1.63**	**+1.14**	**+1.92**
DST [6]	78.40±1.92	82.84±1.46	84.48±1.16	85.47±1.22	82.80±1.43	45.40±0.34	49.74±0.39	51.68±0.51	53.17±0.73	50.00±0.48
DST + NAF	80.70±1.22	83.46±1.42	84.87±1.58	85.53±1.63	83.64±1.46	48.73±0.41	52.28±0.77	54.10±0.76	54.93±0.89	52.51±0.70
Δ	**+2.29**	**+0.62**	**+0.40**	**+0.06**	**+0.84**	**+3.33**	**+2.54**	**+2.42**	**+1.77**	**+2.51**
LERM [52]	60.03±1.88	62.42±1.99	63.81±2.09	64.77±2.07	62.76±2.00	48.10±0.47	50.13±0.46	50.83±0.34	51.66±0.27	50.18±0.38
LERM + NAF	66.01±1.71	67.34±1.67	67.83±1.74	68.00±1.61	67.30±1.68	49.42±0.19	51.06±0.34	51.65±0.40	51.97±0.53	51.03±0.34
Δ	**+5.98**	**+4.92**	**+4.02**	**+3.23**	**+4.54**	**+1.32**	**+0.93**	**+0.82**	**+0.31**	**+0.84**

samples is zero, both ERM and NAF perform poorly. For ERM, the absence of labeled target samples hinders the effective learning of unlabeled samples, resulting in significant performance degradation. In NAF, since the noise is not from the same sample space as the target domain and lacks an inherent relationship with the target samples, its discriminative knowledge cannot be effectively transferred. (2) When the number of labeled target samples is non-zero, NAF consistently outperforms ERM across all scenarios. Those results indicate that NAF effectively leverages both the labeled target samples and noise to further enhance the generalization performance of the target domain.

Table 3: Accuracy (%) comparison on CIFAR-100 using ResNet-18 with different numbers of labeled target samples per category.

# Labeled target samples per category	0	4	8	12	16	20
ERM	0.97	42.24	54.11	58.27	61.64	63.85
NAF	1.34	49.98	59.51	62.21	64.23	66.45

Q5. How do $\mathcal{L}_n$ and $\mathcal{L}_{n,t}$ influence the performance of NAF? To further investigate the effectiveness of $\mathcal{L}_n$ and $\mathcal{L}_{n,t}$, we examine two NAF variants: (1) NAF (w/o $\mathcal{L}_n$), which ablates $\mathcal{L}_n$; and (2) NAF (w/o $\mathcal{L}_{n,t}$), which removes $\mathcal{L}_{n,t}$. Additionally, ERM can be seen as a NAF variant that eliminates both losses. The results on CIFAR-100 using ResNet-18 are shown in Table 4. We observe that NAF significantly outperforms all variants, indicating that both losses are beneficial. Moreover, NAF (w/o $\mathcal{L}_n$) outperforms NAF (w/o $\mathcal{L}_{n,t}$), suggesting that reducing distributional divergence between domains is more crucial.

Table 4: Accuracy (%) of NAF variants on CIFAR-100 using ResNet-18.

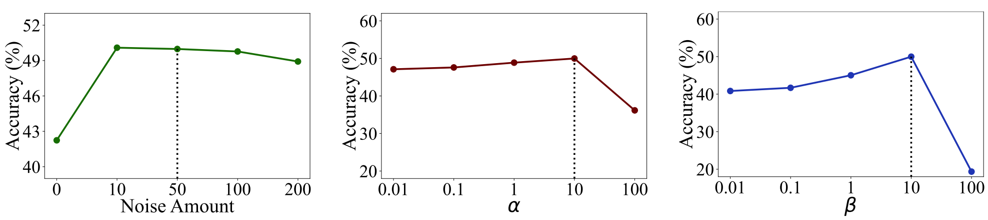
ERM	NAF (w/o $\mathcal{L}_n$)	NAF (w/o $\mathcal{L}_{n,t}$)	NAF
42.24	47.33	40.64	49.98

Q6: How do other distribution alignment methods influence NAF? In the implementation of NAF, we employed NDS to quantify the distributional divergence between domains. Next, we introduce several alternative mechanisms for comparison. (1) Negative Sample Similarity (NSS): It takes the negative sum of the cosine similarities between all individual sample pairs belonging to the same category across domains. (2) Negative Contrastive Domain Similarity (NCDS): It utilizes a contrastive loss [37] to calculate a negative similarity for same-category mean pairs and a positive similarity for different-category mean pairs. (3) Negative Contrastive Sample Similarity (NCSS): It utilizes a contrastive loss [37] to compute a negative similarity for same-category sample pairs and a positive similarity for different-category sample pairs. (4) Euclidean Domain Distance (EDD): It calculates the Euclidean distance between the global means of two domains, along with the sum of Euclidean distances between their corresponding category-wise means. Table 5 presents the results on CIFAR-100 using ResNet-18. As observed, NAF (NDS) achieves the best performance, suggesting that NDS is an effective metric for capturing distributional divergence between domains. Conversely, NAF (EDD) yields the worst performance, indicating that Euclidean distance is less suitable than cosine similarity for measuring distributional divergence in the domain-shared representation space. Therefore, we empirically adopt NDS in the implementation of NAF.

Table 5: Accuracy (%) of NAF with various distributional alignment mechanisms on CIFAR-100 using ResNet-18.

NAF (NDS)	NAF (NSS)	NAF (NCDS)	NAF (NCSS)	NAF (EDD)
49.98	48.65	47.20	44.27	20.03

Q7: How does the amount of noise impact NAF? We vary the amount of noise per category (*i.e.*, 0, 10, 50, 100, 200) to evaluate its impact on NAF. The results on CIFAR-100 using ResNet-18 are shown in Figure 5a. As can be observed, when the amount of noise is zero, NAF degenerates to ERM, resulting in poor performance. As the noise increases, the performance remains relatively stable, showing no noticeable improvement. When the amount of noise per category reaches 200, the performance slightly declines. This suggests that excessive noise may not be beneficial as it could increase learning difficulty.



(a) (b) (c)

Figure 5: Accuracy (%) comparison on CIFAR-100 using ResNet-18 with varying (a) amounts of noise, (b) values of α, and (c) values of β.

Q8: How do the hyper-parameters α and β influence NAF? We analyze the parameter sensitivity of α and β on CIFAR-100 using ResNet-18. Figures 5b and 5c present the performance of NAF under varying values of α and β, respectively. The results show that NAF performs well and remains relatively stable when α and β are close to the default value of 10. However, when either parameter increases to 100, a significant performance drop is observed, suggesting that overemphasizing the learning of the noise domain while neglecting the target domain can be detrimental.

Q9: Is there another method to learn the noise domain within NAF in the representation space? In all the above experiments, we utilize a noise projector g_n to learn an optimal noise domain distribution that aligns with the target domain distribution in the domain-shared representation space. As an alternative approach, we also explore learning the mean μ and standard deviation σ, and apply the reparameterization trick [24] to transform samples from a standard normal distribution into a Gaussian distribution $\mathcal{N}(\mu, \sigma^2\mathbf{I})$ in the representation space. We evaluate this method on CIFAR-10 using ResNet-18, achieving an accuracy of 70.60%, which is comparable to the performance of NAF of 71.83%, and exceeds ERM by 12.45%. Those results indicate that modeling a parametric noise distribution via reparameterization is also a feasible and effective strategy, which may inspire further promising research.

6 Conclusion

In this paper, we formulate the SSNA problem, which leverages a synthetic noise domain to facilitate the learning task in the target domain. To address this problem, we first derive a generalization bound for the target domain that offers a theoretical understanding of how incorporating a noise domain can influence generalization performance. Building on this bound, we propose the NAF, which jointly minimizes the empirical risks on both the noise and target domains while reducing their distributional divergence within a domain-shared representation space. Extensive experiments on five benchmark datasets demonstrate that NAF effectively tightens the generalization bound of the target domain, resulting in improved performance. Our work explores the potential of noise domains as a complementary source for improving the generalization performance of the target domain. In future work, we intend to extend SSNA to more diverse real-world scenarios.

References

[1] Shuanghao Bai, Min Zhang, Wanqi Zhou, Siteng Huang, Zhirong Luan, Donglin Wang, and Badong Chen. Prompt-based distribution alignment for unsupervised domain adaptation. In *AAAI*, volume 38, pages 729–737, 2024.

[2] Runxue Bao, Yiming Sun, Yuhe Gao, Jindong Wang, Qiang Yang, Zhi-Hong Mao, and Ye Ye. A recent survey of heterogeneous transfer learning. *arXiv preprint arXiv:2310.08459*, 2023.

[3] Manel Baradad Jurjo, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. In *NeurIPS*, volume 34, pages 2556–2569, 2021.

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

[5] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NeurIPS*, volume 19, 2006.

[6] Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debiased self-training for semi-supervised learning. In *NeurIPS*, 2022.

[7] Zhiming Cheng, Shuai Wang, Defu Yang, Jie Qi, Mang Xiao, and Chenggang Yan. Deep joint semantic adaptation network for multi-source unsupervised domain adaptation. *Pattern Recognition*, 151:110409, 2024.

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014.

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, pages 702–703, 2020.

[10] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*, pages 3941–3950, 2020.

[11] Oscar Day and Taghi M. Khoshgoftaar. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4(1):29, 2017.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[13] Brian Everitt. *The Cambridge Dictionary of Statistics*. Cambridge University Press, Cambridge, UK, 2002.

[14] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, pages 178–178, 2004.

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(59):1–35, 2016.

[16] Zhiqiang Gao, Shufei Zhang, Kaizhu Huang, Qiufeng Wang, and Chaoliang Zhong. Gradient distribution alignment certificates better adversarial domain adaptation. In *ICCV*, pages 8937–8946, October 2021.

[17] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, volume 17, 2004.

[18] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem, 2006.

[19] Xiang Gu, Yucheng Yang, Wei Zeng, Jian Sun, and Zongben Xu. Keypoint-guided optimal transport with applications in heterogeneous domain adaptation. volume 35, pages 14972–14985, 2022.

[20] Qian Gui, Hong Zhou, Na Guo, and Baoning Niu. A survey of class-imbalanced semi-supervised learning. *Machine Learning*, 113(8):5057–5086, 2024.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[22] Sida Huang, Hongyuan Zhang, and Xuelong Li. Enhance vision-language alignment with noise. In *AAAI*, volume 39, pages 17449–17457, 2025.

[23] Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. Transferability in deep learning: A survey, 2022.

[24] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

[26] Xuelong Li. Positive-incentive noise. *TNNLS*, 2022.

[27] Meihan Liu, Zeyu Fang, Zhen Zhang, Ming Gu, Sheng Zhou, Xin Wang, and Jiajun Bu. Rethinking propagation for unsupervised graph domain adaptation. In *AAAI*, volume 38, pages 13963–13971, 2024.

[28] Xiaofeng Liu, Zhenhua Guo, Site Li, Fangxu Xing, Jane You, C.-C. Jay Kuo, Georges El Fakhri, and Jonghye Woo. Adversarial unsupervised domain adaptation with conditional and label shift: Infer, align and iterate. In *ICCV*, pages 10367–10376, October 2021.

[29] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Transferable representation learning with deep adaptation networks. *TPAMI*, 41(12):3071–3085, 2019.

[30] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[31] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. volume 31, 2018.

[32] Mingsheng Long, Jianmin Wang, Guiguang Ding, Sinno Jialin Pan, and Philip S Yu. Adaptation regularization: A general framework for transfer learning. *TKDE*, 26(5):1076–1089, 2013.

[33] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. In *NeurIPS*, pages 5972–5984, 2021.

[34] Lakmal Meegahapola, Hamza Hassoune, and Daniel Gatica-Perez. M3bat: Unsupervised domain adaptation for multimodal mobile sensing with multi-branch adversarial training. *IMWUT*, 8(2), 2024.

[35] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[36] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[38] Chuan-Xian Ren, Yiming Zhai, You-Wei Luo, and Hong Yan. Towards unsupervised domain adaptation via domain-transformer. *IJCV*, 132(12):6163–6183, 2024.

[39] Lianghe Shi and Weiwei Liu. Adversarial self-training improves robustness and generalization for gradual domain adaptation. *NeurIPS*, 36:37321–37333, 2023.

[40] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, volume 33, pages 596–608, 2020.

[41] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *ICML*, volume 162, pages 21111–21132, 17–23 Jul 2022.

[42] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008.

[43] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[44] Bocheng Wang, Chusheng Zeng, Mulin Chen, and Xuelong Li. Towards learnable anchor for deep multi-view clustering. In *AAAI*, volume 39, pages 21044–21052, 2025.

[45] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *CVPR*, pages 14647–14657, 2022.

[46] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, volume 33, pages 6256–6268, 2020.

[47] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, pages 5423–5432, 2018.

[48] Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer learning*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[49] Yuan Yao, Xiaopu Zhang, Yu Zhang, Jian Jin, and Qiang Yang. Noise may contain transferable knowledge: Understanding semi-supervised heterogeneous domain adaptation from an empirical perspective. *arXiv preprint arXiv:2502.13573*, 2025.

[50] Yuan Yao, Yu Zhang, Xutao Li, and Yunming Ye. Heterogeneous domain adaptation via soft transfer network. In *ACM MM*, page 1578–1586, 2019.

[51] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *NeurIPS*, volume 34, pages 18408–18419, 2021.

[52] Yulong Zhang, Yuan Yao, Shuhao Chen, Pengrong Jin, Yu Zhang, Jian Jin, and Jiangang Lu. Rethinking guidance information to utilize unlabeled samples: a label encoding perspective. In *ICML*, 2024.

A Limitations and Broader Impacts

Limitations. While randomly sampled noise has shown promising potential in SSNA, our exploration to date has focused primarily on classification tasks. Due to differences in the objectives of generative tasks such as text generation, directly applying the noise domain to such tasks remains non-trivial. Consequently, investigating the role of noise in generative modeling is a promising direction for future research.

Broader Impacts. This work demonstrates the broad potential of noise as an alternative source domain to facilitate learning tasks in label-scarce target domains. This insight is particularly valuable in real-world scenarios where acquiring labeled source samples is often infeasible due to privacy regulations, confidentiality constraints, or copyright restrictions. Furthermore, It may offer new perspectives on the underlying principles of transfer learning. We expect this work to inspire more meaningful and in-depth research.

B Detailed Parameter Configuration

Table 6 outlines the detailed parameter configurations utilized in this paper. For the main experiments conducted in Section 5.2, we report the average classification accuracy over three independent runs with random seeds 0, 1, and 2 for all datasets, except for ImageNet, where we use seed 0 exclusively due to its substantial computational demands. As for the analysis experiments performed in Section 5.3, we fix the random seed to 0 to strike a balance between methodological rigor and computational runtime, ensuring reproducibility and consistency of results.

Table 6: Detailed parameter configuration utilized in this paper.

Method	Dataset	Backbone	α	β	λ	learning rate	iterations
NAF	CIFAR-10 / DTD	ResNet-50 / ResNet-18	1	1	0.7	0.03	10000
	CIFAR-100	ResNet-50 / ResNet-18	10	10		0.01	
	Caltech-101	ResNet-50 / ResNet-18	1	10		0.003	
	ImageNet-1K	ResNet-18	1	10		0.01	80000
LERM + NAF	CIFAR-10	ResNet-18	1	1	0.99	0.03	10000
Others + NA	CIFAR-10 / CIFAR-100		10	10	0.99 / 0.7	0.03 / 0.01	
	DTD		1	5	0.7	0.03	
	Caltech-101		1	10	0.7	0.003	

C Additional Experimental Results

We provide additional results for SOTA + NAF on DTD and Caltech-101 using ResNet-18. As shown in Table 7, SOTA + NAF consistently outperforms the standalone SOTA methods across most scenarios, further demonstrating the effectiveness of NAF in leveraging the noise domain to enhance the performance of the target domain.

Table 7: Accuracy (%) comparison on DTD and Caltech-101 using ResNet-18. Here, $\mathbf{\Delta}$ indicates the performance gain introduced by NAF.

Datasets	DTD					Caltech-101				
Epoch	5	10	15	20	Average	5	10	15	20	Average
UDA [46]	46.28 ± 0.20	46.81 ± 0.10	46.90 ± 0.76	47.32 ± 0.47	46.83 ± 0.35	79.20 ± 0.81	79.61 ± 0.44	80.00 ± 0.41	80.28 ± 0.44	79.77 ± 0.51
UDA + NAF	46.88 ± 1.76	47.89 ± 1.72	49.10 ± 1.67	49.22 ± 1.51	48.27 ± 1.66	80.98 ± 0.74	81.40 ± 0.33	81.21 ± 0.36	81.43 ± 0.46	81.25 ± 0.45
Δ	**+0.60**	**+1.08**	**+2.20**	**+1.90**	**+1.45**	**+1.78**	**+1.79**	**+1.21**	**+1.14**	**+1.48**
FixMatch [40]	46.51 ± 0.49	47.78 ± 0.61	48.09 ± 1.05	48.23 ± 0.88	47.65 ± 0.75	80.13 ± 0.43	80.27 ± 0.19	80.28 ± 0.34	79.99 ± 0.27	80.17 ± 0.27
FixMatch + NAF	48.85 ± 0.94	49.57 ± 0.98	50.12 ± 0.72	49.86 ± 1.07	49.60 ± 0.92	80.96 ± 0.39	80.96 ± 0.27	80.42 ± 0.17	80.42 ± 0.04	80.69 ± 0.17
Δ	**+2.34**	**+1.79**	**+2.04**	**+1.63**	**+1.95**	**+0.82**	**+0.70**	**+0.15**	**+0.44**	**+0.53**
FlexMatch [51]	50.66 ± 0.36	51.29 ± 0.79	50.94 ± 0.63	50.69 ± 0.75	50.90 ± 0.63	82.74 ± 0.53	83.83 ± 0.65	83.61 ± 0.49	83.70 ± 0.25	83.47 ± 0.47
FlexMatch + NAF	50.51 ± 0.88	50.87 ± 0.86	51.03 ± 0.83	51.35 ± 0.82	50.94 ± 0.84	83.22 ± 0.60	84.08 ± 0.51	83.74 ± 0.54	83.77 ± 0.37	83.70 ± 0.50
Δ	**-0.14**	**-0.43**	**+0.09**	**+0.66**	**+0.04**	**+0.48**	**+0.26**	**+0.13**	**+0.08**	**+0.23**
DebiasMatch [45]	45.67 ± 0.60	45.99 ± 0.66	45.46 ± 0.61	46.42 ± 0.64	45.89 ± 0.63	80.87 ± 0.17	81.09 ± 0.28	81.29 ± 0.21	81.60 ± 0.21	81.21 ± 0.07
DebiasMatch + NAF	49.01 ± 1.07	49.79 ± 0.81	50.02 ± 0.95	50.09 ± 0.91	49.73 ± 0.93	82.46 ± 0.55	82.62 ± 0.46	82.77 ± 0.51	82.60 ± 0.17	82.61 ± 0.38
Δ	**+3.33**	**+3.79**	**+4.56**	**+3.67**	**+3.84**	**+1.58**	**+1.53**	**+1.48**	**+1.00**	**+1.40**
DST [6]	49.84 ± 0.63	51.68 ± 1.17	52.27 ± 0.72	51.93 ± 0.99	51.43 ± 0.87	80.75 ± 0.64	81.85 ± 0.46	82.19 ± 0.42	82.16 ± 0.42	81.74 ± 0.45
DST + NAF	51.08 ± 0.77	52.00 ± 0.92	52.54 ± 0.64	52.55 ± 0.84	52.04 ± 0.79	81.70 ± 0.56	82.72 ± 0.49	82.85 ± 0.39	82.87 ± 0.33	82.53 ± 0.43
Δ	**+1.24**	**+0.32**	**+0.27**	**+0.62**	**+0.61**	**+0.95**	**+0.87**	**+0.66**	**+0.71**	**+0.80**
LERM [52]	47.20 ± 0.52	47.50 ± 0.34	48.03 ± 0.28	48.42 ± 0.65	47.79 ± 0.41	82.36 ± 0.28	83.06 ± 0.18	82.98 ± 0.13	83.13 ± 0.13	82.88 ± 0.17
LERM + NAF	48.85 ± 1.06	48.83 ± 0.81	48.87 ± 0.94	48.92 ± 1.04	48.87 ± 0.96	83.14 ± 0.48	83.59 ± 0.63	83.23 ± 0.39	83.06 ± 0.31	83.26 ± 0.45
Δ	**+1.65**	**+1.33**	**+0.83**	**+0.50**	**+1.08**	**+0.78**	**+0.53**	**+0.26**	**-0.07**	**+0.38**

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: We summarize our main contributions in Section 1.

   Guidelines:

   - The answer NA 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 NA 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: We discuss the limitations of this work in Appendix A.

   Guidelines:

   - The answer NA 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: We analyze the generalization bound of the target domain in Section 4, along with the corresponding proof.

Guidelines:

- The answer NA 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: We present the experimental setup in Section 5.1, with additional details available in Appendix B.

Guidelines:

- The answer NA 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 conduct experiments on five publicly available benchmark datasets: CIFAR-10 [25], CIFAR-100 [25], DTD [8], Caltech-101 [14], and ImageNet [12]. The codes are available at `https://anonymous.4open.science/R/SSNA`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We present the experimental setup in Section 5.1, with additional details available in Appendix B.

Guidelines:

- The answer NA 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: The results in Table 1, Table 2, and Table 7 (see Appendix C) are reported as the mean and standard error over three random trials.

Guidelines:

- The answer NA 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: All experiments are conducted on NVIDIA V100 series GPUs, and the computational resources utilized in this work are described in Section 5.1.

   Guidelines:

   - The answer NA 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: This work adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: We discuss the broader impacts of this work in Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No such risks arise from this work.

    Guidelines:

    - The answer NA 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: All open-source resources used in this work are properly cited in Section 5.1.

    Guidelines:

    - The answer NA 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: [Yes]

    Justification: Our codes and a detailed README file have been released at `https://anonymous.4open.science/R/SSNA`.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No crowdsourcing or human subject research is involved in this work.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No crowdsourcing or human subject research is involved in this work.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This work uses the LLM solely for language-related tasks such as writing, editing, and formatting.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.