RESIDUAL FEATURE INTEGRATION IS SUFFICIENT TO PREVENT NEGATIVE TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Transfer learning has become a central paradigm in modern machine learning, yet it suffers from the long-standing problem of negative transfer, where leveraging source representations can harm rather than help performance on the target task. Although empirical remedies have been proposed, there remains little theoretical understanding of how to reliably avoid negative transfer. In this article, we investigate a simple yet remarkably effective strategy: augmenting frozen, pretrained source-side features with a trainable target-side encoder that adapts target features to capture residual signals overlooked by models pretrained on the source data. We show this residual feature integration strategy is sufficient to provably prevent negative transfer, by establishing rigorous theoretical guarantees that it never performs worse than training from scratch on the target data, and that the convergence rate can transition seamlessly from nonparametric to near-parametric when source representations are informative. To our knowledge, this is the first theoretical work that ensures protection against negative transfer. We carry out extensive numerical experiments across image, text and tabular benchmarks, and empirically verify that the method consistently safeguards performance under distribution shift, label noise, semantic perturbation, and class imbalance. Our study thus advances the theory of safe transfer learning, and provides a principled approach that is simple, robust, architecture-agnostic, and broadly applicable.

1 INTRODUCTION

Transfer learning provides a fundamental paradigm in modern machine learning, where knowledge acquired from one task (source domain) is leveraged to enhance performance on another related task (target domain). It encompasses a wide range of applications, from adapting models across different sources or domains, to distilling knowledge from large, pretrained models into smaller, task-specific models. Yet, a critical and persistent challenge is negative transfer: the phenomenon where transferring knowledge degrades performance compared to simply training on the target data from scratch. This issue, which arises from mismatches between source and target distributions, has been documented across numerous scenarios [26; 5; 23; 16; 40; 38; 32]. It is especially concerning in high-stakes applications such as healthcare, where transferring from broad datasets like ImageNet to medical imaging can be detrimental [30; 5]. Despite its prevalence, there remains little theoretical understanding of how to reliably avoid negative transfer.

In this article, we identify and validate a simple yet remarkably effective strategy that provably prevents negative transfer, i.e., augmenting frozen, pretrained source-side features with a trainable target-side encoder that adapts target features to capture residual signals overlooked by models pretrained on the source data. We call this strategy Residual Feature Integration (REFINE). Its implementation is straightforward: after obtaining the transferred representation $f_{\text{rep}}(x)$ from the source domain, instead of relying solely on $f_{\text{rep}}(x)$, we further introduce a residual connection with a trainable feature encoder $h(x)$ that is learned from the target domain. We then combine $f_{\text{rep}}(x)$ and $h(x)$, and fit a *shallow* neural network on the concatenated representation $(f_{\text{rep}}(x), h(x))$. Intuitively, while $f_{\text{rep}}(x)$ captures transferable features, it may omit target-specific signals that are critical for accurate prediction in the target domain. The residual connection via $h(x)$ compensates for this omission, ensuring that key information in the target domain is preserved. Furthermore, because $f_{\text{rep}}(x)$ already encodes a substantial portion of the predictive signal, learning from the joint representations $(f_{\text{rep}}(x), h(x))$ can potentially be achieved with a much simpler class of functions than

learning from x or $h(x)$ alone. We demonstrate, both theoretically and empirically, that this strategy is *sufficient to prevent negative transfer* across a broad range of settings.

Our contributions are threefold. First, we identify the residual connection, a widely adopted structural component originally devised to address optimization challenges in deep neural networks [11; 17], as a powerful mechanism for provably avoiding negative transfer. This strategy in turn offers a lightweight, robust, architecture-agnostic, and broadly applicable enhancement to transfer learning pipelines. Second, we formally justify this simple yet remarkably effective approach through a rigorous theoretical analysis, which is the main contribution of this article. Specifically, we show that augmenting any frozen f_{rep} with a trainable $h(x)$ guarantees that the resulting predictor achieves a convergence rate of prediction risk that is never worse than that obtained by training from scratch on the target data alone. In other words, REFINE is inherently robust against negative transfer in the worst-case scenario. Moreover, our prediction risk bound seamlessly transitions from a nonparametric convergence rate to a near-parametric rate when source representations are informative. Finally, we conduct extensive experiments on benchmark datasets spanning image, text, and tabular domains, and compare REFINE with multiple alternative solutions. We empirically verify that our method consistently mitigates negative transfer, especially under significant representational mismatch or task divergence.

2 RELATED WORK

Transfer learning. Linear probing [21], and adapter-based feature extraction [14] are two of the most widely used transfer learning approaches. Both methods operate by extracting penultimate-layer features from a pretrained model in the source domain, followed by fine-tuning the final layer using data in the target domain. The main difference between the two is that linear probing employs a linear layer, while the adapter method uses a shallow neural network. Both are computationally efficient, but both are vulnerable to negative transfer. Knowledge distillation is another widely used transfer learning technique, where a large pretrained foundation model (the teacher) transfers knowledge to a simpler model (the student) that is typically fine-tuned in the target domain with substantially reduced complexity [12]. However, distillation remains vulnerable to negative transfer, especially when the teacher is poorly aligned with the target domain or when the transferred knowledge is too complex for the student to absorb effectively [7]. Our approach is applicable not only to knowledge transfer in foundation models, but also to general transfer learning settings.

Negative transfer mitigation. To mitigate negative transfer, various empirical remedies have been proposed, most of which focus on developing metrics that estimate similarity between source and target domains [8; 24; 39; 1]. Yet in practice, such similarity measures are often difficult to quantify, and sometimes require specialized loss functions or architectures, which limits their applicability [13]. [22] proposed SAFEW, which constructs an ensemble of source-domain models using a min–max framework. While theoretically sound, this method is computationally intensive and relies on the assumption that the optimal predictor can be expressed as a convex combination of source classifiers. [37] introduced DANN-GATE, a state-of-the-art solution that reduces negative transfer by combining adversarial training with a gating mechanism to filter out misleading source samples. While practically effective, this method requires direct access to source data and is primarily empirical, lacking theoretical guarantees. In contrast, our method does not require access to original training data in the source domain and comes with rigorous theoretical guarantees.

Residual learning, stacking, and parameter-efficient fine-tuning. Several methods are conceptually related to REFINE, although they do not explicitly target negative transfer in transfer learning. Residual learning, a core idea in architectures such as ResNet [11] and algorithms like gradient boosting [17], was originally developed to ease optimization challenges or improve prediction. Its potential for addressing negative transfer, however, remains unexplored. Stacking is an ensemble technique that combines predictions from multiple base models through a meta-learner trained on validation outputs. This approach is generally more robust than simple model averaging [3], but it assumes that all external models are reliable [6; 9], and requires aligned output spaces, which restricts its applicability across different types of tasks. Parameter-efficient fine-tuning methods, such as LoRA [15], insert lightweight, trainable modules into pretrained models to enable domain adaptation without modifying the original weights. Such an approach is effective and significantly reduces parameter costs, but struggles when source representations misalign with the target domain.

Besides, it requires access to pretrained model weights and computational graphs, limiting their flexibility, particularly in the multi-source transfer setting.

3 Problem Formulation and Algorithm

Transfer learning aims to leverage knowledge from a source task to improve performance on a related target task. A common practice is to use a representation function f_{rep} learned from a large source dataset D^{s} under a source distribution $\mathbb{P}^{\text{s}}$ as an extracted feature for the target task. However, if f_{rep} does not align well with the target distribution $\mathbb{P}^{\text{t}}$, naively reusing it can lead to negative transfer, resulting in degraded performance compared to using the target data alone.

We formalize the Residual Feature Integration (ReFine) approach. The objective is to construct a method such that, when f_{rep} aligns well with the target distribution, effectively leverage transferred knowledge and outperform models trained from scratch on target data only, and when f_{rep} misaligns with the target distribution, safeguard against negative transfer and outperform models that rely solely on $f_{\text{rep}}(x)$. We focus on the supervised learning task. Let $D^{\text{t}} = \{(x_i, y_i)\}_{i=1}^{n} \sim \mathbb{P}^{\text{t}}$ denote the labeled dataset from the target task. Assume access to a frozen extracted feature $f_{\text{rep}} : \mathcal{X} \to \mathbb{R}^p$ trained on an external source data D^{s}. Define a class $\mathcal{H}$ of trainable feature encoders $h : \mathcal{X} \to \mathbb{R}^q$ and a class $\mathcal{W}$ of trainable adapters $w : \mathbb{R}^{p+q} \to \mathbb{R}^k$ on top of $(f_{\text{rep}}(x), h(x))$. Let w_{ft} be the trained adapter on top of the baseline model, and let g_{sc} be the model trained from scratch on x. We seek to learn both the encoder h and the adapter w, such that

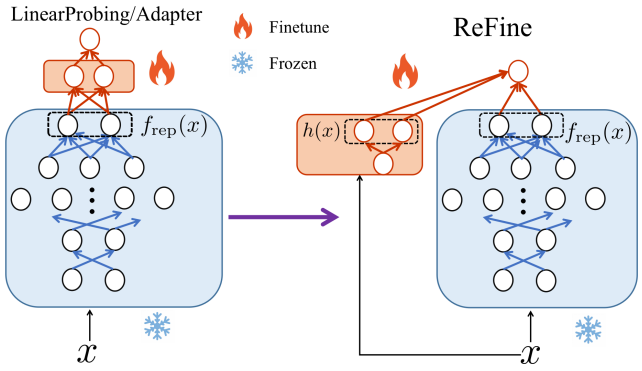


Figure 1: A schematic overview of ReFine.

$$\mathcal{R}_{\mathbb{P}^{\text{t}}}(w \circ (f_{\text{rep}}, h)) \leq \min\{\mathcal{R}_{\mathbb{P}^{\text{t}}}(w_{\text{ft}} \circ f_{\text{rep}}),\ \mathcal{R}_{\mathbb{P}^{\text{t}}}(g_{\text{sc}})\} \tag{1}$$

where $\mathcal{R}_{\mathbb{P}^{\text{t}}}$ denotes the expected loss under the distribution $\mathbb{P}^{\text{t}}$.

Algorithm 1 outlines the ReFine approach. It extracts $f_{\text{rep}}(x)$ from the penultimate layer of a frozen pretrained model, and combines it with the residual connection $h(x)$. The concatenated features $(f_{\text{rep}}(x), h(x))$ are passed to a linear classifier for prediction, where only $h(x)$ and the adapter w are updated, whereas the pretrained model and $f_{\text{rep}}(x)$ remain unchanged. This design allows ReFine to efficiently complement transferred knowledge with adapted features from the target data, and thus recover potentially lost information during the forward pass in the frozen source model. Figure 1 gives a schematic overview of ReFine.

Algorithm 1 The residual feature integration (ReFine) method.

1: **Input:** Training data $\mathcal{D}_{\text{train}} = (X_i, Y_i)_i$, test data $\mathcal{D}_{\text{test}}$, pretrained model f, loss function ℓ.
2: **Output:** Prediction of the label $\hat{y}(x_0)$ for $x_0 \in \mathcal{D}_{\text{test}}$.
3: **Training Phase:**
4: (a) Extract $f_{\text{rep}}(x)$ from the penultimate-layer of a frozen pretrained model f.
5: (b) Construct the concatenated features $C_h(x) := (f_{\text{rep}}(x), h(x))$.
6: (c) Let $(\hat{w}, \hat{h})$ be the minimizer of $\sum_i \ell(w(C_h(X_i)), Y_i)$ while freezing f_{rep}.
7: **Prediction Phase:**
8: (a) Compute $C_h(x_0)$ with the frozen f.
9: (b) Obtain the final prediction $\hat{y}(x_0)$ based on $\hat{w}(C_{\hat{h}}(x_0))$.

4 THEORETICAL ANALYSIS

We provide a theoretical analysis to rigorously prove that REFINE is robust to negative transfer. The intuition and core insight is that the residual connection provides a natural transition: if the external representation f_{rep} is uninformative, the residual network h can still learn the target function from the raw input, recovering the performance of training from scratch. Conversely, if f_{rep} is informative, h only needs to learn the simpler residual function, reducing the effective complexity of the problem and accelerating the learning. This allows REFINE to adaptively interpolate between training from scratch and near-parametric transfer, depending on the quality of external representation.

We formalize this intuition within the framework of nonparametric regression. We consider the model with a *trainable* residual feature encoder h:

$$g(x) = uh(x) + v^\top f_{\text{rep}}(x),$$

where $h(x)$ is a (clipped) ReLU network over raw input, combined with a linear probe on the feature $f_{\text{rep}}(x)$. We establish the risk bound demonstrating that, for any capacity of h, REFINE's learning rate is never worse than the standard nonparametric rate. Furthermore, when the capacity of h is tuned to the difficulty of the residual task, the rate adapts and improves, showcasing its ability to effectively leverage useful prior information from $f_{\text{rep}}(x)$.

Formal Setup. We consider the nonparametric regression setup adopted in the statistical analysis of deep neural networks [33; 31; 19]. Specifically, we observe n i.i.d. pairs $(X_i, Y_i)_{i\in[n]} \sim \mathbb{P}^{\text{t}}$ with support on $[0,1]^d \times \mathbb{R}$ following the model

$$Y_i = f^*(X_i) + \epsilon_i,$$

where $f^* : [0,1]^d \to [-1,1]$ is the ground-truth regression function, and $(\epsilon_i)_{i\in[n]}$ are i.i.d. Gaussian with variance $\sigma^2 = O(1)$, independent of $(X_i)_{i\in[n]}$. We assume the marginal distribution $\mathbb{P}^{\text{t}}_X$ of $\mathbb{P}^{\text{t}}$ on X admits a positive continuous density on $[0,1]^d$ upper bounded by an absolute constant. Under this set-up, the expected loss for a given function g is $\mathcal{R}_{\mathbb{P}^{\text{t}}}(g) = \mathbb{E}_{(X,Y)\sim\mathbb{P}^{\text{t}}}[(g(X) - Y)^2]$.

To facilitate the theoretical analysis, following the standard setup of nonparametric regression, we consider f^* to be Hölder smooth. Specifically, for a non-integer $\beta > 0$, the Hölder norm for f^* that are $\lfloor\beta\rfloor$-times differentiable on $[0,1]^d$ is

$$\|f\|_{\mathcal{C}^\beta} := \max\Big\{\max_{a\in\mathbb{N}^d:\|a\|_1\le\lfloor\beta\rfloor}\sup_{x\in[0,1]^d}|\partial^a f(x)|,\ \max_{a\in\mathbb{N}^d:\|a\|_1=\lfloor\beta\rfloor}\sup_{x\neq x'}\frac{|\partial^a f(x) - \partial^a f(x')|}{\|x - x'\|^{\beta-\lfloor\beta\rfloor}}\Big\}.$$

The unit ball is $\mathcal{C}^\beta_{\text{u}} := \{f : [0,1]^d \to \mathbb{R} : f \text{ is } \lfloor\beta\rfloor\text{-times differentiable and } \|f\|_{\mathcal{C}^\beta} \le 1\}$.

Further, we assume the residual connection $h : \mathbb{R}^d \to \mathbb{R}$ is realized by a ReLU network with width at most W, depth at most L, and weight magnitude at most B:

$$h(x) = A_{L'}x^{(L'-1)} + b_{L'}, \quad x^{(\ell)} = \sigma(A_\ell x^{(\ell-1)} + b_\ell)\ (\ell \in [L'-1]), \quad x^{(0)} = x, \tag{2}$$

for some $L' \le L$, where $d_0 = d$, $d_{L'} = 1$, and $d_\ell \le W$. Here $\sigma(z) = \max\{0, z\}$ is applied element-wise, $A_\ell \in [-B, B]^{d_\ell \times d_{\ell-1}}$, and $b_\ell \in [-B, B]^{d_\ell}$. The class is $\mathcal{H}_d(W, L, B)$, and we use its clipped counterpart $\bar{\mathcal{H}}_d(W, L, B) := \{x \mapsto \min\{1, \max\{-1, h(x)\}\} : h \in \mathcal{H}_d(W, L, B)\}$.

Empirical risk minimization for REFINE. We consider squared loss $\ell(y, y') = (y - y')^2$. Let $f_{\text{rep}} : [0,1]^d \to \mathcal{B}_p(1)$ be an external representation with $\mathcal{B}_p(R) = \{u \in \mathbb{R}^p : \|u\|_2 \le R\}$. Define the REFINE class

$$\mathcal{G}_{d,p}(W, L, B; f_{\text{rep}}) = \Big\{g : [0,1]^d \to \mathbb{R} \ \Big|\ g(x) = v^\top f_{\text{rep}}(x) + uh(x),\ |u| \le 1,\ \|v\| \le 1,\ h \in \bar{\mathcal{H}}_d(W, L, B)\Big\}.$$

We train $\hat{g}$ via empirical risk minimization,

$$\hat{g} = \operatorname*{arg\,min}_{g\in\mathcal{G}_{d,p}(W,L,B;f_{\text{rep}})} \frac{1}{n}\sum_{i\in[n]} \ell(g(X_i), Y_i). \tag{3}$$

The effectiveness of REFINE depends on the quality of f_{rep}. We quantify this by defining the best possible linear probe and the corresponding residual. Specifically, for any $f_{\text{rep}} : [0,1]^d \to \mathcal{B}_p(1)$, the best linear probe is defined as

$$v^* = \arg\min_{v \in \mathbb{R}^p} \mathbb{E}\big[\{v^\top f_{\text{rep}}(X_1) - f^*(X_1)\}^2\big].$$

The difficulty of learning the residual is then captured by its Hölder norm, which we denote as $\rho^* := \|v^{*\top} f_{\text{rep}} - f^*\|_{\mathcal{C}^\beta}$. A small ρ^* indicates that $f$rep is highly informative for the target task.

Our main theory bounds the excess risk of the REFINE estimator. It shows that the learning rate depends on both the standard nonparametric complexity and the quality of external representation ρ^*.

Theorem 4.1 (Generalization Error of REFINE)**.** *Suppose $\|v^*\| \le 1$ and the residual $f^* - v^{*\top} f_{rep}$ lies in the unit Hölder ball $\mathcal{C}_u^\beta$ for a non-integer $\beta > 0$. Let $\rho > 0$ be a tuning parameter, which serves as a proxy for the residual norm, and choose the network parameters for h as*

$$L = c_1, \qquad W = c_2 \max\{n^{d/(2\beta+d)} \rho^{2d/(2\beta+d)}, 1\}, \qquad B = \max\{n\rho^2, 1\}^{c_3}, \tag{4}$$

where $c_1, c_2, c_3 > 0$ depend on β and d. Let $\hat{g}$ be the empirical risk minimizer in (3) with the parameter specified as in (4). Then there exists $C > 0$, which depends on β, d, such that

$$\mathbb{E}[\mathcal{R}_{\mathbb{P}'}(\hat{g}) - \mathcal{R}_{\mathbb{P}'}(f^*)] \le C\Big\{\big(\rho^{2d/(2\beta+d)} \log n + \rho^{*2} \rho^{-4\beta/(2\beta+d)}\big) n^{-2\beta/(2\beta+d)} + \frac{p \log n}{n}\Big\}. \tag{5}$$

The bound in (5) splits into a parametric term $p \log n / n$ for learning v^* on top of f_{rep}, and a nonparametric term with the standard minimax rate $n^{-2\beta/(2\beta+d)}$ for learning the residual modulated by the tuning parameter ρ and the residual difficulty ρ^*. The tuning radius ρ controls the effective capacity of h via W and B in (4). That is, a larger ρ increases the approximation power, achieving a smaller bias, but worsens the estimation, resulting in a larger variance factor $\rho^{2d/(2\beta+d)}$. On the other hand, a smaller ρ regularizes h, which is preferable when the residual is genuinely small.

We further discuss two direct implications of Theorem 4.1.

Corollary 4.2 (Fixed ρ)**.** *For any fixed choice of $\rho > 0$, the bound in (5) implies that*

$$\mathbb{E}[\mathcal{R}_{\mathbb{P}'}(\hat{g}) - \mathcal{R}_{\mathbb{P}'}(f^*)] = \tilde{O}\Big(n^{-2\beta/(2\beta+d)} + \frac{p}{n}\Big).$$

This corollary indicates that, by introducing an additional residual connection h, REFINE never has a worse rate than $n^{-2\beta/(2\beta+d)}$ when p is bounded, which is the standard minimax-optimal rate when training from scratch on $(X_i, Y_i)_{i\in[n]}$ for β-Hölder f^* [36].

Corollary 4.3 (Tuned ρ)**.** *Balancing (5) by choosing $\rho = \rho^*$ yields*

$$\mathbb{E}[\mathcal{R}_{\mathbb{P}'}(\hat{g}) - \mathcal{R}_{\mathbb{P}'}(f^*)] = \tilde{O}\Big(\rho^{*2d/(2\beta+d)} n^{-2\beta/(2\beta+d)} + \frac{p}{n}\Big). \tag{6}$$

This corollary indicates that, when f_{rep} is well aligned with the target, i.e., a small ρ^*, choosing $\rho = \rho^*$ effectively regularizes the residual network h via the parameter choice in (4), which shrinks the nonparametric term so that the bound is dominated by the near-parametric p/n term. Conversely, when f_{rep} is misaligned, i.e., a large ρ^*, the nonparametric component dominates and the rate reverts to the classical β-Hölder minimax rate $n^{-2\beta/(2\beta+d)}$.

Putting together, these two corollaries explain why REFINE avoids negative transfer under proper regularization: it leverages external source representations whenever they are informative, while retaining the fallback guarantee of nonparametric learning when they are not. This adaptivity ensures that external knowledge is never harmful and is properly utilized when it is beneficial.

Proof sketch of Theorem 4.1 For any v, decompose

$$f^*(x) = \underbrace{f^*(x) - v^\top f_{\text{rep}}(x)}_{\text{residual}} + \underbrace{v^\top f_{\text{rep}}(x)}_{\text{linear in } f_{\text{rep}}(x)}.$$

The first term is fit by h and the second by a linear probe on f_{rep}. Approximation results for ReLU networks over $\mathcal{C}^\beta$ functions give the residual term at rate $n^{-2\beta/(2\beta+d)}$ with a capacity-dependent

multiplier governed by ρ. A standard linear estimation yields the p/n term for v. Choosing (W, L, B) as in (4) implements this bias-variance trade-off. The full proof is deferred to Appendix A.

We remark that our theoretical results are derived under the squared-loss objective, following a long line of work that analyzes classification problems through regression surrogates [10; 41]. This approach aligns with common practice in the machine learning theory community, where regression surrogates are employed to derive insights for classification algorithms.

5 Numerical Experiments

5.1 Experiment setup

We demonstrate that ReFine consistently mitigates negative transfer through extensive numerical experiments across image, text, and tabular modalities, using benchmark datasets including CIFAR-10, CIFAR-100 [20], STL [4], Clipart, Sketch [28], USPS, MNIST, Books, Kitchen, DVD, and Electronics [2]. We evaluate performance using classification accuracy, area under ROC (AUC), F1 score, and minimum class accuracy.

We also compare ReFine with a number of alternative solutions. In particular, NoTrans serves as a no-transfer baseline, reusing pretrained features without any adaptation. LinearProbe [21] trains only a linear classifier on top of frozen features, offering a lightweight baseline. Adapter [14] inserts a small trainable module into pretrained models, enabling efficient adaptation with limited parameters. Distillation [12] transfers knowledge from a frozen teacher to a student model through a combination of hard labels and soft predictions. LoRA [15] applies low-rank adaptations to weight matrices, achieving parameter-efficient fine-tuning. DANN-Gate [37] combines adversarial training with gating to encourage domain-invariant representations.

We consider a variety of experiment settings. In Section 5.2, we evaluate ReFine on datasets that exhibit natural distribution shift. In Section 5.3, we deliberately construct more challenging scenarios to stress-test various transfer learning methods. In Section 5.4, we investigate multi-source transfer. Furthermore, in Appendix C.2, we consider a tabular data setting with four tabular benchmark datasets.

In our implementations, we train all models using stochastic gradient descent with a learning rate 0.01 and momentum 0.9, with pretraining for 60 epochs and fine-tuning for 30 epochs. We consider both CNNs and transformers architectures for pretrain model f_{rep} and the encoders h. We also carry out an ablation study in Appendix C.3 regarding the complexity of the encoder h, showing that ReFine remains effective across different choices of the model parameters for h.

We provide more details about the experiment setup and implementations in Appendix D.

5.2 Single-source transfer with natural distribution shift

In the first experiment setting, we evaluate ReFine on datasets that exhibit natural distribution shift. To provide a comprehensive assessment, we consider transfer tasks spanning both image and language, thereby covering cross-domain as well as cross-modality adaptation. For image, we include CIFAR-10, CIFAR-100, and STL-10, which offer complementary object recognition tasks with varying class granularity and image resolution. We further incorporate artistic domains, specifically, Clipart and Sketch, to capture substantial stylistic diversity, along with digit recognition benchmarks, USPS and MNIST, which provide structured and well-curated handwritten digits. For text, we adopt the datasets, Books, DVD, Electronics, and Kitchen, which span heterogeneous product categories and exhibit rich linguistic variations. We process the image datasets using convolutional neural networks (CNNs), and process the text datasets using transformers. This design allows us to assess transfer across distribution and domain shifts, and also under cross-modality and cross-models. Collectively, these datasets constitute a broad and rigorous benchmark for evaluating transfer learning methods.

We use the notation $A \to B$ to denote transfer learning from source domain A to target domain B. Our evaluation covers diverse scenarios. Specifically, CIFAR100→10 and CIFAR10→100 test transfers across datasets with overlapping but non-identical class spaces and label granularity; CIFAR10→STL reflects natural distribution shift due to resolution and dataset construc-

Dataset	Method	Accuracy	AUC	F1	Min CAcc
CIFAR100→10	NoTrans	$\mathbf{56.5820 \pm 0.3659}$	$\mathbf{0.9005 \pm 0.0012}$	$\mathbf{0.5634 \pm 0.0046}$	$\mathbf{37.2000 \pm 3.4117}$
	LinearProb	38.9260 ± 0.5463	0.8284 ± 0.0017	0.3815 ± 0.0051	16.9400 ± 3.7441
	Adapter	38.2320 ± 0.3111	0.8247 ± 0.0016	0.3754 ± 0.0071	16.4600 ± 5.4544
	LoRA	43.1360 ± 0.3239	0.8603 ± 0.0003	0.4237 ± 0.0046	20.1400 ± 4.1020
	DANN-Gate	43.2220 ± 0.1295	0.8605 ± 0.0005	0.4214 ± 0.0040	17.4800 ± 4.7755
	REFINE	54.4000 ± 0.3336	0.8942 ± 0.0026	0.5406 ± 0.0051	33.6200 ± 2.8273
CIFAR10→100	NoTrans	18.3200 ± 0.5254	0.8140 ± 0.0050	0.1774 ± 0.0052	1.0000 ± 0.8944
	LinearProbe	7.0140 ± 0.3347	0.7489 ± 0.0011	0.0496 ± 0.0034	0.0000 ± 0.0000
	Adapter	6.5640 ± 0.2875	0.7499 ± 0.0008	0.0459 ± 0.0026	0.0000 ± 0.0000
	LoRA	6.8240 ± 0.1037	0.7558 ± 0.0010	0.0463 ± 0.0015	0.0000 ± 0.0000
	DANN-Gate	5.1980 ± 0.3924	0.7341 ± 0.0055	0.0285 ± 0.0033	0.0000 ± 0.0000
	REFINE	$\mathbf{18.5880 \pm 0.5494}$	$\mathbf{0.8276 \pm 0.0053}$	$\mathbf{0.1787 \pm 0.0057}$	$\mathbf{1.4000 \pm 0.8000}$
CIFAR10→STL	NoTrans	48.6925 ± 0.6338	0.8683 ± 0.0032	0.4831 ± 0.0089	$\mathbf{26.8000 \pm 4.9006}$
	LinearProbe	50.2725 ± 0.3016	0.8795 ± 0.0015	0.4955 ± 0.0067	18.9250 ± 6.1546
	Adapter	49.2900 ± 0.7344	0.8773 ± 0.0008	0.4865 ± 0.0096	15.6750 ± 6.6340
	LoRA	50.7550 ± 0.3793	0.8813 ± 0.0016	0.4930 ± 0.0040	5.6750 ± 2.6933
	DANN-Gate	47.7050 ± 0.6586	0.8659 ± 0.0013	0.4712 ± 0.0104	13.9250 ± 5.3424
	REFINE	$\mathbf{53.4175 \pm 0.3628}$	$\mathbf{0.8944 \pm 0.0013}$	$\mathbf{0.5301 \pm 0.0053}$	25.9750 ± 3.5693
Clipart→Sketch	NoTrans	18.8804 ± 1.3709	0.7170 ± 0.0117	0.1828 ± 0.0119	0.0000 ± 0.0000
	LinearProbe	18.3430 ± 0.8649	0.7290 ± 0.0065	0.1727 ± 0.0087	0.0000 ± 0.0000
	Adapter	18.2356 ± 0.5807	$\mathbf{0.7369 \pm 0.0059}$	0.1549 ± 0.0040	0.0000 ± 0.0000
	LoRA	16.9010 ± 0.6906	0.6937 ± 0.0043	0.1671 ± 0.0069	0.0000 ± 0.0000
	DANN-Gate	16.5786 ± 0.4868	0.6942 ± 0.0021	0.1544 ± 0.0048	0.0000 ± 0.0000
	REFINE	$\mathbf{20.3403 \pm 0.4768}$	0.7338 ± 0.0043	$\mathbf{0.1968 \pm 0.0059}$	$\mathbf{0.5263 \pm 1.0526}$
USPS→MNIST	NoTrans	62.0740 ± 8.7771	0.9566 ± 0.0073	0.5967 ± 0.0969	9.2863 ± 12.1512
	LinearProbe	66.9960 ± 1.0095	0.9469 ± 0.0050	0.6563 ± 0.0086	9.1576 ± 3.5478
	Adapter	61.8660 ± 3.0334	0.9375 ± 0.0085	0.5952 ± 0.0441	8.8750 ± 7.2427
	LoRA	64.8240 ± 0.8520	0.9333 ± 0.0045	0.6435 ± 0.0135	29.3265 ± 13.5652
	DANN-Gate	52.2080 ± 3.6669	0.9012 ± 0.0185	0.4853 ± 0.0482	0.0198 ± 0.0396
	REFINE	$\mathbf{70.0460 \pm 2.1721}$	$\mathbf{0.9582 \pm 0.0053}$	$\mathbf{0.6954 \pm 0.0194}$	$\mathbf{31.6157 \pm 14.5527}$
Books→Kitchen	NoTrans	71.6600 ± 1.3632	0.7848 ± 0.0155	0.7161 ± 0.0137	$\mathbf{68.6000 \pm 2.9719}$
	LinearProbe	66.7400 ± 3.1455	0.7568 ± 0.0278	0.6571 ± 0.0401	51.5600 ± 9.7336
	Adapter	71.3400 ± 0.1356	0.7839 ± 0.0008	0.7111 ± 0.0015	62.8800 ± 2.9027
	LoRA	66.9600 ± 0.2154	0.7279 ± 0.0018	0.6695 ± 0.0022	65.6400 ± 0.4079
	DANN-Gate	66.6000 ± 0.0894	0.7330 ± 0.0006	0.6659 ± 0.0009	64.6800 ± 0.6997
	REFINE	$\mathbf{72.7200 \pm 1.6522}$	$\mathbf{0.8147 \pm 0.0133}$	$\mathbf{0.7248 \pm 0.0189}$	65.5200 ± 6.4778
DVD→Electronics	NoTrans	68.5200 ± 2.8979	0.7585 ± 0.0304	0.6806 ± 0.0338	59.8000 ± 9.8298
	LinearProbe	66.0600 ± 0.5122	0.7266 ± 0.0017	0.6580 ± 0.0072	58.3600 ± 4.5579
	Adapter	65.8600 ± 0.3200	0.7206 ± 0.0008	0.6577 ± 0.0037	61.4400 ± 2.5935
	LoRA	66.5600 ± 0.3555	0.7170 ± 0.0013	0.6656 ± 0.0036	$\mathbf{65.4000 \pm 0.4899}$
	DANN-Gate	66.9000 ± 0.1897	0.7196 ± 0.0013	0.6686 ± 0.0019	63.5600 ± 0.2653
	REFINE	$\mathbf{70.3400 \pm 0.9972}$	$\mathbf{0.7886 \pm 0.0115}$	$\mathbf{0.6995 \pm 0.0122}$	61.7200 ± 7.5181

Table 1: Single-source transfer learning with natural distribution shift.

tion; Clipart→Sketch represents cross-style adaptation between artistic domains; USPS→MNIST examines digit recognition under handwriting and design difference; and Books→Kitchen and DVD→Electronics capture cross-topic sentiment transfer, where vocabulary and linguistic style vary considerably. We exclude knowledge distillation [12] in this comparison, as it requires identical class spaces across source and target, which do not apply here.

Table 1 reports the results. REFINE consistently achieves competitive or superior performance compared to alternative methods across all scenarios. On transfers with large label-space difference, including CIFAR100→10 and CIFAR10→100, REFINE improves accuracy by over $10 - 15\%$ relative to Adapter, LoRA, and DANN-Gate, substantially narrowing the gap to the no-transfer baseline while remaining robust to negative transfer. On transfers under natural resolution or stylistic shift, including CIFAR10→STL, Clipart→Sketch, REFINE achieves $3-4\%$ accuracy gains over the strongest alternative, along with consistent improvements in AUC and F1. On transfers with digit benchmarks, including USPS→MNIST, it yields $5 - 10\%$ accuracy gains, and much higher minimum class accuracy, indicating stronger preservation of performance on underrepresented classes. On transfers with cross-topics, including Books→Kitchen, DVD→Electronics), REFINE delivers $2 - 4\%$ improvements across all metrics. Overall, REFINE not only avoids the severe degradation observed in other methods, but also provides reliable accuracy lifts of $5 - 15\%$ across image and text domains under diverse settings of distribution shifts.

Dataset	Setting	Method	Acc	AUC	F1	MinCAcc
CIFAR-10	40% flips	NoTrans	56.05 ± 0.64	0.9037 ± 0.0028	0.5580 ± 0.0080	32.40 ± 5.84
		LinearProbe	65.54 ± 0.06	0.9378 ± 0.0003	0.6561 ± 0.0008	42.82 ± 1.45
		Adapter	65.78 ± 0.19	0.9376 ± 0.0007	0.6581 ± 0.0024	$\mathbf{45.20 \pm 2.29}$
		Distill	57.01 ± 0.58	0.9115 ± 0.0016	0.5674 ± 0.0032	34.84 ± 4.53
		LoRA	65.47 ± 0.12	0.9374 ± 0.0004	0.6545 ± 0.0018	42.38 ± 0.89
		DANN-Gate	65.40 ± 0.15	0.9353 ± 0.0006	0.6539 ± 0.0016	43.40 ± 2.22
		REFINE	$\mathbf{66.23 \pm 0.32}$	$\mathbf{0.9388 \pm 0.0006}$	$\mathbf{0.6625 \pm 0.0036}$	43.94 ± 3.78
	80% flips	NoTrans	56.57 ± 0.64	0.9057 ± 0.0033	0.5622 ± 0.0055	33.60 ± 3.04
		LinearProbe	19.46 ± 0.75	0.6895 ± 0.0011	0.1177 ± 0.0108	0.00 ± 0.00
		Adapter	18.49 ± 0.46	0.6906 ± 0.0006	0.1219 ± 0.0156	0.00 ± 0.00
		Distill	53.51 ± 0.79	0.8982 ± 0.0021	0.5269 ± 0.0091	26.80 ± 2.49
		LoRA	22.92 ± 1.73	0.7202 ± 0.0079	0.1911 ± 0.0308	0.76 ± 1.52
		DANN-Gate	20.83 ± 1.32	0.7097 ± 0.0084	0.1341 ± 0.0253	0.00 ± 0.00
		REFINE	$\mathbf{56.58 \pm 0.33}$	$\mathbf{0.9067 \pm 0.0019}$	$\mathbf{0.5655 \pm 0.0041}$	$\mathbf{36.90 \pm 2.94}$
	Schematic confusion	NoTrans	56.53 ± 0.77	0.9006 ± 0.0021	0.5639 ± 0.0056	35.76 ± 2.75
		LinearProbe	48.54 ± 0.42	0.8987 ± 0.0008	0.4757 ± 0.0046	18.44 ± 7.89
		Adapter	47.17 ± 0.82	0.8998 ± 0.0006	0.4479 ± 0.0148	7.42 ± 6.47
		Distill	57.80 ± 0.44	$\mathbf{0.9068 \pm 0.0009}$	0.5772 ± 0.0037	35.92 ± 3.00
		LoRA	49.96 ± 0.26	0.9039 ± 0.0005	0.4864 ± 0.0116	16.34 ± 9.91
		DANN-Gate	49.04 ± 0.33	0.9028 ± 0.0006	0.4719 ± 0.0059	11.40 ± 1.53
		REFINE	$\mathbf{58.65 \pm 0.47}$	0.9034 ± 0.0011	$\mathbf{0.5861 \pm 0.0048}$	$\mathbf{38.40 \pm 3.10}$
	Class imbalance	NoTrans	56.44 ± 0.48	0.9055 ± 0.0019	0.5599 ± 0.0051	32.80 ± 4.54
		LinearProbe	53.15 ± 1.04	0.8883 ± 0.0145	0.5238 ± 0.0215	28.36 ± 14.04
		Adapter	51.64 ± 0.99	0.8960 ± 0.0022	0.5130 ± 0.0150	19.52 ± 8.32
		Distill	54.89 ± 0.49	0.9063 ± 0.0013	0.5492 ± 0.0065	$\mathbf{41.96 \pm 3.43}$
		LoRA	53.21 ± 0.19	0.8975 ± 0.0005	0.5338 ± 0.0022	33.76 ± 5.38
		DANN-Gate	53.05 ± 0.28	0.8964 ± 0.0009	0.5281 ± 0.0055	32.62 ± 3.60
		REFINE	$\mathbf{56.54 \pm 0.73}$	$\mathbf{0.9103 \pm 0.0012}$	$\mathbf{0.5619 \pm 0.0103}$	31.58 ± 10.31

Table 2: Single-source transfer learning with label noise, semantic perturbation, and class imbalance for CIFAR-10 using CNNs.

5.3 SINGLE-SOURCE TRANSFER UNDER LABEL NOISE, SEMANTIC PERTURBATION, AND CLASS IMBALANCE

In the second experiment setting, we deliberately construct challenging scenarios to stress-test various transfer learning methods. Using CIFAR-10 with CNNs, we introduce four types of challenges in the pretraining data while keeping the target domain fixed: (i) heavy label noise with 40% random label flips, (ii) extreme label noise with 80% flips, (iii) semantic perturbation created by paired-class flipping combined with additive image noise, and (iv) class imbalance induced by resampling to a long-tailed distribution. In addition, we repeat the experiments on CIFAR-100 and also evaluate both CIFAR-10 and CIFAR-100 with transformer-based models. We report the corresponding results in Appendix C.1.

Table 2 summarizes the results. REFINE consistently mitigates severe degradation and outperforms competing methods across all stress-test scenarios. In the moderate noise setting with 40% label flips, it achieves the best overall balance, improving accuracy and F1 by about 1% over Adapter and LoRA, while maintaining competitive minimum class accuracy. In the more extreme noise setting with 80% flips, most baselines collapse, with LinearProbe, Adapter, and DANN-Gate drop below 25% accuracy, whereas REFINE remains close to the no-transfer baseline, improving accuracy by nearly 35% over the strongest adaptive alternative. In the semantic confusion setting, with paired-class flips plus image noise, REFINE gains 1-2% in accuracy and F1 over NoTrans, while all other adaptive baselines perform worse, highlighting the robustness of REFINE to perturbed label semantics. In the class imbalance setting, it surpasses LinearProbe, Adapter, and LoRA by 3-5% in accuracy and F1, achieving the strongest overall results aside from a slightly lower minimum class accuracy than Distillation. Overall, REFINE avoids the catastrophic failures common to existing transfer strategies under noise, semantic perturbation, and class imbalance, while consistently delivering performance gains across all stress-test conditions.

We also briefly remark that, an important advantage of REFINE is that its complexity can be flexibly tuned through the choice of the encoder h. Such a design keeps it comparable in parameter efficiency to methods such as Adapter and Distillation. For instance, in this setting, for REFINE, the number of trainable parameters is 4.88% of the total number of parameters in the frozen source

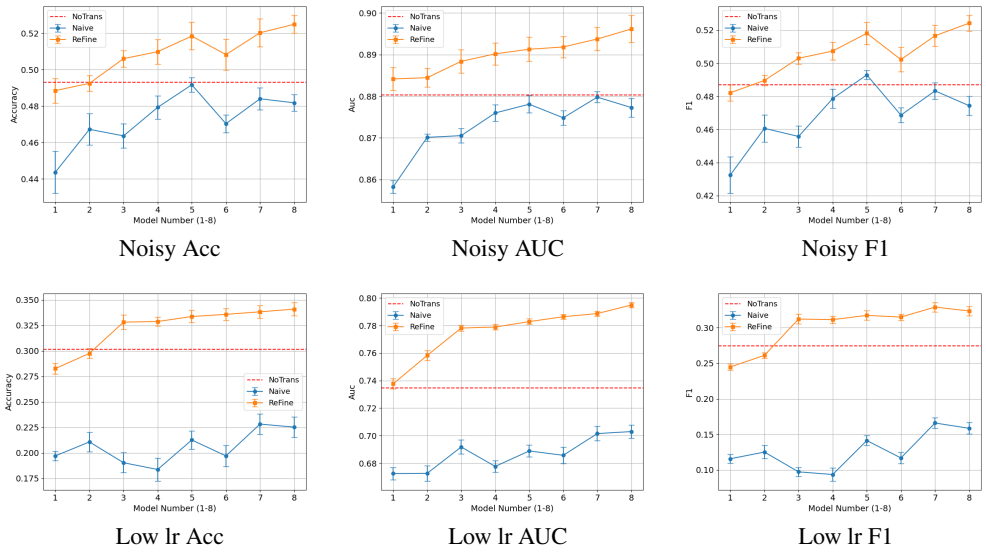


Figure 2: Results of multi-source transfer learning under noisy and low-learning-rate conditions.

model, for Adapter, it is 5.46%, and for Distillation, 4.68%. Thus REFINE achieves a comparable parameter efficiency, but clearly outperforms in mitigating negative transfer. The ablation study in Appendix C.3 further shows that the performance of REFINE remains stable across different parameter choices of h, indicating that the overall parameter complexity has relatively little impact. By contrast, increasing Adapter's complexity fails to resolve negative transfer, suggesting that its limitation stems from design rather than capacity.

5.4 MULTI-SOURCE TRANSFER

In the third experiment setting, we investigate multi-source transfer, an important yet underexplored setting where knowledge is drawn from multiple heterogeneous sources to achieve better generalization than any single source alone. Despite its practical relevance, most existing approaches, such as LinearProbe, Adapter, and Distillation, are designed for single-source transfer and do not naturally extend to the multi-source case. To provide a fair comparison, we implement a Naive baseline that assigns each source domain its own feature extractor, concatenates the resulting representations, and trains a classifier on top of the joint embedding. This straightforward strategy captures the most natural way of leveraging multiple sources in the absence of specialized methods. For our experiments, we partition CIFAR-10 into eight disjoint subsets of 2000 samples each, treating them as distinct source domains and training separate CNNs on each. REFINE then integrates the corresponding penultimate representations through its modular structure, mimicking multi-source transfer while keeping inference overhead modest. This setup enables a direct evaluation of principled multi-source integration against naive concatenation.

Figure 2 reports the results under two stress conditions, a noisy case with 50% label corruption, testing robustness to unreliable label supervision, and a low learning rate case, testing training stability and efficiency. In the noisy case, REFINE significantly outperforms both Naive and NoTrans as more external sources are integrated. With all eight sources, REFINE achieves classification accuracy 52.5%, AUC 0.8962, and F1 0.5242, compared to Naive's 48.2%, 0.8773, and 0.4744, and NoTrans's 49.3%, 0.8803, and 0.4871. Notably, Naive consistently performs worse than NoTrans, indicating negative transfer when external information is not integrated effectively. In the low learning rate case, REFINE again improves steadily over NoTrans as the number of sources increases, while Naive suffers severe degradation. With all eight sources, REFINE reaches 34.09% classification accuracy, surpassing NoTrans's 30.16% and Naive's 22.53%. Overall, these results demonstrate that REFINE effectively integrates multiple sources, and remains robust under adverse supervision and training conditions. It avoids the pitfalls of naive concatenation and provides a stable approach for multi-source transfer.

Ethics Statement

This research does not involve human subjects, personally identifiable information, or sensitive data. The datasets used are publicly available and widely used in the community. We are not aware of direct applications of our method that raise ethical concerns. Nevertheless, as with any machine learning system, there is a potential risk of misuse if deployed in contexts where fairness or bias are critical. We encourage future work to examine these dimensions before deployment in such settings.

Reproducibility Statement

We have made efforts to ensure the reproducibility of our results. Detailed descriptions of datasets, preprocessing steps, and hyperparameters, optimizers are provided in Section 5 and Appendix D. All proofs for theoretical claims are provided in Section 4 and Appendix A. An anonymized version of our source code is included in the supplementary materials and will be released publicly upon acceptance.

References

[1] Muhammad Jamal Afridi, Arun Ross, and Erik M. Shapiro. On automated source selection for transfer learning in convolutional neural networks. *Pattern Recognition*, 73:65–75, 2018. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2017.07.019. URL https://www.sciencedirect.com/science/article/pii/S0031320317302881.

[2] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Annie Zaenen and Antal van den Bosch (eds.), *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-1056/.

[3] Aurélien Bibaut Cheng Ju and Mark van der Laan. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800–2818, 2018. doi: 10.1080/02664763.2018.1441383. URL https://doi.org/10.1080/02664763.2018.1441383. PMID: 31631918.

[4] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

[5] Rhys Compton, Lily Zhang, Aahlad Puli, and Rajesh Ranganath. When more is less: Incorporating additional datasets can hurt performance by introducing spurious correlations, 2023. URL https://arxiv.org/abs/2308.04431.

[6] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning, 2019. URL https://arxiv.org/abs/1904.02868.

[7] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[8] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(null):723–773, March 2012. ISSN 1532-4435.

[9] Pratham Grover, Kunal Chaturvedi, Xing Zi, Amit Saxena, Shiv Prakash, Tony Jan, and Mukesh Prasad. Ensemble transfer learning for distinguishing cognitively normal and mild cognitive impairment patients using mri. *Algorithms*, 16(8), 2023. ISSN 1999-4893. doi: 10.3390/a16080377. URL https://www.mdpi.com/1999-4893/16/8/377.

[10] XY Han, Vardan Papyan, and David L Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

[13] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim. Transfer learning: a friendly introduction. *Journal of Big Data*, 9(1):102, 2022. doi: 10.1186/s40537-022-00652-w. URL https://doi.org/10.1186/s40537-022-00652-w. Epub 2022 Oct 22.

[14] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019. URL https://arxiv.org/abs/1902.00751.

[15] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[16] Rohan Jha, Charles Lovering, and Ellie Pavlick. Does data augmentation improve generalization in nlp?, 2020. URL https://arxiv.org/abs/2004.15012.

[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[18] Ronny Kohavi and Barry Becker. Adult income dataset. https://www.kaggle.com/datasets/uciml/adult-census-income, 1996. Originally from the UCI Machine Learning Repository. Kaggle version shared by user 1251, updated 2016.

[19] Michael Kohler and Sophie Langer. On the rate of convergence of fully connected deep neural network regression estimates. *The Annals of Statistics*, 49(4):2231–2249, 2021.

[20] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. Technical Report.

[21] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution, 2022. URL https://arxiv.org/abs/2202.10054.

[22] Yu-Feng Li, Lan-Zhe Guo, and Zhi-Hua Zhou. Towards safe weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):334–346, 2021. doi: 10.1109/TPAMI.2019.2922396.

[23] Chi-Heng Lin, Chiraag Kaushik, Eva L. Dyer, and Vidya Muthukumar. The good, the bad and the ugly sides of data augmentation: An implicit spectral regularization perspective. *J. Mach. Learn. Res.*, 25:91:1–91:85, 2022. URL https://api.semanticscholar.org/CorpusID:252815719.

[24] Y. P. Lin and T. P. Jung. Improving eeg-based emotion classification using conditional transfer learning. *Frontiers in Human Neuroscience*, 11:334, 2017. doi: 10.3389/fnhum.2017.00334. URL https://doi.org/10.3389/fnhum.2017.00334. Published on June 27, 2017.

[25] Ryumei Nakada and Masaaki Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *Journal of Machine Learning Research*, 21(174): 1–38, 2020.

[26] Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip, 2023. URL https://arxiv.org/abs/2208.05516.

[27] Rohan Paris. Credit score classification. https://www.kaggle.com/datasets/rohanparis/credit-score-classification, 2022. Kaggle Dataset, CC0: Public Domain.

[28] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

[29] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

[30] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in Neural Information Processing Systems (NeurIPS 2019)*, pp. 3347–3357, Red Hook, NY, USA, 2019. Curran Associates, Inc.

[31] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 2020.

[32] Michael J. Sorocky, Siqi Zhou, and Angela P. Schoellig. Experience selection using dynamics similarity for efficient multi-source transfer learning between robots. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2739–2745, 2020. doi: 10.1109/ICRA40945.2020.9196744.

[33] Taiji Suzuki. Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. *arXiv preprint arXiv:1810.08033*, 2018.

[34] Tedo. Students performance: analysis and classification. https://www.kaggle.com/code/tedo/students-performance-analysis-and-classification, 2018. Kaggle Notebook, Version 4, Apache 2.0 License.

[35] Karthik Chowdary Tsaliki. Diabetes classification (pima indians diabetes database). https://www.kaggle.com/competitions/diabetes-classification, 2019. Kaggle Community Prediction Competition.

[36] A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, 2008. ISBN 9780387790527. URL https://books.google.com/books?id=mwB8rUBsbqoC.

[37] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11285–11294, 2019. doi: 10.1109/CVPR.2019.01155.

[38] Dongrui Wu. Online and offline domain adaptation for reducing bci calibration effort. *IEEE Transactions on Human-Machine Systems*, 47(4):550–563, 2017. doi: 10.1109/THMS.2016.2608931.

[39] Ge Xie, Yu Sun, Minlong Lin, and Ke Tang. A selective transfer learning method for concept drift adaptation. In Fengyu Cong, Andrew Leung, and Qinglai Wei (eds.), *Advances in Neural Networks - ISNN 2017*, pp. 353–361, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59081-3.

[40] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2023. doi: 10.1109/JAS.2022.106004.

[41] Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. In *International Conference on Machine Learning*, pp. 27179–27202. PMLR, 2022.

APPENDICES

In the appendices, we provide additional technical and empirical details. Appendix A provides the proof of the main theorem, supported by auxiliary lemmas in Appendix B. Appendix C expands the empirical evaluations, including additional results on more benchmark data, tabular data, and an ablation study. Appendix D documents the experiment setup and implementation details for reproducibility. Together, they offer a complete account of the theory, validation, and practical details underlying our work.

A PROOFS OF MAIN RESULTS

In this section, we prove the main results in Section 4.

Additional Notation. Let $\|\cdot\|_{L_q}$ denote the L_q norm under the probability measure $\mathbb{P}^{\mathrm{t}}_X$ for any $q \in [1,\infty]$, where $\mathbb{P}^{\mathrm{t}}_X$ is the distribution of X_i in the training data. For $a, b \in \mathbb{R}$, we define $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$.

In addition, we would like to recall $\mathcal{R}_{\mathbb{P}^{\mathrm{t}}}(g) = \mathbb{E}_{(X,Y)\sim\mathbb{P}^{\mathrm{t}}}[(g(X) - Y)^2]$. As a result,

$$\begin{aligned}\mathcal{R}_{\mathbb{P}^{\mathrm{t}}}(g) - \mathcal{R}_{\mathbb{P}^{\mathrm{t}}}(f^*) &= \mathbb{E}_{(X,Y)\sim\mathbb{P}^{\mathrm{t}}}[(g(X) - f^*(X) - \epsilon)^2] - \sigma^2 \\ &= \mathbb{E}_{(X,Y)\sim\mathbb{P}^{\mathrm{t}}}[(g(X) - f^*(X))^2] \asymp \|g - f^*\|^2_{L_2}, \end{aligned} \tag{S.1}$$

where the last asymptotic equivalence is due to the fact that we assume X has positive continuous density on $[0,1]^d$ bounded by an absolute value. As $[0,1]^d$ is a compact space and the density function of X is continuous, this implies that the density function is both upper and lower bounded by absolute constants.

We now prove the main theorem on the prediction risk of REFINE. The results of the two corollaries can be obtained straightforwardly, and we thus omit their proofs.

Proof of Theorem 4.1.

Proof. Recall that v^* is the optimal linear probe of f_{rep}, i.e.,

$$v^* = \arg\min_{v\in\mathbb{R}^p} \mathbb{E}[\{f^*(X) - v^\top f_{\text{rep}}(X)\}^2].$$

We begin by observing that the difficulty of the estimation problem is governed by the residual $r^* := f^* - v^{*\top} f_{\text{rep}}$, since f_{rep} is assumed to be known, and $v^{*\top} f_{\text{rep}}$ can be seen as a linear function of the known quantity. By appropriately choosing the parameters W, L, and B, we control the complexity of the neural network, and the bias of estimating r^*.

Specifically, choose

$$L = (2 + \lceil \log_2 \beta \rceil)\left(11 + \frac{\beta}{d}\right), \;\; W = c_1' \epsilon^{-d/\beta}, \;\; B = \epsilon^{-c_2'}, \tag{S.2}$$

where $c_1', c_2' > 0$ are constants appearing in Lemma B.2. Define $\rho^* := \|r^*\|_{\mathcal{C}^\beta}$. Set

$$\epsilon := n^{-\beta/(2\beta+d)} \rho^{-2\beta/(2\beta+d)} \wedge 1, \tag{S.3}$$

where $\rho > 0$ is some tuning parameter. The choices in (4) are realized by taking $\epsilon = n^{-\beta/(2\beta+d)}\rho^{-2\beta/(2\beta+d)}$ and setting $c_1 := (2 + \lceil \log_2 \beta \rceil)(11 + \beta/d)$, $c_2 := c_1'$, $c_3 := c_2'$.

Note that $\sup_{g\in\mathcal{G}_{d,p}(W,L,B;f_{\text{rep}})} \|g\|_{L_\infty} \le 2$. From Lemma B.3 with the choice $\delta \leftarrow 1/n$, we have

$$\mathbb{E}[\|\hat{g} - f^*\|^2_{L_2}] \lesssim \left(\inf_{g\in\mathcal{G}} \|g - f^*\|^2_{L_2} + \frac{\log \mathcal{N}(1/n, \mathcal{G}_{d,p}(W,L,B;f_{\text{rep}}), \|\cdot\|_{L_\infty})}{n} + \frac{1}{n}\right).$$

Next we compute the first term and the second term separately.

Part 1: Bounding the first term. Notice that $\rho^* \leq 1$ by assumption $r^* = f^* - v^{*\top} f_{\text{rep}} \in \mathcal{C}_{\text{u}}^{\beta}$. Rescale the residual by noting that $(1/\rho^*)r^* \in \mathcal{C}_{\text{u}}^{\beta}$. Then, by Lemma B.2, there exists a neural network $r_{\text{NN}} \in \mathcal{H}_d(W, L, B)$ such that

$$\|r_{\text{NN}} - (1/\rho^*)r^*\|_{L_2} \lesssim \epsilon. \tag{S.4}$$

This inequality provides the approximation error of the ReLU network class. To translate this result to the bias term $\|g - f^*\|_{L_2}^2$, we proceed as follows. Write

$$r_{\text{NN}} = r_{\text{NN},L} \circ r_{\text{NN},L-1} \circ \cdots \circ r_{\text{NN},1}(x),$$

where $r_{\text{NN},\ell}(x) = \sigma(A_\ell x + b_\ell)$ for $\ell \in [L-1]$ and $r_{\text{NN},L}(x) = A_L x + b_L$. Define $r'_{\text{NN},L}(x) = (\rho^* A_L)x + (\rho^* b_L)$ to approximate $\rho^* r_{\text{NN}}$. Then, it follows that the function

$$g^\circ(x) := 1 \wedge ((-1) \vee r'_{\text{NN},L} \circ r_{\text{NN},L-1} \circ \cdots \circ r_{\text{NN},1}(x)) + v^{*\top} f_{\text{rep}}(x)$$

belongs to $\mathcal{G}_{d,p}(W, L, B; f_{\text{rep}})$ since $\rho^* \leq 1$ and $\|v^*\| \leq 1$. Moreover, we can write g° as

$$g^\circ(x) = 1 \wedge ((-1) \vee \rho^* r_{\text{NN}}(x)) + v^{*\top} f_{\text{rep}}(x).$$

Using (S.4), we have

$$\begin{aligned}
\mathbb{E}[\{g^\circ(X_1) - f^*(X_1)\}^2]^{1/2} &= \|1 \wedge ((-1) \vee \rho^* r_{\text{NN}}) + v^{*\top} f_{\text{rep}} - f^*\|_{L_2} \\
&= \rho^* \left\| \frac{1}{\rho^*} \wedge \left(-\frac{1}{\rho^*} \vee r_{\text{NN}} \right) - \frac{1}{\rho^*} r^* \right\|_{L_2} \\
&\leq \rho^* \left\| r_{\text{NN}} - \frac{1}{\rho^*} r^* \right\|_{L_2} \\
&\lesssim \rho^* \epsilon,
\end{aligned}$$

where we used the fact that $\|r^*/\rho^*\|_{L_\infty} \leq \|r^*/\rho^*\|_{\mathcal{C}^\beta} \leq 1/\rho^*$. Thus,

$$\inf_{g \in \mathcal{G}_{d,p}(W,L,B;f_{\text{rep}})} \mathbb{E}[\|g - f^*\|_{L_2}^2] \leq \mathbb{E}[\|g^\circ - f^*\|_{L_2}^2] \lesssim \rho^{*2} \epsilon^2.$$

Part 2: Bounding the second term. The covering number bound from Lemma B.4 with the choice of W, L, B in (S.2), we have

$$\frac{\log \mathcal{N}(1/n, \mathcal{G}_{d,p}(W, L, B; f_{\text{rep}}), \|\cdot\|_{L_\infty})}{n} \leq \frac{C'}{n} (\epsilon^{-d/\beta} + p) \log\left(\frac{n}{\epsilon}\right),$$

where C' is a constant depending on d and β.

Part 3: Balancing terms. Finally, we combine the results from part 1 and part 2. Recalling the choice of ϵ in (S.3), we consider two cases depending on the value of ρ.

When $1/\sqrt{n} \leq \rho$, we have $\epsilon = (n\rho^2)^{-\beta/(2\beta+d)}$. In this case, the bound becomes

$$\begin{aligned}
\mathbb{E}[\|\hat{g} - f^*\|_{L_2}^2] &\leq \rho^{*2} \rho^{-4\beta/(2\beta+d)} n^{-2\beta/(2\beta+d)} + C' \left(\rho^{2d/(2\beta+d)} n^{-2\beta/(2\beta+d)} + \frac{p}{n} \right) \log n \\
&\leq (C'+1) \left((\rho^{*2} \rho^{-4\beta/(2\beta+d)} + \rho^{2d/(2\beta+d)} \log n) n^{-2\beta/(2\beta+d)} + \frac{p \log n}{n} \right).
\end{aligned} \tag{S.5}$$

When $\rho \leq 1/\sqrt{n}$ (so that $\epsilon = 1$), the bound becomes

$$\mathbb{E}[\|\hat{g} - f^*\|_{L_2}^2] \leq \rho^{*2} + C' \frac{p \log n}{n} \leq (C'+1) \left(\rho^{*2} \rho^{-4\beta/(2\beta+d)} n^{-2\beta/(2\beta+d)} + \frac{p \log n}{n} \right). \tag{S.6}$$

Combining the bounds in (S.5) and (S.6) with (S.1), we obtain the desired result.

This completes the proof of Theorem 4.1. □

B AUXILIARY LEMMAS

In this section, we provide some auxiliary lemmas.

The next lemma is about the entropy bound for $\mathcal{H}_d(W, L, B)$.

Lemma B.1 (Lemma 21 from Nakada & Imaizumi [25]). *Fix any W, L, and $B > 0$. Then, we have the covering number bound*

$$\log \mathcal{N}(\epsilon, \mathcal{H}_d(W, L, B), \|\cdot\|_{L_\infty}) \leq W \log\left(\frac{2LB^L(W+1)^L}{\epsilon}\right).$$

The next lemma is modified from Petersen & Voigtlaender [29], adapted to consider L_2 approximation error with respect to the probability measure $\mathbb{P}^{\mathrm{t}}_X$ over the domain $[0,1]^d$, rather than the original L_2 error with a uniform measure on $[-1/2, 1/2]^d$.

Lemma B.2 (Modification of Theorem 3.1 from Petersen & Voigtlaender [29]). *Fix $d \in \mathbb{N}_+$ and $\beta > 0$. Suppose that $\mathbb{P}^{\mathrm{t}}_X$ has a density bounded by $O(1)$. Then, there exist constants $c_1', c_2' > 0$, depending on d and β, such that for any $\epsilon \in (0, 1/2)$, if one chooses W, L, and B satisfying*

$$L \leq (2 + \lceil \log_2 \beta \rceil)\left(11 + \frac{\beta}{d}\right), \quad W \leq c_1' \epsilon^{-d/\beta}, \quad B \leq \epsilon^{-c_2'},$$

then

$$\sup_{f^\# \in \mathcal{C}_u^\beta} \inf_{f_{NN} \in \mathcal{H}_d(W,L,B)} \|f_{NN} - f^\#\|_{L_2} \lesssim \epsilon.$$

The next lemma provides a bound on the prediction risk of the empirical risk minimizer in terms of the covering number of the function class and the approximation error.

Lemma B.3 (Modification to Lemma 4 from Schmidt-Hieber [31]). *Let $\mathcal{G}$ be a function class, and let $\hat{g}$ be the minimizer of the empirical risk $(1/n)\sum_{i\in[n]} \ell(\hat{g}(X_i), Y_i)$ over $\mathcal{G}$ under the data generating process introduced in Section 4. Suppose that $\{f^*\} \cup \mathcal{G} \subset \{[0,1]^d \to [-F, F]\}$ for some $F \geq 1$. Then there exists a universal constant $C_0 > 0$ such that*

$$\mathbb{E}[\|\hat{g} - f^*\|_{L_2}^2] \leq C_0\left(\inf_{g\in\mathcal{G}} \|g - f^*\|_{L_2}^2 + F^2 \frac{\log \mathcal{N}(\delta, \mathcal{G}, \|\cdot\|_{L_\infty})}{n} + \delta F\right).$$

The next lemma provides a bound on the covering number of the REFINE class $\mathcal{G}_{d,p}(W, L, B; f_{\text{rep}})$.

Lemma B.4. *Fix $W \in \mathbb{N}_+$, $L \in \mathbb{N}_+$, $B > 0$, and $\delta > 0$. Then, there exists a universal constant $C > 0$ such that*

$$\log \mathcal{N}(\delta, \mathcal{G}_{d,p}(W, L, B; f_{rep}), \|\cdot\|_{L_\infty}) \leq C\left\{W \log\left(\frac{LB^L(W+1)^L}{\delta}\right) + p \log\left(\frac{1}{\delta}\right)\right\}.$$

Proof. We next bound the covering number $\mathcal{N}(\delta, \mathcal{G}_{d,p}(W, L, B; f_{\text{rep}}), \|\cdot\|_{L_\infty})$. Note that for any $\delta > 0$, we have

$$\begin{aligned}
&\log \mathcal{N}(\delta, \mathcal{G}_{d,p}(W, L, B; f_{\text{rep}}), \|\cdot\|_{L_\infty}) \\
&\quad \leq \log \mathcal{N}\left(\frac{\delta}{2}, \{x \mapsto uh(x) \mid u \in [-1,1], h \in \bar{\mathcal{H}}_d(W, L, B)\}, \|\cdot\|_{L_\infty}\right) \\
&\qquad + \log \mathcal{N}\left(\frac{\delta}{2}, \{x \mapsto v^\top f_{\text{rep}}(x) \mid v \in \mathcal{B}_p(1)\}, \|\cdot\|_{L_\infty}\right). \qquad \text{(S.7)}
\end{aligned}$$

Recall that $f_{\text{rep}} : [0,1]^d \to \mathcal{B}_p(1)$. Since $\|v^\top f_{\text{rep}} - v'^\top f_{\text{rep}}\|_{L_\infty} \leq \|v - v'\|_2$ for any $v, v' \in \mathcal{B}_p(1)$, a standard argument shows that

$$\mathcal{N}\left(\frac{\delta}{2}, \{x \mapsto v^\top f_{\text{rep}}(x) \mid v \in \mathcal{B}_p(1)\}, \|\cdot\|_{L_\infty}\right) \leq \mathcal{N}\left(\frac{\delta}{2}, \mathcal{B}_p(1), \|\cdot\|_2\right) \leq \left(\frac{6}{\delta}\right)^p. \qquad \text{(S.8)}$$

Furthermore, since $\|u_1 h_1 - u_2 h_2\|_{L_\infty} \le \|h_1 - h_2\|_{L_\infty} + |u_1 - u_2|$ for any $u_1, u_2 \in [-1, 1]$ and $h_1, h_2 \in \bar{\mathcal{H}}_d(W, L, B)$, we have

$$\begin{aligned}
&\mathcal{N}(\frac{\delta}{2}, \{x \mapsto uh(x) \mid u \in [-1,1], h \in \bar{\mathcal{H}}_d(W,L,B)\}, \|\cdot\|_{L_\infty}) \\
&\quad \le \mathcal{N}\left(\frac{\delta}{4}, [-1,1], |\cdot|\right) \mathcal{N}\left(\frac{\delta}{4}, \bar{\mathcal{H}}_d(W,L,B), \|\cdot\|_{L_\infty}\right) \\
&\quad \lesssim \frac{1}{\delta} \mathcal{N}\left(\frac{\delta}{4}, \mathcal{H}_d(W,L,B), \|\cdot\|_{L_\infty}\right). \qquad \text{(S.9)}
\end{aligned}$$

Note that clipping does not increase the covering number of $mH_d(W, L, B)$. Using (S.7), (S.8) and (S.9), combined with Lemma B.1, we obtain

$$\log \mathcal{N}(\delta, \mathcal{G}_{d,p}(W,L,B; f_{\text{rep}}), \|\cdot\|_{L_\infty}) \lesssim W \log\left(\frac{LB^L(W+1)^L}{\delta}\right) + p \log\left(\frac{1}{\delta}\right).$$

This completes the proof of Lemma B.4. □

C More Numerical Experiments

In this section, we present additional results that complement Section 5.

C.1 Single-source transfer under challenging scenarios

Similar to the setting considered in Section 5.3 for CIFAR-10, we run the experiments on CIFAR-100. Moreover, in addition to CNNs, we also evaluate both CIFAR-10 and CIFAR-100 with transformer-based models.

Table S1 reports the results on CIFAR-100 with CNNs. Similar to CIFAR-10, ReFine consistently outperforms the baseline methods under all four stress scenarios. In particular, in the extreme noise setting with 80% label flips, most competing methods collapse to near-random performance, whereas ReFine remains stable and comparable to the no-transfer baseline. In the semantic confusion and class imbalance settings, ReFine achieves the strongest improvements in classification accuracy and F1, highlighting its ability to mitigate negative transfer even when pretraining data is severely perturbed.

Table S2 and Table S3 report the results on CIFAR-10 and CIFAR-100, respectively, with transformer-based models. Similar to CNNs, existing adaptation methods degrade sharply under noisy or imbalanced pretraining, whereas ReFine maintains stable and superior performance in accuracy, AUC, and F1.

Together, these results demonstrate that the advantages of ReFine are not tied to a specific model architecture or dataset size. By design, it reliably suppresses negative transfer and delivers consistent gains under challenging pretraining conditions.

C.2 Tabular data

We demonstrate that ReFine is equally effective in handling tabular data. We consider three binary-class datasets, Adult [18], Credit [27], Diabetes [35], and one multi-class dataset, Performance [34]. Each raw training data contains $K \times 100$ samples, where K is the number of classes. To assess model complexity, we design two multilayer perceptron (MLP) architectures: MLP1 with a lower complexity, and MLP2 with a more complex structure. We also compare to DirectAug, which refers to directly combining additional data with the raw data to train the classifier.

Table S4 reports the results using the original data, and Table S5 reports the results using the noisy data with 80% flips of class labels. In both settings, ReFine consistently improves accuracy, AUC, and F1 over using the raw data alone. Although DirectAug can sometimes perform better through full data merging, ReFine surpasses it on several datasets, including Credit and Performance, confirming its ability to exploit useful auxiliary information without over-relying on data merging. In the presence of heavy label noise, DirectAug suffers severe degradation, whereas ReFine maintains

Dataset	Setting	Method	Acc	AUC	F1	MinCAcc
CIFAR-100	40% flips	NoTrans	17.82 ± 0.36	0.8259 ± 0.0068	0.1684 ± 0.0039	$\mathbf{0.60 \pm 0.49}$
		LinearProbe	17.35 ± 0.27	$\mathbf{0.8605 \pm 0.0015}$	0.1472 ± 0.0043	0.00 ± 0.00
		Adapter	16.19 ± 0.33	0.8578 ± 0.0019	0.1303 ± 0.0037	0.00 ± 0.00
		Distill	18.73 ± 0.22	$\mathbf{0.8605 \pm 0.0035}$	0.1631 ± 0.0024	0.00 ± 0.00
		LoRA	17.24 ± 0.33	0.8568 ± 0.0018	0.1463 ± 0.0053	0.00 ± 0.00
		DANN-Gate	15.02 ± 0.39	0.8472 ± 0.0020	0.1239 ± 0.0041	0.00 ± 0.00
		REFINE	$\mathbf{19.28 \pm 0.34}$	0.8555 ± 0.0042	$\mathbf{0.1805 \pm 0.0043}$	0.40 ± 0.80
	80% flips	NoTrans	$\mathbf{17.52 \pm 0.60}$	$\mathbf{0.8252 \pm 0.0059}$	$\mathbf{0.1663 \pm 0.0047}$	$\mathbf{0.60 \pm 0.49}$
		LinearProbe	1.00 ± 0.00	0.6740 ± 0.0019	0.0002 ± 0.0000	0.00 ± 0.00
		Adapter	1.00 ± 0.00	0.5250 ± 0.0058	0.0002 ± 0.0000	0.00 ± 0.00
		Distill	15.11 ± 0.49	0.8174 ± 0.0069	0.1227 ± 0.0039	0.00 ± 0.00
		LoRA	2.01 ± 0.18	0.6251 ± 0.0032	0.0026 ± 0.0006	0.00 ± 0.00
		DANN-Gate	1.00 ± 0.00	0.5754 ± 0.0113	0.0002 ± 0.0000	0.00 ± 0.00
		REFINE	17.37 ± 1.09	0.8239 ± 0.0060	0.1641 ± 0.0109	0.20 ± 0.40
	Schematic confusion	NoTrans	18.13 ± 0.74	0.8129 ± 0.0044	0.1747 ± 0.0073	1.20 ± 0.75
		LinearProbe	20.81 ± 0.13	0.8316 ± 0.0003	0.2006 ± 0.0038	0.60 ± 0.80
		Adapter	19.99 ± 0.24	0.8308 ± 0.0012	0.1895 ± 0.0052	0.00 ± 0.00
		Distill	20.06 ± 0.89	$\mathbf{0.8361 \pm 0.0077}$	0.1959 ± 0.0080	1.00 ± 0.63
		LoRA	20.05 ± 0.18	0.8246 ± 0.0017	0.1953 ± 0.0035	0.60 ± 0.80
		DANN-Gate	17.56 ± 0.33	0.8122 ± 0.0023	0.1720 ± 0.0032	0.00 ± 0.00
		REFINE	$\mathbf{21.76 \pm 0.60}$	0.8308 ± 0.0072	$\mathbf{0.2139 \pm 0.0067}$	$\mathbf{2.00 \pm 1.10}$
	Class imbalance	NoTrans	17.58 ± 0.24	0.8271 ± 0.0033	0.1656 ± 0.0046	$\mathbf{1.00 \pm 0.00}$
		LinearProbe	22.41 ± 0.48	0.8687 ± 0.0011	0.2133 ± 0.0048	0.00 ± 0.00
		Adapter	22.66 ± 0.30	0.8676 ± 0.0014	0.2102 ± 0.0025	0.00 ± 0.00
		Distill	19.59 ± 0.61	0.8659 ± 0.0034	0.1752 ± 0.0072	0.00 ± 0.00
		LoRA	22.56 ± 0.39	0.8535 ± 0.0009	0.2129 ± 0.0022	0.00 ± 0.00
		DANN-Gate	20.72 ± 0.24	0.8432 ± 0.0021	0.1966 ± 0.0031	0.00 ± 0.00
		REFINE	$\mathbf{23.31 \pm 0.42}$	$\mathbf{0.8719 \pm 0.0010}$	$\mathbf{0.2264 \pm 0.0032}$	0.40 ± 0.49

Table S1: Single-source transfer learning with label noise, semantic perturbation, and class imbalance for CIFAR-100 using CNNs.

Dataset	Setting	Method	Acc	AUC	F1	MinCAcc
CIFAR-10	80% flips	NoTrans	45.17 ± 1.39	0.8678 ± 0.0028	0.4391 ± 0.0183	16.24 ± 4.52
		LinearProbe	20.65 ± 0.44	0.6826 ± 0.0025	0.1410 ± 0.0083	0.00 ± 0.00
		Adapter	17.88 ± 0.73	0.6682 ± 0.0066	0.1248 ± 0.0111	0.00 ± 0.00
		Distill	40.19 ± 0.57	0.8445 ± 0.0022	0.3827 ± 0.0068	8.00 ± 5.22
		LoRA	21.69 ± 0.49	0.6831 ± 0.0010	0.1511 ± 0.0059	0.00 ± 0.00
		DANN-Gate	21.37 ± 0.27	0.6829 ± 0.0015	0.1468 ± 0.0075	0.00 ± 0.00
		REFINE	$\mathbf{45.53 \pm 0.95}$	$\mathbf{0.8694 \pm 0.0047}$	$\mathbf{0.4463 \pm 0.0105}$	$\mathbf{18.68 \pm 4.97}$
	Domain mismatch	NoTrans	44.37 ± 0.74	0.8628 ± 0.0035	0.4375 ± 0.0055	20.80 ± 4.86
		LinearProbe	46.04 ± 0.71	0.8643 ± 0.0015	0.4544 ± 0.0080	23.46 ± 4.74
		Adapter	44.87 ± 0.55	0.8514 ± 0.0029	0.4445 ± 0.0059	26.74 ± 1.89
		LoRA	47.74 ± 0.38	$\mathbf{0.8752 \pm 0.0015}$	$\mathbf{0.4750 \pm 0.0032}$	27.96 ± 2.61
		DANN-Gate	$\mathbf{47.79 \pm 0.40}$	0.8750 ± 0.0019	0.4733 ± 0.0036	28.12 ± 4.52
		REFINE	44.85 ± 0.38	0.8524 ± 0.0011	0.4474 ± 0.0035	$\mathbf{29.68 \pm 1.78}$
	Schematic confusion	NoTrans	45.36 ± 0.59	0.8662 ± 0.0033	0.4455 ± 0.0081	18.98 ± 7.49
		LinearProbe	53.45 ± 0.44	0.9090 ± 0.0002	0.5259 ± 0.0078	26.28 ± 6.59
		Adapter	52.67 ± 0.33	0.9089 ± 0.0008	0.5195 ± 0.0050	30.84 ± 4.96
		Distill	46.01 ± 1.11	0.8736 ± 0.0028	0.4435 ± 0.0143	14.00 ± 6.94
		LoRA	52.35 ± 0.42	0.9024 ± 0.0008	0.5176 ± 0.0053	32.50 ± 0.97
		DANN-Gate	52.13 ± 0.35	0.9021 ± 0.0009	0.5141 ± 0.0036	33.28 ± 4.33
		REFINE	$\mathbf{54.62 \pm 0.45}$	$\mathbf{0.9134 \pm 0.0010}$	$\mathbf{0.5431 \pm 0.0056}$	$\mathbf{33.90 \pm 3.34}$
	Class imbalanace	NoTrans	45.36 ± 1.39	0.8678 ± 0.0028	0.4391 ± 0.0183	16.24 ± 4.52
		LinearProbe	48.44 ± 0.37	0.8749 ± 0.0008	0.4805 ± 0.0052	25.94 ± 6.98
		Adapter	47.57 ± 0.27	0.8678 ± 0.0029	0.4689 ± 0.0045	25.26 ± 4.53
		Distill	42.25 ± 0.63	0.8650 ± 0.0035	0.3996 ± 0.0051	3.86 ± 0.82
		LoRA	$\mathbf{48.99 \pm 0.30}$	0.8759 ± 0.0007	$\mathbf{0.4866 \pm 0.0036}$	30.92 ± 3.71
		DANN-Gate	48.94 ± 0.41	$\mathbf{0.8766 \pm 0.0009}$	0.4860 ± 0.0051	$\mathbf{31.62 \pm 1.52}$
		REFINE	47.81 ± 0.23	0.8691 ± 0.0007	0.4755 ± 0.0026	29.44 ± 3.26

Table S2: Single-source transfer learning with label noise, semantic perturbation, and class imbalance for CIFAR-10 using transformers.

or slightly improves performance. Overall, these results show that REFINE is effective on tabular data, and offers a safe and reliable mechanism for leveraging additional data compared to direct augmentation.

Dataset	Setting	Method	Acc	AUC	F1	MinCAcc
CIFAR-100	80% flips	NoTrans	15.32 ± 0.33	$\mathbf{0.8449 \pm 0.0021}$	0.1358 ± 0.0041	0.00 ± 0.00
		LinearProbe	6.70 ± 0.27	0.7377 ± 0.0011	0.0390 ± 0.0014	0.00 ± 0.00
		Adapter	6.54 ± 0.16	0.7405 ± 0.0011	0.0348 ± 0.0009	0.00 ± 0.00
		Distill	11.83 ± 0.26	0.8130 ± 0.0027	0.0835 ± 0.0024	0.00 ± 0.00
		LoRA	6.97 ± 0.07	0.7390 ± 0.0015	0.0428 ± 0.0014	0.00 ± 0.00
		DANN-Gate	6.91 ± 0.23	0.7392 ± 0.0016	0.0429 ± 0.0014	0.00 ± 0.00
		REFINE	$\mathbf{15.50 \pm 0.79}$	0.8437 ± 0.0041	$\mathbf{0.1378 \pm 0.0067}$	0.00 ± 0.00
	Domain mismatch	NoTrans	11.28 ± 0.52	0.8023 ± 0.0034	0.0984 ± 0.0033	0.00 ± 0.00
		LinearProbe	13.32 ± 0.52	0.8186 ± 0.0015	0.1175 ± 0.0049	0.00 ± 0.00
		Adapter	12.64 ± 0.32	0.8267 ± 0.0006	0.1052 ± 0.0030	0.00 ± 0.00
		LoRA	14.22 ± 0.26	0.8466 ± 0.0010	0.1289 ± 0.0028	0.00 ± 0.00
		DANN-Gate	14.08 ± 0.37	0.8465 ± 0.0012	0.1280 ± 0.0023	0.00 ± 0.00
		REFINE	$\mathbf{14.38 \pm 0.54}$	$\mathbf{0.8291 \pm 0.0032}$	$\mathbf{0.1329 \pm 0.0039}$	0.00 ± 0.00
	Schematic confusion	NoTrans	$\mathbf{16.24 \pm 0.58}$	$\mathbf{0.8471 \pm 0.0036}$	$\mathbf{0.1485 \pm 0.0075}$	0.00 ± 0.00
		LinearProbe	11.88 ± 0.28	0.7950 ± 0.0016	0.1067 ± 0.0015	0.00 ± 0.00
		Adapter	11.17 ± 0.43	0.7936 ± 0.0027	0.0918 ± 0.0040	0.00 ± 0.00
		Distill	15.01 ± 0.64	0.8266 ± 0.0028	0.1260 ± 0.0081	0.00 ± 0.00
		LoRA	11.36 ± 0.18	0.7899 ± 0.0013	0.0991 ± 0.0015	0.00 ± 0.00
		DANN-Gate	11.46 ± 0.21	0.7893 ± 0.0013	0.0989 ± 0.0017	0.00 ± 0.00
		REFINE	14.94 ± 0.49	0.8282 ± 0.0026	0.1402 ± 0.0026	0.00 ± 0.00
	Class imbalance	NoTrans	15.43 ± 0.32	0.8474 ± 0.0025	0.1386 ± 0.0012	0.00 ± 0.00
		LinearProbe	$\mathbf{25.82 \pm 0.28}$	0.8877 ± 0.0010	$\mathbf{0.2529 \pm 0.0020}$	3.60 ± 0.80
		Adapter	24.48 ± 0.32	0.8847 ± 0.0010	0.2320 ± 0.0027	0.60 ± 0.80
		Distill	16.01 ± 0.13	0.8721 ± 0.0017	0.1252 ± 0.0021	0.00 ± 0.00
		LoRA	23.52 ± 0.09	0.8669 ± 0.0015	0.2250 ± 0.0023	0.00 ± 0.00
		DANN-Gate	23.48 ± 0.13	0.8671 ± 0.0018	0.2264 ± 0.0019	0.00 ± 0.00
		REFINE	25.54 ± 0.43	$\mathbf{0.8879 \pm 0.0013}$	0.2524 ± 0.0039	$\mathbf{4.80 \pm 0.75}$

Table S3: Single-source transfer learning with label noise, semantic perturbation, and class imbalance for CIFAR-100 using transformers.

Dataset	Metric	Classifier					
		MLP1			MLP2		
		Raw	DirectAug	REFINE	Raw	DirectAug	REFINE
Adult	Accuracy	0.807 ± 0.008	0.831 ± 0.006	0.821 ± 0.004	0.800 ± 0.011	0.833 ± 0.005	0.814 ± 0.010
	AUC	0.832 ± 0.008	0.878 ± 0.006	0.852 ± 0.008	0.833 ± 0.010	0.883 ± 0.005	0.854 ± 0.008
	F1	0.547 ± 0.037	0.619 ± 0.015	0.595 ± 0.030	0.570 ± 0.028	0.627 ± 0.021	0.612 ± 0.021
Credit	Accuracy	0.723 ± 0.028	0.735 ± 0.017	0.740 ± 0.022	0.717 ± 0.027	0.732 ± 0.015	0.726 ± 0.020
	AUC	0.730 ± 0.024	0.738 ± 0.013	0.745 ± 0.018	0.725 ± 0.025	0.754 ± 0.020	0.736 ± 0.023
	F1	0.490 ± 0.043	0.524 ± 0.030	0.520 ± 0.038	0.515 ± 0.041	0.541 ± 0.030	0.535 ± 0.037
Diabetes	Accuracy	0.565 ± 0.015	0.573 ± 0.008	0.571 ± 0.008	0.561 ± 0.015	0.596 ± 0.007	0.572 ± 0.010
	AUC	0.582 ± 0.019	0.597 ± 0.008	0.591 ± 0.012	0.576 ± 0.018	0.626 ± 0.008	0.593 ± 0.013
	F1	0.505 ± 0.028	0.533 ± 0.014	0.523 ± 0.022	0.501 ± 0.029	0.534 ± 0.017	0.522 ± 0.027
Performance	Accuracy	0.684 ± 0.019	0.724 ± 0.011	0.711 ± 0.014	0.683 ± 0.018	0.668 ± 0.084	0.702 ± 0.022
	AUC	0.857 ± 0.011	0.878 ± 0.009	0.869 ± 0.009	0.858 ± 0.011	0.830 ± 0.070	0.865 ± 0.011
	F1	0.478 ± 0.025	0.557 ± 0.024	0.521 ± 0.029	0.478 ± 0.027	0.494 ± 0.090	0.507 ± 0.035

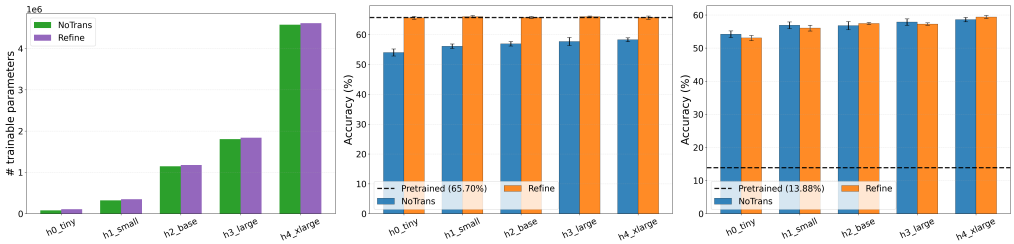
Table S4: Single-source transfer learning with original tabular data.

C.3 AN ABLATION STUDY

We conduct an ablation study to investigate the effect of complexity of the encoder h in REFINE, by varying the width and depth of the neural network models used. Figure S1 reports the performance of REFINE under five different models with increasing complexity for h. The left panel reports the total number of trainable parameters, the middle panel reports the classification accuracy using the original data, and the right panel using the noisy data. On the original data, REFINE consistently outperforms NoTrans across all levels of complexity by a considerable margin, demonstrating its ability to leverage useful pretrained features. On the noisy data, REFINE performs on par with NoTrans regardless of the complexity of h, confirming its robustness to negative transfer. Together, these results show that REFINE offers robust and reliable safeguarding against negative transfer.

Dataset	Metric	Classifier					
		MLP1			MLP2		
		Raw	DirectAug	REFINE	Raw	DirectAug	REFINE
Adult	Accuracy	0.808 ± 0.007	0.615 ± 0.046	0.805 ± 0.008	0.800 ± 0.010	0.641 ± 0.052	0.791 ± 0.016
	AUC	0.834 ± 0.009	0.612 ± 0.051	0.832 ± 0.010	0.834 ± 0.013	0.639 ± 0.052	0.828 ± 0.014
	F1	0.549 ± 0.046	0.383 ± 0.039	0.555 ± 0.029	0.564 ± 0.032	0.395 ± 0.047	0.562 ± 0.027
Credit	Accuracy	0.723 ± 0.027	0.581 ± 0.035	0.705 ± 0.028	0.716 ± 0.027	0.599 ± 0.045	0.705 ± 0.028
	AUC	0.728 ± 0.027	0.578 ± 0.045	0.705 ± 0.028	0.720 ± 0.026	0.599 ± 0.045	0.687 ± 0.028
	F1	0.483 ± 0.049	0.417 ± 0.048	0.481 ± 0.035	0.512 ± 0.041	0.433 ± 0.041	0.493 ± 0.034
Diabetes	Accuracy	0.587 ± 0.007	0.516 ± 0.007	0.575 ± 0.006	0.614 ± 0.006	0.551 ± 0.016	0.609 ± 0.004
	AUC	0.580 ± 0.020	0.516 ± 0.014	0.554 ± 0.016	0.577 ± 0.017	0.585 ± 0.019	0.567 ± 0.020
	F1	0.503 ± 0.032	0.489 ± 0.025	0.498 ± 0.022	0.503 ± 0.026	0.483 ± 0.037	0.514 ± 0.025
Performance	Accuracy	0.682 ± 0.020	0.637 ± 0.088	0.696 ± 0.023	0.684 ± 0.018	0.650 ± 0.079	0.696 ± 0.023
	AUC	0.857 ± 0.011	0.805 ± 0.074	0.862 ± 0.011	0.859 ± 0.010	0.814 ± 0.068	0.863 ± 0.012
	F1	0.476 ± 0.028	0.464 ± 0.096	0.499 ± 0.036	0.480 ± 0.029	0.472 ± 0.088	0.500 ± 0.035

Table S5: Single-source transfer learning with noisy tabular data.



Parameter Number | Accuracy (Original data) | Accuracy (Heavy noise)

Figure S1: Ablation study for the encoder h with varying complexity.

D MORE DETAILS ON EXPERIMENT SETUP AND IMPLEMENTATIONS

We provide additional details on experiment setup and implementations for better reproducibility. All experiments are conducted on an NVIDIA A10G (Ampere) GPU with 23 GB of GDDR6 memory, driver version 535.183.01, and CUDA 12.2. For semantic confusion in CIFAR-10 and CIFAR-100, we construct 4 and 47 pairs of related classes, respectively, and flip 50% of each pair's samples to its counterpart, while also injecting white noise into image attributes with $\sigma = 0.2$. For class imbalance, we create each imbalanced pretrained subset by first sampling 10,000 images from the full training split with a fixed seed (42). In CIFAR-10, classes 0-9 are sampled with proportions $[0.35, 0.30, 0.10, 0.07, 0.06, 0.045, 0.03, 0.02, 0.015, 0.01]$, yielding 3,500 to 100 images per class. In CIFAR-100, the first 10 classes are designated as majority, with 400 images each, and the remaining 90 as minority, with 100 images each, truncated to a total of 10,000 samples. Table S6 summarizes the experiment settings.

Dataset	Pretrain Model	Base Model	Pretrain Size	Fine-tune Size	Adapter Para (%)	REFINE Para (%)
CIFAR-CNN-related	CNN	CNN	10000	4000	5.46	4.88
CIFAR-TF-related	Transformer	Transformer	10000	4000	6.49	4.63
CIFAR-10→STL	CNN	CNN	10000	4000	5.46	4.88
Clipart→Sketch	ResNet18	ResNet10	3000	1000	1.36	44.2
USPS→MNIST	CNN	CNN	5000	100	5.46	4.88
Books→Kitchen	Transformer	Transformer	2000	400	2.25	96.58
DVD→Electronics	Transformer	Transformer	2000	400	2.25	96.58

Table S6: Experiment settings for all data examples.