WEIGHTED DEEP ENSEMBLE UNDER MISSPECIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks are supported by the universal approximation theorem, which guarantees that sufficiently large architectures can approximate smooth functions. In practice, however, this guarantee holds only under restrictive conditions, and violations of these conditions give rise to model misspecification. We categorize such misspecification into three sources: variable misspecification, arising from insufficiently informative features; structural misspecification, stemming from the limited width and depth of networks that cannot fully capture the underlying complexity; and inherent misspecification, occurring when the true model possesses properties such as discontinuities that cannot be faithfully represented. To mitigate the impact of these forms of misspecification, ensemble methods have become a common strategy for enhancing predictive performance. However, standard ensembles composed of identically architected and equally weighted models may suffer from "collective blindness", where shared errors are amplified and lead to systematically biased predictions with high confidence. To mitigate this issue, we introduce weighted deep ensemble method that learns the optimal weights. We prove that our method provably attains the convergence rate of the best single model in the ensemble and asymptotically achieves oracle-level predictive risk. To the best of our knowledge, this is the first work to provide rigorous theoretical guarantees for weighted deep ensemble under both well-specified and misspecified settings.

1 INTRODUCTION

Model misspecification in statistics arises from the omission of relevant variables, inclusion of irrelevant variables, incorrect functional forms and incorrect distributional assumptions (Maasoumi, 1990; White, 1982). When a model suffers from such misspecification, the best possible approximation $f^* \in \mathcal{F}$, with $\mathcal{F}$ denoting the function class used for estimation, may incur a significant approximation error from the true function f_0. In deep learning, the neural networks are always assumed to be well-specified (Barron, 1994; Elbrächter et al., 2019). As shown in the universal approximation theorem, sufficiently large neural networks have the ability to approximate any continuous function, which in principle allows the approximation error $\|f_0 - f^*\|$ to approach zero (Hornik et al., 1989; Park et al., 2020; Lu et al., 2021). Therefore, existing studies always focus on overcoming challenges in optimization and estimation errors (Barron, 1994; Soltanolkotabi et al., 2018; Adcock & Dexter, 2021).

However, the assumption that neural networks are well-specified is frequently violated in practice, as model misspecification is common. Unlike in traditional statistics, misspecification in deep learning manifests in several distinct forms. First, it may arise from an information deficit, where the input features and their latent representations lack the necessary information to capture the true data-generating process. Second, practical constraints on network depth and width impose finite capacity, leading to non-negligible approximation error when the true function is highly complex. Finally, misspecification can occur when the true function has properties such as discontinuities, which cannot be exactly represented by neural networks and can only be approximated with non-vanishing error at the discontinuity points.

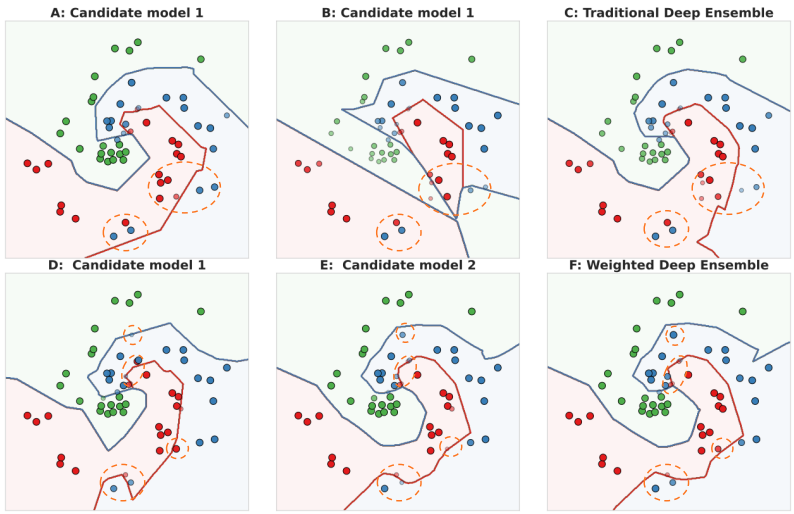


Figure 1: Comparison of the decision boundaries and confidence levels of different ensemble methods, with darker shading indicating higher confidence. The Traditional Deep Ensemble (C) shows "collective blindness" by having lower confidence in correctly classified areas but higher confidence in the misclassified areas, e.g., blue points within the red region. Weighted Deep Ensemble (F) corrects these errors by maintaining high confidence in correct areas while showing low confidence at uncertain boundaries. The key difference is highlighted in the circled area.

"All models are wrong, but some are useful (Box, 1976)." Ensemble methods are the most intuitive method to leverage the useful parts of multiple wrong models (Fort et al., 2019; Huang et al., 2024). However, it still raises a critical question: if a neural network model is misspecified, can the estimators from an ensemble of such models still be trusted? We find that traditional deep ensembles, which consist of models with identical architectures (Hansen & Salamon, 2002; Lakshminarayanan et al., 2017; Zhang et al., 2020; Schweighofer et al., 2024), tend to learn in highly correlated ways when faced with the same misspecification. By deviating in the same incorrect direction, they produce an adverse effect we term "collective blindness". This phenomenon is caused by the ensemble reinforcing, rather than correcting, the biases shared by all members. As illustrated in Figure 1 (C), in the orange circle, the traditional deep ensemble produces misclassified predictions with high confidence. The root of the problem is that traditional ensemble methods not only employ similar model architectures but, more critically, typically aggregate predictions using equal weights. When all models are plagued by the same misspecification, this simplistic averaging only serves to amplify their shared error. To address this, we propose a novel and effective solution: an optimally weighted deep ensemble built upon architectural diversity. By ensuring that exploitable differences exist among the models, we can theoretically derive data-driven weights that minimize the ensemble's prediction error on a held-out validation set. As illustrated in Figure 1 (F), the use of optimal weights helps the ensemble aggregate complementary strengths of its constituent models and produce more accurate predictions than relying on a single misspecified model. Our primary contributions are as follows:

(1) We are the first to systematically define and categorize misspecification in deep learning into variable, structural, and inherent forms. We propose weighted deep ensemble to mitigate the "collective blindness" effect seen in traditional ensembles and provide a theoretical guarantee for our weighted deep ensemble for well-specified and misspecified models.

(2) We establish an asymptotic error bound for the weighted deep ensemble estimator and show that the bound converges at the same rate as the smallest error bound among all candidate networks. This guarantees that the ensemble inherits the speed of the best individual model, so including slower or misspecified models can never slow it down, while any rapidly converging model immediately improves the overall rate. Furthermore, we provide detailed analyses for common networks such as MLP, CNN, and RNN.

(3) We prove that the weight vector yields a prediction risk that converges to the oracle minimum, even though the oracle weight itself depends on unknown population quantities and cannot be computed. Thus the proposed weighted choice method recovers the infeasible optimal weight asymptotically, giving the first rigorous guarantee that weighted deep ensemble can attain oracle-level accuracy

using only observable data. To the best of our knowledge, this is the first study to offer a theoretical guarantee for weighted deep ensemble.

2 Related work

Misspecifiction. Model misspecification, which occurs when a chosen model fails to accurately capture the true data-generating process, is a common challenge in statistics (McGuirk et al., 1993; Cerreia-Vioglio et al., 2025). Misspecification is categorized into several types: omitted varaible bias, where the exclusion of a relevant variables leads to biased and inconsistent parameter estimates (Gospodinov & Maasoumi, 2021), incorrect functional form, such as assuming a linear relationship when the true function is nonlinear (Gerds & Schumacher, 2001; Kasparis, 2011), and mismatch distribution when the assumed probability distribution for the error term or the response variable in models is incorrect (Masiha et al., 2021; Kuang et al., 2020). These types of misspecification always degrade model predictive performance (Lanzani, 2025). In deep learning, the universal approximation theorem states that a sufficiently large neural network can approximate any continuous function (Raghu et al., 2017; Kratsios et al., 2021). Therefore, previous research always assumed that deep models are well-specified. However, misspecification is a widespread issue in practice due to limited information and finite model capacity of neural networks with limited width and depth. Our work is the first to provide a clear definition for misspecification in deep learning and theoretical guarantees for deep ensembles under misspecified conditions.

Ensemble Learning. Deep ensembles, typically composed of identical architectures with different random initialization, have been shown to outperform single deep learning models in terms of accuracy (Lakshminarayanan et al., 2017; Mohammed & Kora, 2023). However, relying solely on the same model structure may limit the effectiveness of the ensemble. To address it, several works have introduced greater diversity by varying neural network architectures (Zhang et al., 2020) and training methods (Gontijo-Lopes et al., 2022). Recent studies have begun exploring weighted deep ensembles (Kim et al., 2018; Matena & Raffel, 2022), but these works provide only empirical evidence. Existing theoretical results for traditional ensemble learning (Wolpert, 1992; Van der Laan et al., 2007) primarily derive from classical settings, where convergence rates are often slower than $n^{-1/2}$ (Stone, 1982; Schmidt-Hieber, 2020) and typically apply to linear models rather than deep neural networks. Moreover, the existing diversity literature shows that uniformly averaged ensembles achieve a risk no worse than the average individual risk, but this line of work does not compare the ensemble to the best individual model and only focus on equal weighting (Zhang et al., 2020; Wood et al., 2023; Abe et al., 2022). PAC-Bayesian theory further provides generalization bounds for deep ensembles (Masegosa et al., 2020; Ortega et al., 2022). However, they fundamentally require the prior distribution that cannot be learned from the data and cannot identify optimal weights. In contrast, we extend validation-based weighting in traditional stacking methods to deep neural networks and establish the first asymptotic optimality result for weighted deep ensembles. We show that, using only observable validation data, a properly weighted deep ensemble can asymptotically achieve oracle-level predictive accuracy.

3 Methodology

3.1 Problem Setup

We consider a general model where the input features $\boldsymbol{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$ and the output Y can be either real-valued or categorical. In regression tasks, $Y \in \mathbb{R}$ and follows the model $Y = f_0(\boldsymbol{X}) + \varepsilon$, where f_0 is an unknown true function and ε is a noise term satisfying $\mathbb{E}(\varepsilon|\boldsymbol{X}) = 0$. In classification tasks with C classes, $\boldsymbol{Y} = (Y_1, \ldots, Y_C)^\top \in \{0,1\}^C$ is the one-hot vector, where only one entry is 1 indicating the true class and all others are 0. The conditional probability of $\boldsymbol{Y}$ is modeled as $P(\boldsymbol{Y} \mid \boldsymbol{X}) = f_0(\boldsymbol{X})$ for $c = 1, \ldots, C$, where $f_0(\boldsymbol{X}) = (f_{0,1}(\boldsymbol{X}), \ldots, f_{0,C}(\boldsymbol{X}))^\top$ and $f_{0,c}(\boldsymbol{X}) = P(Y_c = 1 \mid \boldsymbol{X})$. We assume that n independent observable samples $(\boldsymbol{X}_i, Y_i)$ are drawn from a joint distribution over $(\boldsymbol{X}, Y)$. The supremum norm is defined as $\|f\|_\infty = \sup_{\boldsymbol{X}} |f(\boldsymbol{X})|$, while the L^2 norm is $\|f\|_{L^2} = (\int |f(\boldsymbol{X})|^2 \, dP_{\boldsymbol{X}}(\boldsymbol{X}))^{1/2}$.

Model training. The observable data is split into two parts, a training set with size n_{train} for training the neural network and the other $n_{\text{val}} = n - n_{\text{train}}$ for choosing weights. Specifically, a base model $\widehat{f}$

is obtained by empirical risk minimization: $\widehat{f} = \arg\min_{f \in \mathcal{F}} 1/n_{\text{train}} \sum_{i=1}^{n_{\text{train}}} \ell(f(\boldsymbol{X}_i), Y_i)$, where $\mathcal{F}$ denotes the model function class and ℓ is the loss function. And f^* is defined as the minimizer of the true expected risk: $f^* = \arg\min_{f \in \mathcal{F}} \mathbb{E}\left[\ell(f(\boldsymbol{X}), Y)\right]$.

3.2 Misspecification in deep learning

In this section, we systematically define three types of misspecification in deep learning. Let $f_0 : \mathcal{X}_{\text{true}} \to \mathcal{Y}$ denote the true data-generating function.

Definition 1 (Variable Misspecification)**.** *Let $\mathcal{X}_{\text{model}}$ be the feature space available to a given model. Define the projection map $\pi : \mathcal{X}_{\text{true}} \to \mathcal{X}_{\text{model}}$ that restricts each $x \in \mathcal{X}_{\text{true}}$ to its coordinates in $\mathcal{X}_{\text{model}}$. We say that the model exhibits **variable misspecification** if f_0 cannot be expressed as a composition of π with a function on $\mathcal{X}_{\text{model}}$. Formally,*

$$\nexists g : \mathcal{X}_{\text{model}} \to \mathcal{Y} \quad \textit{such that} \quad f_0(x) = g(\pi(\boldsymbol{X})) \quad \textit{for almost every } \boldsymbol{X} \in \mathcal{X}_{\text{true}}.$$

Example. Consider an image classification task where the true label depends on latent features $Z = (Z_1, Z_2)$ (e.g., ear shape and nose shape). The true function is $f_0(Z) = \mathbb{I}_{Z_1 + Z_2 > 0}$ (e.g., predicting cat if positive, dog if negative). If the model's feature space $\mathcal{X}_{\text{model}}$ corresponds only to Z_2 due to missing feature input or incomplete feature extraction, the model suffers from variable misspecification.

Definition 2 (Structural Misspecification)**.** *Let $\mathcal{H}$ be the function class corresponding to a given neural network architecture. Let $\mathcal{L}$ be a loss function and define the risk*

$$R(h) = \mathbb{E}_{\boldsymbol{X} \sim P_{\boldsymbol{X}}}\left[\mathcal{L}\big(h(\boldsymbol{X}), f_0(\boldsymbol{X})\big)\right].$$

*For a tolerance level $\delta > 0$, we say that the architecture exhibits **structural misspecification** if the minimal achievable risk within $\mathcal{H}$ exceeds this tolerance. Formally,*

$$\inf_{h \in \mathcal{H}} R(h) \; > \; \delta.$$

Example. Consider ReLU neural networks with both depth and width restricted to the order of $\log(n)$. For β-smooth functions in dimension d, it is known that the approximation error in this class satisfies

$$\inf_{h \in \mathcal{H}} R(h) \; = \; c \, \log(n)^{-4\beta/d},$$

for some constant $c > 0$ depending only on β and d (Jiao et al., 2023). We set the tolerance level to $\delta = n^{-1/4}$, which corresponds to the rate condition commonly required in double machine learning (Chernozhukov et al., 2018). For sufficiently large n, we have

$$\inf_{h \in \mathcal{H}} R(h) \; = \; c \, \log(n)^{-4\beta/d} \; > \; \delta,$$

so the architecture is structurally misspecified relative to the tolerance δ.

Definition 3 (Inherent Misspecification)**.** *Let $\mathcal{H}$ be the function class corresponding to a given neural network architecture. Approximation results for neural networks typically require f_0 to belong to a smoothness class, such as a Hölder or Sobolev space, in order to guarantee vanishing L^2 approximation error. We say that the architecture exhibits **inherent misspecification** if f_0 does not satisfy the required smoothness conditions, so that the minimal achievable L^2 error is bounded away from zero. Formally, for some tolerance level $\delta > 0$,*

$$\inf_{h \in \mathcal{H}} \|h - f_0\|_{L^2} > \delta.$$

Example. Consider the Dirichlet function

$$f_0(x) = \begin{cases} 1, & x \in \mathbb{Q} \cap [0,1], \\ 0, & x \in (\mathbb{R} \setminus \mathbb{Q}) \cap [0,1]. \end{cases}$$

This function is nowhere continuous and does not belong to any Hölder or Sobolev class. As a result, neural networks cannot approximate f_0 in L^2 with vanishing error, and the approximation gap remains strictly positive. Therefore, the model class $\mathcal{H}$ suffers from inherent misspecification.

3.3 WEIGHTED DEEP ENSEMBLE

First, we introduce the standard deep ensemble method.

Deep ensembles. A standard *deep ensemble* consists of M candidate models $\widehat{f}_1(\cdot), \ldots, \widehat{f}_M(\cdot)$. The *deep ensemble prediction* $\bar{f}(\boldsymbol{X})$ is: $\bar{f}(\boldsymbol{X}) = \sum_{m=1}^{M} w_m \widehat{f}_m(\boldsymbol{X})$, where all weights are set equally as $w_m = 1/M$. Typically, the candidate models share the same neural network architecture and are trained independently on the same dataset using the same loss function $\ell(f_m(\boldsymbol{X}), Y)$, which corresponds to minimizing the empirical risk: $\mathcal{L}_{\text{avg}} = \mathbb{E}\left[\ell(\bar{f}(\boldsymbol{X}), Y)\right]$.

As stated before, traditional deep ensembles may suffer from "collective blindness" in the presence of variable, structural, or inherent misspecification. This motivates us to apply a more flexible weighting scheme.

Weighted deep ensemble. In this paper, we propose a method called *Weighted Deep Ensemble (WDE)*, which combines multiple neural networks with different structures into a single predictive model. Let $\boldsymbol{w} = (w_1, \ldots, w_M)^{\mathsf{T}}$ be the vector of ensemble weights. We restrict the weights to the simplex $\mathcal{W} = \{\boldsymbol{w} \in [0,1]^M \ : \ \sum_{m=1}^{M} w_m = 1\}$. The *weighted deep ensemble prediction* is defined as $\widehat{f}(\boldsymbol{X}; \boldsymbol{w}) = \sum_{m=1}^{M} w_m \widehat{f}_m(\boldsymbol{X})$. In practice, the weight vector $\boldsymbol{w}$ is unknown and need to be estimated from observable data.

Weight choice criterion. We adopt a *validation-risk minimization* (VRM) criterion: the estimator $\widehat{\boldsymbol{w}}$ is chosen to minimize the empirical loss on a validation set, under the simplex constraints

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w} \in \mathcal{W}}{\arg\min} \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \ell(\hat{f}(\boldsymbol{X}_i; \boldsymbol{w}), Y_i).$$

This VRM strategy links the weighting scheme directly to out-of-sample performance, introducing no extra hyperparameters beyond the standard validation split. Specifically, we consider two general tasks: (i) Regression: with the squared loss $\ell_{\text{reg}}\big(f(\boldsymbol{X}; \boldsymbol{w}), Y\big) = \big(f(\boldsymbol{X}; \boldsymbol{w}) - Y\big)^2$, the VRM objective reduces to a strictly convex quadratic program on the simplex $\mathcal{W}$ (Li et al., 2023; Qu et al., 2025). (ii) Classification: Using cross-entropy loss $\ell_{\text{class}}\big(f(\boldsymbol{X}; \boldsymbol{w}), \boldsymbol{Y}\big) = -\boldsymbol{Y}^\top \log(f(\boldsymbol{X}; \boldsymbol{w}))$, where $\boldsymbol{Y} \in \{0,1\}^C$ denotes the one-hot encoding of the true class, and $f(\boldsymbol{X}; \boldsymbol{w})$ denotes the predicted probability. Since the mapping $\boldsymbol{w} \mapsto f(\boldsymbol{X}; \boldsymbol{w})$ is affine and $-\log(\cdot)$ is convex, the loss function remains convex in $\boldsymbol{w}$ and can be optimized using projected-gradient (Boyd & Vandenberghe, 2004) methods. So we construct the *weighted deep ensemble estimator* $\widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}}) = \sum_{m=1}^{M} \widehat{w}_m \widehat{f}_m(\boldsymbol{X})$. Due to the convexity of the loss function with respect to $\boldsymbol{w}$, we can obtain the global optimal solution for the ensemble weights.

3.4 ASYMPTOTIC ERROR BOUNDS

Our goal is to establish the asymptotic error bound of the proposed weighted deep ensemble estimator. Specifically, we aim to show that our proposed weighted deep ensemble effectively combines multiple small neural networks to achieve an asymptotic error bound at least as fast as that of the best candidate. By appropriately combining these models, the estimator adapts to various data structures and retains the ability to capture intricate features without resorting to a large, monolithic network. Consequently, the weighted deep ensemble benefits from more flexible modeling choices while still maintaining an asymptotic error bound that is as fast as the best candidate. Importantly, the presence of misspecified or poorly performing candidate models does not slow down the convergence rate of the estimator. We provide theoretical guarantees for both regression (Theorem 1) and classification (Theorem 2) settings. To establish these results, we begin by introducing the following condition:

Condition 1. *(i). There exists a positive constant C such that $\|f_0(\boldsymbol{X})\|_\infty < C$, $\|\widehat{f}_m(\boldsymbol{X})\|_\infty < C$ for $m = 1, \ldots, M$; (ii). $\mathbb{E}(\varepsilon|\boldsymbol{X}) = 0$, and ε is sub-Gaussian with parameter σ.*

This condition restricts the upper bound of f_0, f^*, and that ε has mean zero and sub-Gaussian tails. This condition is also widely used in the literature; see Schmidt-Hieber (2020) and Jiao et al. (2023) for example.

Theorem 1. *Suppose Condition 1 holds and assume that the candidate model with the fastest convergence rate has an asymptotic error bound of order S, i.e., the candidate model with the fastest*

asymptotic error bound (denoted as $\tilde{f}$ without loss of generality) satisfies $\|f_0(\boldsymbol{X}) - \widetilde{f}(\boldsymbol{X})\|_{L^2} = O_p(S)$, then our weighted deep ensemble estimator can also achieve this rate asymptotically:

$$\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}})\|_{L^2} = O_p(S + \sqrt{\frac{\log(M)}{n}}).$$

Besides the asymptotic error bound S, the result also includes an additional term of order $\sqrt{\log(M)/n}$. In practice, the minimax rate for nonparametric methods such as neural networks is typically of order $n^{-\beta/(2\beta+d)}$ for some smoothness β and input dimension d (Stone, 1982), which is slower than $\sqrt{\log(M)/n}$ when the number of candidate model is fixed; hence the overall asymptotic error bound is dominated by the nonparametric estimation error. Therefore, Theorem 1 establishes that the asymptotic error bound of the weighted deep ensemble estimator is no worse than that of the best individual candidate model. In other words, regardless of which candidate model achieves the smallest asymptotic error bound, the ensemble procedure guarantees at least comparable asymptotic performance and will not converge more slowly than that benchmark.

Theorem 1 presents the asymptotic error bound of the deep ensemble estimator under regression tasks. In fact, we can establish similar properties for classification tasks. Before presenting the theorem, we define $R_{0/1}(f) = \mathbb{E}\{I(\arg\max_c Y_c \neq \arg\max_c f_c(\boldsymbol{X}))\} - \mathbb{E}\{I(\arg\max_c Y_c \neq \arg\max_c f_{0,c}(\boldsymbol{X}))\}$ to be the excess misclassification rate.

Theorem 2. *Suppose $f_0(\boldsymbol{X})$ is uniformly bounded away from 0 and 1 and assume that the best candidate model has a misclassification rate of S, i.e., the candidate model with the smallest misclassification rate (denoted as $\tilde{f}$ without loss of generality) satisfies $R_{0/1}(\tilde{f}) = O_p(S)$, then our weighted deep ensemble estimator can also achieve this misclassification rate asymptotically:*

$$R_{0/1}(\widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}})) = O_p(S + \sqrt{\frac{\log(M)}{n}}).$$

Theorems 1 and 2 show that, in both regression and classification tasks, the asymptotic error bound of our weighted deep ensemble estimator matches the asymptotic error bound of the best single candidate model. Detailed proofs are provided in the Appendix.

In addition, when the pool of candidate models is altered, the estimator's attainable asymptotic error bound necessarily shifts in response to the new composition. For better understanding, the next three corollaries provide the asymptotic error bound of the weighted deep ensemble estimator when the candidate models are chosen from MLP-based networks, CNN-based networks, and RNN-based networks. It is worth noting that Theorem 1 holds under the listed moment conditions, but additional assumptions are implicitly embedded in the asymptotic error bound $O_p(S)$. Since different neural network architectures require different conditions to guarantee convergence, we do not enumerate them explicitly here and instead present the result in terms of the general order $O_p(S)$. Accordingly, the subsequent corollaries impose further conditions on the candidate models, ensuring that they can indeed attain the corresponding asymptotic error bound in those specific settings. Define $\widetilde{O}_p(\cdot)$ as the rate by ignoring logarithmic factors.

Corollary 1 (MLP case)**.** *If all $M = O(1)$ candidate models are MLP-based models, f_0 is β-Hölder smooth with $\beta > 1$, and Conditions of Theorem 4.2 in Jiao et al. (2023) holds, then with some specifically designed candidate models, the weighted deep ensemble estimators can achieve asymptotic error bound of*

$$\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}})\|_{L^2}^2 = \widetilde{O}_p(n^{-2\beta/(d+2\beta)}).$$

Corollary 2 (CNN case)**.** *If all $M = O(1)$ candidate models are CNN-based models, f_0 is β-Hölder smooth with $\beta > 1$, and Conditions of Theorem 4.6 in Shen et al. (2022) holds, then with some specifically designed candidate models, the weighted deep ensemble estimators can achieve asymptotic error bound of*

$$\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}})\|_{L^2}^2 = \widetilde{O}_p(n^{-2\beta/(d+2\beta)}).$$

Corollary 3 (RNN case)**.** *If all $M = O(1)$ candidate models are RNN-based models, f_0 is β-Hölder smooth with $\beta > 1$ and Conditions of Theorem 6 in Jiao et al. (2024) holds, then with some*

specifically designed candidate models, the weighted deep ensemble estimator can achieve asymptotic error bound of

$$\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}})\|_{L^2}^2 = \widetilde{O}_p(n^{-2\beta/(dl+2\beta)}),$$

where l is the length of the input sequence, i.e., the number of time steps processed by the RNN.

These results can also be extended to ResNet, Transformer, and other network structures. As long as we have the asymptotic error bound of a single network, Theorems 1 and 2 ensure that the weighted deep ensemble estimator can achieve the same asymptotic error bound as the fastest candidate model.

3.5 Asymptotic Optimality under Model Misspecification

Unlike traditional ensembles with uniform weights, our method learns data-dependent weights, ensuring that the ensemble performs at least as well as the best candidate asymptotically. Since equal weights lie within our admissible weight space, our approach is guaranteed to match or exceed the performance of equal-weighted averaging asymptotically. In practice, the oracle weight vector is unknown and cannot be directly computed from observable data. By instead using the VRM criterion, our method significantly reduces computational cost while maintaining competitive efficiency, making it particularly suitable for large-scale scenarios. In the following, we present theoretical results showing that the proposed weighted deep ensemble estimator achieves asymptotic optimality and attains oracle-level accuracy using only observable data.

Let $R(\boldsymbol{w}) = \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\|_{L^2}^2$, $R^*(\boldsymbol{w}) = \|f_0(\boldsymbol{X}) - f^*(\boldsymbol{X}; \boldsymbol{w})\|_{L^2}^2$, $\xi_n = \inf_{\boldsymbol{w}\in\mathcal{W}} R^*(\boldsymbol{w})$, and $\phi_n = \sup_{\boldsymbol{w}\in\mathcal{W}} \|\widehat{f}(\boldsymbol{X}; \boldsymbol{w}) - f^*(\boldsymbol{X}; \boldsymbol{w})\|_{L^2}$. To establish the asymptotic optimality of $\widehat{\boldsymbol{w}}$, we require the following condition.

Condition 2. *(i).* $\xi_n^{-1} n^{-1/2} M^{1/2} = o(1)$*; (ii).* $\xi_n^{-1}\phi_n = o_p(1)$.

This condition regulates the divergence speed of ξ_n, and it is frequently used in FMA research, such as Ando & Li (2014); Zhang et al. (2016). Condition 2 requires ξ_n to grow faster than $\sqrt{1/n}$ and ϕ_n. Importantly, this condition should be interpreted as a *misspecification condition*, ensuring that the oracle risk ξ_n does not vanish too quickly relative to the estimation error. It naturally aligns with the three forms of misspecification introduced above:

(1) **Variable misspecification.** When crucial variables are missing, or when the essential features of f_0 cannot be generated from the available input space, f^* cannot converge to f_0. In this case, the approximation error remains bounded away from zero, which implies that the oracle risk ξ_n does not vanish and Condition 2 (i) is satisfied. This requirement should be viewed as a stronger form of variable misspecification, as it excludes cases where the omitted variables have only negligible influence on f_0, for instance when their contribution diminishes asymptotically.

(2) **Structural misspecification.** When networks have limited depth or width, their approximation error remains bounded away from zero. In this case ξ_n decreases at a much slower rate than $n^{-1/2}$, so Condition 2 (i) is satisfied.

(3) **Inherent misspecification.** When the true function lies outside the smoothness classes typically required for neural network approximation, the minimal L^2 error remains strictly positive. Consequently, ξ_n is bounded away from zero, and Condition 2 (i) holds.

Unlike ξ_n, the term ϕ_n measures the estimation error between $\widehat{f}$ and f^*, which depends on sample size rather than model specification. Condition 2 (ii) therefore requires that ϕ_n converges to zero at a faster rate than ξ_n, ensuring that estimation error does not dominate the asymptotic behavior.

Theorem 3. *Suppose that Conditions 1 and 2 hold, then*

$$\frac{R(\widehat{\boldsymbol{w}})}{\inf_{\boldsymbol{w}\in\mathcal{W}} R(\boldsymbol{w})} \to 1$$

in probability as $n \to \infty$.

This theorem states that under Conditions 1 and 2, as the sample size n goes to infinity, the ratio of the risk of $\widehat{\boldsymbol{w}}$ to the infimum of the risk over all possible weights converges to 1. In other words, although the weight that minimizes the risk is infeasible, the proposed weight choice criterion identifies a

weight $\widehat{\boldsymbol{w}}$ whose risk becomes asymptotically equal to the minimal risk. This means our procedure performs asymptotically as well as this ideal benchmark. This provides strong theoretical support for our weight selection method, assuring that no asymptotic loss is incurred compared to the unattainable optimum. All theoretical proofs are provided in the Appendix B.

4 Numerical Results

In this section, we investigate the models that are well-specified or suffer from variable misspecification, structural misspecification, and inherent misspecification.

Baselines. We compare several ensemble strategies: (1) Deep Ensemble (DE): homogeneous MLPs with different initializations and equal weights; (2) Equal-Weight Heterogeneous (EW): heterogeneous MLPs with equal weights; (3) Our Method, WDE: heterogeneous MLPs with optimal weights.

Experimental details. Datasets are split 6:2:2 for training, validation, and testing. We use Adam with early stopping (patience=20 epochs). Hyperparameters: learning rate searched in [0.001, 0.1], batch size 128, max 5000 epochs. Our method uses 4 heterogeneous networks with total parameters matching a 4×MLP (2 hidden layers of 30 nodes) DE for fair comparison.

Table 1: Performance comparison across different complexity types with varying missing variables, with sample size 5000.

Task	# Missing	DE	EW	WDE	Δ_{DE}	Δ_{EW}
Nested	0	7.452 ± 0.319	7.711 ± 0.527	$\mathbf{7.025 \pm 0.247}$	6.07%	9.76%
	1	8.175 ± 0.472	8.686 ± 0.957	$\mathbf{8.028 \pm 1.236}$	1.84%	8.19%
	3	8.390 ± 0.681	8.825 ± 0.814	$\mathbf{8.077 \pm 1.211}$	3.88%	9.27%
	5	8.440 ± 0.788	8.493 ± 0.482	$\mathbf{7.792 \pm 0.287}$	8.31%	9.00%
	7	9.122 ± 0.498	8.687 ± 0.510	$\mathbf{7.967 \pm 0.431}$	14.50%	9.04%
Interaction	0	1.975 ± 0.330	1.854 ± 0.187	$\mathbf{1.773 \pm 0.131}$	11.35%	4.57%
	1	3.251 ± 0.740	3.640 ± 0.782	$\mathbf{2.949 \pm 0.778}$	10.22%	23.43%
	3	3.949 ± 0.991	4.434 ± 1.046	$\mathbf{3.890 \pm 0.984}$	20.76%	13.98%
	5	5.239 ± 1.149	5.223 ± 1.176	$\mathbf{5.077 \pm 1.099}$	22.36%	2.87%
	7	5.747 ± 1.206	5.617 ± 1.266	$\mathbf{5.548 \pm 1.168}$	13.27%	1.24%
Periodic	0	2.154 ± 0.101	2.197 ± 0.121	$\mathbf{2.023 \pm 0.116}$	6.47%	8.59%
	1	2.578 ± 0.175	2.844 ± 0.419	$\mathbf{2.474 \pm 0.193}$	4.17%	14.93%
	3	2.994 ± 0.213	3.148 ± 0.439	$\mathbf{2.901 \pm 0.251}$	3.19%	8.48%
	5	3.631 ± 0.254	3.666 ± 0.410	$\mathbf{3.362 \pm 0.273}$	7.98%	9.04%
	7	3.841 ± 0.316	3.932 ± 0.481	$\mathbf{3.714 \pm 0.331}$	12.52%	5.87%

Well-specified Models and Variable Misspecification. We consider some simple data-generating processes (DGPs) that can be well-approximated by simple MLPs, i.e., the approximation error is small. Let $\boldsymbol{X} \in \mathbb{R}^p$ be a random vector where each feature X_j is independently sampled from $\mathcal{N}(0,1)$, with the number of features $p = 10$. We define the different DGPs: (i) Nested: $f_0(\boldsymbol{x}) = \sin(\sum_{j=1}^{10} x_j^2) + \sum_{j=1}^{10}(\cos x_j)^2$; (ii) Interaction: $f_0(\boldsymbol{X}) = \frac{1}{2}\sum_{j=1}^{5}\sum_{k=6}^{10} X_j X_k$; (iii) Periodic: $f_0(\boldsymbol{X}) = \sum_{j=1}^{5}\sin(X_j) + \sum_{j=6}^{10}\cos(X_j)$.

When all relevant features are included in the model, the setting is considered well-specified. We introduce variable misspecification by randomly dropping a subset of features during training, with the number of dropped features controlling the degree of misspecification. The results are summarized in Table 2. Our method WDE consistently achieves the lowest MSE.

Structural Misspecification. To investigate structural misspecification, we define the DGP as a convex combination of a simple function and a more complex function: $f_0(\boldsymbol{X}) = \alpha f_{\text{simple}}(\boldsymbol{X}) + (1-\alpha) f_{\text{complex}}(\boldsymbol{X})$, where $f_{\text{simple}} = \text{MLP}_{2\times 30}$ and $f_{\text{complex}} = \text{MLP}_{3\times 100}$. The function used for fitting is $f_{\text{simple}}(\boldsymbol{X})$. The parameter $\alpha \in [0,1]$ controls the degree of misspecification, from well-specified ($\alpha = 0$) to totally misspecified ($\alpha = 1$). We report the relative MSE under varying degrees of misspecification with a sample size of $N = 5000$ in Table 2. The results show that as the misspecification degree increases, the MSE of DE grows rapidly. In contrast, our method (WDE) significantly mitigates this performance degradation and maintains robust accuracy even under high misspecification.

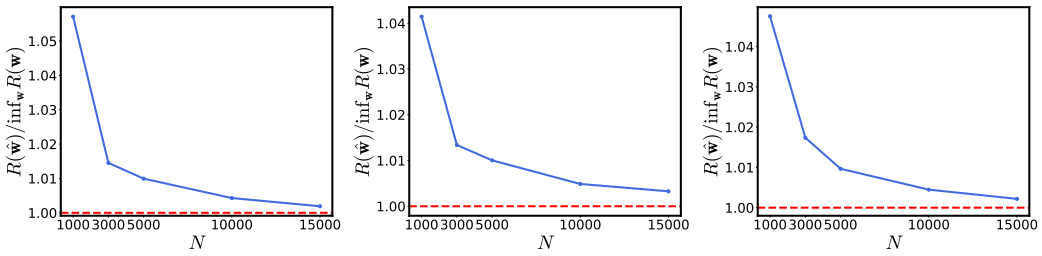
Table 2: Performance comparison across different parameter discrepancies with sample size 5000

Δ	α	**DE**	**EW**	**WDE**	Δ_{DE}	Δ_{EW}
30000	0.2	0.598 ± 0.222	0.402 ± 0.085	$\mathbf{0.341 \pm 0.034}$	75.47%	17.83%
	0.5	0.630 ± 0.259	0.438 ± 0.179	$\mathbf{0.397 \pm 0.146}$	58.59%	10.37%
	0.7	0.634 ± 0.230	0.473 ± 0.128	$\mathbf{0.398 \pm 0.083}$	59.53%	18.87%
	0.9	0.654 ± 0.157	0.524 ± 0.185	$\mathbf{0.417 \pm 0.089}$	56.83%	25.76%
50000	0.2	0.587 ± 0.159	0.374 ± 0.075	$\mathbf{0.339 \pm 0.032}$	73.10%	10.22%
	0.5	0.632 ± 0.193	0.425 ± 0.098	$\mathbf{0.399 \pm 0.091}$	58.31%	6.49%
	0.7	0.701 ± 0.201	0.457 ± 0.099	$\mathbf{0.419 \pm 0.067}$	67.21%	9.17%
	0.9	0.790 ± 0.171	0.488 ± 0.074	$\mathbf{0.437 \pm 0.061}$	80.65%	11.64%
100000	0.2	0.576 ± 0.206	0.335 ± 0.031	$\mathbf{0.331 \pm 0.026}$	74.31%	1.51%
	0.5	0.582 ± 0.175	0.350 ± 0.050	$\mathbf{0.340 \pm 0.039}$	71.42%	3.15%
	0.7	0.638 ± 0.139	0.353 ± 0.052	$\mathbf{0.346 \pm 0.043}$	84.26%	1.84%
	0.9	0.677 ± 0.226	0.350 ± 0.044	$\mathbf{0.348 \pm 0.043}$	94.50%	0.50%

Table 3: Comparison with relative improvement percentages (Δ) across DE, EW and WDE.

Complexity	N	**DE**	**EW**	**WDE**	Δ_{DE}	Δ_{EW}
Square Wave	5000	0.96068 ± 0.02707	0.94702 ± 0.01385	$\mathbf{0.93171 \pm 0.02422}$	3.11%	1.64%
	10000	0.81132 ± 0.02665	0.80467 ± 0.02623	$\mathbf{0.76520 \pm 0.03430}$	6.03%	5.16%
	15000	0.74766 ± 0.03754	0.71625 ± 0.03140	$\mathbf{0.64963 \pm 0.06092}$	15.09%	10.25%
Infinite	5000	0.00137 ± 0.00021	0.00141 ± 0.00010	$\mathbf{0.00117 \pm 0.00012}$	17.34%	20.40%
	10000	0.00085 ± 0.00015	0.00086 ± 0.00013	$\mathbf{0.00067 \pm 0.00014}$	27.38%	28.86%
	15000	0.00057 ± 0.00013	0.00065 ± 0.00010	$\mathbf{0.00044 \pm 0.00012}$	30.13%	49.55%

Inherent Misspecification. We consider two discontinuous DGPs to simulate inherent misspecification: (i) Infinite Discontinuity: $f_0(\boldsymbol{X}) = \sum_{j=1}^{d} \frac{1}{j} \left(\sum_{k=1}^{K} \frac{1}{k^2} \cdot \mathbb{I}\left(X_j > \frac{1}{k}\right) \right)$ is the indicator function and $K = 1000$; (ii) Square Wave: $y = \sum_{i=1}^{d} \frac{1}{i+1} \cdot \text{sgn}\left(\sin\left(\frac{2\pi(i+1)}{d} x_i + \frac{i\pi}{d}\right)\right)$, where $\text{sgn}(\cdot)$ is the signum function. The results in Table 3 demonstrate that WDE achieves substantial improvement.



(a) Nested (b) Interaction (c) Periodic

Figure 2: The ratio of MSE of WDE to the oracle weight as sample size increases.

Theoretical results. To further validate Theorem 3, we plot the ratio of the MSE of our proposed weighted deep ensemble estimator to that of the oracle weight as the sample size increases, as shown in Figure 2. We can observe that as the sample size grows, the weighted deep ensemble estimator asymptotically approaches the optimal oracle weight using only the observed data.

Comparing with other weighted methods. We compare our method against two alternatives in Table 4: (i) Greedy Ensemble: Sequentially adds models to minimize validation loss, then uses equal weighting. (ii) In-sample Ensemble: Weights are optimized directly on the training MSE.

More simulation results can be found in Section D in the Appendix.

5 Conclusion

Our paper formally defines the misspecification problem in deep learning and establishes a theoretical foundation for the weighted deep ensemble, including its error bound and asymptotic optimality. Extensive experiments demonstrate that our method consistently outperforms traditional deep ensembles across various misspecification scenarios and significantly mitigates the adverse effects of model misspecification.

Table 4: Performance comparison of weighted methods in variable misspecification

Complexity	# Missing	WDE	In-sample Ensemble	Greedy Ensemble
Nested	0	$\mathbf{7.025 \pm 0.247}$	7.258 ± 0.329	7.136 ± 0.232
	1	$\mathbf{8.028 \pm 1.236}$	8.149 ± 1.150	8.108 ± 1.187
	3	$\mathbf{8.077 \pm 1.211}$	8.186 ± 1.208	8.177 ± 1.217
	5	$\mathbf{7.792 \pm 0.287}$	7.963 ± 0.338	7.904 ± 0.346
	7	$\mathbf{7.967 \pm 0.431}$	8.076 ± 0.393	8.045 ± 0.400
Interaction	0	$\mathbf{1.773 \pm 0.131}$	1.869 ± 0.161	1.794 ± 0.116
	1	$\mathbf{2.949 \pm 0.778}$	2.990 ± 0.813	2.948 ± 0.778
	3	$\mathbf{3.890 \pm 0.984}$	3.914 ± 0.995	3.908 ± 0.981
	5	$\mathbf{5.077 \pm 1.099}$	5.114 ± 1.038	5.125 ± 1.084
	7	$\mathbf{5.548 \pm 1.168}$	5.651 ± 1.124	5.547 ± 1.170
Periodic	0	$\mathbf{2.023 \pm 0.116}$	2.086 ± 0.137	2.051 ± 0.126
	1	$\mathbf{2.474 \pm 0.193}$	2.535 ± 0.178	2.518 ± 0.226
	3	$\mathbf{2.901 \pm 0.251}$	2.961 ± 0.261	2.929 ± 0.260
	5	$\mathbf{3.362 \pm 0.273}$	3.400 ± 0.270	3.378 ± 0.279
	7	$\mathbf{3.714 \pm 0.331}$	3.751 ± 0.310	3.730 ± 0.336

6 Ethics Statement

Our work is committed to the highest standards of scientific excellence, grounded in the principles of honesty, reliability, and transparency. The core technical contribution of this paper is to address the challenge of model misspecification in deep learning. This is not merely a technical problem but an ethical imperative. A misspecified model can produce unreliable predictions, perpetuate and amplify societal biases, and ultimately cause harm if deployed in critical real-world applications such as healthcare, finance, or autonomous systems. By developing methods to better understand, identify, and correct for misspecification, our research aims to contribute to the creation of more robust, fair, and trustworthy AI systems. We believe that this work is a necessary step toward the responsible development of artificial intelligence, ensuring that its benefits can be realized while minimizing potential negative societal consequences.

7 Reproducibility Statement

We are committed to the full reproducibility of our work. To this end, we have included the complete and detailed derivations of our theoretical proofs in the Appendix. Furthermore, all experimental results presented in this paper can be fully reproduced according to experimental details and and the source code will be made available in our code repository. [1]

References

Taiga Abe, Estefany Kelly Buchanan, Geoff Pleiss, Richard Zemel, and John P Cunningham. Deep ensembles work, but are they necessary. *Advances in Neural Information Processing Systems.*, 35: 33646–33660, 2022.

Ben Adcock and Nick Dexter. The gap between theory and practice in function approximation with deep neural networks. *SIAM Journal on Mathematics of Data Science*, 3(2):624–655, 2021.

Tomohiro Ando and Ker-Chau Li. A model-averaging approach for high-dimensional regression. *Journal of the American Statistical Association.*, 109(505):254–265, 2014.

Andrew R Barron. Approximation and estimation bounds for artificial neural networks. *Machine learning*, 14(1):115–133, 1994.

Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356): 791–799, 1976.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press., 2004.

[1] Source code are available at: `https://anonymous.4open.science/r/WDE-547B`

Simone Cerreia-Vioglio, Lars Peter Hansen, Fabio Maccheroni, and Massimo Marinacci. Making decisions under model misspecification. *Review of Economic Studies*, pp. rdaf046, 2025.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, pp. C1–C68, 2018.

Dennis Elbrächter, Dmytro Perekrestenko, Philipp Grohs, and Helmut Bölcskei. Deep neural network approximation theory. *arXiv preprint arXiv:1901.02220*, 2019.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757.*, 2019.

TA Gerds and Martin Schumacher. On functional misspecification of covariates in the cox regression model. *Biometrika*, 88(2):572–580, 2001.

Raphael Gontijo-Lopes, Yann Dauphin, and Ekin Dogus Cubuk. No one representation to rule them all: Overlapping features of training methods. In *International Conference on Learning Representations.*, 2022.

Nikolay Gospodinov and Esfandiar Maasoumi. Generalized aggregation of misspecified models: With an application to asset pricing. *Journal of Econometrics*, 222(1):451–467, 2021.

Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 2002.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition.*, pp. 770–778, 2016.

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

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision.*, pp. 646–661. Springer, 2016.

Yichong Huang, Xiaocheng Feng, Baohang Li, Yang Xiang, Hui Wang, Ting Liu, and Bing Qin. Ensemble learning for heterogeneous large language models with deep parallel collaboration. *Advances in Neural Information Processing Systems.*, 37:119838–119860, 2024.

Yuling Jiao, Guohao Shen, Yuanyuan Lin, and Jian Huang. Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics.*, 51(2):691–716, 2023.

Yuling Jiao, Yang Wang, and Bokai Yan. Approximation bounds for recurrent neural networks with application to regression. *arXiv preprint arXiv:2409.05577.*, 2024.

Ioannis Kasparis. Functional form misspecification in regressions with a unit root. *Econometric Theory*, 27(2):285–311, 2011.

Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *European conference on computer vision.*, pp. 736–751, 2018.

Anastasis Kratsios, Behnoosh Zamanlooy, Tianlin Liu, and Ivan Dokmanić. Universal approximation under constraints is possible with transformers. *arXiv preprint arXiv:2110.03303*, 2021.

Kun Kuang, Ruoxuan Xiong, Peng Cui, Susan Athey, and Bo Li. Stable prediction with model misspecification and agnostic distribution shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4485–4492, 2020.

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

Giacomo Lanzani. Dynamic concern for misspecification. *Econometrica*, 93(4):1333–1370, 2025.

Shaoze Li, Zhibin Deng, Cheng Lu, Junhao Wu, Jinyu Dai, and Qiao Wang. An efficient global algorithm for indefinite separable quadratic knapsack problems with box constraints. *Computational Optimization and Applications.*, 86(1):241–273, 2023.

Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021.

Esfandiar Maasoumi. How to live with misspecification if you must. *Journal of Econometrics*, 44 (1-2):67–86, 1990.

Andrés Masegosa, Stephan Lorenzen, Christian Igel, and Yevgeny Seldin. Second order pac-bayesian bounds for the weighted majority vote. *Advances in Neural Information Processing Systems*, 33: 5263–5273, 2020.

Mohammad Saeed Masiha, Amin Gohari, Mohammad Hossein Yassaee, and Mohammad Reza Aref. Learning under distribution mismatch and model misspecification. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2912–2917. IEEE, 2021.

Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems.*, 35:17703–17716, 2022.

Anya M McGuirk, Paul Driscoll, and Jeffrey Alwang. Misspecification testing: a comprehensive approach. *American Journal of Agricultural Economics*, 75(4):1044–1055, 1993.

Ammar Mohammed and Rania Kora. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University-Computer and Information Sciences.*, 35 (2):757–774, 2023.

Luis A Ortega, Rafael Cabañas, and Andres Masegosa. Diversity and generalization in neural network ensembles. In *International Conference on Artificial Intelligence and Statistics*, pp. 11720–11743. PMLR, 2022.

Sejun Park, Chulhee Yun, Jaeho Lee, and Jinwoo Shin. Minimum width for universal approximation. *arXiv preprint arXiv:2006.08859*, 2020.

Guangtai Qu, Shaoze Li, Zhibin Deng, and Cheng Lu. A fast global algorithm for multi-linearly constrained separable binary quadratic program. *Journal of Industrial and Management Optimization.*, 21(2):1456–1473, 2025.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *international conference on machine learning*, pp. 2847–2854. PMLR, 2017.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Anselm Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of statistics*, 48(4):1875–1897, 2020.

Kajetan Schweighofer, Adrian Arnaiz-Rodriguez, Sepp Hochreiter, and Nuria Oliver. The disparate benefits of deep ensembles. *arXiv preprint arXiv:2410.13831.*, 2024.

Guohao Shen, Yuling Jiao, Yuanyuan Lin, and Jian Huang. Approximation with cnns in sobolev space: with applications to classification. *Advances in Neural Information Processing Systems.*, 35:2876–2888, 2022.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

Charles J Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, pp. 1040–1053, 1982.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Ambuj Tewari and Peter L Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. 2007.

Aad W Van Der Vaart and Jon A Wellner. Weak convergence. In *Weak convergence and empirical processes: with applications to statistics*, pp. 16–28. Springer, 1996.

Halbert White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the econometric society*, pp. 1–25, 1982.

David H Wolpert. Stacked generalization. *Neural Networks.*, 5(2):241–259, 1992.

Danny Wood, Tingting Mu, Andrew M Webb, Henry WJ Reeve, Mikel Lujan, and Gavin Brown. A unified theory of diversity in ensemble learning. *Journal of Machine Learning Research.*, 24(359): 1–49, 2023.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Shaofeng Zhang, Meng Liu, and Junchi Yan. The diversified ensemble neural network. *Advances in Neural Information Processing Systems.*, 33:16001–16011, 2020.

Xinyu Zhang. *Model averaging and its applications*. PhD thesis, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 2010.

Xinyu Zhang, Dalei Yu, Guohua Zou, and Hua Liang. Optimal model averaging estimation for generalized linear models and generalized linear mixed-effects models. *Journal of the American Statistical Association.*, 111(516):1775–1790, 2016.

A Large Language Model Usage Disclosure

In this work, we made limited use of a large language model (LLM) as an auxiliary tool. In particular:

Language polishing: We used ChatGPT-5 to improve the readability, grammar, and fluency of the English text. The authors reviewed all edits and manually adjusted phrasing as needed.
Code assistance: We asked ChatGPT-5 to assist in generating boilerplate code for data preprocessing, but in a minimal and constrained way; the authors carefully verified, tested, modified, and adapted all generated code to ensure correctness.

We emphasize that all content in the submission is attributed to the authors. We take full responsibility for the correctness of all claims and any content originally generated by the LLM that contained errors or inconsistencies that were revised or removed. We confirm that the LLM was not included as an author, and no portion of the submission is entirely generated without human oversight.

B Proof of theorems

To facilitate the proof of the theorem, we begin by stating a useful lemma.

B.1 Lemma 1

Lemma 1 (Lemma 1 in Zhang (2010))**.** *Let*

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \{R(\boldsymbol{w}) + a_n(\boldsymbol{w}) + b_n\}.$$

If

$$\sup_{\boldsymbol{w}\in\mathcal{W}} \frac{|a_n(w)|}{R^*(w)} = o_p(1)$$

and

$$\sup_{w\in W} \frac{|R(\boldsymbol{w}) - R^*(\boldsymbol{w})|}{R^*(\boldsymbol{w})} = o_p(1),$$

and there exists a positive constant c so that $\lim_{n\to\infty} \inf_{\boldsymbol{w}\in\mathcal{W}} R^*(\boldsymbol{w}) \geq c$ almost surely, then we have

$$\frac{R(\widehat{\boldsymbol{w}})}{\inf\limits_{w\in\mathcal{W}} R(\boldsymbol{w})} \to 1$$

in probability.

B.2 Proof of Theorem 1

The weight choice criterion can be decomposed as

$$\begin{aligned}
& \mathcal{L}(\boldsymbol{w}) \\
= \;& \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{Y_i - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w})\}^2 \\
= \;& \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{Y_i - f_0(\boldsymbol{X}_i) + f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w})\}^2 \\
= \;& \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{Y_i - f_0(\boldsymbol{X}_i)\}^2 + \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w})\}^2 \\
& + \frac{2}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{Y_i - f_0(\boldsymbol{X}_i)\}\{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w})\}.
\end{aligned}$$

We first analyze $1/n_{\text{val}} \sum_{i=1}^{n_{\text{val}}} \{Y_i - f_0(\boldsymbol{X}_i)\}\{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w})\}$. Let $\mathcal{G}_r = \{g_{\boldsymbol{w}}(\boldsymbol{X}) = f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}) : \boldsymbol{w} \in \mathcal{W}, \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\|_{L^2} \leq r\}$, and let $\mathcal{D} = \{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{validation}}\}$ collect all observed samples. By the multiplier inequality (Van Der Vaart & Wellner, 1996; Bartlett et al., 2005), we have

$$\sup_{g\in\mathcal{G}_r} [\frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} (Y_i - f_0(\boldsymbol{X}_i))g(\boldsymbol{X}_i) - \mathbb{E}\{(Y - f_0(\boldsymbol{X}))g(\boldsymbol{X})|\mathcal{D}\}] = O_p\Big(\sigma \mathbb{E}\mathcal{R}_{n_{\text{val}}}\mathcal{G}_r + \sigma \frac{1}{\sqrt{n_{\text{val}}}} r\Big), \quad (1)$$

where $\mathcal{R}_{n_{\text{val}}}\mathcal{G}_r$ is the Rademacher complexity of $\mathcal{G}_r$, and σ is the sub-Gaussian parameter of the noise $\varepsilon = Y - f_0(\boldsymbol{X})$. Given that $\mathcal{W} = \{\boldsymbol{w} \in [0,1]^M, \sum_{m=1}^M w_m = 1\}$, we have $\mathcal{R}_{n_{\text{val}}}\mathcal{G}_r \leq r\sqrt{2\log(M)/n}$. Since σ is finite, and n_{val} has the same order as the total sample size n, the first term on the right hand side of (1) is $O_p(\sigma \mathbb{E}\mathcal{R}_n \mathcal{G}_r) = O_p(r\sqrt{\log(M)}/\sqrt{n})$, and the second term is $O_p(\sigma\sqrt{\log(n_{\text{val}})/n_{\text{val}}} r) = O_p(r/\sqrt{n})$. Then (1) becomes

$$\sup_{g\in\mathcal{G}_r} [\frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} (Y_i - f_0(\boldsymbol{X}_i))g(\boldsymbol{X}_i) - \mathbb{E}\{(Y - f_0(\boldsymbol{X}))g(\boldsymbol{X})|\mathcal{D}\}] = O_p\Big(\frac{r\sqrt{\log(M)}}{\sqrt{n}}\Big). \quad (2)$$

Let $\boldsymbol{w}_0$ be the one-hot vector with entry 1 at the position corresponding to the model with the fastest convergence rate and 0 elsewhere. Then it is straightforward to show $\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0)\|_{L^2}^2 = \|f_0(\boldsymbol{X}) - \widetilde{f}(\boldsymbol{X})\|_{L^2}^2 = O_p(S^2)$. Moreover, since $\widehat{f}$ only depends on $\mathcal{D}$, and $\boldsymbol{X}$ is an independent sample drawn from the same distribution but independent of $\mathcal{D}$, we have

$$\begin{aligned}
& \mathbb{E}\{(Y - f_0(\boldsymbol{X}))(f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0))|\mathcal{D}\} \\
= \;& \mathbb{E}\left[\mathbb{E}\left\{(Y - f_0(\boldsymbol{X}))(f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0))|\boldsymbol{X}, \mathcal{D}\right\}|\mathcal{D}\right] \\
= \;& \mathbb{E}\left\{\mathbb{E}(Y - f_0(\boldsymbol{X})|\boldsymbol{X}, \mathcal{D})(f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0))|\mathcal{D}\right\} \\
= \;& \mathbb{E}\{\mathbb{E}(\varepsilon|\boldsymbol{X})(f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0))|\mathcal{D}\} \\
= \;& 0. \qquad (3)
\end{aligned}$$

Taking $r = S$, we have $g_{\boldsymbol{w}_0} \in \mathcal{G}_S$ and thus by (2) and (3), we have

$$\begin{aligned}
& \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{Y_i - f_0(\boldsymbol{X}_i)\}\{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w}_0)\} \\
= & \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} (Y_i - f_0(\boldsymbol{X}_i))(f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w}_0)) - \mathbb{E}\{(Y - f_0(\boldsymbol{X}))(f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0))|\mathcal{D}\} \\
& + \mathbb{E}\{(Y - f_0(\boldsymbol{X}))(f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0))|\mathcal{D}\} \\
\leq & \sup_{g \in \mathcal{G}_r} [\frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} (Y_i - f_0(\boldsymbol{X}_i)) g(\boldsymbol{X}_i) - \mathbb{E}\{(Y - f_0(\boldsymbol{X})) g(\boldsymbol{X})|\mathcal{D}\}] \\
= & O_p(\sqrt{\frac{\log(M)}{n}} S), \qquad (4)
\end{aligned}$$

Note that (3) remains valid when $\boldsymbol{w}_0$ is replaced by $\widehat{\boldsymbol{w}}$, because $\widehat{\boldsymbol{w}}$ is entirely determined by $\mathcal{D}$, and hence is independent of the new sample $\boldsymbol{X}$. Taking $r = \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}})\|_{L^2}$, we have

$$\begin{aligned}
& \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{Y_i - f_0(\boldsymbol{X}_i)\}\{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \widehat{\boldsymbol{w}})\} \\
= & \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} (Y_i - f_0(\boldsymbol{X}_i))(f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \widehat{\boldsymbol{w}})) - \mathbb{E}\{(Y - f_0(\boldsymbol{X}))(f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}}))|\mathcal{D}\} \\
& + \mathbb{E}\{(Y - f_0(\boldsymbol{X}))(f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}}))|\mathcal{D}\} \\
\leq & \sup_{g \in \mathcal{G}_r} [\frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} (Y_i - f_0(\boldsymbol{X}_i)) g(\boldsymbol{X}_i) - \mathbb{E}\{(Y - f_0(\boldsymbol{X})) g(\boldsymbol{X})|\mathcal{D}\}] \\
= & O_p(\sqrt{\frac{\log(M)}{n}} \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \widehat{\boldsymbol{w}})\|_{L^2}). \qquad (5)
\end{aligned}$$

Then we analyze $1/n_{\text{val}} \sum_{i=1}^{n_{\text{val}}} \{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w})\}^2$. Let $\mathcal{H}_r = \{h_{\boldsymbol{w}} = \{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\}^2 : \boldsymbol{w} \in \mathcal{W}, \text{Var}[\{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\}^2|\mathcal{D}] \leq r^2\}$. Similar to (1), we obtain

$$\sup_{h \in \mathcal{H}_r} \left[\frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} h(\boldsymbol{X}_i)^2 - \mathbb{E}\left\{h(\boldsymbol{X})^2|\mathcal{D}\right\} \right] = O_p\Big(\mathbb{E}\mathcal{R}_n \mathcal{H}_r + \sqrt{\tfrac{1}{n_{\text{val}}}} r\Big) = O_p\Big(\sqrt{\tfrac{\log(M)}{n}} r\Big). \quad (6)$$

Since f_0 and $\widehat{f}_m$ are uniformly bounded, there exists a positive constant C such that

$$\begin{aligned}
& \text{Var}[\{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\}^2|\mathcal{D}] \\
& \leq \mathbb{E}[\{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\}^4|\mathcal{D}] \\
& \leq \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\|_{L^2}^2 \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\|_{\infty}^2 \\
& \leq C^2 \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w})\|_{L^2}^2. \qquad (7)
\end{aligned}$$

When taking $\boldsymbol{w} = \boldsymbol{w}_0$ and $r = CS$ in $\mathcal{H}_r$, we have $h_{\boldsymbol{w}_0} = \{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0)\}^2 \in \mathcal{H}_{CS}$ because $\text{Var}[\{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0)\}^2|\mathcal{D}] \leq C^2 \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0)\|_{L^2}^2 = C^2 S^2$ from (7). By (6), it follows that

$$\begin{aligned}
& \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w}_0)\}^2 \\
= & \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i; \boldsymbol{w}_0)\}^2 - \mathbb{E}\left[\{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0)\}^2|\mathcal{D}\right] + \mathbb{E}\left[\{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0)\}^2|\mathcal{D}\right] \\
\leq & \sup_{h \in \mathcal{H}_{CS}} \left[\frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} h(\boldsymbol{X}_i)^2 - \mathbb{E}\left\{h(\boldsymbol{X})^2|\mathcal{D}\right\} \right] + \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X}; \boldsymbol{w}_0)\|_{L^2}^2 \\
= & O_p(\sqrt{\tfrac{\log(M)}{n}} S + S^2). \qquad (8)
\end{aligned}$$

Similarly, taking $\boldsymbol{w} = \widehat{\boldsymbol{w}}$ and $r_{\widehat{\boldsymbol{w}}} = C\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2}$, we have $h_{\widehat{\boldsymbol{w}}} = \{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\}^2 \in \mathcal{H}_{r_{\widehat{\boldsymbol{w}}}}$ and thus by (6), the following bound holds:

$$\begin{aligned}
&\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}\{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i;\widehat{\boldsymbol{w}})\}^2 \\
= \;& \frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}\{f_0(\boldsymbol{X}_i) - \widehat{f}(\boldsymbol{X}_i;\widehat{\boldsymbol{w}})\}^2 - \mathbb{E}\left[\{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\}^2|\mathcal{D}\right] + \mathbb{E}\left[\{f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\}^2|\mathcal{D}\right] \\
\leq \;& \sup_{h\in\mathcal{H}_{r_{\widehat{\boldsymbol{w}}}}}\left[\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}h(\boldsymbol{X}_i)^2 - \mathbb{E}\left\{h(\boldsymbol{X})^2|\mathcal{D}\right\}\right] + \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2}^2 \\
= \;& O_p(\sqrt{\tfrac{\log(M)}{n}}\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2}) + \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2}^2. \qquad (9)
\end{aligned}$$

Although $\widehat{\boldsymbol{w}}$ depends on the validation data, the bound still holds because the supremum is taken over the class $\mathcal{H}_{r_{\widehat{\boldsymbol{w}}}}$. Combining (5) and (9), $\mathcal{L}(\widehat{\boldsymbol{w}})$ can be written as

$$\mathcal{L}(\widehat{\boldsymbol{w}}) = \frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}\{Y_i - f_0(\boldsymbol{X}_i)\}^2 + \|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2}^2 + O_p(\sqrt{\frac{\log(M)}{n}})\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2}^2, \quad (10)$$

and according to (4) and (8), $\mathcal{L}(\boldsymbol{w}_0)$ can be written as

$$\mathcal{L}(\boldsymbol{w}_0) = \frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}\{Y_i - f_0(\boldsymbol{X}_i)\}^2 + O_p(S^2 + \sqrt{\frac{\log(M)}{n}}S). \qquad (11)$$

Using the expansions in (10) and (11), together with the fact that $\mathcal{L}(\widehat{\boldsymbol{w}})$ minimizes the validation loss, i.e., $\mathcal{L}(\widehat{\boldsymbol{w}}) \leq \mathcal{L}(\boldsymbol{w}_0)$, we have

$$\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2}^2 + O_p(\sqrt{\frac{\log(M)}{n}})\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2} = O_p(S^2 + \sqrt{\frac{\log(M)}{n}}S). \quad (12)$$

By completing the square, (12) can be written as

$$\left[\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2} + O_p(\sqrt{\frac{\log(M)}{n}})\right]^2 = \{O_p(S + \sqrt{\frac{\log(M)}{n}})\}^2,$$

i.e.,

$$\|f_0(\boldsymbol{X}) - \widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\|_{L^2} = O_p(S + \sqrt{\frac{\log(M)}{n}}).$$

This completes the proof of Theorem 1.

B.3 Proof of Theorem 2

The regression and multiclass-classification problems differ only in the choice of loss function. In classification problems, the performance measure of primary interest is the misclassification error (the 0–1 loss). Because this loss is discontinuous, it is typically replaced by a continuous surrogate, most commonly the cross-entropy loss. The surrogate is smooth and differentiable, which facilitates gradient-based optimization. Importantly, since cross-entropy is a calibrated surrogate, its excess risk dominates the squared excess misclassification risk; see Tewari & Bartlett (2007) for example. Let $\boldsymbol{w}_0$ be the one-hot vector that places 1 on the coordinate corresponding to the model with the fastest convergence rate and 0 elsewhere. Thus, if the excess misclassification risk of $\boldsymbol{w}_0$ is of order S, the corresponding cross-entropy excess risk satisfies

$$\mathbb{E}\Big\{f_0(\boldsymbol{X})\log\frac{f_0(\boldsymbol{X})}{\widehat{f}(\boldsymbol{X};\boldsymbol{w}_0)}|\mathcal{D}\Big\} = O_p(S^2).$$

Let

$$\mathcal{H}_r = \Big\{\, h_{\boldsymbol{w}}(\boldsymbol{X}) = \{\log f_0(\boldsymbol{X}) - \log\widehat{f}(\boldsymbol{X};\boldsymbol{w})\} :\; w\in\mathcal{W},\; \text{Var}\{h_{\boldsymbol{w}}(\boldsymbol{X})|\mathcal{D}\} \leq r^2 \,\Big\}.$$

By the multiplier inequality,

$$\sup_{h\in\mathcal{H}_r}\Big|\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}\{Y_i-f_0(\boldsymbol{X}_i)\}h(\boldsymbol{X}_i)-\mathbb{E}[\{Y-f_0(\boldsymbol{X})\}h(\boldsymbol{X})|\mathcal{D}]\Big|=O_p\Big(\gamma\,\mathbb{E}\mathcal{R}_{n_{\text{val}}}\mathcal{H}_r+\gamma\sqrt{\tfrac{\log(M)}{n}}\,r\Big),\quad(13)$$

where $\gamma^2=\text{Var}(Y-f_0(\boldsymbol{X}))$ is finite because $\text{Var}(Y-f_0(\boldsymbol{X}))\le 1/4$ in classification tasks. Moreover, we have $\mathbb{E}\mathcal{R}_{n_{\text{val}}}\mathcal{H}_r\le Cr\sqrt{\log(M)/n_{\text{val}}}$, and n_{val} has the same order as n, then (13) becomes

$$\sup_{h\in\mathcal{H}_r}\Big|\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}\{Y_i-f_0(\boldsymbol{X}_i)\}h(\boldsymbol{X}_i)-\mathbb{E}[\{Y-f_0(\boldsymbol{X})\}h(\boldsymbol{X})|\mathcal{D}]\Big|=O_p\Big(\sqrt{\tfrac{\log(M)}{n}}\,r\Big).\quad(14)$$

Similarly,

$$\sup_{h\in\mathcal{H}_r}\Big|\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}f_0(\boldsymbol{X}_i)h(\boldsymbol{X}_i)-\mathbb{E}[f_0(\boldsymbol{X})h(\boldsymbol{X})|\mathcal{D}]\Big|=O_p\Big(\sqrt{\tfrac{\log(M)}{n}}\,r\Big).\quad(15)$$

Moreover, we have

$$\mathbb{E}[\{Y-f_0(\boldsymbol{X})\}h(\boldsymbol{X})|\mathcal{D}]=\mathbb{E}[\mathbb{E}\{Y-f_0(\boldsymbol{X})|\boldsymbol{X}\}h(\boldsymbol{X})|\mathcal{D}]=0.\quad(16)$$

By (14)-(16), we have

$$\begin{aligned}
&\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}Y_ih(\boldsymbol{X}_i)\\
=\;&\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}\{Y_i-f_0(\boldsymbol{X}_i)\}h(\boldsymbol{X}_i)+\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}f_0(\boldsymbol{X}_i)h(\boldsymbol{X}_i)\\
\le\;&\sup_{h\in\mathcal{H}_r}\Big|\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}\{Y_i-f_0(\boldsymbol{X}_i)\}h(\boldsymbol{X}_i)-\mathbb{E}[\{Y-f_0(\boldsymbol{X})\}h(\boldsymbol{X})|\mathcal{D}]\Big|\\
&+\sup_{h\in\mathcal{H}_r}\Big|\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}f_0(\boldsymbol{X}_i)h(\boldsymbol{X}_i)-\mathbb{E}[f_0(\boldsymbol{X})h(\boldsymbol{X})|\mathcal{D}]\Big|\\
&+\mathbb{E}[\{Y-f_0(\boldsymbol{X})\}h(\boldsymbol{X})|\mathcal{D}]+\mathbb{E}[f_0(\boldsymbol{X})h(\boldsymbol{X})|\mathcal{D}]\\
=\;&\mathbb{E}[f_0(\boldsymbol{X})h(\boldsymbol{X})|\mathcal{D}]+O_p\Big(\sqrt{\tfrac{\log(M)}{n}}r\Big).
\end{aligned}\quad(17)$$

Taking $r=S$ and $h(\boldsymbol{X})=\log(f_0(\boldsymbol{X})/f_0(\boldsymbol{X};\boldsymbol{w}_0))$, (17) becomes

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}_0)&=\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}Y_i\log(\widehat{f}(\boldsymbol{X}_i;\boldsymbol{w}_0))\\
&=\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}Y_i\log\frac{f_0(\boldsymbol{X}_i)}{\widehat{f}(\boldsymbol{X}_i;\boldsymbol{w}_0)}-\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}Y_i\log(f_0(\boldsymbol{X}_i))\\
&=\Big|\mathbb{E}\{f_0(\boldsymbol{X})\log\frac{f_0(\boldsymbol{X})}{\widehat{f}(\boldsymbol{X};\boldsymbol{w}_0)}|\mathcal{D}\}\Big|+O_p\Big(\sqrt{\tfrac{\log(M)}{n}}S\Big)-\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}Y_i\log(f_0(\boldsymbol{X}_i))\\
&=O_p\Big(S^2+\sqrt{\tfrac{\log(M)}{n}}S\Big)-\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}Y_i\log(f_0(\boldsymbol{X}_i))
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
\mathcal{L}(\widehat{\boldsymbol{w}})&=\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}Y_i\log(\widehat{f}(\boldsymbol{X}_i;\widehat{\boldsymbol{w}}))\\
&=\mathbb{E}\{f_0(\boldsymbol{X})\log\frac{f_0(\boldsymbol{X})}{\widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})}|\mathcal{D}\}+O_p(\sqrt{\frac{\log(n)}{n}})\sqrt{\mathbb{E}\{f_0(\boldsymbol{X})\log\frac{f_0(\boldsymbol{X})}{\widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})}|\mathcal{D}\}}\\
&\quad-\frac{1}{n_{\text{val}}}\sum_{i=1}^{n_{\text{val}}}Y_i\log(f_0(\boldsymbol{X}_i)).
\end{aligned}$$

Since $\mathcal{L}(\widehat{\boldsymbol{w}}) \leq \mathcal{L}(\boldsymbol{w}_0)$, This implies

$$\mathbb{E}\{f_0(\boldsymbol{X}) \log \frac{f_0(\boldsymbol{X})}{\widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})}|\mathcal{D}\} + O_p(\sqrt{\frac{\log(M)}{n}})\sqrt{\mathbb{E}\{f_0(\boldsymbol{X}) \log \frac{f_0(\boldsymbol{X})}{\widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})}|\mathcal{D}\}} = O_p\Big(S^2 + \sqrt{\tfrac{\log(M)}{n}}S\Big),$$

and thus $\sqrt{\mathbb{E}\{f_0(\boldsymbol{X}) \log\{f_0(\boldsymbol{X})/\widehat{f}(\boldsymbol{X};\widehat{\boldsymbol{w}})\}|\mathcal{D}\}} = O_p(\sqrt{\frac{\log(M)}{n}} + S)$. This further implies that the misclassification rate $R_{0/1}(\widehat{\boldsymbol{w}}) = O_p(S + \sqrt{\frac{\log(M)}{n}})$.

B.4 Proof of Corollaries

In addition to the moment conditions discussed in the main text, specific network architectures require additional structural assumptions, such as smoothness conditions, to achieve the desirable convergence rates. Below we present the concrete sets of assumptions for MLPs, CNNs, and RNNs under which the corresponding rates can be attained.

Now we introduce the definition of Hölder class. Hölder class $\mathcal{H}^\beta([0,1]^d, \mathbb{R}, B_0)$ is defined as

$$\mathcal{H}^\beta([0,1]^d, \mathbb{R}, B_0) = \Big\{f : [0,1]^d \to \mathbb{R} : \max_{\|\boldsymbol{\alpha}\|_1 \leq \lfloor\beta\rfloor} \|\partial^{\boldsymbol{\alpha}} f\|_\infty \leq B_0,\ \max_{\|\boldsymbol{\alpha}\|_1 = \lfloor\beta\rfloor} \sup_{x \neq y} \frac{|\partial^{\boldsymbol{\alpha}} f(x) - \partial^{\boldsymbol{\alpha}} f(y)|}{\|x-y\|^{\beta - \lfloor\beta\rfloor}} \leq B_0\Big\},$$

where $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1} \cdots \partial^{\alpha_d}$ with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)^\top \in \mathbb{N}_0^d$ and $\|\boldsymbol{\alpha}\|_1 = \sum_{i=1}^d \alpha_i$.

Condition 3. *(i). The true target function f_0 belongs to a Hölder class $\mathcal{H}^\beta([0,1]^d, \mathbb{R}, B_0)$, and $\beta > 0$, $B_0 > 0$.*

(ii). The candidate DNNs are in the function class of ReLU MLPs:

$$\mathcal{F}_{B_0,\mathcal{W},\mathcal{D},\mathcal{S},d} = \{f : [0,1]^d \to \mathbb{R} : \textit{width } \mathcal{W},\ \textit{depth } \mathcal{D},\ \textit{total parameters } \mathcal{S},\ \|f\|_\infty \leq B_0\}, \quad (18)$$

and at least one candidate model satisfies

$$\mathcal{W} = O\Big(n^{\frac{d}{4(d+2\beta)}} \log_2(n)\Big), \quad \mathcal{D} = O\Big(n^{\frac{d}{4(d+2\beta)}} \log_2(n)\Big), \quad \mathcal{S} = O\Big(n^{\frac{3d}{4(d+2\beta)}} (\log n)^4\Big).$$

Condition 4. *(i). The true target function f_0 belongs to the Hölder class $\mathcal{H}^\beta([0,1]^d, \mathbb{R}, B_0)$ with $\beta > 0$ and $B_0 > 0$.*

(ii). The candidate CNNs are in the function class of CNNs:

$$\mathcal{F}_{B_0,\mathcal{W},\mathcal{D},\mathcal{S},s_{\min},s_{\max},d} = \Big\{f_{\text{CNN}} : [0,1]^d \to \mathbb{R} : \textit{width } \mathcal{W}, \textit{depth } \mathcal{D},\ \textit{total parameters } \mathcal{S},\ \textit{filter lengths } s_{\min} \leq s^{(i)} \leq s_{\max},\ \|f\|_\infty \leq B_0\Big\},$$

and at least one candidate model satisfies

$$\mathcal{D} \leq 42(|\beta|+1)^2 M \Big\lceil \log_2(8M) \Big\lceil \frac{\mathcal{W}-1}{s_{\min}-1} \Big\rceil \Big\rceil, \qquad 2 \leq s_{\min} \leq s_{\max} \leq \mathcal{W}, \qquad \mathcal{S} \leq 8\mathcal{W}\mathcal{D},$$

where

$$\mathcal{W} = 38^2(|\beta|+1)^4 d^{2|\beta|+2} N^2 \lceil \log_2(8N) \rceil^2.$$

(iii). The conditional probability $\mathbb{P}\{Y = 1 \mid \boldsymbol{X}\}$ is continuous on the support of $\mathcal{X}$, and the probability measure of $\boldsymbol{X}$ is absolutely continuous with respect to the Lebesgue measure.

Condition 5. *(i). The true target function f_0 belongs to the Hölder class $f_0 \in H^\beta([0,1]^{d\times l}, \mathbb{R}^{d_y}, B_0)$ with $\beta > 0$ and $B_0 > 0$..*

(ii). The candidate RNNs are in the function class of RNNs:

$$\mathcal{F}_{B_0,\mathcal{W},\mathcal{D},d,d_y,l} = \Big\{f_{\text{RNN}} : [0,1]^{d\times l} \to \mathbb{R}^{d_y} : \textit{width } \mathcal{W}, \textit{depth } \mathcal{D}, \|f\|_\infty \leq B_0\Big\},$$

and there exists at least one candidate RNN satisfying

$$\mathcal{W} = n^\alpha \log n, \qquad \mathcal{D} = n^{\frac{dl}{2dl+4\beta} - \alpha} \log n.$$

Corollary 5.3 in Jiao et al. (2023) guarantees that under Condition 3, the empirical risk minimizer $\widehat{f}$ satisfies

$$\mathbb{E}\|\widehat{f}(\boldsymbol{X}) - f_0(\boldsymbol{X})\|_{L^2}^2 \leq \mathcal{B}_0^5(\lfloor\beta\rfloor+1)^4 d^{2\lfloor\beta\rfloor+\beta\vee 1} n^{-2\beta/(d+2\beta)} (\log n)^{11} = \widetilde{O}(n^{-2\beta/(d+2\beta)}).$$

Substituting $S = n^{-\beta/(d+2\beta)}$ into Theorem 1 implies that, when the candidate models are MLPs and include one with width $\mathcal{W}$, depth $\mathcal{D}$, and size $\mathcal{S}$, the asymptotic error bound of the model averaging estimator is $\widetilde{O}_p(n^{-2\beta/(d+2\beta)})$.

For the CNN and RNN settings, the results are analogous. Under Conditions 1, 4-5, the convergence rates of CNNs and RNNs is $O_p(n^{-2\beta/(d+2\beta)})$ and $O_p(n^{-2\beta/(dl+2\beta)})$ according to Theorem 4.6 in Shen et al. (2022) and Theorem 6 in Jiao et al. (2024), respectively. Substituting these rates into Theorem 1 yields Corollary 2 and 3.

B.5 Proof of Theorem 3

Let

$$\widetilde{\mathcal{L}}(\boldsymbol{w}) = \mathcal{L}(\boldsymbol{w}) - \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} (Y_i^2 - f_0(\boldsymbol{X}_i)^2).$$

Observe that the newly added component is unrelated to $\boldsymbol{w}$, so we have

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \mathcal{L}(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \{R(\boldsymbol{w}) + \widetilde{\mathcal{L}}(\boldsymbol{w}) - R(\boldsymbol{w})\}.$$

According to Lemma 1, to prove Theorem 3, it is sufficient to prove

$$\sup_{\boldsymbol{w}\in\mathcal{W}} \frac{|R(\boldsymbol{w}) - R^*(\boldsymbol{w})|}{R^*(\boldsymbol{w})} = o_p(1) \tag{19}$$

and

$$\sup_{\boldsymbol{w}\in\mathcal{W}} \frac{|\widetilde{\mathcal{L}}(\boldsymbol{w}) - R(\boldsymbol{w})|}{R^*(\boldsymbol{w})} = o_p(1). \tag{20}$$

For (19), we have

$$\begin{aligned}
& \sup_{\boldsymbol{w}\in\mathcal{W}} \frac{|R(\boldsymbol{w}) - R^*(\boldsymbol{w})|}{R^*(\boldsymbol{w})} \\
\leq\ & \frac{\sup_{\boldsymbol{w}\in\mathcal{W}} |R(\boldsymbol{w}) - R^*(\boldsymbol{w})|}{\inf_{w\in\mathcal{W}} R^*(\boldsymbol{w})} \\
=\ & \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} |\|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2}^2 - \|f^*(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2}^2| \\
\leq\ & \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} |(\|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2} + \|f^*(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2}) \|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f^*(\boldsymbol{X};\boldsymbol{w})\|_{L^2}| \\
\leq\ & C\xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f^*(\boldsymbol{X};\boldsymbol{w})\|_{L^2} \\
=\ & O_p(\xi_n^{-1}\phi_n) \\
=\ & o_p(1),
\end{aligned}$$

where the last step comes from Condition 2. Thus (19) is obtained. Next, we will prove (20):

$$\begin{aligned}
&\sup_{\boldsymbol{w}\in\mathcal{W}} \frac{|\widetilde{\mathcal{L}}(\boldsymbol{w}) - R(\boldsymbol{w})|}{R^*(\boldsymbol{w})} \\
\leq\ & \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} |\widetilde{\mathcal{L}}(\boldsymbol{w}) - R(\boldsymbol{w})| \\
=\ & \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \Big\{ y_i - \widehat{f}(\boldsymbol{x}_i;\boldsymbol{w}) \Big\}^2 - \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{ y_i^2 - f_0(\boldsymbol{x}_i)^2 \} - \|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2}^2 \Big| \\
\leq\ & \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \Big\{ \widehat{f}(\boldsymbol{x}_i;\boldsymbol{w}) - f_0(\boldsymbol{X}_i) \Big\}^2 - \|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2}^2 \Big| \\
&+ \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{\widehat{f}(\boldsymbol{X}_i;\boldsymbol{w}) - f^*(\boldsymbol{X}_i;\boldsymbol{w})\}\{Y_i - f_0(\boldsymbol{X}_i)\} \Big|. \\
&+ \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} f^*(\boldsymbol{X}_i;\boldsymbol{w})\{Y_i - f_0(\boldsymbol{X}_i)\} \Big|. \qquad (21)
\end{aligned}$$

To prove (20), it is sufficient to prove the next three equations

$$\xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \Big\{ \widehat{f}(\boldsymbol{x}_i;\boldsymbol{w}) - f_0(\boldsymbol{X}_i) \Big\}^2 - \|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2}^2 \Big| = o_p(1), \qquad (22)$$

$$\xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{\widehat{f}(\boldsymbol{X}_i;\boldsymbol{w}) - f^*(\boldsymbol{X}_i;\boldsymbol{w})\}\{Y_i - f_0(\boldsymbol{X}_i)\} \Big| = o_p(1), \qquad (23)$$

and

$$\xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} f^*(\boldsymbol{X}_i;\boldsymbol{w})\{Y_i - f_0(\boldsymbol{X}_i)\} \Big| = o_p(1). \qquad (24)$$

Let $\mathcal{H}_r = \{h_{\boldsymbol{w}} = \{\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\}^2 : \boldsymbol{w} \in \mathcal{W}\}$. By the multiplier inequality, we have

$$\begin{aligned}
&\sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \Big\{ \widehat{f}(\boldsymbol{x}_i;\boldsymbol{w}) - f_0(\boldsymbol{X}_i) \Big\}^2 - \|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2}^2 \Big| \\
=\ & \sup_{h\in\mathcal{H}_r} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} h(\boldsymbol{X}_i) - \mathbb{E}\{h(\boldsymbol{X})|\mathcal{D}\} \Big| \\
=\ & O_p\Big(\sqrt{\tfrac{\log(M)}{n}}\Big), \qquad (25)
\end{aligned}$$

The difference between (6) and (25) is that we set $r = 1$ here. This is because we take the supremum over all $\boldsymbol{w} \in \mathcal{W}$, instead of localizing to a particular neighborhood of $\boldsymbol{w}$. And the bound in (25) is also bigger than that in (6).

Therefore, by Condition 2, (22) can be proved by

$$\xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \Big\{ \widehat{f}(\boldsymbol{x}_i;\boldsymbol{w}) - f_0(\boldsymbol{X}_i) \Big\}^2 - \|\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f_0(\boldsymbol{X})\|_{L^2}^2 \Big| = O_p(\xi_n^{-1}\log(M)/\sqrt{n}) = o_p(1). \qquad (26)$$

For (23), we have

$$\begin{aligned}
& \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{\widehat{f}(\boldsymbol{X}_i;\boldsymbol{w}) - f^*(\boldsymbol{X}_i;\boldsymbol{w})\}\{Y_i - f_0(\boldsymbol{X}_i)\} \Big| \\
= \; & \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \{\widehat{f}(\boldsymbol{X}_i;\boldsymbol{w}) - f^*(\boldsymbol{X}_i;\boldsymbol{w})\}\{Y_i - f_0(\boldsymbol{X}_i)\} \\
& -\mathbb{E}[\{\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f^*(\boldsymbol{X};\boldsymbol{w})\}\{Y - f_0(\boldsymbol{X})\}|\mathcal{D}] \Big| \\
& + \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \mathbb{E}[\{\widehat{f}(\boldsymbol{X};\boldsymbol{w}) - f^*(\boldsymbol{X};\boldsymbol{w})\}\{Y - f_0(\boldsymbol{X})\}|\mathcal{D}] \Big| \\
= \; & \xi_n^{-1} O_p(\sqrt{\log(M)/n}) \\
= \; & o_p(1). \qquad (27)
\end{aligned}$$

For (24), according to the Chebyshev's inequality, we have

$$\begin{aligned}
& \Pr\left\{ \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} f^*(\boldsymbol{X}_i;\boldsymbol{w})\varepsilon_i \Big| > \nu \right\} \\
= \; & \Pr\left\{ \xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \sum_{m=1}^{M} w_m f_m^*(\boldsymbol{X}_i)\varepsilon_i \Big| > \nu \right\} \\
\leq \; & \sum_{m=1}^{M} \Pr\left\{ \xi_n^{-1} \Big| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} f_m^*(\boldsymbol{X}_i)\varepsilon_i \Big| > \nu \right\} \\
\leq \; & \sum_{m=1}^{M} \xi_n^{-2}\nu^{-2} \text{Var}\left\{ \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} f_m^*(\boldsymbol{X}_i)\varepsilon_i \right\} \\
= \; & \xi_n^{-2}\nu^{-2} n^{-1} M \text{Var}\left\{ f_m^*(\boldsymbol{X})\varepsilon \right\} \\
= \; & M\xi_n^{-2}\nu^{-2} n^{-1} \\
= \; & o_p(1) \qquad (28)
\end{aligned}$$

according to Condition 2, which means $\xi_n^{-1} \sup_{\boldsymbol{w}\in\mathcal{W}} \left| \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} f^*(\boldsymbol{X}_i;\boldsymbol{w})\varepsilon_i \right| = o_p(1)$ by Condition 2.

Equations (26)-(28) imply that (22)-(24) hold, and thus we obtain (20). Since (19) and (20) hold, we complete the proof of Theorem 3.

C More Experimental Details

Training resouces. We use the A800 80G for training the models with PyTorch version 2.5.1.

D Additional Results

We report the ratio of WDE MSE to oracle MSE across parameter discrepancies and sample sizes in structural misspecification in Table 5.

We compare with other weighted methods in structural misspecification from Table 6 to Table 8.

E Future Work

While the current theoretical results establish asymptotic error bounds for the proposed weighted deep ensemble estimator, extending these results to non-asymptotic settings remains an important direction for future research. Such analysis could provide tighter guarantees in finite-sample regimes, which are often relevant in practical applications. Another promising avenue is to investigate principled

Table 5: Ratio of WDE MSE to oracle MSE across parameter discrepancies and sample sizes

α	$\Delta = 30000$			$\Delta = 50000$			$\Delta = 100000$		
	N			N			N		
	5000	10000	15000	5000	10000	15000	5000	10000	15000
0.9	1.0066	1.0038	1.0020	1.0061	1.0033	1.0013	1.0076	1.0032	1.0031
0.8	1.0062	1.0031	1.0022	1.0046	1.0022	1.0020	1.0054	1.0034	1.0025
0.7	1.0045	1.0030	1.0023	1.0078	1.0041	1.0021	1.0061	1.0034	1.0021
0.6	1.0056	1.0020	1.0025	1.0087	1.0034	1.0018	1.0068	1.0038	1.0022
0.5	1.0059	1.0022	1.0014	1.0055	1.0036	1.0025	1.0040	1.0030	1.0014
0.4	1.0061	1.0028	1.0034	1.0081	1.0027	1.0025	1.0051	1.0041	1.0010
0.3	1.0058	1.0031	1.0022	1.0057	1.0027	1.0019	1.0046	1.0038	1.0036
0.2	1.0091	1.0037	1.0024	1.0078	1.0027	1.0018	1.0066	1.0028	1.0014
0.1	1.0088	1.0029	1.0017	1.0062	1.0028	1.0015	1.0058	1.0049	1.0021
0.0	1.0069	1.0020	1.0015	1.0036	1.0034	1.0021	1.0079	1.0045	1.0023

Table 6: Relative MSE comparison for sample size $N = 5,000$

Δ	$1-\alpha$	**WDE**	**In-sample Ensemble**	**Greedy Ensemble**
		Mean ± Std	**Mean ± Std**	**Mean ± Std**
30,000	0.1	0.417 ± 0.089	0.421 ± 0.090	0.418 ± 0.091
	0.2	0.408 ± 0.081	0.416 ± 0.081	0.413 ± 0.083
	0.3	0.398 ± 0.083	0.402 ± 0.086	0.400 ± 0.085
	0.4	0.387 ± 0.095	0.394 ± 0.097	0.391 ± 0.098
	0.5	0.397 ± 0.146	0.399 ± 0.146	0.399 ± 0.146
	0.6	0.360 ± 0.048	0.366 ± 0.052	0.363 ± 0.049
	0.7	0.347 ± 0.043	0.356 ± 0.048	0.351 ± 0.044
	0.8	0.341 ± 0.034	0.349 ± 0.037	0.343 ± 0.036
	0.9	0.337 ± 0.031	0.343 ± 0.034	0.340 ± 0.032
	1.0	0.334 ± 0.031	0.342 ± 0.030	0.336 ± 0.031
50,000	0.1	0.437± 0.061	0.442 ± 0.063	0.439 ± 0.063
	0.2	0.476 ± 0.192	0.483 ± 0.190	0.479 ± 0.191
	0.3	0.419 ± 0.067	0.424 ± 0.070	0.423 ± 0.070
	0.4	0.391 ± 0.050	0.396 ± 0.051	0.394 ± 0.051
	0.5	0.399 ± 0.091	0.382 ± 0.041	0.402 ± 0.103
	0.6	0.354 ± 0.041	0.361 ± 0.041	0.357 ± 0.041
	0.7	0.344 ± 0.030	0.350 ± 0.032	0.345 ± 0.031
	0.8	0.339 ± 0.032	0.345 ± 0.035	0.340 ± 0.033
	0.9	0.334 ± 0.027	0.342 ± 0.029	0.337 ± 0.028
	1.0	0.330 ± 0.027	0.335 ± 0.027	0.334 ± 0.027
100,000	0.1	0.348 ± 0.043	0.357 ± 0.044	0.349 ± 0.043
	0.2	0.349 ± 0.042	0.357 ± 0.047	0.352 ± 0.044
	0.3	0.346 ± 0.043	0.355 ± 0.043	0.348 ± 0.043
	0.4	0.342 ± 0.042	0.348 ± 0.043	0.344 ± 0.042
	0.5	0.340 ± 0.039	0.347 ± 0.040	0.342 ± 0.040
	0.6	0.335 ± 0.035	0.343 ± 0.038	0.338 ± 0.036
	0.7	0.332 ± 0.034	0.339 ± 0.037	0.335 ± 0.035
	0.8	0.331 ± 0.026	0.336 ± 0.028	0.331 ± 0.026
	0.9	0.330 ± 0.025	0.337 ± 0.026	0.334 ± 0.027
	1.0	0.327 ± 0.024	0.336 ± 0.021	0.331 ± 0.026

Table 7: Relative MSE comparison for sample size $N = 10,000$

		WDE	In-sample Ensemble	Greedy Ensemble
Δ	$1-\alpha$	**Mean ± Std**	**Mean ± Std**	**Mean ± Std**
30,000	0.1	0.399 ± 0.108	0.404 ± 0.108	0.404 ± 0.110
	0.2	0.396 ± 0.117	0.400 ± 0.116	0.399 ± 0.120
	0.3	0.377 ± 0.094	0.380 ± 0.093	0.379 ± 0.094
	0.4	0.369 ± 0.069	0.371 ± 0.068	0.371 ± 0.069
	0.5	0.351 ± 0.053	0.355 ± 0.051	0.352 ± 0.054
	0.6	0.339 ± 0.044	0.344 ± 0.045	0.342 ± 0.047
	0.7	0.328 ± 0.038	0.331 ± 0.038	0.330 ± 0.038
	0.8	0.327 ± 0.034	0.330 ± 0.034	0.329 ± 0.036
	0.9	0.318 ± 0.032	0.321 ± 0.032	0.319 ± 0.032
	1.0	0.320 ± 0.034	0.323 ± 0.035	0.323 ± 0.036
50,000	0.1	0.399 ± 0.052	0.405 ± 0.052	0.401 ± 0.052
	0.2	0.398 ± 0.063	0.401 ± 0.065	0.400 ± 0.064
	0.3	0.384 ± 0.060	0.388 ± 0.059	0.387 ± 0.061
	0.4	0.396 ± 0.105	0.398 ± 0.104	0.397 ± 0.105
	0.5	0.349 ± 0.048	0.353 ± 0.048	0.351 ± 0.049
	0.6	0.334 ± 0.039	0.338 ± 0.040	0.336 ± 0.040
	0.7	0.324 ± 0.036	0.327 ± 0.036	0.325 ± 0.036
	0.8	0.318 ± 0.034	0.321 ± 0.033	0.319 ± 0.034
	0.9	0.314 ± 0.035	0.320 ± 0.039	0.316 ± 0.036
	1.0	0.311 ± 0.029	0.314 ± 0.030	0.313 ± 0.029

Table 8: Relative MSE comparison for sample size $N = 15,000$

		WDE	In-sample Ensemble	Greedy Ensemble
Δ	$1-\alpha$	**Mean ± Std**	**Mean ± Std**	**Mean ± Std**
30,000	0.1	0.381 ± 0.086	0.384 ± 0.084	0.384 ± 0.089
	0.2	0.381 ± 0.106	0.384 ± 0.105	0.386 ± 0.111
	0.3	0.350 ± 0.052	0.354 ± 0.052	0.351 ± 0.053
	0.4	0.340 ± 0.047	0.343 ± 0.046	0.342 ± 0.047
	0.5	0.328 ± 0.040	0.333 ± 0.039	0.330 ± 0.041
	0.6	0.323 ± 0.032	0.327 ± 0.032	0.326 ± 0.032
	0.7	0.323 ± 0.062	0.326 ± 0.062	0.325 ± 0.065
	0.8	0.309 ± 0.026	0.314 ± 0.029	0.312 ± 0.029
	0.9	0.335 ± 0.106	0.338 ± 0.105	0.336 ± 0.106
	1.0	0.303 ± 0.023	0.307 ± 0.024	0.305 ± 0.024
50,000	0.1	0.388 ± 0.049	0.390 ± 0.049	0.389 ± 0.050
	0.2	0.379 ± 0.052	0.382 ± 0.053	0.383 ± 0.055
	0.3	0.384 ± 0.071	0.387 ± 0.071	0.387 ± 0.071
	0.4	0.360 ± 0.049	0.363 ± 0.049	0.362 ± 0.049
	0.5	0.335 ± 0.040	0.339 ± 0.040	0.337 ± 0.041
	0.6	0.331 ± 0.040	0.334 ± 0.042	0.333 ± 0.042
	0.7	0.312 ± 0.025	0.314 ± 0.025	0.313 ± 0.026
	0.8	0.306 ± 0.024	0.308 ± 0.025	0.306 ± 0.024
	0.9	0.301 ± 0.023	0.303 ± 0.023	0.303 ± 0.023
	1.0	0.302 ± 0.023	0.304 ± 0.022	0.303 ± 0.022
100,000	0.1	0.306 ± 0.027	0.308 ± 0.027	0.307 ± 0.027
	0.2	0.303 ± 0.025	0.306 ± 0.025	0.305 ± 0.026
	0.3	0.304 ± 0.027	0.306 ± 0.026	0.305 ± 0.027
	0.4	0.301 ± 0.023	0.304 ± 0.024	0.302 ± 0.024
	0.5	0.299 ± 0.024	0.302 ± 0.024	0.300 ± 0.024
	0.6	0.297 ± 0.022	0.299 ± 0.022	0.298 ± 0.022
	0.7	0.297 ± 0.021	0.299 ± 0.021	0.298 ± 0.021
	0.8	0.294 ± 0.019	0.296 ± 0.019	0.295 ± 0.019
	0.9	0.297 ± 0.018	0.299 ± 0.018	0.298 ± 0.018
	1.0	0.299 ± 0.020	0.301 ± 0.021	0.300 ± 0.021

approaches for determining the number and composition of candidate models in the ensemble. Understanding how model diversity and ensemble size affect performance could lead to more efficient and adaptive ensemble design strategies.

F REAL-WORLD DATASETS

Datasets. (i) CIFAR-10: 60,000 32x32 color images in 10 classes for image classification. (ii) CIFAR-10-C: A corrupted benchmark derived from CIFAR-10 for evaluating model robustness. It applies 15 common corruptions and each at 5 severity levels. It is widely used to test the out-of-distribution (OOD) robustness of image classifiers and to measure performance degradation under real-world perturbations.

Candidate models. We consider the following neural networks (i) ResNet18 (He et al., 2016), (ii)PreActResNet18, (iii) StochasticDepth18 (Huang et al., 2016), (iv) MobileNetV2 (Sandler et al., 2018), and (v) GoogleNet (Szegedy et al., 2015).

Evaluation metrics. (i) Accuracy (Acc), which reports the fraction of correctly classified samples. (ii) Expected Calibration Error (ECE), which measures how well model confidence aligns with its actual accuracy, with lower values indicating better calibration.

Table 9: Comparison of Accuracy on CIFAR10 and CIFAR10-C

Corruption	DE					EW	WDE
	PreActResNet18	ResNet18	StochasticDepth18	GoogleNet	MobileNetV2		
clean	91.18%	91.10%	91.20%	90.99%	87.81%	92.48%	**92.76%**
brightness	29.20%	27.44%	27.51%	28.61%	25.78%	31.61%	**32.45%**
contrast	16.05%	15.58%	15.36%	12.17%	13.71%	13.21%	**16.53%**
defocus blur	21.07%	18.79%	18.95%	17.95%	19.00%	20.42%	**22.03%**
elastic transform	20.89%	18.02%	17.85%	18.13%	19.95%	20.43%	**21.83%**
fog	16.68%	17.30%	14.43%	14.16%	16.16%	15.68%	**17.48%**
frost	23.75%	22.87%	19.67%	22.75%	20.57%	24.17%	**26.31%**
gaussian blur	19.97%	18.52%	18.91%	17.06%	17.45%	19.33%	**20.40%**
gaussian noise	22.17%	19.85%	18.17%	27.91%	12.45%	27.91%	**28.39%**
glass blur	21.31%	23.65%	19.13%	21.29%	18.90%	24.58%	**25.88%**
impulse noise	21.44%	23.52%	20.45%	26.46%	14.44%	27.45%	**27.54%**
jpeg compression	24.84%	20.98%	19.14%	21.12%	22.09%	23.68%	**25.90%**
motion blur	19.55%	17.20%	18.44%	14.83%	16.16%	17.26%	**20.88%**
pixelate	26.56%	23.28%	21.61%	23.95%	22.71%	26.59%	**27.76%**
saturate	29.53%	28.26%	27.10%	28.26%	26.27%	31.14%	**33.72%**
shot noise	23.00%	21.30%	18.83%	27.49%	14.30%	26.16%	**27.92%**
snow	25.47%	26.29%	22.08%	26.51%	22.27%	29.24%	**31.57%**
spatter	26.18%	26.12%	23.81%	26.14%	22.47%	29.65%	**32.79%**
speckle noise	22.46%	21.27%	18.42%	26.66%	14.49%	26.73%	**27.65%**
zoom blur	**19.27%**	17.47%	17.80%	15.96%	16.78%	18.47%	19.23%

Table 10: Comparison of ECE on CIFAR10 and CIFAR10-C

Corruption	DE					EW	WDE
	PreActResNet18	ResNet18	StochasticDepth18	GoogleNet	MobileNetV2		
clean	0.0599	0.0564	0.0599	0.0583	0.0768	0.0507	**0.0438**
brightness	0.5594	0.6016	0.5594	0.5166	0.4912	0.5545	**0.4475**
contrast	0.4831	0.5829	0.4831	0.6327	0.7247	0.5056	**0.4517**
defocusblur	0.5684	0.6574	0.5684	0.5504	0.5937	0.6046	**0.4743**
elastictransform	0.5881	0.7021	0.5881	0.5715	0.5672	0.6227	**0.4946**
fog	0.5403	0.6363	0.5403	0.6132	0.6530	0.6498	**0.5311**
frost	0.5944	0.6404	0.5944	0.5708	0.5288	0.6138	**0.4933**
gaussian blur	0.5580	0.6273	0.5580	0.5378	0.6269	0.5963	**0.4671**
gaussian noise	0.6481	0.7243	0.6481	0.5931	0.5971	0.6559	**0.5394**
glass blur	0.6309	0.6251	0.6309	0.5893	0.5350	0.6615	**0.4907**
impulse noise	0.6776	0.6509	0.6776	0.5364	0.6188	0.6989	**0.5331**
jpeg compression	0.5825	0.6811	0.5825	0.5798	0.5303	0.6073	**0.4991**
motion blur	0.5636	0.6607	0.5636	0.6221	0.6631	0.5909	**0.5379**
pixelate	0.5703	0.6418	0.5703	0.5532	0.5129	0.5828	**0.4795**
saturate	0.5611	0.5903	0.5611	0.5179	0.4896	0.5666	**0.4341**
shot noise	0.6394	0.6945	0.6394	0.5199	0.5783	0.6510	**0.5142**
snow	0.6056	0.6146	0.6056	0.5410	0.4925	0.6036	**0.4615**
spatter	0.6006	0.6121	0.6006	0.5478	0.4979	0.6087	**0.4383**
speckle noise	0.6452	0.6837	0.6452	0.5210	0.5766	0.6609	**0.5105**
zoom blur	0.5722	0.6627	0.5722	0.5528	0.6187	0.6084	**0.4817**

Additional comparison in pre-trained model ensembles on ImageNet. We apply the pre-trained model used in Wortsman et al. (2022), where different random initializations of the same architecture (CLIP ViT-B/32) are provided. In Table 11, we report the results on ImageNet, as well as the performance averaged over four natural distribution shifts: ImageNetV2, ImageNet-R, ImageNet-Sketch, and ImageNet-A.

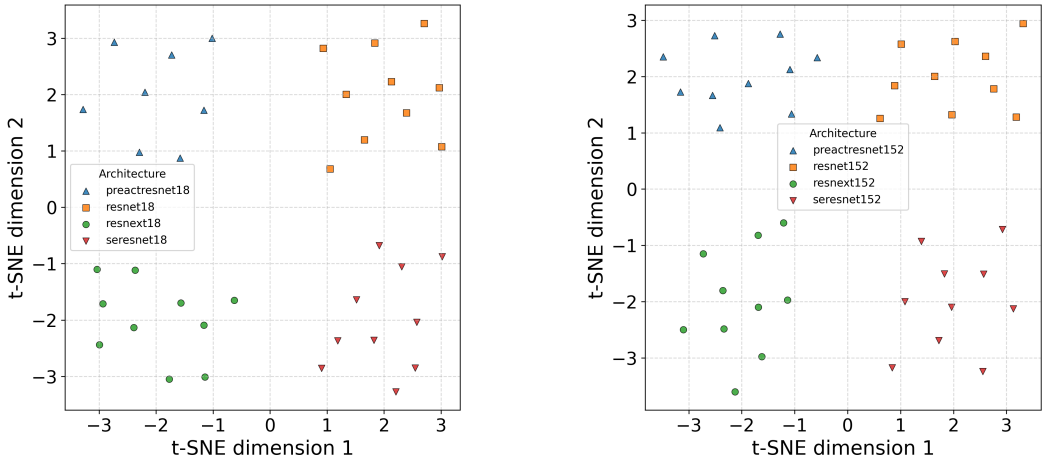
We also note that Under this homogeneous-architecture setting, our weighted ensembling achieves clear improvements.

Method	ImageNet	Dist.shifts
Best individual model	80.39	49.31
Second best model	79.89	45.50
Excluding soup results	–	–
Ensemble	81.19	52.14
Greedy ensemble	81.90	50.19
WDE (homogeneous)	**81.96**	**52.39**

Table 11: Performance comparison on ImageNet and distribution shifts.

G Collective Blindness

Diversity in predictions: We begin by initializing each architecture 10 times, including both shallower models (e.g., depth 18) and deeper ones (e.g., depth 152), and generating predictions on the CIFAR-10 test set. We then visualize the similarity among these predictions using t-SNE. As shown in the Figure 3, models with the same architecture tend to cluster together, whereas different architectures produce more diverse prediction.



(a) Shallower model

(b) Deeper model

Figure 3: t-SNE visualization of CIFAR-10 predictions

Evaluation on collective blindness. To quantify this effect, we introduce three metrics that measure different levels of error correlation within the ensemble:

For each model m, We define the error set on slightly hard samples to ensure that correlations are measured only on non-trivial mistakes: $E_m = \{n : |\hat{f}_m(\mathbf{X}_n) - Y_n| > \tau\}$, and for the ensemble, $E_{\text{ens}} = \{n : |\hat{f}(\mathbf{X}_n; \mathbf{w}) - Y_n| > \tau\}$. To measure how two models make errors on the same samples, we define their error correlation $\text{CoErr}(i, j) = \frac{|E_i \cap E_j|}{|E_i \cup E_j|}$, and for each model m, the correction with the ensemble is $\text{CoErr}(\text{ens}, m) = \frac{|E_{\text{ens}} \cap E_m|}{|E_{\text{ens}} \cup E_m|}$.

(i) Model–Model Error Correlation (mCo-Err). The average error correction across all model pairs is

$$\text{mCoErr} = \frac{1}{M(M-1)} \sum_{i \neq j} \text{CoErr}(i, j),$$

which measures the typical correction of errors across individual models.

(ii) Ensemble–Model Error Correlation (EnCo-Err). The weighted ensemble–model correction is

$$\mathrm{EnCoErr} = \sum_{m=1}^{M} w_m \,\mathrm{CoErr}(\mathrm{ens}, m),$$

which measures the correction between ensemble errors and those of its base models.

(iii) Tail Error Rate (TER).

$$\mathrm{TER} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\Big(|\hat{f}(\mathbf{X}_n; \mathbf{w}) - Y_n| > \tau\Big),$$

which measures the proportion of predictions that incur non-trivial errors.

We report the corresponding results under variable, inherent and structural misspecification in Tables 12 - 17. These metrics jointly indicate that (a) collective blindness is weaker in diverse architectures than in single-architecture settings. (b) weighted ensembles reduce error correlations with individual models, and (c) extreme prediction errors become less frequent. These results offer a quantitative assessment of collective blindness and demonstrate that WDE effectively mitigates it.

Task	# Missing	mCoErr ↓		EnCoErr ↓		
		Diverse	Same	DE	EW	WDE
Nested	0	0.6837 ± 0.0565	0.7669 ± 0.0352	0.8220 ± 0.0748	0.7765 ± 0.0366	0.5467 ± 0.0691
	1	0.7137 ± 0.0654	0.7835 ± 0.0407	0.8303 ± 0.0598	0.7774 ± 0.0405	0.5854 ± 0.0788
	3	0.7850 ± 0.0535	0.8210 ± 0.0316	0.8864 ± 0.0626	0.8262 ± 0.0320	0.6819 ± 0.0655
	5	0.8335 ± 0.0162	0.8507 ± 0.0216	0.9158 ± 0.0598	0.8616 ± 0.0457	0.7481 ± 0.0210
	7	0.8935 ± 0.0430	0.9120 ± 0.0257	0.9532 ± 0.0358	0.9342 ± 0.0263	0.8376 ± 0.0592
Interaction	0	0.5813 ± 0.0968	0.6711 ± 0.0748	0.7009 ± 0.1175	0.6825 ± 0.0666	0.4435 ± 0.0802
	1	0.7014 ± 0.0851	0.7743 ± 0.0346	0.7709 ± 0.0431	0.7577 ± 0.0352	0.5798 ± 0.0852
	3	0.7674 ± 0.0536	0.8206 ± 0.0235	0.8509 ± 0.0262	0.8415 ± 0.0355	0.6566 ± 0.0667
	5	0.7965 ± 0.0475	0.8459 ± 0.0285	0.9014 ± 0.0394	0.8599 ± 0.0380	0.6991 ± 0.0633
	7	0.8959 ± 0.0436	0.9208 ± 0.0421	0.9517 ± 0.0417	0.9313 ± 0.0452	0.8471 ± 0.0673
Nonlinear	0	0.6294 ± 0.0348	0.7030 ± 0.0596	0.7324 ± 0.0733	0.7067 ± 0.0620	0.4797 ± 0.0359
	1	0.6990 ± 0.0682	0.7484 ± 0.0668	0.7608 ± 0.0391	0.7358 ± 0.0655	0.5626 ± 0.0860
	3	0.7963 ± 0.0896	0.8589 ± 0.0373	0.8804 ± 0.0511	0.8431 ± 0.0406	0.7054 ± 0.0924
	5	0.8736 ± 0.0364	0.8882 ± 0.0395	0.9429 ± 0.0528	0.9039 ± 0.0463	0.8071 ± 0.0513
	7	0.9102 ± 0.0191	0.9320 ± 0.0204	0.9562 ± 0.0213	0.9489 ± 0.0122	0.8582 ± 0.0255

Table 12: Comparison of mCoErr and EnCoErr under variable misspecification when $N = 5000$.

Task	# Missing	TER ↓		
		DE	EW	WDE
Nested	0	0.2311 ± 0.0207	0.2118 ± 0.0118	0.2008 ± 0.0131
	1	0.2332 ± 0.0184	0.2220 ± 0.0138	0.2079 ± 0.0134
	3	0.2474 ± 0.0117	0.2398 ± 0.0076	0.2368 ± 0.0126
	5	0.2616 ± 0.0102	0.2608 ± 0.0113	0.2559 ± 0.0137
	7	0.2781 ± 0.0100	0.2749 ± 0.0113	0.2742 ± 0.0104
Interaction	0	0.0359 ± 0.0082	0.0325 ± 0.0061	0.0315 ± 0.0053
	1	0.1012 ± 0.0088	0.0988 ± 0.0089	0.0979 ± 0.0083
	3	0.1844 ± 0.0119	0.1834 ± 0.0131	0.1802 ± 0.0094
	5	0.2157 ± 0.0084	0.2167 ± 0.0112	0.2160 ± 0.0135
	7	0.2287 ± 0.0084	0.2294 ± 0.0090	0.2292 ± 0.0074
Nonlinear	0	0.0377 ± 0.0050	0.0384 ± 0.0035	0.0357 ± 0.0044
	1	0.0636 ± 0.0289	0.0657 ± 0.0286	0.0633 ± 0.0281
	3	0.1148 ± 0.0238	0.1150 ± 0.0242	0.1138 ± 0.0257
	5	0.1735 ± 0.0353	0.1730 ± 0.0356	0.1718 ± 0.0352
	7	0.2183 ± 0.0305	0.2169 ± 0.0279	0.2164 ± 0.0293

Table 13: Comparison of TER under variable misspecification when $N = 5000$.

Task	N	mCoErr ↓		EnCoErr ↓		
		Diverse	Same	DE	EW	WDE
Infinite	5000	0.3420 ± 0.0267	0.3775 ± 0.0131	0.6000 ± 0.0516	0.5804 ± 0.0501	0.3126 ± 0.0277
	10000	0.2649 ± 0.0292	0.3257 ± 0.0282	0.4880 ± 0.0515	0.4457 ± 0.0477	0.1954 ± 0.0293
	15000	0.1869 ± 0.0147	0.2900 ± 0.0134	0.4458 ± 0.0492	0.4393 ± 0.0486	0.1679 ± 0.0117
Square	5000	0.6582 ± 0.1324	0.7105 ± 0.1117	0.9472 ± 0.0430	0.9313 ± 0.0668	0.6359 ± 0.1310
	10000	0.7795 ± 0.2307	0.8026 ± 0.2062	0.9543 ± 0.0669	0.9597 ± 0.0520	0.6942 ± 0.1659
	15000	0.8015 ± 0.2164	0.8210 ± 0.2017	0.9840 ± 0.0210	0.9676 ± 0.0455	0.6263 ± 0.0934

Table 14: Comparison of mCoErr and EnCoErr under inherent misspecification.

Task	N	TER ↓		
		DE	EW	WDE
Infinite	5000	0.1304 ± 0.0110	0.1104 ± 0.0037	0.0037 ± 0.0001
	10000	0.0892 ± 0.0072	0.0826 ± 0.0023	0.0024 ± 0.0005
	15000	0.0769 ± 0.0070	0.0632 ± 0.0006	0.0014 ± 0.0002
Square	5000	0.3034 ± 0.0517	0.2945 ± 0.0403	0.2466 ± 0.0576
	10000	0.2906 ± 0.0307	0.2792 ± 0.0346	0.2313 ± 0.0453
	15000	0.2971 ± 0.0552	0.2854 ± 0.0277	0.2222 ± 0.0271

Table 15: Comparison of TER under inherent misspecification for varying sample sizes N.

Δ	$1-\alpha$	mCoErr ↓		EnCoErr ↓		
		Diverse	Same	DE	EW	WDE
30000	0.1	0.7060 ± 0.0547	0.7413 ± 0.0939	0.8054 ± 0.1079	0.7808 ± 0.0543	0.4887 ± 0.0839
	0.2	0.7124 ± 0.0677	0.7648 ± 0.1372	0.7883 ± 0.0903	0.8091 ± 0.0833	0.5250 ± 0.0875
	0.3	0.7012 ± 0.0927	0.7624 ± 0.1667	0.7968 ± 0.1450	0.7152 ± 0.1279	0.4903 ± 0.0708
	0.4	0.6753 ± 0.0287	0.6914 ± 0.0353	0.7552 ± 0.0500	0.7374 ± 0.0307	0.4697 ± 0.0772
	0.5	0.6855 ± 0.0798	0.7262 ± 0.1486	0.7733 ± 0.1060	0.7708 ± 0.0918	0.5121 ± 0.1009
	0.6	0.7473 ± 0.1006	0.7841 ± 0.1530	0.8098 ± 0.1163	0.7734 ± 0.1133	0.5152 ± 0.1103
	0.7	0.7161 ± 0.1135	0.7465 ± 0.1424	0.8243 ± 0.1241	0.7639 ± 0.0996	0.4345 ± 0.0399
	0.8	0.7240 ± 0.1123	0.7700 ± 0.1610	0.8369 ± 0.1331	0.8326 ± 0.1305	0.5365 ± 0.0829
	0.9	0.7191 ± 0.1260	0.7267 ± 0.1506	0.8246 ± 0.1291	0.8980 ± 0.1400	0.5612 ± 0.1236
	1.0	0.7032 ± 0.0575	0.7119 ± 0.1166	0.7635 ± 0.0954	0.7809 ± 0.1000	0.5012 ± 0.1087
50000	0.1	0.7612 ± 0.0834	0.8114 ± 0.1346	0.8323 ± 0.1084	0.9278 ± 0.0777	0.5572 ± 0.1032
	0.2	0.7273 ± 0.0802	0.8123 ± 0.1617	0.8272 ± 0.1297	0.8264 ± 0.0742	0.5081 ± 0.0566
	0.3	0.7474 ± 0.1498	0.8093 ± 0.1650	0.8514 ± 0.1299	0.9076 ± 0.1326	0.5570 ± 0.1704
	0.4	0.7313 ± 0.1133	0.7505 ± 0.1365	0.8420 ± 0.1185	0.7808 ± 0.1083	0.5016 ± 0.0989
	0.5	0.7533 ± 0.1086	0.7914 ± 0.1482	0.8281 ± 0.1172	0.8556 ± 0.1043	0.5234 ± 0.1279
	0.6	0.7196 ± 0.1370	0.7357 ± 0.1439	0.7859 ± 0.1278	0.7754 ± 0.1375	0.4910 ± 0.1022
	0.7	0.7331 ± 0.1097	0.7871 ± 0.1531	0.8479 ± 0.1241	0.7444 ± 0.1161	0.4888 ± 0.0877
	0.8	0.7503 ± 0.1289	0.7773 ± 0.1593	0.8249 ± 0.1233	0.7848 ± 0.1387	0.4843 ± 0.0969
	0.9	0.6789 ± 0.0921	0.6962 ± 0.1296	0.8073 ± 0.1111	0.7546 ± 0.0664	0.4915 ± 0.1199
	1.0	0.7032 ± 0.0575	0.7119 ± 0.1166	0.7635 ± 0.0954	0.7809 ± 0.1000	0.5012 ± 0.1087
100000	0.1	0.6972 ± 0.0614	0.7365 ± 0.1425	0.7828 ± 0.1284	0.7319 ± 0.0341	0.4785 ± 0.1068
	0.2	0.6764 ± 0.0604	0.6667 ± 0.0677	0.7569 ± 0.1221	0.6963 ± 0.0618	0.4494 ± 0.0851
	0.3	0.6821 ± 0.0627	0.7055 ± 0.1094	0.7603 ± 0.1318	0.7845 ± 0.0961	0.4989 ± 0.1032
	0.4	0.7349 ± 0.0615	0.7698 ± 0.1324	0.8319 ± 0.1166	0.7595 ± 0.1053	0.4746 ± 0.0602
	0.5	0.6648 ± 0.0564	0.7063 ± 0.1106	0.7871 ± 0.1185	0.7393 ± 0.0723	0.4433 ± 0.0573
	0.6	0.7252 ± 0.0907	0.7847 ± 0.1508	0.8142 ± 0.1339	0.7955 ± 0.0997	0.4958 ± 0.0939
	0.7	0.6417 ± 0.0349	0.6840 ± 0.1144	0.7664 ± 0.1049	0.7311 ± 0.0324	0.4406 ± 0.0701
	0.8	0.6768 ± 0.0569	0.7048 ± 0.1127	0.8013 ± 0.1238	0.7503 ± 0.0461	0.4779 ± 0.1099
	0.9	0.6833 ± 0.0752	0.7117 ± 0.1135	0.7576 ± 0.1141	0.7071 ± 0.0697	0.4727 ± 0.1140
	1.0	0.7032 ± 0.0575	0.7119 ± 0.1166	0.7635 ± 0.0954	0.7809 ± 0.1000	0.5012 ± 0.1087

Table 16: Comparison of mCoErr and EnCoErr under structural misspecification when $N = 5000$.

Δ	$1-\alpha$	TER ↓		
		DE	EW	WDE
30000	0.1	0.1504 ± 0.0443	0.1191 ± 0.0363	0.1159 ± 0.0352
	0.2	0.1405 ± 0.0505	0.1160 ± 0.0337	0.1101 ± 0.0325
	0.3	0.1502 ± 0.0458	0.1061 ± 0.0178	0.1021 ± 0.0178
	0.4	0.1334 ± 0.0263	0.1028 ± 0.0141	0.0985 ± 0.0146
	0.5	0.1356 ± 0.0482	0.0984 ± 0.0141	0.0947 ± 0.0152
	0.6	0.1417 ± 0.0511	0.0958 ± 0.0157	0.0923 ± 0.0150
	0.7	0.1367 ± 0.0386	0.0923 ± 0.0152	0.0874 ± 0.0145
	0.8	0.1325 ± 0.0595	0.0943 ± 0.0186	0.0910 ± 0.0197
	0.9	0.1374 ± 0.0701	0.1033 ± 0.0377	0.0999 ± 0.0382
	1.0	0.1146 ± 0.0350	0.0927 ± 0.0144	0.0882 ± 0.0165
50000	0.1	0.1743 ± 0.0556	0.1441 ± 0.0416	0.1392 ± 0.0421
	0.2	0.1629 ± 0.0421	0.1261 ± 0.0192	0.1212 ± 0.0214
	0.3	0.1704 ± 0.0614	0.1382 ± 0.0574	0.1338 ± 0.0568
	0.4	0.1636 ± 0.0491	0.1175 ± 0.0254	0.1136 ± 0.0265
	0.5	0.1583 ± 0.0606	0.1202 ± 0.0470	0.1165 ± 0.0471
	0.6	0.1436 ± 0.0513	0.1044 ± 0.0279	0.1018 ± 0.0267
	0.7	0.1508 ± 0.0535	0.0972 ± 0.0160	0.0945 ± 0.0177
	0.8	0.1498 ± 0.0544	0.0917 ± 0.0117	0.0878 ± 0.0137
	0.9	0.1220 ± 0.0397	0.0874 ± 0.0155	0.0832 ± 0.0158
	1.0	0.1146 ± 0.0350	0.0927 ± 0.0144	0.0882 ± 0.0165
100000	0.1	0.1193 ± 0.0348	0.0873 ± 0.0210	0.0825 ± 0.0175
	0.2	0.1233 ± 0.0366	0.0900 ± 0.0183	0.0870 ± 0.0186
	0.3	0.1279 ± 0.0471	0.0909 ± 0.0207	0.0863 ± 0.0217
	0.4	0.1317 ± 0.0546	0.0937 ± 0.0211	0.0912 ± 0.0208
	0.5	0.1227 ± 0.0363	0.0883 ± 0.0183	0.0848 ± 0.0170
	0.6	0.1376 ± 0.0495	0.0946 ± 0.0211	0.0902 ± 0.0204
	0.7	0.1071 ± 0.0206	0.0867 ± 0.0134	0.0822 ± 0.0149
	0.8	0.1118 ± 0.0311	0.0864 ± 0.0119	0.0809 ± 0.0124
	0.9	0.1150 ± 0.0388	0.0891 ± 0.0150	0.0836 ± 0.0136
	1.0	0.1146 ± 0.0350	0.0927 ± 0.0144	0.0882 ± 0.0165

Table 17: Comparison of TER under structural misspecification when $N = 5000$.