Temporal Experts Averaging for Large-Scale Temporal Domain Generalization

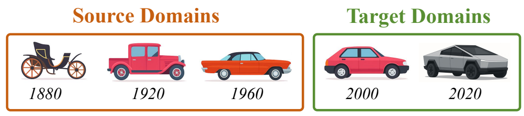
Anonymous ACL submission

Abstract

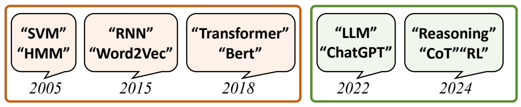
Temporal Domain Generalization (TDG) aim at generalizing across temporal distribution shifts, e.g., lexical change over time, by predicting future models. Due to the prohibitive full model prediction cost on large-scale scenarios, recent TDG works only predict the classifier, but this limits generalization potential by failing to adjust other model components. To address this, we propose Temporal Experts Averaging (TEA), a novel TDG framework based on weight averaging that adjusts the entire model to maximize generalization potential while maintaining minimal computational overhead when scaling to large-scale datasets and models. Our theoretical analysis of weight averaging for TDG guided us to develop two steps that enhance generalization to future domains. First, we create expert models with functional diversity yet parameter similarity by fine-tuning a domain-agnostic base model on individual temporal domains while constraining weight changes. Second, we optimize the bias-variance tradeoff through adaptive averaging coefficients derived from modeling temporal weight trajectories in a principal component subspace and weighting experts based on their projected proximity to future domains in the subspace. Extensive experiments across 7 TDG benchmarks, 5 models, and 2 TDG settings reports TEA outperforms prior TDG methods by up to 69% while being up to 60x more efficient.

1 Introduction

Temporal Domain Generalization (TDG) (ying Bai et al., 2022; Nasery et al., 2021; Qin et al., 2022; Xie et al., 2024c,a; Yong et al., 2023; Xie et al., 2024b) aims at generalizing to unseen future data under temporal distribution shift without retraining the models, as illustrated in Fig. 1. Unlike traditional Domain Generalization (DG) lacking target domain information (Li et al., 2017a; Muandet et al., 2013; Li et al., 2018a,b), TDG could leverage temporal patterns for prediction, such as



(a) Temporal distribution shift in vehicle appearance.



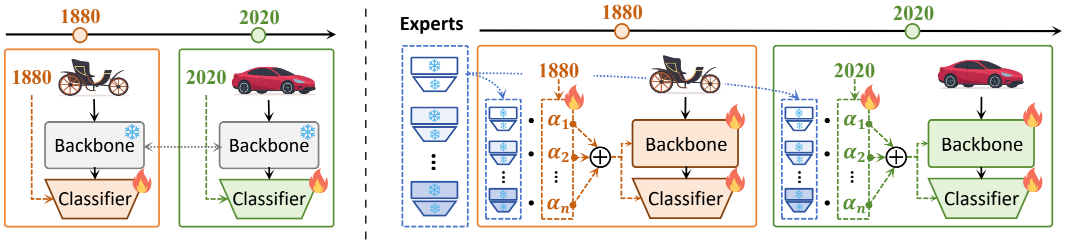
(b) Temporal distribution shift in NLP paper keywords.

Figure 1: Examples of temporal domain generalization (TDG) span both vision and language tasks. TDG aims at enabling models trained on historical data to directly generalize to future data without retraining.

predicting NLP research trends (Yao et al., 2022a), to better adapt the models for future domains. However, prior work faces limitations in scaling. Early brute-force approaches predict entire models but encounter prohibitive computational costs on large-scale models and datasets (Nasery et al., 2021; ying Bai et al., 2022; Qin et al., 2022). As shown in Fig. 2a, recent methods improve efficiency by only predicting classifier (Xie et al., 2024c,b), but sacrifice generalization potential by failing to adjust other model components. In large-scale benchmarks (Yao et al., 2022a; Lin et al., 2022), these methods struggle to surpass basic ERM baselines.

To address the scaling challenges, we propose Temporal Experts Averaging (TEA), a TDG framework based on weight averaging (WA) that predicts the averaging coefficients of temporal experts for future domains. As standard WA methods for DG, e.g., (Cha et al., 2021; Rame et al., 2022; Wortsman et al., 2022), lack mechanisms to exploit temporal patterns in TDG, we identify two key strategies to leverage temporal patterns in WA through a bias-variance-covariance-locality decomposition analysis of generalization error: a) creating weights with functional diversity yet parameter similarity, and b) optimizing averaging coefficients to achieve better bias-variance tradeoffs than uniform averaging.

Thus, our TEA first creates a set of temporal



(a) Classifier-only TDG framework (b) Our Temporal Expert Averaging (TEA) framework

Figure 2: TDG framework comparison. **(a) Classifier-only TDG** (Xie et al., 2024c,b) only predicts future classifiers to reduce computational costs in large-scale scenarios, but limits generalization potential by neglecting other model components. **(b) Our Temporal Expert Averaging (TEA)** enables higher generalization potential by adjusting the entire model through predicting future averaging coefficients of temporal experts capturing diverse functionalities. The low-dimensional nature of these coefficients ensures TEA's efficiency in large-scale scenarios.

experts with functional diversity yet parameter similarity by training a domain-agnostic base model on all source domains, followed by constrained incremental fine-tuning on each individual domain. To create adaptive averaging coefficients, we then extract principal components from the deviations between expert weights and the base model, creating a low-dimensional subspace to model temporal weight trajectories. This enables forecasting future domain positions and averaging experts based on their projected proximity to the future domain. This enables TEA to temporally-adapt all model parameters with computational costs comparable to standard ERM training, offering higher generalization potential than merely adjusting the classifier.

The superiority of TEA is demonstrated through comprehensive evaluation across 7 diverse TDG benchmarks and 5 different models, covering both vision and language tasks. Beyond standard TDG with simultaneous access to all source domains, we also evaluate on Continual Domain Generalization over Temporal Drift (CDGTD) settings, where new domains arrive sequentially. Across this extensive evaluation, TEA consistently achieves new state-of-the-art results, outperforming prior TDG methods by up to 69% while being up to 60x more efficient.

Our contributions can be summarized as follows:

- We propose TEA, a novel weight-averaging-based TDG framework that efficiently enhances generalization across temporal shifts with broad model/dataset compatibility.
- We provide valuable theoretical insights on the under-explored WA-TDG integration, design our method based on these insights, and validate our insights and method through superior generalization performance across various benchmarks.
- We enhance TDG evaluation comprehensiveness by addressing both TDG and CDGTD, unlike prior work that typically focused on just one setting. We also introduce CLEAR-10 and CLEAR-100 as new evaluation benchmarks for TDG.

2 Related Work

Temporal Domain Generalization (TDG) (Ortiz-Jiménez et al., 2019; Mancini et al., 2019; Wang et al., 2020; ying Bai et al., 2022; Nasery et al., 2021; Zeng et al., 2023; Wang et al., 2022; Xie et al., 2024c,a; Yong et al., 2023; Xie et al., 2024b) exploits temporal patterns in ordered domains with smooth distribution shifts to enhance generalization to future domains. Early approaches like GI (Nasery et al., 2021) and DRAIN (ying Bai et al., 2022) predict entire model parameters but face computational challenges with large-scale models, while recent methods like EvoS (Xie et al., 2024c) and W-Diff (Xie et al., 2024b) reduce costs by only adjusting classifiers, potentially limiting generalization. TDG encompasses multiple settings: the original setting with simultaneous access to all source domains, Continual Domain Generalization over Temporal Drift (CDGTD) with sequentially available domains, and Continuous Temporal Domain Generalization (CTDG) for continuously distributed temporal data. We focus on the original TDG and CDGTD settings as CTDG remains impractical for most realistic benchmarks.

Domain Adaptation and Generalization. Enabling models to perform well on out-of-distribution (OOD) data has been a crucial challenge in machine learning. Two specific tasks highly relevant to our work are Domain Adaptation (DA) and Domain Generalization (DG). DA methods (Saenko et al., 2010; Sun et al., 2015; Sun and Saenko, 2016; Gong et al., 2012; Tzeng et al., 2017; Li et al., 2016) typically adapt models against distribution shift by utilizing data from the target domain. In contrast, DG methods (Li et al., 2017a; Muandet et al., 2013; Li et al., 2018a, 2017b; Gulrajani and Lopez-Paz, 2021; Li et al., 2018b, 2019) operate without target domain information, solely leveraging source domain patterns to enhance OOD generalization.

Weight Averaging (WA) (Cha et al., 2021, 2022; Rame et al., 2022; Wortsman et al., 2022) proves effective for Domain Generalization, with DiWA (Rame et al., 2022) showing reduced variance against marginal distribution shifts. While WA is also used in Multi-task Learning (Ilharco et al., 2022b; Yadav et al., 2023; Ortiz-Jimenez et al., 2023; Wang et al., 2024; Stoica et al., 2023) with our design partly inspired by task arithmetic (Ilharco et al., 2022a), fundamental differences between MTL and TDG make direct application impractical.

3 Temporal Experts Averaging

Let $\mathcal{X}$ be the input space, $\mathcal{Y}$ the label space, $\ell : \mathcal{Y}^2 \to \mathbb{R}^+$ a loss function, $\{D_i\}$ a sequence of domains with timestamps $t_i \in \mathcal{T}$ and distributions p_i. Given source domains $\mathbf{D}_S = \{D_i\}_{i=1}^S$, where $t_1 < \ldots < t_S$, and a neural network $f(\cdot, \theta) : \mathcal{X} \to \mathcal{Y}$ with weights θ, we aim to minimize the generalization error at future time $t_f > t_S$:

$$\mathcal{E}_f(\theta) = \mathbb{E}_{(x,y)\sim p_f}[\ell(f(x,\theta), y)]. \quad (1)$$

We obtain the weights of S temporal expert models $\{\theta_i\}_{i=1}^S = \{\theta(l_i)\}_{i=1}^S$, where θ_i is optimized for domain D_i while using data from other domains, with learning procedure noted as $l_i = \{\{D_i\}_{i=1}^S, t_i, c\}$ and other configurations (e.g., hyper-parameters) as c. We leverage temporal patterns to derive adaptive coefficients $\{\alpha_i\}_{i=1}^S$, where $\sum_{i=1}^S \alpha_i = 1$ and $\alpha_i \geq 0$, for combining expert weights into the final weight θ_{TEA}, formulated as:

$$f_{\text{TEA}} \triangleq f(\cdot, \theta_{\text{TEA}}),$$

$$\theta_{\text{TEA}} \triangleq \sum_{i=1}^S \alpha_i \left(\{t_i\}_{i=1}^S, \{\theta_i\}_{i=1}^S, t_f\right) \cdot \theta_i. \quad (2)$$

To leverage temporal shift patterns for reducing future generalization error, we gain insight into TEA through theoretical analysis in Section 3.1. Following the insights, we implement our TEA by creating functionally diverse yet parametrically similar experts $\{\theta_i\}_{i=1}^S$ (Section 3.2) and determining coefficients $\{\alpha_i\}_{i=1}^S$ based on expert-future proximity (Section 3.3). Section 3.4 details TEA's application to the CDGTD setting.

3.1 Theoretical Analysis and Insights

To gain insight into TEA, we extend DiWA's (Rame et al., 2022) theoretical analysis developed for DG to our WA-TDG integration setting. Since our primary goal is to guide method design, we briefly summarize the theoretical analysis and results in the main text, with complete derivations and proofs available in the supplementary.

Bias-variance-covariance-locality Decomposition. Similar to DiWA (Rame et al., 2022), we introduce the bias-variance-covariance-locality (BVCL) decomposition of generalization error for TDG and TEA by leveraging the similarity between averaging in weight space and function space. Denoting $\mathbb{E}_f = \mathbb{E}_{(x,y)\sim p_f}$, $\mathbf{l} = \{l_1, \ldots, l_S\}$, $\bar{f}_i(x) = \mathbb{E}_{l_i}[f(x, \theta(l_i))]$, $\text{bias}_i = y - \bar{f}_i(x)$, $\text{var}_i = \mathbb{E}_{l_i}\left[\left(f(x,\theta(l_i)) - \bar{f}_i(x)\right)^2\right]$, $\text{cov}_{i,j} = \mathbb{E}_{l_i,l_j}\left[\left(f(x,\theta(l_i)) - \bar{f}_i(x)\right)\left(f(x,\theta(l_j)) - \bar{f}_j(x)\right)\right]$ and $\Delta_{\{\theta\}} = \max_{i=1}^S \|\theta_i - \theta_{\text{TEA}}\|_2$, the expected generalization error on future timestamp t_f of $\theta_{\text{TEA}} = \sum_{i=1}^S \alpha_i \theta_i$ over the joint distribution of the learning procedures is:

$$\mathbb{E}_{\mathbf{l}}[\mathcal{E}_f(\theta_{\text{TEA}})] = \mathbb{E}_f\left[\mathcal{B} + \mathcal{V} + \mathcal{C}\right] + O(\bar{\Delta}^2), \quad (3)$$

$$\mathcal{B} = \left(\sum_{i=1}^S \alpha_i \cdot \text{bias}_i\right)^2, \quad \mathcal{V} = \sum_{i=1}^S \alpha_i^2 \cdot \text{var}_i,$$

$$\mathcal{C} = \sum_{i \neq j} \alpha_i \alpha_j \text{cov}_{i,j}, \quad \bar{\Delta}^2 = \mathbb{E}_{\mathbf{l}}\left[\Delta_{\{\theta\}}\right].$$

To reduce future generalization error in Equation 3, we can control learning procedures $\{l_i\}_{i=1}^S$ affecting expert weights $\{\theta_i\}_{i=1}^S$ and modify averaging coefficients $\{\alpha_i\}_{i=1}^S$, which constitute the key differences between our TEA and WA for typical DG. While finding optimal solutions remains challenging due to real-world complexity, qualitative analysis provides valuable insights summarized as two tradeoffs implemented through experts and coefficients respectively. Detailed analysis and assumptions are in the supplementary material.

Insight 1 *Tradeoff between Functional Diversity and Parameter Similarity among Experts.* Covariance $\mathcal{C}$ reduction necessitates functional diversity among experts, while the locality constraint $\bar{\Delta}^2$ demands parameter similarity among experts.

Insight 2 *Tradeoff between Bias and Variance via Averaging Coefficients.* Reducing variance $\mathcal{V}$ requires averaging weights evenly, while reducing bias $\mathcal{B}$ demands concentrating coefficients on experts with lower bias magnitudes on future data.

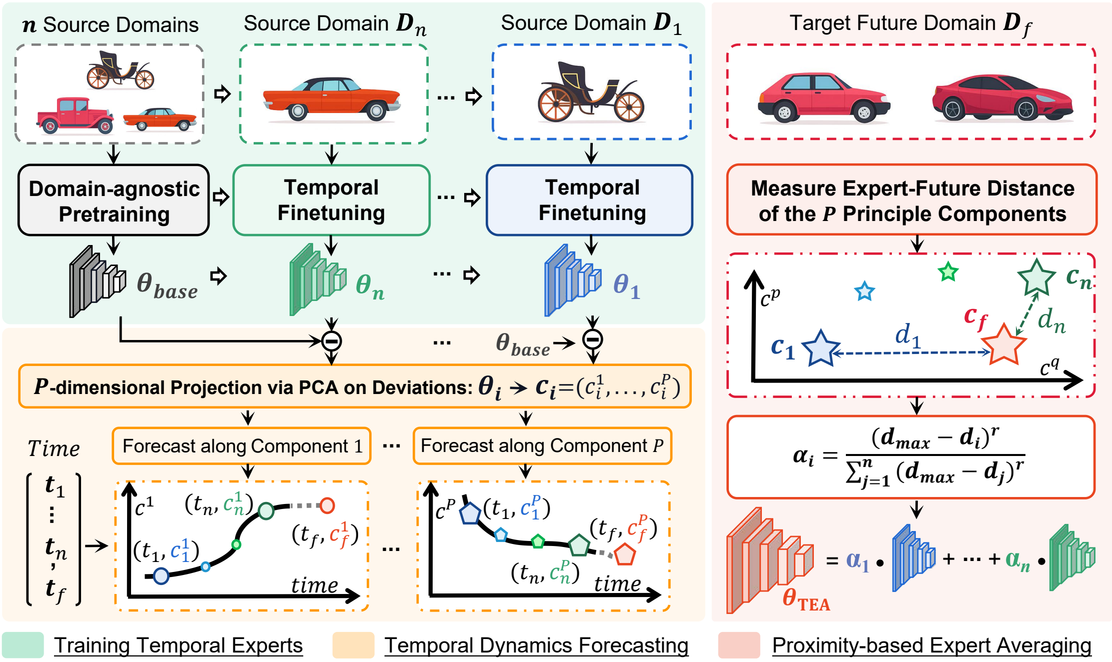


Figure 3: Overview of our TEA framework. Firstly, we obtain a base model θ_{base} through domain-agnostic pretraining on all source domains, then derive experts $\theta_1, \dots, \theta_n$ via constrained domain-specific incremental finetuning in reverse temporal order. Secondly, we apply PCA to expert weight deviations $\{\theta_i - \theta_{\text{base}}\}_{i=1}^n$, forecast future positions along the P most significant components with Autoregressive Integrated Moving Average (ARIMA), effectively projecting experts into a low-dimensional space for prediction. Finally, we assign averaging coefficients based on projected expert-future proximity, where closer experts receive higher coefficients.

3.2 Training Temporal Experts

TDG assumes smooth temporal distribution shifts with moderate changes between adjacent domains. This allows an expert to be fine-tuned for learning domain-specific functionality of neighboring domains with minimal parameter adjustments. Therefore, we can satisfy Insight 1 through incremental domain-specific fine-tuning while constraining minimal parameter changes. However, a prerequisite is that experts must have already thoroughly learned the intrinsic distribution.

A "Pretraining-Finetuning" approach is adopted for our expert training that efficiently generates diverse temporal experts with similar parameters. The overall process can be formulated as:

$$\theta_{\text{base}} = \theta_{S+1} = \theta(l_{\text{ERM}}(\mathbf{D}_S)), \qquad \text{(Pretraining)}$$
$$\theta_i = \theta(l_{\text{SI}}(\{D_{t_i}\}, \theta_{i+1})), \qquad \text{(Finetuning)}$$

where $i \in \{S, \dots, 1\}, \mathbf{D}_S = \{D_1, \dots, D_S\}$, l_{ERM} represents the Empirical Risk Minimization (ERM) learning process, and l_{SI} represents the learning process with Synaptic Intelligence (SI) (Zenke et al., 2017) constraining parameter changes.

Pretraining aims to capture intrinsic, time-invariant distributions. We apply standard ERM training with source domains $\{D_1, \dots, D_S\}$. No temporal information is incorporated during this stage. Unlike WA for DG (Cha et al., 2021, 2022; Rame et al., 2022; Wortsman et al., 2022), we update normalization layers during pretraining to prevent underfitting, as TDG exhibits smaller distribution differences than DG settings.

Temporal Finetuning sequentially adapts the base model to capture time-varying distributions. We freeze the normalization layers and proceed in reverse temporal order ($t_S \rightarrow \dots \rightarrow t_1$) in this stage. For each domain D_i, we uniformly sample K weights during finetuning, $\{\theta_i^k\}_{k=1}^K$, and expert θ_i is obtained by uniform averaging: $\theta_i = \sum_{k=1}^K \frac{1}{K}\theta_i^k$

SI (Zenke et al., 2017) is used to constrain parameter changes, which also prevent catastrophic forgetting of intrinsic distributions. Other continual learning methods can also be used. Since later fine-tuning stages are influenced by previous ones, we use reverse temporal order (recent to earliest) to better capture distributions from recent domains that more likely resemble future test domains under smooth distribution shift assumptions.

3.3 Adaptive Weight Averaging

If future weights are available, we could satisfy Insight 2 by assigning coefficients based on expert-future weight proximity, although we could directly use future weights. However, precisely predicting the future in high-dimensional weight space is both hard and computationally prohibitive. As we only need relative rankings of expert-future proximity, we can project experts into a low-dimensional space that captures the principal components of weight temporal evolution, enabling us to predict future positions and measure expert-future proximity efficiently for assigning averaging coefficients.

PCA over Temporal Weight Deviation. The weight deviations $\{\delta\theta_i\}_{i=1}^{S},\ \delta\theta_i = \theta_i - \theta_{\text{base}}$ of all experts captures weight dynamics under temporal distribution shifts, though these dynamics typically stem from multiple underlying factors and contain noise. We apply PCA to $\{\delta\theta_i\}_{i=1}^{S}$ to decompose the principal components of weight temporal evolution and reduce noise. By considering only the P most significant components $\{v_p\}_{p=1}^{P}$, we can obtain a P-dimensional space and project the experts into points in this space, where $\mathbf{c}_i = (c_i^1, ..., c_i^P)$ is the projection of θ_i:

$$\begin{aligned}\mathbf{c}_i &= (c_i^1, ..., c_i^P) \\ &= (\langle \theta_i - \theta_{\text{base}}, v_1\rangle, \ldots, \langle \theta_i - \theta_{\text{base}}, v_P\rangle)\end{aligned} \tag{4}$$

Principal Component Trajectory Forecasting. We could construct a temporal evolution trajectory of the P principle components with all experts' projected points and their timestamps, $\{(\mathbf{c}_i, t_i)\}_{i=1}^{S}$. Then we predict the future domain position in this P-dimensional space by forecasting along this temporal evolution trajectory. As we often have limited temporal domains available leading to few historical points in the trajectory, we simply model the P-dimensional trajectory as P separate time series, $\{(c_i^p, t_i)\}_{i=1}^{P}$ for $p \in \{1, ..., P\}$, by treating all the dimensions independently. For prediction, we apply the Autoregressive Integrated Moving Average (ARIMA) model to each time series:

$$c^p(t_f) = \text{ARIMA}(\{(c_i^p, t_i)\}_{i=1}^{S}, t_f) \tag{5}$$

where $p \in \{1, ..., P\}$ and t_f is the future domain's timestamp. The predicted future point in the principle component space is $\mathbf{c}_f = (c^1(t_f), ..., c^P(t_f))$.

Distance-based Averaging Coefficients Based on Insight 2, we assign higher averaging coefficients to experts with greater expert-future proximity (lower expert-future distance) in the principal component space. Specifically, for expert θ_i with projected point $\mathbf{c}_i = (c_i^1, ..., c_i^P)$ and our predicted future point $\mathbf{c}_f = (c^1(t_f), ..., c^P(t_f))$, we calculate distance $d_i = \|\mathbf{c}_i - \mathbf{c}_f\|$. We then assign the averaging coefficient for θ_i as:

$$\alpha_i = \frac{(d_{\max} - d_i)^r}{\sum_{j=1}^{n}(d_{\max} - d_j)^r}, \tag{6}$$

where $d_{\max} = \max(d_1, ..., d_n)$ and r is a hyperparameter controlling the sharpness of the weighting distribution. Higher r concentrates the averaging more on experts closer to the predicted future.

3.4 TWA for CDTDG

The TWA method described above targets the original TDG setting with access to all source domains, and cannot be directly applied to the CDTDG setting with sequentially available domains. Simply sampling models during the incremental learning process fails because adjacent domains have variations not only from temporal distribution shifts but also from newly introduced data, which inevitably leads to significant parameter differences even between models from adjacent domains, thereby violating the locality constraints.

We therefore slightly relax the CDTDG constraints by maintaining small memory buffers (e.g., 10%) of used training data $\{d_1, d_2, \ldots, d_S\}$ from each domain. After training on the final source domain, we can access these buffers, which is reasonable in practice with minimal impacts on fairness by using only previously seen data. Based on this relaxation, we apply the original TWA framework:

$$\theta_{\text{base}} = \theta_{S+1} = \theta(l_{\text{IncERM}}(\{\mathbf{D}_S\})) \quad \text{(Pretraining)}$$
$$\theta_i = \theta(l_{\text{SI}}(\{d_{t_i}\}, \theta_{i+1})), \quad \text{(Finetuning)}$$

where $i \in \{S, \ldots, 1\}$, l_{IncERM} is the incremental learning process with ERM, and l_{SI} is the learning process with SI constraining parameter changes.

4 Experimental Results

4.1 Experimental Setup

We first introduce the major experimental setups, with detailed configurations provided in the supplementary material. Note that for overlapped benchmarks, we follow the configurations from Xie et al. (2024c,b) for fair and consistent comparisons.

Benchmarks. We include 4 benchmarks from Yao et al. (2022a) (Huffpost, Arxiv, Yearbook and FMoW), 2 benchmarks from Lin et al. (2022) (CLEAR-10/100), and Rotated MNIST (RMNIST). Huffpost and Arxiv are text benchmarks; others are image benchmarks. RMNIST and Yearbook are small-scale; others are large-scale. Each dataset is

Dataset	Metric	Method									TEA (ours)
		ERM	IRM	CORAL	Mixup	LISA	GI§	LSSAE§	SWAD	DiWA	
Yearbook (Yao et al., 2022a)	D_{S+1}	89.30	97.09	95.94	94.98	95.51	97.42	93.93	97.18	97.66	**97.71**
	$\mathrm{OOD}_{\mathrm{avg.}}$	88.46	94.52	91.79	91.12	92.97	**96.37**	92.12	95.00	95.36	95.95
	$\mathrm{OOD}_{\mathrm{worst}}$	86.81	92.58	88.84	88.35	91.29	**95.73**	88.75	93.89	94.42	94.80
RMNIST	D_{S+1}	98.15	95.10	93.04	97.11	96.21	97.78	96.73	97.93	97.67	**98.61**
	$\mathrm{OOD}_{\mathrm{avg.}}$	92.14	85.05	79.10	89.66	87.04	91.00	90.36	94.51	92.06	**94.47**
	$\mathrm{OOD}_{\mathrm{worst}}$	83.89	72.52	62.96	79.63	75.15	82.46	82.13	84.89	84.31	**88.83**
FMoW (Yao et al., 2022a)	D_{S+1}	72.43	64.77	62.14	70.27	70.05	61.62	59.15	71.59	73.85	**75.63**
	$\mathrm{OOD}_{\mathrm{avg.}}$	59.76	54.92	51.42	57.73	55.52	50.83	48.66	59.96	60.77	**62.45**
	$\mathrm{OOD}_{\mathrm{worst}}$	49.85	46.51	42.19	48.04	44.61	42.78	41.38	50.48	51.00	**52.45**
CLEAR-10 (Lin et al., 2022)	D_{S+1}	80.83	77.50	77.57	78.57	71.50	72.73	55.63	69.20	81.03	**83.53**
	$\mathrm{OOD}_{\mathrm{avg.}}$	81.20	77.03	77.89	78.21	70.89	71.31	55.74	68.14	81.17	**83.16**
	$\mathrm{OOD}_{\mathrm{worst}}$	80.83	76.60	77.47	76.90	70.27	70.33	54.83	67.53	80.60	**82.43**
CLEAR-100 (Lin et al., 2022)	D_{S+1}	63.92	57.74	61.95	62.96	53.80	51.87	39.82	47.38	65.64	**67.39**
	$\mathrm{OOD}_{\mathrm{avg.}}$	63.19	56.79	60.53	62.42	52.82	51.06	39.41	46.04	64.71	**66.96**
	$\mathrm{OOD}_{\mathrm{worst}}$	62.62	56.24	59.46	61.93	52.08	50.32	38.87	45.18	63.96	**66.43**
Huffpost (Yao et al., 2022a)	D_{S+1}	72.74	71.04	71.34	73.34	72.19	68.06	-	73.40	73.31	**73.43**
	$\mathrm{OOD}_{\mathrm{avg.}}$	71.50	70.31	70.08	71.16	70.24	66.32	-	71.59	71.51	**72.12**
	$\mathrm{OOD}_{\mathrm{worst}}$	69.63	68.97	68.68	69.29	68.60	64.64	-	70.10	70.18	**70.64**
Arxiv (Yao et al., 2022a)	D_{S+1}	57.49	51.11	50.98	57.58	56.53	53.43	-	57.08	57.21	**59.28**
	$\mathrm{OOD}_{\mathrm{avg.}}$	52.38	45.89	45.77	52.77	52.41	49.19	-	52.96	52.80	**55.23**
	$\mathrm{OOD}_{\mathrm{worst}}$	49.28	42.86	42.71	49.62	49.67	46.13	-	50.09	49.92	**52.31**
Overall Avg.	D_{S+1}	76.41	73.48	73.28	76.40	73.68	71.70	-	73.25	77.91	**79.37**
	$\mathrm{OOD}_{\mathrm{avg.}}$	72.66	69.22	68.08	71.87	69.13	67.87	-	69.74	74.77	**75.76**
	$\mathrm{OOD}_{\mathrm{worst}}$	69.13	64.90	62.04	67.54	65.95	64.63	-	65.71	70.55	**72.56**

Table 1: Accuracy (%) on all benchmarks under TDG setting. Baselines include ERM, IRM (Arjovsky et al., 2019), CORAL (Sun and Saenko, 2016), Mixup (Zhang et al., 2018), LISA (Yao et al., 2022b), GI (Nasery et al., 2021), LSSAE (Qin et al., 2022), SWAD (Cha et al., 2021), and DiWA (Rame et al., 2022). Best and second-best results are **bolded** and underlined. For FMoW, CLEAR-10&100, Huffpost and Arxiv, we only apply GI to classifiers due to backbone size limitations. LSSAE only applies to image benchmarks. § indicates TDG baselines.

divided into first S source and last F target domains with ratios $S : F$ of: Yearbook (16:5), RMNIST (6:3), FMoW (13:3), Huffpost (4:3), Arxiv (9:7), and CLEAR-10/100 (5:5). Each source domain uses a random 90%-10% train-validation split.

Model Architectures. We use: 4-layer CNN for Yearbook, ConvNet (Qin et al., 2022) for RMNIST, DenseNet-121 (Huang et al., 2017) for FMoW, DistilBERT (Sanh et al., 2019) for Huffpost/Arxiv, and ResNet-18/50 (He et al., 2016) for CLEAR-10/100.

Baselines: For TDG, we evaluate against ERM, IRM (Arjovsky et al., 2019), CORAL (Sun and Saenko, 2016), Mixup (Zhang et al., 2018), LISA (Yao et al., 2022b), GI (Nasery et al., 2021), LSSAE (Qin et al., 2022), SWAD (Cha et al., 2021), and DiWA (Rame et al., 2022), where GI and LSSAE are representative TDG methods and SWAD and DiWA are representative weight averaging approaches. For CDGTD, we include Incremental ERM (IncERM), Mixup (Zhang et al., 2018), SimCLR (Chen et al., 2020), SwAV (Caron et al., 2020), EWC (Kirkpatrick et al., 2017), SI (Zenke et al., 2017), A-GEM (Chaudhry et al., 2018), DRAIN (ying Bai et al., 2022), EvoS (Xie et al., 2024c), and W-Diff (Xie et al., 2024b), with EvoS and W-Diff being state-of-the-art CDGTD methods. Due to computational constraints (e.g., GI finetuning costs 400 GPU hours per epoch), full GI and DRAIN are applied only on Yearbook and RMNIST. For larger benchmarks, we use GI without finetuning and apply DRAIN only to the classifier.

Method Configurations. Overlapped baselines use configurations from (Xie et al., 2024c,b). TEA maintain equivalent total training steps (e.g., 25 baseline epochs = 20 pretraining + 5 finetuning for TEA). Other details including CLEAR configurations are in the supplement.

4.2 Main Results

TDG setting results and comparisons are presented in Table 1. Our TEA outperforms all baseline methods on both image and text benchmarks. Specifically, we observe: a). Prior TDG baselines (GI (Nasery et al., 2021) and LSSAE (Qin et al., 2022)) perform well on small-scale benchmarks (RMNIST and Yearbook (Yao et al., 2022a)) but degrade significantly on other large-scale benchmarks (Lin et al., 2022; Yao et al., 2022a). While GI's poor performance potentially stems from computational constraints preventing finetuning stage, LSSAE was fully applied, indicating that prior

Dataset	Metric	Method										TEA (ours)
		IncERM	Mixup	SimCLR	SwAV	EWC	SI	A-GEM	DRAIN§	EvoS§	W-Diff§	
Yearbook (Yao et al., 2022a)	D_{S+1}	96.61	90.21	95.94	97.37	97.18	97.09	94.36	96.23	97.37	97.32	**97.75**
	$\text{OOD}_{\text{avg.}}$	94.72	89.83	93.07	94.27	95.12	94.67	90.96	94.71	**95.53**	95.03	$\underline{95.29}$
	$\text{OOD}_{\text{worst}}$	93.48	88.43	89.65	91.44	93.64	93.48	88.88	93.73	**94.78**	94.05	$\underline{94.40}$
RMNIST	D_{S+1}	98.62	98.43	98.23	98.08	98.56	98.61	95.99	98.52	98.64	$\underline{98.70}$	**98.74**
	$\text{OOD}_{\text{avg.}}$	92.80	92.38	90.98	90.85	92.02	93.27	86.95	93.09	$\underline{93.84}$	**94.12**	93.76
	$\text{OOD}_{\text{worst}}$	84.61	83.45	81.05	80.96	82.80	85.65	75.45	85.75	87.04	**87.36**	$\underline{87.05}$
FMoW (Yao et al., 2022a)	D_{S+1}	65.52	64.84	64.97	66.47	66.23	66.61	54.54	67.22	67.18	**68.80**	$\underline{67.87}$
	$\text{OOD}_{\text{avg.}}$	53.99	52.00	53.20	54.51	54.55	54.89	47.61	55.05	54.64	**55.86**	$\underline{55.21}$
	$\text{OOD}_{\text{worst}}$	45.23	42.54	44.71	45.29	45.80	$\underline{46.46}$	41.13	46.24	45.86	**46.51**	46.27
CLEAR-10 (Lin et al., 2022)	D_{S+1}	75.90	74.97	78.43	77.53	75.07	76.73	60.67	74.40	77.03	68.00	**79.20**
	$\text{OOD}_{\text{avg.}}$	75.82	74.99	**78.41**	$\underline{78.05}$	73.71	76.07	59.49	74.52	77.06	67.85	77.87
	$\text{OOD}_{\text{worst}}$	74.83	74.10	**77.73**	77.13	72.30	75.00	58.17	73.97	76.87	66.03	$\underline{77.43}$
CLEAR-100 (Lin et al., 2022)	D_{S+1}	56.73	51.68	**60.52**	58.89	56.22	31.76	23.61	54.74	57.02	52.33	$\underline{58.93}$
	$\text{OOD}_{\text{avg.}}$	55.67	50.86	**59.67**	57.59	55.20	30.82	22.55	53.16	56.09	51.92	$\underline{58.43}$
	$\text{OOD}_{\text{worst}}$	54.47	50.32	**58.65**	56.53	54.30	30.35	21.64	51.90	55.47	51.65	$\underline{57.70}$
Huffpost (Yao et al., 2022a)	D_{S+1}	73.57	73.07	-	-	73.64	72.58	72.23	73.42	73.42	$\underline{73.91}$	**73.99**
	$\text{OOD}_{\text{avg.}}$	71.98	71.52	-	-	71.53	71.50	71.16	71.75	$\underline{72.36}$	72.29	**72.40**
	$\text{OOD}_{\text{worst}}$	69.80	69.44	-	-	68.99	69.61	69.10	69.69	70.19	$\underline{70.40}$	**70.61**
Arxiv (Yao et al., 2022a)	D_{S+1}	56.22	56.64	-	-	56.60	49.98	52.02	56.04	56.60	$\underline{56.66}$	**57.34**
	$\text{OOD}_{\text{avg.}}$	52.43	52.95	-	-	52.78	47.27	48.91	52.07	53.15	$\underline{53.43}$	**54.20**
	$\text{OOD}_{\text{worst}}$	49.37	49.97	-	-	49.73	44.77	46.03	48.97	50.19	$\underline{50.70}$	**51.41**
Overall Avg.	D_{S+1}	73.88	72.01	-	-	74.79	69.34	64.77	74.29	$\underline{75.32}$	73.67	**76.26**
	$\text{OOD}_{\text{avg.}}$	71.23	69.55	-	-	71.88	66.48	61.34	70.99	$\underline{71.81}$	70.07	**72.45**
	$\text{OOD}_{\text{worst}}$	67.59	65.46	-	-	67.84	63.40	57.09	67.94	$\underline{68.63}$	66.67	**69.27**

Table 2: Accuracy (%) under CDGTD setting. Baselines include: ERM (IncERM), Mixup (Zhang et al., 2018), SimCLR (Chen et al., 2020), SwAV (Caron et al., 2020), EWC (Kirkpatrick et al., 2017), SI (Zenke et al., 2017), A-GEM (Chaudhry et al., 2018), DRAIN (ying Bai et al., 2022), EvoS (Xie et al., 2024c), and W-Diff (Xie et al., 2024b). Best and second-best results are **bolded** and $\underline{\text{underlined}}$. For FMoW, CLEAR-10, CLEAR-100, Huffpost and Arxiv, we only apply DRAIN to classifiers due to backbone size limitations. SimCLR and SwAV only apply to image benchmarks. § indicates TDG baselines.

TDG methods also struggle to model temporal distribution shifts on large-scale tasks beyond computational limitations. In contrast, TEA consistently improves performance across all scales, outperforming GI by up to 30% and LSSAE by up to 69%; b). TEA also consistently outperforms weight averaging methods (DiWA (Rame et al., 2022) and SWAD (Cha et al., 2021)), validating that our approach not only benefits from sampling experts with functional diversity and parameter similarity but further leverages adaptive averaging coefficients to specifically address temporal shifts, thereby enhancing temporal generalization beyond standard weight averaging techniques.

CDGTD setting results and comparisons are presented in Table 2. Our TEA still achieves the best performance on average, outperforming state-of-the-art CDGTD baselines, EvoS (Xie et al., 2024c) and W-Diff (Xie et al., 2024b). On text benchmarks, our TEA consistently performs the best, while on image benchmarks, although different benchmarks favor different methods, our TEA generally ranks within the top two. These results demonstrate the superiority and flexibility of TEA, showing that TEA can effectively improve temporal generalization even under imited data access constraints.

Training Cost Analysis is presented in Table 3. Early TDG methods (GI (Nasery et al., 2021), LSSAE (Qin et al., 2022), and DRAIN (ying Bai et al., 2022)) significantly increase training costs (see Yearbook and RMNIST for full costs). Even classifier-only W-Diff averages 81× the training cost. In contrast, our TEA only slightly increases cost by 33% over ERM in both TDG and CDGTD, being up to 60x more efficient that prior TDG/CDGTD baselines.

4.3 Ablation Study and Analysis

Single Model Ablation results are shown in Table 4. The *Random Expert* average accuracies from randomly selected temporal experts, while *Last Expert* shows accuracies from the last domain experts. *Random Expert* performs worse than ERM, indicating that our method does not simply improve domain-agnostic convergence during finetuning. *Last Expert* outperforms *Random Expert*, demonstrating that our temporal finetuning enables the model to learn domain-specific distributions, achieving functional diversity among experts.

Weight Averaging Ablation are shown in Table 4. Recall that TEA optimizes two tradeoffs: (1) functional diversity vs. parameter similarity (with tem-

Method	Yearbook	RMNIST	CLEAR-10	CLEAR-100	FMoW	HuffPost	Arxiv	Overall	Rel. Cost
				TDG setting					
ERM	0.02	0.02	0.30	1.58	2.34	3.05	7.92	2.18	1.00
GI	0.21	1.31	0.32*	3.54*	5.35*	3.87*	9.75*	3.48	12.01
LSSAE	0.19	0.22	2.19	9.43	12.05	-	-	-	7.78
TEA	0.04	0.04	0.33	1.62	2.43	3.23	8.57	2.32	1.33
				CDGTD setting					
IncERM	0.02	0.02	0.30	1.58	2.34	3.03	7.95	2.18	1.00
DRAIN	0.05	0.13	0.33†	1.75†	2.45†	3.07†	8.86	2.38	2.05
EvoS	0.07	0.07	0.38	1.67	2.56	3.08	9.04	2.41	1.80
W-Diff	3.12	6.74	3.47	32.35	65.31	13.18	77.93	28.87	81.01
TEA	0.04	0.04	0.32	1.64	2.46	3.19	8.65	2.33	1.33

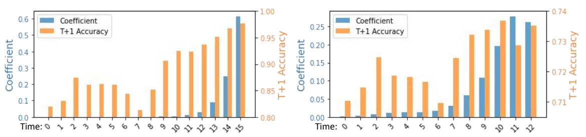
Table 3: Training cost (hours on A40 GPU) for each method. Rel. Cost is the computational cost ratio vs. ERM/IncERM, averaged across all tasks. *GI without finetuning. †Classifier-only DRAIN.

Configuration	Yearbook	RMNIST	FMoW	CLEAR-10	CLEAR-100	Huffpost	Arxiv	Overall
Single Model								
- ERM	88.46	92.14	59.76	81.20	63.19	71.50	52.38	72.66
- *Random Expert*	87.46	82.29	59.37	81.63	63.26	71.34	52.05	71.06
- *Last Expert*	95.42	92.17	60.49	81.53	63.16	71.33	54.57	74.10
Weight Averaging								
- *Only Temporal Experts*	95.41	92.64	60.54	82.12	66.32	71.43	53.79	74.61
- *Only Adaptive Averaging*	94.03	92.92	60.83	83.07	66.85	71.73	53.26	74.67
Full TEA (ours)	**95.95**	**94.47**	**62.45**	**83.16**	**66.96**	**72.12**	**55.23**	**75.76**

Table 4: Ablation study of TEA components under the TDG setting with OOD average accuracy (%).

Coefficients	Yearbook	RMNIST	Arxiv	Overall
ERM	88.46	92.14	52.38	72.66
Correct	95.95	94.47	55.23	75.76
Reversed	82.95	77.03	50.13	70.05

Table 5: Temporal Sanity Check with OOD Avg. Accuracy (%). Overall is averaged across the 7 benchmarks.



(a) Yearbook. (b) FMoW.

Figure 4: Visualization of averaging coefficients and accuracies of experts on target domain D_{S+1}.

poral experts), and (2) bias vs. variance (with adaptive averaging). *Only Temporal Experts* uses uniform coefficients to average experts, optimizing only tradeoff 1, while *Only Adaptive Averaging* samples domain-agnostic weights then trains Time2Vec (Kazemi et al., 2019) for adaptive coefficients (detailed in supp.), optimizing only tradeoff 2. Both variants underperform full TEA, validating the necessity of both design choices.

Temporal Sanity Check are shown in Figure 4 and Table 5. Our adaptive averaging should increase coefficients for better-performing experts on future domains while decreasing coefficients for poor performers. Figure 4 confirms this design by showing higher coefficients for higher-performing models on domain D_{S+1}. Table 5 validates our design by showing that reversing coefficient order leads to worse OOD accuracy than ERM.

5 Conclusion

This work addresses Temporal Domain Generalization (TDG), enabling models to generalize across temporal distribution shifts. We propose Temporal Expert Averaging (TEA), an efficient weight averaging framework for large-scale TDG. Based on theoretical insights, TEA uses constrained temporal finetuning to create functionally diverse yet parameter-similar experts, then adaptively averages them using coefficients derived from temporal dynamics of weight deviation principal components. Comprehensive evaluation demonstrates TEA's superior performance and efficiency across TDG and CDGTD settings. Since prior TDG work focuses on small-scale scenarios, we hope this encourages research on large-scale temporal generalization.

Limitations. Like prior TDG methods, our TEA relies on smooth distribution shift assumptions and cannot guarantee performance with abrupt shifts. Since most large-scale TDG benchmarks use discrete domains, we only explore discrete settings, though TEA could theoretically extend to Continuous Temporal Domain Generalization (CTDG).

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. 2021. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418.

Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. 2022. Domain generalization by mutual-information regularization with pre-trained models. *European Conference on Computer Vision (ECCV)*.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2018. Efficient lifelong learning with a-gem. *ArXiv*, abs/1812.00420.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2023. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. *arXiv preprint arXiv:2311.17667*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Li Deng. 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29:141–142.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning. *arXiv preprint arXiv:2406.09170*.

Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. 2015. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–7.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073.

Ishaan Gulrajani and David Lopez-Paz. 2021. In search of lost domain generalization. In *International Conference on Learning Representations*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022a. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. 2022b. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 35:29262–29277.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. 2019. Time2vec: Learning a vector representation of time. *ArXiv*, abs/1907.05321.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Ron Kohavi, David H Wolpert, and 1 others. 1996. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–283. Citeseer.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. 2017a. Deeper, broader and artier domain generalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5543–5551.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. 2017b. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*.

Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. 2019. Episodic training for domain generalization. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1446–1455.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex Chichung Kot. 2018a. Domain generalization with adversarial feature learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. 2018b. Deep domain generalization via conditional invariant adversarial networks. In *European Conference on Computer Vision*.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. 2016. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*.

Zhiqiu Lin, Jia Shi, Deepak Pathak, and Deva Ramanan. 2022. The clear benchmark: Continual learning on real-world imagery. *ArXiv*, abs/2201.06289.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *NeurIPS*.

Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. 2019. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *Computer Vision and Pattern Recognition (CVPR)*.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*.

Anshul Nasery, Soumyadeep Thakur, Vihari Piratla, Abir De, and Sunita Sarawagi. 2021. Training for the future: A simple gradient interpolation loss to generalize along time. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2023. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36:66727–66754.

Guillermo Ortiz-Jiménez, Mireille El Gheche, Effrosyni Simou, Hermina Petric Maretic, and Pascal Frossard. 2019. Cdot: Continuous domain adaptation using optimal transport. *ArXiv*, abs/1909.11448.

Tiexin Qin, Shiqi Wang, and Haoliang Li. 2022. Generalizing to evolving domains with latent structure-aware sequential autoencoder. In *International Conference on Machine Learning*, pages 18062–18082. PMLR.

Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. 2022. Diverse weight averaging for out-of-distribution generalization. In *NeurIPS*.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 213–226. Springer.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *NeurIPS*.

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. 2023. Zipit! merging models from different tasks without training. *arXiv preprint arXiv:2305.03053*.

Baochen Sun, Jiashi Feng, and Kate Saenko. 2015. Return of frustratingly easy domain adaptation. *ArXiv*, abs/1511.05547.

Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971.

Hao Wang, Hao He, and Dina Katabi. 2020. Continuously indexed domain adaptation. In *International Conference on Machine Learning*.

Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari. 2024. Sam-clip: Merging vision foundation models towards semantic and spatial understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3635–3647.

William Wei Wang, Gezheng Xu, Ruizhi Pu, Jiaqi Li, Fan Zhou, Changjian Shui, Charles Ling, Christian Gagné, and Boyu Wang. 2022. Evolving domain generalization. *arXiv preprint arXiv:2206.00047*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*.

Binghui Xie, Yongqiang Chen, Jiaqi Wang, Kaiwen Zhou, Bo Han, Wei Meng, and James Cheng. 2024a. Enhancing evolving domain generalization through dynamic latent representations. *arXiv preprint arXiv:2401.08464*.

Mixue Xie, Shuang Li, Binhui Xie, Chi Liu, Jian Liang, Zixun Sun, Ke Feng, and Chengwei Zhu. 2024b. Weight diffusion for future: Learn to generalize in non-stationary environments. *Advances in Neural Information Processing Systems*, 37:6367–6392.

Mixue Xie, Shuang Li, Longhui Yuan, Chi Liu, and Zehui Dai. 2024c. Evolving standardization for continual domain generalization over temporal drift. *Advances in Neural Information Processing Systems*, 36.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. *arXiv preprint arXiv:2401.06853*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115.

Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei Koh, and Chelsea Finn. 2022a. Wild-time: A benchmark of in-the-wild distribution shift over time. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. 2022b. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pages 25407–25437. PMLR.

Guang ying Bai, Ling Chen, and Liang Zhao. 2022. Temporal domain generalization with drift-aware dynamic neural network. *ArXiv*, abs/2205.10664.

LIN Yong, Fan Zhou, Lu Tan, Lintao Ma, Jianmeng Liu, HE Yansu, Yuan Yuan, Yu Liu, James Y Zhang, Yujiu Yang, and 1 others. 2023. Continuous invariance learning. In *The Twelfth International Conference on Learning Representations*.

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024. Back to the future: Towards explainable temporal reasoning with large language models. In *Proceedings of the ACM Web Conference 2024*, pages 1963–1974.

Qiuhao Zeng, W. Wang, Fan Zhou, Charles X. Ling, and Boyu Wang. 2023. Foresee what you will learn: Data augmentation for domain generalization in non-stationary environments. *ArXiv*, abs/2301.07845.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987–3995.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. Mixup: Beyond empirical risk minimization. In *ICLR*.

A Experimental Setup Details

A.1 Benchmark Introduction

Huffpost (Ginosar et al., 2015) is a text classification benchmark comprising news headlines from The Huffington Post spanning 2012-2018. The task requires classifying headlines into 11 news categories: "Black Voices", "Business", "Comedy", "Crime", "Entertainment", "Impact", "Queer Voices", "Science", "Sports", "Tech", and "Travel". This temporal dataset captures evolving journalistic styles and content trends in digital media over six years. We adopt a temporal split using the first 4 years as training domains and the final 3 years as test domains for evaluating temporal generalization. Sample distributions across domains are detailed in Table 6.

Arxiv (Ginosar et al., 2015) is a text classification benchmark containing paper titles and their corresponding primary categories spanning 2007-2022. The task requires classifying research papers into one of 172 categories based solely on their titles. This temporal dataset reflects the dynamic evolution of research fields, with changing academic trends and emerging disciplines captured across the 16-year timespan. We adopt a temporal split using the first 9 years as training domains and the final 7 years as test domains for evaluating temporal generalization. Sample distributions across domains are presented in Table 7.

Yearbook dataset, sourced from Yao et al. (2022a) and built upon the MIT-licensed Portraits dataset (Ginosar et al., 2015), comprises 32×32 grayscale yearbook portraits from 128 American high schools across 27 states. Spanning eight decades (1930-2013), this temporal dataset captures the evolution of fashion trends and societal changes, making it particularly suitable for evaluating algorithmic performance on temporal domain shift. We formulate the task as binary gender classification, partitioning the timeline into 4-year intervals to create 21 distinct domains. Following standard practice, we allocate the initial 16 domains for training and reserve the final 5 domains for out-of-domain evaluation. Sample distributions across domains are detailed in Table 8.

Rotated MNIST (RMNIST) derives from the classic MNIST dataset (Deng, 2012) by systematically applying rotational transformations from 0° to 80° in 10° increments, creating 9 sequential domains that simulate temporal distribution shift. This benchmark evaluates 10-class digit classification performance on 28×28 grayscale images under gradually increasing rotational distortion. We adopt a 6-3 domain split, utilizing the initial six domains for model training and evaluating generalization on the final three domains.

FMoW (Ginosar et al., 2015) contains 224×224 RGB satellite imagery spanning 2002-2017 across 200 countries. This temporal benchmark captures natural evolution in visual features driven by human development and environmental changes over time. The classification task involves predicting functional land use across 62 categories, ranging from residential areas to industrial facilities. We partition the dataset temporally with each year constituting a distinct domain, yielding 16 total domains. Training utilizes the first 13 domains, while the final 3 domains serve as out-of-distribution test sets. Domain-wise sample distributions are provided in Table 9.

CLEAR-10&100 (Lin et al., 2022) contain user-uploaded images from 2007-2014 with natural temporal shifts of visual concepts. Samples are organized into 10 chronologically ordered domains. CLEAR-10 comprises 10 classes with 3,000 samples per domain (300 per class), while CLEAR-100 contains 100 classes with 10,000 samples per domain (100 per class). We set the image input shape as $(224, 224, 3)$ and use the first 5 domains as source domains and the final 5 domains as target domains for temporal generalization evaluation.

Domain	Year	Training Split	Validation Split	All
1	2012	6701	744	7446
2	2013	7492	832	8325
3	2014	9539	1059	10599
4	2015	11826	1313	13140
5	2016	10548	1172	11721
6	2017	7907	878	8786
7	2018	3501	388	3890
Total	2012–2018	**57514**	**6386**	**63907**

Table 6: Domain Sizes for Huffpost (Yao et al., 2022a)

A.2 Method Configurations

Huffpost (Yao et al., 2022a) uses pretrained DistilBERT base model (Sanh et al., 2019) as the backbone. All baseline methods are trained on 90% randomly split training data from source domains for 50 epochs with learning rate 2e-5 (except A-GEM which uses 1e-7). Other baseline configurations follow Xie et al. (2024c,b).

Domain	Year	Training Split	Validation Split	All
1	2007	131550	14616	146167
2	2008	62460	6939	69400
3	2009	206244	22916	229161
4	2010	50665	5629	56295
5	2011	55741	6193	61935
6	2012	51678	5741	57420
7	2013	64951	7216	72168
8	2014	79498	8833	88332
9	2015	193979	21553	215533
10	2016	120682	13409	134092
11	2017	111024	12336	123361
12	2018	123891	13765	137657
13	2019	142767	15862	158630
14	2020	166014	18445	184460
15	2021	201241	22360	223602
16	2022	89765	9973	99739
Total	2007–2022	**1852150**	**205786**	**2057952**

Table 7: Domain Size for Arxiv (Yao et al., 2022a)

Domain	Interval	Training Split	Validation Split	All
1	1930 – 1933	758	87	845
2	1934 – 1937	1149	130	1279
3	1938 – 1941	949	108	1057
4	1942 – 1945	2353	263	2616
5	1946 – 1949	1229	138	1367
6	1950 – 1953	1082	122	1204
7	1954 – 1957	1646	185	1831
8	1958 – 1961	1295	146	1441
9	1962 – 1965	1468	166	1634
10	1966 – 1969	2227	249	2476
11	1970 – 1973	1634	183	1817
12	1974 – 1977	2238	250	2488
13	1978 – 1981	1553	175	1728
14	1982 – 1985	2331	261	2592
15	1986 – 1989	1792	201	1993
16	1990 – 1993	1729	195	1924
17	1994 – 1997	1882	211	2093
18	1998 – 2001	2136	239	2375
19	2002 – 2005	1868	210	2078
20	2006 – 2009	1010	114	1124
21	2010 – 2013	1102	125	1227
Total	1930 – 2013	**33431**	**3758**	**37189**

Table 8: Domain Sizes for Yearbook (Yao et al., 2022a)

TEA for Huffpost uses the same DistilBERT backbone. Under TDG setting, TEA first trains on all source domain training splits using ERM for 45 epochs with learning rate 2e-5 during the pretraining stage, then performs temporal finetuning for 5 epochs on each domain in reverse temporal order (from 2015 to 2012) using SI with learning rate 5e-6 and constraint strength $c_{si} = 0.1$. Under CDGTD setting, we adopt 47-epoch incremental ERM training on each domain (from 2012 to 2015) with learning rate 2e-5, followed by 30 temporal finetuning epochs on each domain in reverse temporal order (from 2015 to 2012) using SI with learning rate 5e-6 and constraint strength $c_{si} = 0.1$. Note that temporal finetuning under CDGTD uses only 10% of the data, so the total training cost remains 47+30×0.1=50 epochs. During temporal finetuning on each domain, we sample model weights at $K = 5$ evenly spaced training steps and uniformly average them to obtain expert model weights. For PCA on expert deviations, we use the top 10 principal components. For ARIMA estimation, we employ an ARIMA(1,1,1) model. When computing averaging coefficients, we set the sharpness hyperparameter $r = 5$.

Domain	Year	Training Split	Validation Split	All
1	2002	1676	227	1903
2	2003	2279	276	2555
3	2004	1755	240	1995
4	2005	2512	324	2836
5	2006	3155	406	3561
6	2007	1497	190	1687
7	2008	2261	298	2559
8	2009	7439	935	8374
9	2010	18957	2456	21413
10	2011	22111	2837	24948
11	2012	24704	3138	27842
12	2013	3465	385	3850
13	2014	5572	620	6192
14	2015	8885	988	9873
15	2016	14363	1596	15959
16	2017	5534	615	6149
Total	2002–2017	**126165**	**15531**	**141696**

Table 9: Domain Sizes for FMoW (Yao et al., 2022a)

Arxiv (Yao et al., 2022a) also uses pretrained DistilBERT base model (Sanh et al., 2019) as the backbone. All baseline methods are trained on 90% randomly split training data from source domains for 5 epochs with learning rate 2e-5 (except A-GEM which uses 1e-6). Other baseline configurations follow Xie et al. (2024c,b).

TEA for Arxiv uses the same DistilBERT backbone. Under TDG setting, TEA first trains on all source domain training splits using ERM for 4 epochs with learning rate 2e-5 during the pretraining stage, then performs temporal finetuning for 1 epoch on each domain in reverse temporal order (from 2015 to 2007) using SI with learning rate 5e-6 and constraint strength $c_{si} = 0.1$. Under CDGTD setting, we adopt 4-epoch incremental ERM training on each domain (from 2007 to 2015) with learning rate 2e-5, followed by 10 temporal finetuning epochs on each domain in reverse temporal order (from 2015 to 2007) using SI with learning rate 5e-6 and constraint strength $c_{si} = 0.1$. Note that temporal finetuning under CDGTD uses only 10% of the data, so the total training cost remains 4+10×0.1=5 epochs. During temporal finetuning on each domain, we sample model weights at $K = 5$ evenly spaced training steps and uniformly average them to obtain expert model weights. For PCA on expert deviations, we use the top 10 principal components. For ARIMA estimation, we employ an ARIMA(1,1,1) model. When computing averaging coefficients, we set the sharpness hyperparameter $r = 5$.

Yearbook (Yao et al., 2022a) uses a 4-layer convolutional network from Yao et al. (2022a). All baseline methods are trained on 90% randomly split training data from source domains for 50 epochs with learning rate 1e-3. Other baseline configurations follow Xie et al. (2024c,b).

TEA for Yearbook uses the same 4-layer convolutional network from Yao et al. (2022a). Under TDG setting, TEA first trains on all source domain training splits using ERM for 40 epochs with learning rate 1e-3 during the pretraining stage, then performs temporal finetuning for 10 epochs on each domain in reverse temporal order (from D_{16} to D_1) using SI with learning rate 5e-4 and constraint strength $c_{si} = 0.1$. Under CDGTD setting, we adopt 48-epoch incremental ERM training on each domain (from D_1 to D_{16}) with learning rate 1e-3, followed by 20 temporal finetuning epochs on each domain in reverse temporal order (from D_{16} to D_1) using SI with learning rate 5e-4 and constraint strength $c_{si} = 0.1$. Note that temporal finetuning under CDGTD uses only 10% of the data, so the total training cost remains 48+20×0.1=50 epochs. During temporal finetuning on each domain, we sample model weights at $K = 5$ evenly spaced training steps and uniformly average them to obtain expert model weights. For PCA on expert deviations, we use the top 10 principal components. For ARIMA estimation, we employ an ARIMA(1,1,1) model. When computing averaging coefficients, we set the sharpness hyperparameter $r = 5$.

RMNIST adopts the ConvNet in Qin et al. (2022). All baseline methods are trained on 90% randomly split training data from source domains for 50 epochs with learning rate 1e-3 (except A-GEM which uses 1e-5). Other baseline configurations follow Xie et al. (2024c,b).

TEA for RMNIST uses the same ConvNet. Under TDG setting, TEA first trains on all source domain training splits using ERM for 40 epochs with learning rate 1e-3 during the pretraining stage, then performs temporal finetuning for 10 epochs on each domain in reverse temporal order (from D_6 to D_1) using SI with learning rate 2e-4 and constraint strength $c_{si} = 0.1$. Under CDGTD setting, we adopt 48-epoch incremental ERM training on each domain (from D_1 to D_6) with learning rate 1e-3, followed by 20 temporal finetuning epochs on each domain in reverse temporal order (from D_6 to D_1) using SI with learning rate 2e-4 and constraint strength $c_{si} = 0.1$. Note that temporal finetuning under CDGTD uses only 10% of the data, so the total training cost remains 48+20×0.1=50 epochs. During temporal finetuning on each domain, we sample model weights at $K = 5$ evenly spaced training steps and uniformly average them to obtain expert model weights. For PCA on expert deviations, we use the top 10 principal components. For ARIMA estimation, we employ an ARIMA(1,1,1) model. When computing averaging coefficients, we set the sharpness hyperparameter $r = 5$.

FMoW (Yao et al., 2022a) adopts a DenseNet-121 (Huang et al., 2017) backbone pretrained on ImageNet (Deng et al., 2009). All baseline methods are trained on 90% randomly split training data from source domains for 25 epochs with learning rate 2e-4 (except A-GEM which uses 1e-6). Other baseline configurations follow Xie et al. (2024c,b).

TEA for FMoW uses the same DenseNet-121 (Huang et al., 2017). Under TDG setting, TEA first trains on all source domain training splits using ERM for 20 epochs with learning rate 2e-4 during the pretraining stage, then performs temporal finetuning for 5 epochs on each domain in reverse temporal order (from D_{13} to D_1) using SI with learning rate 7e-5 and constraint strength $c_{si} = 0.1$. Under CDGTD setting, we adopt 23-epoch incremental ERM training on each domain (from D_1 to D_{13}) with learning rate 2e-4, followed by 20 temporal finetuning epochs on each domain in reverse temporal order (from D_{13} to D_1) using SI with learning rate 2e-5 and constraint strength $c_{si} = 0.1$. Note that temporal finetuning under CDGTD uses only 10% of the data, so the total training cost remains 23+20×0.1=25 epochs. During temporal finetuning on each domain, we sample model weights at $K = 5$ evenly spaced training steps and uniformly average them to obtain expert model weights. For PCA on expert deviations, we use the top 10 principal components. For ARIMA estimation, we employ an ARIMA(1,1,1) model. When computing averaging coefficients, we set the sharpness hyperparameter $r = 1$.

CLEAR-10 (Lin et al., 2022) adopts a ResNet-18 (He et al., 2016). All baseline methods are trained on 90% randomly split training data from source domains for 50 epochs with batch size 128 and learning rate 1e-3 (except A-GEM which uses 1e-6). Other baseline configurations follow the FMoW configurations from Xie et al. (2024c,b).

TEA for CLEAR-10 uses the same ResNet-18 (He et al., 2016). Batch size is 128. Under TDG setting, TEA first trains on all source domain training splits using ERM for 45 epochs with learning rate 1e-3 during the pretraining stage, then performs temporal finetuning for 5 epochs on each domain in reverse temporal order (from D_5 to D_1) using SI with learning rate 1e-4 and constraint strength $c_{si} = 0.1$. Under CDGTD setting, we adopt 49-epoch incremental ERM training on each domain (from D_1 to D_5) with learning rate 1e-3, followed by 10 temporal finetuning epochs on each domain in reverse temporal order (from D_5 to D_1) using SI with learning rate 1e-4 and constraint strength $c_{si} = 0.1$. Note that temporal finetuning under CDGTD uses only 10% of the data, so the total training cost remains 49+10×0.1=50 epochs. During temporal finetuning on each domain, we sample model weights at $K = 5$ evenly spaced training steps and uniformly average them to obtain expert model weights. For PCA on expert deviations, we use the top 10 principal components. For ARIMA estimation, we employ an ARIMA(1,1,1) model. When computing averaging coefficients, we set the sharpness hyperparameter $r = 0.5$.

CLEAR-100 (Lin et al., 2022) adopts a ResNet-50 (He et al., 2016). All baseline methods are trained on 90% randomly split training data from source domains for 50 epochs with batch size 128 and learning rate 5e-4 (except A-GEM which uses 1e-6). Other baseline configurations follow the FMoW configurations from Xie et al. (2024c,b).

TEA for CLEAR-100 uses the same ResNet-50 (He et al., 2016). Batch size is 128. Under TDG setting, TEA first trains on all source domain training splits using ERM for 45 epochs with learning rate 5e-4 during the pretraining stage, then performs temporal finetuning for 5 epochs on each domain in reverse temporal order (from D_5 to D_1) using SI with learning rate 1e-4 and constraint strength $c_{si} = 0.1$. Under CDGTD setting, we adopt 49-epoch incremental ERM training on each domain (from D_1 to D_5) with learning rate 5e-4, followed by 10 temporal finetuning epochs on each domain in reverse temporal order (from D_5 to D_1) using SI with learning rate 1e-4 and constraint strength $c_{si} = 0.1$. Note that temporal finetuning under CDGTD uses only 10% of the data, so the total training cost remains 49+10×0.1=50 epochs. During temporal finetuning on each domain, we sample model weights at $K = 5$ evenly spaced training steps and uniformly average them to obtain expert model weights. For PCA on expert deviations, we use the top 10 principal components. For ARIMA estimation, we employ an ARIMA(1,1,1) model. When computing averaging coefficients, we set the sharpness hyperparameter $r = 0.5$.

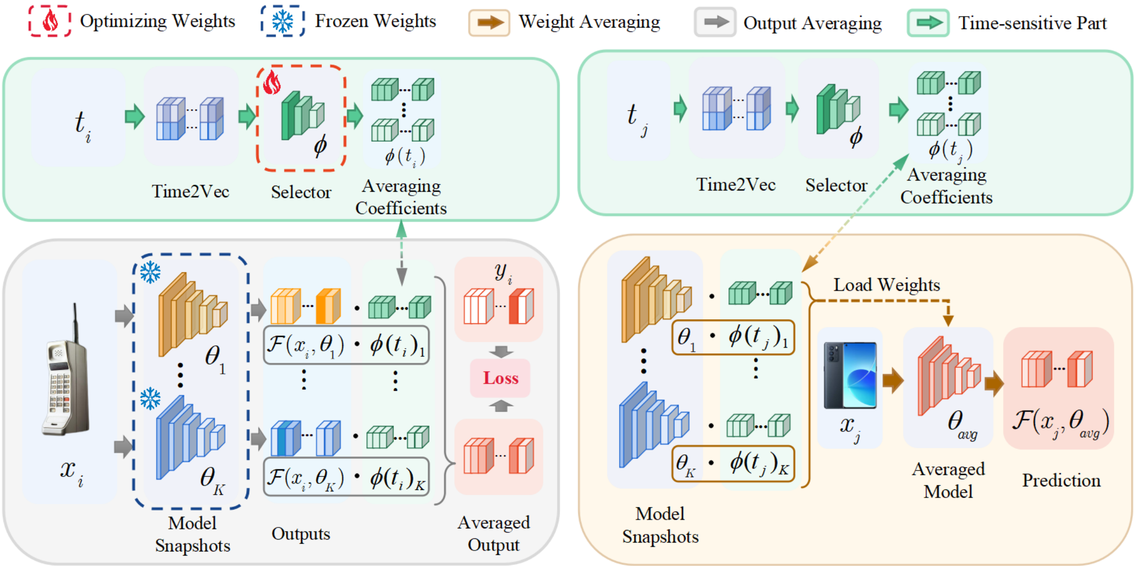
A.3 Ablation Details

Ablation study of TEA components examines four variants: Random Expert, Last Expert, Only Temporal Experts, and Only Adaptive Averaging. The first three involve simple modifications to specific TEA components, while Only Adaptive Averaging represents a more substantially different variant. We briefly describe the first three below and detail Only Adaptive Averaging in the following section:

- **Random Expert**: Randomly selects expert models and reports the average performance across multiple runs, which effectively equals the average performance of all experts.
- **Last Expert**: Uses only the expert from the final domain.
- **Only Temporal Experts**: Identical to TEA except for using uniform averaging coefficients ($1/S$) instead of adaptive coefficients to average all expert weights.

Only Adaptive Averaging shown in 5 aims to use base weights without temporal fine-tuning to achieve functional diversity, capturing temporal shift patterns solely through adaptive weight averaging in the coefficients. This variant cannot be implemented by simply removing TEA components, as our averaging coefficient estimation relies on shift patterns from experts corresponding to different temporal domains. Without temporal differences between base weights, we cannot use TEA's principal component trajectory-based coefficient estimation. Therefore, we adopt a training-based generation approach instead.

We first sample base weights. Following SWA (Izmailov et al., 2018), we randomly sample S weights from the training process, which we call "snapshots". A key challenge arises with normalization layers: on TDG tasks, freezing normalization layers leads to underfitting, while optimizing them results in snapshots with



(a) Training the Selector Network.

(b) Inference.

Figure 5: An overview of our *Only Adaptive Averaging* ablation. (a) When optimizing the selector network in *Only Adaptive Averaging*, we use output averaging as a proxy task, utilizing the estimated coefficients to average the outputs of all snapshots. (b) During inference, we perform weight averaging with the optimized selector network.

different normalization parameters and statistics. Since weight averaging is highly sensitive to normalization differences, excessive variation causes poor performance in the averaged model. We address this using a "late sampling" strategy, as we observe that normalization becomes sufficiently good during intermediate training stages. Specifically, we freeze the normalization layers during the final epoch of each task and sample K snapshots $\{\theta_k\}_{k=1}^{K}$ within this last epoch (noted as K as we use all domain as a unified domain and set $K = S$ for fair ablation). We then generate adaptive averaging coefficients through a training-based approach. Specifically, we use a Time2Vec (Kazemi et al., 2019) module with a 2-layer MLP as the selector network ϕ to generate averaging coefficients. After sampling the snapshots, we randomly select samples with timestamps from the training domains and train the selector network to combine the outputs of these snapshots. We formulate this training process as:

$$\phi^* = \arg\min_{\phi} \sum_{i\in[1,S]} \sum_{(x,t,y)\sim D_i} \ell\left(\sum_{k=1}^{K}\phi(t)_k \cdot f(x,\theta_k),\ y\right)$$
$$\text{s.t.}\quad \{\theta_k\}_{k=1}^{K} \sim \mathcal{S}_{ls}(\arg\min_{\theta} \sum_{i\in[1,S]} \sum_{(X,\cdot,Y)\sim D_i} \ell(f(X,\theta),Y)),$$

where $\mathcal{S}_{ls}$ is the snapshot sampling process with late sampling strategy. We use Adam optimizer for optimizing the selector network with learning rate as 1e-4, batch size as 1 and training steps as 2000.

After training ϕ^*, we use it during inference to generate averaging coefficients for the $K = S$ snapshots: $\boldsymbol{\alpha}^{OAA} = \{\alpha_k^{OAA}\}_{k=1}^{K} = \phi^*(t_f)$.

B Additional Discussion

TDG's Value for NLP Community. On one hand, Temporal Domain Generalization (TDG) (Ortiz-Jiménez et al., 2019; Mancini et al., 2019; Wang et al., 2020; ying Bai et al., 2022; Nasery et al., 2021; Zeng et al., 2023; Wang et al., 2022; Xie et al., 2024c,a; Yong et al., 2023; Xie et al., 2024b) has broad application prospects in NLP tasks, as temporal distribution shifts are prevalent in NLP, such as lexical changes over time and evolving understanding of specific expressions (e.g., memes) across time periods. Particularly in the large language model era, TDG's low-resource generalization nature can reduce the expensive computational and data costs required for LLM retraining or fine-tuning. On the other hand, TDG has already been widely recognized as a valuable direction by the relevant community, with numerous papers published in top-tier conferences, including our baselines: GI (NeurIPS'21) (Nasery et al., 2021), LSSAE (ICML'22) (Qin et al., 2022), DRAIN (ICLR'23) (ying Bai et al., 2022), EvoS (NeurIPS'23) (Xie et al., 2024c), and W-Diff (NeurIPS'24) (Xie et al., 2024b).

Continual Learning. TDG shares similar data configurations with continual learning (Zenke et al., 2017; Lopez-Paz and Ranzato, 2017; Shin et al., 2017; Chaudhry et al., 2018), and our main benchmarks (Yao et al., 2022a; Lin et al., 2022)were originally introduced for continual learning. However, TDG and continual learning differ significantly in their objectives. Standard continual learning primarily focuses on the past, addressing whether learning new tasks causes catastrophic forgetting of previous knowledge. In contrast, TDG focuses on the future, concerned with leveraging past knowledge to enhance generalization to future domains. We incorporate representative continual learning baselines including EWC (Kirkpatrick et al., 2017), SI (Zenke et al., 2017), and A-GEM (Chaudhry et al., 2018), which show no significant generalization improvement on future domains.

Continual Domain Generalization over Temporal Drift (CDGTD) can be viewed as an intersection of standard TDG and continual learning. This represents a reasonable application direction, requiring models to both retain past knowledge and generalize well to future domains. However, this does not diminish the importance of standard TDG, as the core challenge of TDG—how to utilize temporal shift patterns in past data for better future generalization—is orthogonal to CDGTD's additional constraint of sequential domain access. Moreover, CDGTD may complicate the exploration of temporal generalization capabilities by introducing an additional variable. Therefore, we consider both standard TDG and CDGTD equally important, with no priority distinction.

Large Language Models (LLMs). While LLMs (OpenAI, 2023; Touvron et al., 2023; Guo et al., 2025) achieve good generalization through training on massive datasets, this does not conflict with TDG. TDG fundamentally targets low-resource scenarios and has considerable practical value when large training datasets are unavailable. Conversely, in cases of relatively smooth temporal distribution shifts, applying TDG with limited data is more data-efficient than brute-force generalization through massive training. Furthermore, regardless of how much data LLMs are trained on, TDG can be further applied to enhance temporal generalization capabilities. Notably, TDG application to LLMs is particularly promising as it can effectively reduce LLM training costs. However, TDG is still far from being applicable to LLMs, primarily due to scaling limitations. This highlights the value of our work as a solid step toward LLM-scale TDG.

Temporal Reasoning (Xiong et al., 2024; Yuan et al., 2024; Fatemi et al., 2024; Chu et al., 2023). While this may sound related to TDG, the primary connection is that both contain "temporal" in their names. Temporal reasoning focuses on enabling models to understand explicit temporal relationships at the individual sample level, whereas TDG aims to adapt models to implicit temporal distribution shifts at the dataset level. Temporal reasoning could potentially improve TDG performance, but this remains unexplored.

C Theoretical Analysis

C.1 Notations

We denote $\mathcal{X}$ the input space, $\mathcal{Y}$ the label space, and $\ell : \mathcal{Y}^2 \to \mathbb{R}^+$ a loss function. We have a sequence of domains $\{D_i\}$ indexed by timestamps $t_i \in \mathcal{T}$, where $\mathcal{T}$ is a totally ordered set representing time. Each domain D_i has a distribution p_i. For the training (source) domains $\{D_i\}_{i=1}^S$, we have timestamps $t_1 < t_2 < \ldots < t_S$ in $\mathcal{T}$, and corresponding distributions $p_1, p_2, \ldots, p_S$. For simplicity, we will use p_i to refer to the joint, posterior, and marginal distributions of (X, Y) at time t. We note $f_i : \mathcal{X} \to \mathcal{Y}$ as the labeling function at time t_i. We assume there is no noise in the data: f_i is defined on $\mathcal{X}_i \triangleq \{x \in \mathcal{X} \mid p_i(x) > 0\}$ by $\forall (x, y) \sim p_i, f_i(x) = y$.

C.2 Temporal Domain Generalization

We consider a neural network (NN) $f(\cdot, \theta) : \mathcal{X} \to \mathcal{Y}$ made of a fixed architecture f with weights θ. Given observations from source domains at times $t_1, t_2, \ldots, t_S$, we seek θ minimizing the target generalization error at a future time $t_f > t_S$:

$$\mathcal{E}_f(\theta) = \mathbb{E}_{(x,y)\sim p_f}[\ell(f(x, \theta), y)]. \tag{7}$$

$f(\cdot, \theta)$ should approximate f_f on $\mathcal{X}_f$. This is challenging in the TDG setup because we only have data from earlier timestamps, which are related yet different from the future target domain.

The differences between domains at different timestamps are due to distribution shifts (i.e., the fact that $p_i(X, Y) \neq p_j(X, Y)$ for $i \neq j$), which can be decomposed into:

- **Diversity shift:** when marginal distributions differ over time (i.e., $p_i(X) \neq p_j(X)$)
- **Correlation shift:** when posterior distributions differ over time (i.e., $p_i(Y|X) \neq p_j(Y|X)$ and $f_i \neq f_j$)

The weights are typically learned on source domain data $\{D_1, D_2, \ldots, D_S\}$ from timestamps $\{t_1, t_2, \ldots, t_S\}$ (each composed of n_i i.i.d. samples from $p_i(X, Y)$) with a configuration c, which contains all other configurations and sources of randomness in learning. We call $l_{\mathcal{T}} = \{D_1, D_2, \ldots, D_S, c\}$ a learning procedure, and explicitly write $\theta(l_{\mathcal{T}})$ to refer to the weights obtained after stochastic minimization of the appropriate objective function. Specific to our TEA, we define $l_i = \{D_1, D_2, \ldots, D_S, t_i, c\}$ as a temporal expert learning procedure to get expert model $\theta_i = \theta(l_i)$ which is designed to excels on domain D_i while also using data from other domains.

C.3 Temporal Expert Averaging

We study the benefits of combining S individual member weights $\{\theta_i\}_{i=1}^S \triangleq \{\theta(l_i)\}_{i=1}^S$ obtained from S different domains at timestamps $\{t_1, t_2, \ldots, t_S\}$. Each weight θ_i corresponds to an expert model that is more proficient for domain D_i (though not necessarily trained exclusively on that domain).

Unlike traditional weight averaging (Cha et al., 2021; Rame et al., 2022; Wortsman et al., 2022) that uses equal coefficients, for temporal domain generalization, we propose a temporally-weighted averaging scheme that assigns different importance to experts based on their relevance to the target future domain.

Temporal Expert Averaging (TEA) is defined as:

$$f_{\text{TEA}} \triangleq f(\cdot, \theta_{\text{TEA}}),$$

$$\theta_{\text{TEA}} \triangleq \sum_{i=1}^{S} \alpha_i \left(\{t_i\}_{i=1}^S, \{\theta_i\}_{i=1}^S, t_f\right) \cdot \theta_i. \tag{8}$$

where the coefficients $\{\alpha_i\}_{i=1}^S$ satisfy $\sum_{i=1}^S \alpha_i = 1$ and $\alpha_i \geq 0$ for all i. These coefficients are determined based on the temporal shift among the source domain experts $\{\theta_i\}_{i=1}^S$ and temporal information $\{t_i\}_{i=1}^S$ and t_f.

C.4 TEA loss derivation

Following Rame et al. (2022), we decompose TEA's error leveraging the similarity between WA and functional ensembling (ENS) (Lakshminarayanan et al., 2017; Dietterich, 2000), a more traditional way to combine a collection of weights. We also use Mean Squared Error as ℓ for simplicity. For TDG setting, we define Temporal ENS (T-ENS) with coefficients $\{\alpha_i\}_{i=1}^S$ as

$$f_{\text{T-ENS}} \triangleq \sum_{i=1}^{S} \alpha_i f(\cdot, \theta_i). \tag{9}$$

Lemma 1 establishes that f_{TEA} approximates $f_{\text{T-ENS}}$ to first order when $\{\theta_i\}_{i=1}^S$ are close in weight space.

Lemma 1 (TWA and T-ENS)**.** *Given $\{\theta_i\}_{i=1}^S$ with learning procedures for different temporal experts. Denoting $\Delta_{\{\theta\}} = \max_{i=1}^S \|\theta_i - \theta_{TEA}\|_2$, $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$:*

$$f_{TEA}(x) = f_{T\text{-}ENS}(x) + O(\Delta^2_{\{\theta\}}) \tag{10}$$

$$\ell(f_{TEA}(x), y) = \ell(f_{T\text{-}ENS}(x), y) + O(\Delta^2_{\{\theta\}}).$$

Proof. This proof has two components:

- to establish the functional approximation, it performs Taylor expansion of the models' predictions at the first order.
- to establish the loss approximation, it performs Taylor expansion of the loss at the first order.

Functional approximation With a Taylor expansion at the first order of the models' predictions w.r.t. parameters θ:

$$f_{\theta_i} = f_{\text{TEA}} + \nabla f|_{\text{TEA}} \Delta_i + O\left(\|\Delta_i\|_2^2\right)$$

$$f_{\text{T-ENS}} - f_{\text{TEA}}$$

$$= \sum_{i=1}^{S} \alpha_i \nabla f|_{\text{TEA}} \Delta_i + \sum_{i=1}^{S} \alpha_i O\left(\|\Delta_i\|_2^2\right),$$

where $\Delta_i = \theta_i - \theta_{\text{TWA}}$.

Note that unlike in the equal weighting case, we don't have $\sum_{i=1}^S \Delta_i = 0$ for weighted averaging. Instead, we have $\sum_{i=1}^S \alpha_i \Delta_i = 0$. Therefore:

$$f_{\text{T-ENS}} - f_{\text{TEA}}$$

$$= \sum_{i=1}^{S} \alpha_i \nabla f|_{\text{TEA}} \Delta_i + \sum_{i=1}^{S} \alpha_i O\left(\|\Delta_i\|_2^2\right)$$

$$= \nabla f|_{\text{TWA}} \sum_{i=1}^{S} \alpha_i \Delta_i + O\left(\sum_{i=1}^{S} \alpha_i \|\Delta_i\|_2^2\right)$$

$$= O\left(\sum_{i=1}^{S} \alpha_i \|\Delta_i\|_2^2\right)$$

Since $\Delta_i \leq \Delta_{\{\theta\}}$ for all i, and $\sum_{i=1}^{S} \alpha_i = 1$, we have:

$$f_{\text{T-ENS}} - f_{\text{TEA}} = O\left(\sum_{i=1}^{S} \alpha_i \Delta_{\{\theta\}}^2\right)$$
$$= O\left(\Delta_{\{\theta\}}^2 \sum_{i=1}^{S} \alpha_i\right)$$
$$= O\left(\Delta_{\{\theta\}}^2\right)$$

Loss approximation. With a Taylor expansion at the zeroth order of the loss w.r.t. its first input and injecting the functional approximation:

$$\ell(f_{\text{T-ENS}}(x); y) = \ell(f_{\text{TWA}}(x); y) + O(\|f_{\text{T-ENS}}(x) - f_{\text{TEA}}(x)\|_2)$$
$$\ell(f_{\text{T-ENS}}(x); y) = \ell(f_{\text{TEA}}(x); y) + O\left(\Delta_{\{\theta\}}^2\right)$$

□

C.5 Bias-variance-covariance-locality Decomposition for TEA

We can derive the following decomposition of TEA's expected test error in the future domain. The expectation is over the joint distribution describing the S learning procedures $\{l_i\}_{i=1}^{S}$. (*Note that in the temporal domain generalization (TDG) setting, models from different timestamps may have different biases and variances due to the evolution of data distributions over time. This temporal heterogeneity is a key characteristic that distinguishes TDG from standard DG.*)

Proposition 1 (Bias-variance-covariance-locality decomposition for temporal weight averaging)**.** *Denoting $\bar{f}_i(x) = \mathbb{E}_{l_i}[f(x, \theta(l_i))]$ as the expected prediction of an expert model for timestamp t_i, $\mathbb{E}_f = \mathbb{E}_{(x,y)\sim p_f}$ and $\mathbf{l} = \{l_1, \ldots, l_S\}$, the expected generalization error on future domain t_f of $\theta_{TWA} = \sum_{i=1}^{S} \alpha_i \cdot \theta_i$ over the joint distribution of the learning procedures is:*

$$\mathbb{E}_{\mathbf{l}}[\mathcal{E}_f(\theta_{TEA})] = \mathbb{E}_f\left[\mathcal{B} + \mathcal{V} + \mathcal{C}\right] + O(\bar{\Delta}^2), \quad (11)$$

where

$$\mathcal{B} = \left(\sum_{i=1}^{S} \alpha_i \cdot bias_i\right)^2, \; bias_i = y - \bar{f}_i(x),$$
$$\mathcal{V} = \sum_{i=1}^{S} \alpha_i^2 \cdot var_i, \; var_i = \mathbb{E}_{l_i}\left[dev_i^2\right],$$
$$\mathcal{C} = \sum_{i \neq j} \alpha_i \alpha_j cov_{i,j}, \; cov_{i,j} = \mathbb{E}_{\{l_i, l_j\}}\left[dev_i \cdot dev_j\right],$$
$$with\ dev_i = f(x, \theta(l_i)) - \bar{f}_i(x),$$
$$\bar{\Delta}^2 = \mathbb{E}[\Delta_{\{\theta\}}^2] \ with \ \Delta_{\{\theta\}} = \max_{i=1}^{S} \|\theta_i - \theta_{TWA}\|_2.$$

Proof. Following Rame et al. (2022), we use the he bias-variance decomposition in Kohavi et al. (1996) with $f_{\text{T-ENS}} \triangleq \sum_{i=1}^{S} \alpha_i f(\cdot, \theta(l_i))$ to decompose the expected generalization error:

$$\mathbb{E}_{\mathbf{l}}[\mathcal{E}_f(\{\theta(l_i)\}_{i=1}^{S})]$$
$$= \mathbb{E}_f[\text{Bias}\{f_{\text{T-ENS}}|(x,y)\}^2 + \text{Var}\{f_{\text{T-ENS}}|x\}],$$

where bias term becomes:

$$\text{Bias}\{f_{\text{T-ENS}}|(x,y)\}$$
$$= y - \mathbb{E}_{\mathbf{l}}\left[\sum_{i=1}^{S} \alpha_i f(x, \theta(l_i))\right]$$
$$= y - \sum_{i=1}^{S} \alpha_i \mathbb{E}_{\mathbf{l}}[f(x, \theta(l_i))]$$
$$= y - \sum_{i=1}^{S} \alpha_i \bar{f}_i(x)$$
$$= \sum_{i=1}^{S} \alpha_i (y - \bar{f}_i(x))$$
$$= \sum_{i=1}^{S} \alpha_i \text{bias}_i(x, y)$$

Thus, the squared bias term is:

$$\text{Bias}\{f_{\text{T-ENS}}|(x,y)\}^2 = \left(\sum_{i=1}^{S} \alpha_i \text{bias}_i\right)^2$$

For the variance term, denoting $\text{dev}_i = f(x, \theta(l_i)) - \bar{f}_i(x)$, we have:

$$
\begin{aligned}
&\mathrm{Var}\{f_{\text{T-ENS}}|x\} \\
&= \mathbb{E}_{\mathrm{l}}\left[\left(\sum_{i=1}^{S}\alpha_i f(x,\theta(l_i)) - \mathbb{E}_{\mathrm{l}}\left[\sum_{i=1}^{S}\alpha_i f(x,\theta(l_i))\right]\right)^2\right] \\
&= \mathbb{E}_{\mathrm{l}}\left[\left(\sum_{i=1}^{S}\alpha_i (f(x,\theta(l_i)) - \bar{f}_i(x))\right)^2\right] \\
&= \mathbb{E}_{\mathrm{l}}\left[\sum_{i=1}^{S}\sum_{j=1}^{S}\alpha_i\alpha_j \cdot \mathrm{dev}_i \cdot \mathrm{dev}_j\right] \\
&= \sum_{i=1}^{S}\alpha_i^2\mathbb{E}_{\mathrm{l}}[\mathrm{dev}_i^2] + \sum_{i\neq j}\alpha_i\alpha_j\mathbb{E}_{\mathrm{l}}[\mathrm{dev}_i \cdot \mathrm{dev}_j] \\
&= \sum_{i=1}^{S}\alpha_i^2\mathrm{var}_i + \sum_{i\neq j}\alpha_i\alpha_j\mathrm{cov}_{i,j}
\end{aligned}
$$

Combination with Lemma 1 We recall that per our adapted Lemma 1:

$$\ell(f_{\text{TEA}}(x), y) = \ell(f_{\text{T-ENS}}(x), y) + O(\Delta^2_{\{\theta\}}).$$

Taking the expectation over the learning procedures and combining all terms:

$$
\begin{aligned}
\mathbb{E}[\mathcal{E}_f(\theta_{\text{TEA}})] = \mathbb{E}_f &\left[\left(\sum_{i=1}^{S}\alpha_i \mathrm{bias}_i\right)^2\right] \\
&+ \mathbb{E}_f\left[\sum_{i=1}^{S}\alpha_i^2\mathrm{var}_i\right] \\
&+ \mathbb{E}_f\left[\sum_{i\neq j}\alpha_i\alpha_j\ \mathrm{cov}_{i,j}\right] \\
&+ O(\bar{\Delta}^2)
\end{aligned}
$$

□

C.6 Theoretical Insights for TEA

From Equation 11, we can see that generalization error can be reduced by minimizing bias $\mathcal{B}$, variance $\mathcal{V}$, covariance $\mathcal{C}$, and locality $\bar{\Delta}^2$. However, due to the complexity of real-world data and models, finding an optimal analytical solution is nearly impossible. Nevertheless, similar to Rame et al. (2022), we can derive practical insights for designing TEA by analyzing the relationships between these four terms, model properties, and averaging coefficients.

Insight 1 *Tradeoff between Functional Diversity and Parameter Similarity among Experts.* Covariance $\mathcal{C}$ reduction necessitates functional diversity among experts, while the locality constraint $\bar{\Delta}^2$ demands parameter similarity among experts.

The covariance term increases when the predictions of $\{f(\cdot,\theta(l_i))\}_{i=1}^S$ are correlated, suggesting that DiWA's (Rame et al., 2022) approach to reduce covariance by encouraging functional diversity remains effective. However, the locality term $\bar{\Delta}^2$ simultaneously constrains the weights to remain close in parameter space. This tradeoff suggests that when training these expert models, we should find an appropriate balance between encouraging diverse predictions and maintaining parameter similarity.

Insight 2 *Tradeoff between Bias and Variance via Averaging Coefficients.* Reducing variance $\mathcal{V}$ requires averaging weights evenly, while reducing bias $\mathcal{B}$ demands concentrating coefficients on experts with lower bias magnitudes on future data.

Insight 2 is obtained by introducing 2 assumptions specific to the TDG for further discussion about bias and variance.

Assumption 1 (Ordered Bias Magnitudes)**.** *The models can be ordered by expected bias magnitudes on future domains such that* $\mathbb{E}_f\left[bias^2_{m_1}\right] \geq \mathbb{E}_f\left[bias^2_{m_1}\right] \geq \cdots \geq \mathbb{E}_f\left[bias^2_{m_S}\right]$, *with* $\{m_j\}_{j=1}^S$ *being a permutation of* $\{i\}_{i=1}^S$.

Assumption 2 (Equal Variance Experts)**.** *The variance of each expert's prediction is equal across all experts, such that* $\mathbb{E}_f\left[var_i\right] = v$ *for all* $i \in \{1, 2, ..., M\}$.

Lemma 2 (**Optimal Averaging Coefficients for Bias Minimization**)**.** *Let the bias of model i be:*

$$b_i(x,y) = \mathrm{bias}_i(x,y), \sigma_i^2 := \mathbb{E}_f\left[b_i^2\right],$$

and define the root-mean-square magnitude:

$$\sigma_i = \sqrt{\sigma_i^2} \quad (i = 1, \ldots, S).$$

According to Assumption 1, magnitudes are ordered $\sigma_{m_1} \geq \sigma_{m_2} \geq \cdots \geq \sigma_{m_S}$, *where* $\{m_j\}_{j=1}^S$ *is a permutation of* $\{1, \ldots, S\}$. *For convex weights* $\boldsymbol{\alpha} \in \Delta^S := \{\alpha_i \geq 0, \sum_{i=1}^S \alpha_i = 1\}$, *consider the combined bias loss:*

$$L(\boldsymbol{\alpha}) := \mathbb{E}_f\left[\left(\sum_{i=1}^{S}\alpha_i b_i\right)^2\right]. \tag{12}$$

*If **no information** is available on the pairwise bias covariances* $\Sigma_{ij} := \mathbb{E}_f[b_i b_j]$, $(i \neq j)$, *then the minimax problem:*

$$\min_{\boldsymbol{\alpha}\in\Delta^S}\ \max_{\Sigma\ s.t.\ \mathrm{diag}(\Sigma)=\boldsymbol{\sigma}^2} L(\boldsymbol{\alpha}) \tag{13}$$

is solved by:

$$\boxed{\alpha^\star_{m_S} = 1, \quad \alpha^\star_i = 0 \text{ for } i \neq m_S} \tag{14}$$

with $L(\boldsymbol{\alpha}^\star) = \sigma^2_{m_S}$.

Proof. We can write $L(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\top \Sigma \boldsymbol{\alpha}$ with unknown positive-semidefinite matrix Σ satisfying $\Sigma_{ii} = \sigma_i^2$. By the Cauchy-Schwarz inequality, $|\Sigma_{ij}| \leq \sigma_i \sigma_j$. The worst case occurs when all covariances reach the extreme value $\Sigma_{ij} = \sigma_i \sigma_j$, yielding:

$$\max_{\Sigma} L(\boldsymbol{\alpha}) = \left(\sum_{i=1}^{S} \alpha_i \sigma_i \right)^2. \tag{15}$$

Since $\sum_i \alpha_i \sigma_i$ is a convex combination of the ordered set $\{\sigma_{m_j}\}$, its minimum over the simplex Δ^S is attained by placing all weight on the smallest RMS magnitude σ_{m_S}, which gives the stated $\boldsymbol{\alpha}^\star$ and the minimax value $L(\boldsymbol{\alpha}^\star) = \sigma_{m_S}^2$. □

Lemma 3 (Optimal Averaging Coefficients for Variance Minimization)**.** *Consider the variance term with equal variances $\mathbb{E}_f[var_i] = v$ for all $i \in \{1, \ldots, S\}$:*

$$\mathbb{E}_f[\mathcal{V}] = v \sum_{i=1}^{S} \alpha_i^2. \tag{16}$$

For averaging coefficients $\boldsymbol{\alpha} \in \Delta^S := \{\alpha_i \geq 0, \sum_{i=1}^{S} \alpha_i = 1\}$, the variance term is minimized when weights are distributed equally across all models:

$$\boxed{\alpha_i^\star = \frac{1}{S} \textit{ for all } i} \tag{17}$$

with optimal variance $v \cdot \frac{1}{S}$.

Proof. We seek to minimize $\sum_{i=1}^{S} \alpha_i^2$ subject to the constraints $\sum_{i=1}^{S} \alpha_i = 1$ and $\alpha_i \geq 0$. By the Cauchy-Schwarz inequality:

$$\left(\sum_{i=1}^{S} \alpha_i \right)^2 \leq S \sum_{i=1}^{S} \alpha_i^2, \tag{18}$$

with equality if and only if all α_i are equal. Since $\sum_{i=1}^{S} \alpha_i = 1$, we have:

$$1 = \left(\sum_{i=1}^{S} \alpha_i \right)^2 \leq S \sum_{i=1}^{S} \alpha_i^2, \tag{19}$$

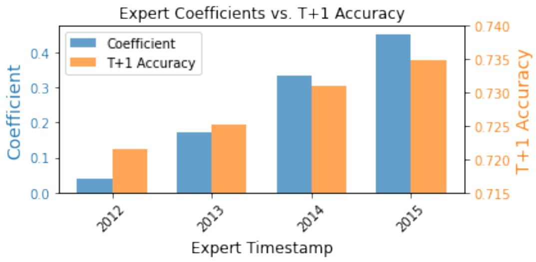
which implies $\sum_{i=1}^{S} \alpha_i^2 \geq \frac{1}{S}$. Equality is achieved when $\alpha_i = \frac{1}{S}$ for all i, giving the optimal solution. □

In summary, Lemma 2 indicates that optimizing the bias term requires concentrating weight on experts with smaller bias magnitude on future domains, while Lemma 3 suggests that minimizing variance requires the opposite approach—distributing weight as evenly as possible across all experts. This creates a fundamental trade-off between bias and variance in the selection of averaging coefficients.

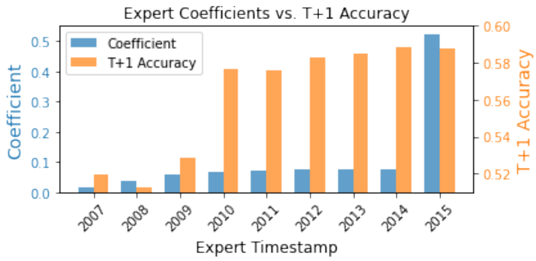
Discussion about Assumptions. Assumption 1 is similar to the smooth distribution shift assumption used by most prior TDG methods (ying Bai et al., 2022; Zeng et al., 2023; Nasery et al., 2021; Xie et al., 2024b,c), allowing us to model distribution change and leverage temporal information to predict future parameter or feature. Assumption 2 is reasonable when all experts share the same architecture, optimization procedure and hyperparameters, differing only in the specific temporal domains they've been optimized to excel in.

D Additional Results

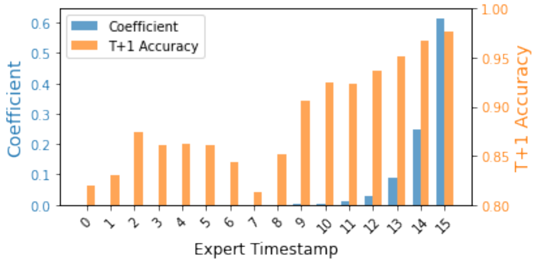
We show the coefficients vs. D_{S+1} accuracy across all benchmarks in Figure 6.



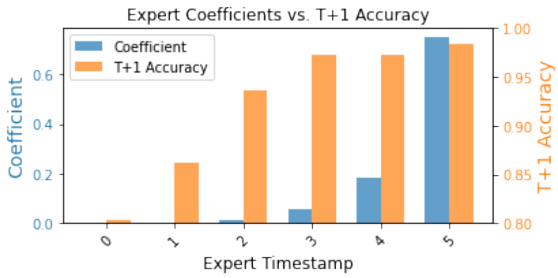
(a) Huffpost.



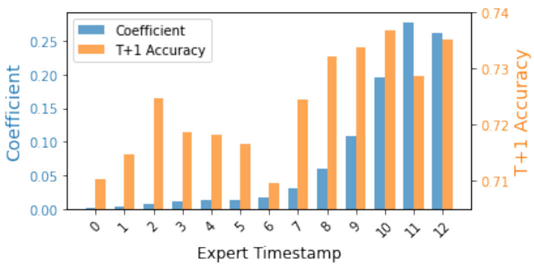
(b) Arxiv.



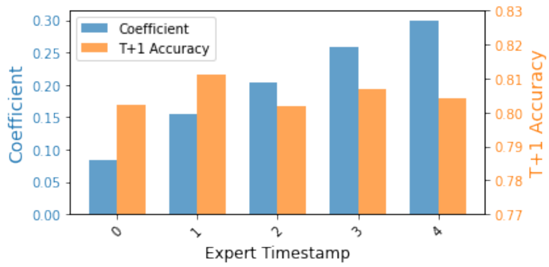
(c) Yearbook.



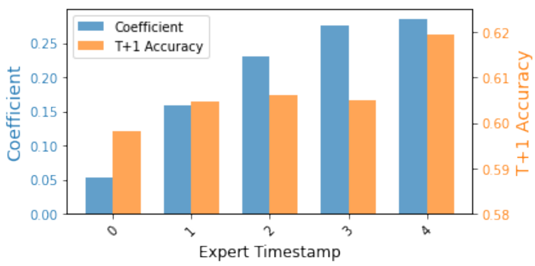
(d) RMNIST.



(e) FMoW.



(f) CLEAR-10.



(g) CLEAR-100.

Figure 6: Visualization of averaging coefficients and accuracies of experts on target domain D_{S+1}.