MoE^2: A Mixture-of-Mixtures of Experts for Ensemble-Free Domain Generalization

Ahmed Radwan, Mahmoud Soliman, Omar Abdelaziz, Ahmad Abdel-Qader, Mohamed S. Shehata,

The University of British Columbia

ahmedm04@student.ubc.ca, mosama97@student.ubc.ca, oabdelaz@student.ubc.ca, aabdelqa@student.ubc.ca, mohamed.sami.shehata@ubc.ca

Abstract

Domain Generalization (DG) requires models to generalize across unseen data distributions. Kernel-based theory reveals a No-Free-Lunch problem: any model with a fixed representation is fundamentally sub-optimal for all possible shifts. While large ensembles mitigate this, they are computationally expensive and remain static once trained, inheriting the same theoretical limitation. We introduce MoE^2 (Mixture-of-Mixtures of Experts), a framework that uses a single frozen backbone to dynamically synthesize a bespoke adapter for each input, allowing it to continuously adapt its effective kernel. We provide a theoretical grounding for this process, proving our routing mechanism is a principled non-parametric estimator for the optimal Bayes mixture of experts. We derive a generalization bound that cleanly separates the router's estimation error from the reduction in a kernel-mismatch penalty achieved via synthesis. MoE^2 matches or exceeds state-of-the-art ensemble baselines on major DG benchmarks while using only a single, compact model. MoE^2 thus provides a theoretically-grounded and lightweight alternative to large-scale ensembles for robust domain generalization.

Code — https://github.com/AhmedMostafaSoliman/MoE2

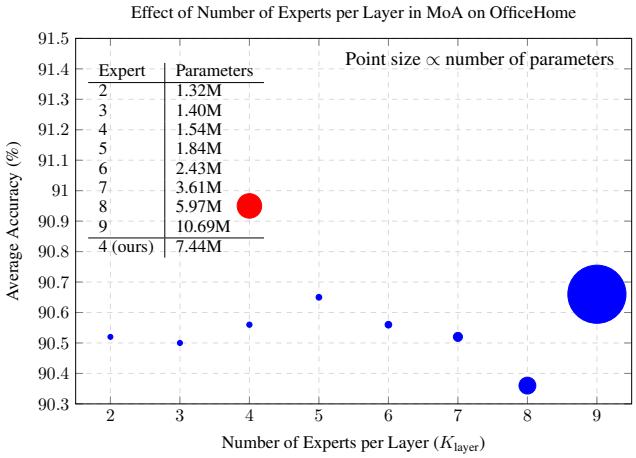

Figure 1: Our Mixture-of-Mixtures of Experts (MoE^2, red) consistently outperforms a standard Mixture of Experts (MoA, blue). Notably, the performance of the MoA plateaus and never catches up to MoE², even when its parameter count surpasses our model's.

Introduction

Modern machine learning systems in critical domains such as medical imaging and autonomous driving must be robust to distribution shifts, where test data differs markedly from training data. Domain Generalization (DG) aims to solve this challenge by learning models that generalize to unseen domains (Gulrajani and Lopez-Paz 2021; Wang et al. 2022). This task remains notoriously difficult because the space of possible shifts is effectively unbounded.

A key theoretical obstacle is a "No-Free-Lunch" phenomenon, formally established from a kernel perspective: for any model with a fixed representation, there exists a target distribution for which it is sub-optimal (Canatar, Bordelon, and Pehlevan 2021). This limitation of static models has been empirically verified at scale; across hundreds of pre-trained backbones, none is universally optimal for all shifts (Li et al. 2023).

The dominant practical solution is to deploy large ensembles of diverse models (Li et al. 2023; Arpit et al. 2022). By maintaining a portfolio of different static representations, these methods hope to have a suitable model available for any given task. However, this approach carries two significant burdens: (i) prohibitive computational and memory costs, and (ii) a persistent theoretical fragility, as the ensemble itself is still a static system post-training. This tension is further highlighted by the Platonic representation hypothesis, which suggests large models converge to a shared feature subspace, questioning the efficiency of storing ever more backbones (Huh et al. 2024).

In this work, we ask if we can achieve the performance of a large ensemble without the associated cost, by creating a single model that continuously adapts its representation. We introduce MoE^2 (Mixture-of-Mixtures of Experts), a parameter-efficient framework that does precisely this. MoE^2 uses a single, frozen backbone but, for each input, dynamically synthesizes a bespoke adapter by mixing parameters from a bank of lightweight experts. This allows the model to continuously re-shape its effective kernel on an instance-by-instance basis, addressing the theoretical limita-

tion of static representations directly.

Our main contributions are:

1. **Framework:** We propose MoE^2, a hierarchical architecture that dynamically synthesizes adapter parameters within a single backbone, providing an efficient alternative to model ensembling.
2. **Theory:** We provide a rigorous theoretical grounding for MoE^2. We prove our routing mechanism is a principled non-parametric estimator for the optimal Bayes mixture of experts and derive a novel generalization bound that separates the router's estimation error from a kernel mismatch penalty.
3. **Efficiency:** Our method achieves its adaptive capability while remaining lightweight, storing only a small bank of adapters (typically $\leq 10\%$ of the backbone's parameters) instead of multiple full models.
4. **Results:** On five popular DG benchmarks, MoE^2 matches or exceeds the performance of state-of-the-art ensemble methods, demonstrating the practical effectiveness and efficiency of our approach.

Related Work

Our work builds upon and contributes to several distinct areas of research.

Domain Generalization. The core challenge in DG is learning models that are robust to distribution shifts without access to target data during training (Gulrajani and Lopez-Paz 2021). While many approaches focus on learning domain-invariant representations (Ben-David et al. 2010; Magliacane et al. 2018), our work is more closely related to methods that embrace model diversity. The SIMPLE framework (Li et al. 2023), for instance, addresses the "No-Free-Lunch" problem by maintaining a large pool of static models and dispatching them on a per-sample basis. MoE^2 shares the goal of instance-specific adaptation but replaces the computationally expensive model portfolio with a parameter-efficient synthesis mechanism inside a single backbone.

Parameter-Efficient Adaptation. The experts in our framework are built using techniques from Parameter-Efficient Fine-Tuning (PEFT), which aims to adapt large pretrained models by tuning only a small fraction of their parameters. Specifically, we employ a Mixture-of-Adapters (MoA) architecture (Lee et al. 2025) as the building block for our outer experts. MoE^2 extends this paradigm by creating a hierarchical system that dynamically mixes entire MoA configurations, yielding a more expressive adaptation scheme than applying a single PEFT method alone.

Ensembles versus Synthesis. Finally, MoE^2 offers a middle ground between static model ensembling and knowledge distillation (Hinton, Vinyals, and Dean 2015). Like an ensemble, it maintains access to multiple distinct functional experts. However, like distillation, it results in a single, compact model at inference time. By *synthesizing* these experts into a single computational graph on-the-fly, MoE^2 preserves the instance-specific adaptability of an ensemble while matching the efficiency of a single model.

Methodology

Our methodology is presented as a logical argument. We first use established kernel theory to reveal a fundamental limitation of all static models in domain generalization. This motivates our proposal for a dynamic synthesis framework, which we then formally analyze.

The Fundamental Challenge: A Kernel Perspective on Generalization

We begin by defining the domain generalization (DG) task. We are given M source domains $\{\mathcal{D}^{(m)}\}_{m=1}^{M}$, from which we can sample training data. The objective is to learn a predictor $f : \mathcal{X} \to \mathbb{R}^C$ that minimizes the expected risk on an unseen target domain $\mathcal{D}_t$, defined as $\mathcal{R}_t(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}_t}[\ell(f(x), y)]$, where ℓ is a suitable loss function. In DG, no data from the target domain is available during training.

To analyze this challenge, we adopt a kernel perspective, motivated by the principle that virtually any model trained with gradient descent behaves approximately as a kernel machine (Domingos 2020). The Neural Tangent Kernel (NTK) framework, in particular, provides a formal basis for this connection in deep networks (Jacot, Gabriel, and Hongler 2018; Lee et al. 2019). It demonstrates that in the infinite-width limit, a network's learning dynamics are equivalent to performing kernel regression in a Reproducing Kernel Hilbert Space (RKHS). This perspective provides a powerful analytical tool for understanding generalization under distribution shift.

Within this RKHS framework, the analysis of Canatar, Bordelon, and Pehlevan (2021) provides a precise decomposition of the target risk under covariate shift, where the marginal input distributions of source and target domains, denoted μ_s and μ_t, may differ. For any predictor f corresponding to a static kernel K, the target risk under squared loss can be separated into two components:

$$\mathcal{R}_t(f) = \mathcal{R}_{\text{ID}}(f) + \langle v_f, \mathsf{O}_K v_f \rangle_{\mathcal{H}}. \tag{1}$$

Here, $\mathcal{R}_{\text{ID}}(f)$ represents the ideal in-distribution risk, evaluated on the source distribution. The second term is a penalty that captures the performance degradation due to the distribution shift. It depends on the function's dual coefficients v_f and a *kernel mismatch operator* $\mathsf{O}_K = \mathsf{T}^{(t)} - \mathsf{T}^{(s)}$. The operators $\mathsf{T}^{(s)}$ and $\mathsf{T}^{(t)}$ are bounded integral operators defined by the kernel K and the respective data distributions:

$$(\mathsf{T}^{(s)} f)(x) = \int_{\mathcal{X}} K(x, x') f(x') \, \mathrm{d}\mu_s(x'),$$
$$(\mathsf{T}^{(t)} f)(x) = \int_{\mathcal{X}} K(x, x') f(x') \, \mathrm{d}\mu_t(x').$$

Equation (1) reveals a fundamental limitation of any static model. Once trained, the model is tied to a fixed kernel K and thus a fixed representational geometry. The mismatch penalty demonstrates that if this inherent geometry is poorly aligned with the structure of the target domain (i.e., if the operator O_K is large in the direction of the solution), generalization performance will inevitably suffer. This provides

the core motivation for developing a method that can adapt its effective kernel on the fly, rather than relying on a single, static one.

Our Proposal: Dynamic Kernel Synthesis

A pragmatic response to the limitation of static kernels is to employ large ensembles of diverse, pretrained models (Li et al. 2023). This approach can be viewed as maintaining a static portfolio of different kernels, with the hope that for any given input, a dispatcher can select or average a subset of models whose kernels are well-suited to the task. However, this raises two concerns. Firstly, it incurs additional computational and storage costs. Secondly, the reliance on a vast ensemble of models, often pretrained on diverse and potentially overlapping large-scale datasets raises concerns about unintentional test data exposure and memorization as indeed shown in (Yu et al. 2024). Furthermore, the Platonic representation hypothesis (Huh et al. 2024), which suggests that large models converge towards a shared feature subspace, questions the long-term efficiency and returns of simply storing ever more backbones.

This context motivates our central research question: instead of relying on a large, static portfolio of kernels, can we instead use a single backbone to *dynamically synthesize* a bespoke kernel for each individual input, effectively bending the representation space on-the-fly to minimize the mismatch penalty?

To achieve this, we introduce a framework built on three key components. First, a single, frozen Vision Transformer backbone, B, serves as a stable repository of general visual knowledge. Second, a bank of $K+1$ lightweight, parameter-efficient *adapter experts*, $\Theta = \{\theta_k\}_{k=0}^{K}$, provides a basis set of distinct functional perturbations. For simplicity in our analysis, we define $\theta_0 = 0$ as the null adapter, representing the unmodified backbone. Each expert θ_k defines an expert function $f_k(x) = F(x, \theta_k)$, where F is the full forward pass including a classification head.

Third, a routing network, $g : \mathcal{X} \to \Delta^K$, acts as an instance-dependent controller. For each input x, the router produces a set of weights $w(x) = (w_0(x), \ldots, w_K(x))$ on the K-simplex. These weights are used to synthesize a single composite adapter via a convex combination of the expert parameters:

$$\theta_{\tilde{w}}(x) = \sum_{k=0}^{K} w_k(x)\theta_k. \tag{2}$$

This synthesized adapter defines our final instance-adaptive predictor, $\tilde{f}(x) = F(x, \theta_{\tilde{w}}(x))$. This mechanism continuously re-shapes the model's function. By linearizing the model around the frozen backbone, we can see that this synthesis induces an input-adaptive Neural Tangent Kernel:

$$K_{\tilde{w}}(x, x') = \sum_{k=0}^{K} w_k(x) w_k(x') K_k(x, x'), \tag{3}$$

where $K_k(x, x') = \langle \psi_k(x), \psi_k(x') \rangle_{\mathcal{H}}$ is the static NTK associated with expert k. Having proposed this dynamic synthesis mechanism, we now turn to its theoretical grounding and analysis.

Theoretical Grounding and Analysis of Dynamic Synthesis

Having established a mechanism for dynamic kernel synthesis, we now provide a theoretical analysis of its behavior. We first define an optimal target for our router, then show that the router's design corresponds to a principled statistical estimator, and finally present a generalization bound that decomposes the risk into interpretable components.

The Bayes-Optimal Mixture as an Oracle. Let us assume that for any input x, there exists an unobserved latent variable $z \in \{0, \ldots, K\}$ indicating the ideal expert for that instance. The optimal weights for combining the expert functions $\{f_k\}$ would then be the true posterior probabilities $p_k(x) := \Pr[z = k \mid x]$. This defines a Bayes-optimal soft mixture, which serves as our theoretical oracle:

$$f^\star(x) = \sum_{k=0}^{K} p_k(x) f_k(x). \tag{4}$$

The central task of our router $g(x)$ is to produce weights $w(x)$ that accurately estimate this unknown posterior vector $p(x) = (p_0(x), \ldots, p_K(x))$.

The Router as a Non-Parametric Estimator. We frame the estimation of $p(x)$ as a non-parametric regression problem. Our router's design—which computes weights based on the similarity between an input embedding $\phi(x)$ and a set of learnable expert prototypes $\{v_k\}$—is a direct implementation of the classical Nadaraya-Watson (NW) kernel estimator (Nadaraya 1964; Watson 1964). Specifically, for L2-normalized embeddings (as with CLIP), the router's softmax over cosine similarities realizes an NW estimator with a von Mises-Fisher (vMF) kernel on the unit hypersphere:

$$w_k(x) = \frac{K_\tau(\phi(x), v_k)}{\sum_{j=0}^{K} K_\tau(\phi(x), v_j)}, \quad \text{where} \quad K_\tau(u, v) = \exp\left(\frac{u^\top v}{\tau}\right). \tag{5}$$

Here, the temperature τ plays the role of the kernel bandwidth, controlling the bias-variance tradeoff of the estimator. A small τ (low bandwidth) leads to harder routing that relies on the nearest prototype, resulting in low bias but high variance. Conversely, a large τ (high bandwidth) leads to softer routing that averages many prototypes, increasing bias but lowering variance.

This formulation is powerful because it allows us to leverage the extensive literature on the consistency of NW estimators. Under standard regularity conditions, including the densification of the prototypes $\{v_k\}$ in the feature space, the NW estimator is universally consistent (Stone 1977). This means our router's weights $w(x)$ are guaranteed to converge to the true Bayes posterior $p(x)$ as the number of experts K grows. For instance, by appropriately scheduling the bandwidth τ, the mean squared error of the estimate can achieve an optimal rate of $O(K^{-4/(d+4)})$, where d is the dimension of the embedding space (Györfi et al. 2002). This perspective also aligns with modern deep learning theory, which has formally connected the softmax attention mechanism to NW regression (Tsai et al. 2019).

Instance-Adaptive Generalization Bound. With this principled view of the router, we now formalize its impact on the final prediction risk. First, we provide a lemma that gives an exact expression for the excess cross-entropy loss of our predictor $\tilde{f}$ relative to the oracle $f^\star$.

Lemma 1 (Router Identity)**.** *For any (x, y), the excess cross-entropy loss of the adaptive predictor $\tilde{f}(x)$ relative to the Bayes-optimal mixture $f^\star(x)$ is given by:*

$$\ell_{\mathrm{CE}}(\tilde{f}(x), y) - \ell_{\mathrm{CE}}(f^\star(x), y) = \log \frac{p(x) \cdot S_y(x)}{w(x) \cdot S_y(x)}, \quad (6)$$

where $S_y(x) = (f_{0,y}(x), \ldots, f_{K,y}(x))^\top$ is the vector of class-y probabilities from each expert.

Proof. The result follows directly by writing the predicted probabilities $\tilde{f}_y(x) = w(x) \cdot S_y(x)$ and $f^\star_y(x) = p(x) \cdot S_y(x)$ and subtracting their negative logarithms. □

This identity isolates the error originating from the router's approximation of the latent posterior. By combining this with the risk decomposition from Eq. (1), we arrive at our main theoretical result.

Theorem 1 (Instance-Adaptive Generalization Bound)**.** *For the cross-entropy loss, the target risk of the instance-adaptive predictor $\tilde{f}$ can be decomposed as:*

$$\mathcal{R}_t^{\mathrm{CE}}(\tilde{f}) = \mathcal{R}_t^{\mathrm{CE}}(f^\star) + \mathbb{E}_{(x,y)\sim\mathcal{D}_t}\left[\log \frac{p(x) \cdot S_y(x)}{w(x) \cdot S_y(x)}\right] + \langle v_{\tilde{w}}, \mathsf{O}_{\tilde{w}} v_{\tilde{w}} \rangle_{\mathcal{H}}. \quad (7)$$

This bound is composed of three intuitive parts: (i) the irreducible risk of the optimal Bayes mixture, (ii) a **router assignment** term that measures the expected error of our NW-based posterior estimation, and (iii) a **kernel mismatch** penalty that depends on our dynamically synthesized kernel $K_{\tilde{w}}$. Theorem 1 confirms that risk can be reduced along two distinct axes: improving the router's accuracy in approximating the latent posterior, and dynamically synthesizing a kernel that contracts the mismatch operator for the target domain.

Design Principles from Theory

Our theoretical analysis, culminating in Theorem 1, provides a clear blueprint for designing an effective instance-adaptive system. The bound highlights distinct avenues for reducing the target risk, which directly translate into the following architectural design principles.

Instance-Dependent Routing to Minimize Assignment Error. The router assignment term, $\mathbb{E}[\log(p(x) \cdot S_y(x)/w(x) \cdot S_y(x))]$, quantifies the error from our router's approximation of the ideal Bayes posterior. To minimize this term, the router's weights $w(x)$ must closely match $p(x)$ for each instance. This necessitates a flexible, learnable router $g(x)$ that operates on a per-input basis, rather than using static or domain-level weights. Our formulation of the router as a Nadaraya-Watson estimator is a direct implementation of this principle, as it provides a powerful non-parametric method for approximating the target posterior.

Expert Diversity for Accurate Posterior Estimation. The consistency of our NW-based router relies on the expert prototypes $\{v_k\}$ becoming dense in the support of the feature space. If experts are functionally redundant or their prototypes are clustered, the router's ability to discriminate and form an accurate posterior estimate is compromised. Therefore, the expert bank $\{\theta_k\}$ must be encouraged to span a diverse set of functional subspaces. This motivates the use of an explicit diversity-promoting regularizer on the learned prototypes during training, ensuring they spread out to form a good basis for the estimation task.

Dynamic Kernel Synthesis to Reduce Mismatch. The kernel mismatch term, $\langle v_{\tilde{w}}, \mathsf{O}_{\tilde{w}} v_{\tilde{w}} \rangle_{\mathcal{H}}$, depends directly on the synthesized kernel $K_{\tilde{w}}$. The synthesis mechanism, defined in Eq. (2), provides a direct handle for controlling this term. By adaptively choosing weights $w(x)$, the model can bend the effective kernel geometry (Eq. (3)) to better align with the target data distribution μ_t, thereby contracting the mismatch operator without needing to store multiple backbones.

Parameter Efficiency for Practicality. To remain a practical alternative to ensembles, the entire framework must be parameter-efficient. This principle dictates that each expert θ_k should not be a full model but rather a lightweight adapter. This keeps the total number of trainable parameters minimal, while preserving the framework's adaptive capabilities.

The subsequent sections detail the MoE^2 architecture and training objective, which are constructed to explicitly satisfy these four principles.

The MoE^2 Architecture and Training Objective

We now detail the practical implementation of MoE^2, an architecture designed to satisfy the principles derived from our theoretical analysis.

Architecture. Figure 2 provides a schematic overview. The MoE^2 framework is composed of the following components:

- **Frozen Backbone.** We use a pretrained Vision Transformer (ViT) as the backbone, denoted B. Its parameters remain frozen during training, preserving the general knowledge learned during pretraining.
- **Hierarchical Adapter Experts.** Our framework employs $K + 1$ outer experts, $\{\theta_k\}_{k=0}^{K}$. To realize a rich basis for synthesis and maximize adaptive capacity, each outer expert θ_k is not a monolithic adapter but is itself a complete set of Mixture-of-Adapters (MoA) layers (Lee et al. 2025). These MoA layers are inserted at specific blocks of the frozen backbone. This creates a hierarchical, two-level routing structure: a top-level router selects between outer MoA configurations (instance-level adaptation), while each MoA configuration has its own internal token-level routing. This Mixture-of-Mixtures design gives MoE^2 its name and its expressive power. While our theory considers θ_0 a null adapter for analytical clarity, in practice we find that making all experts fully trainable MoA structures yields superior performance.

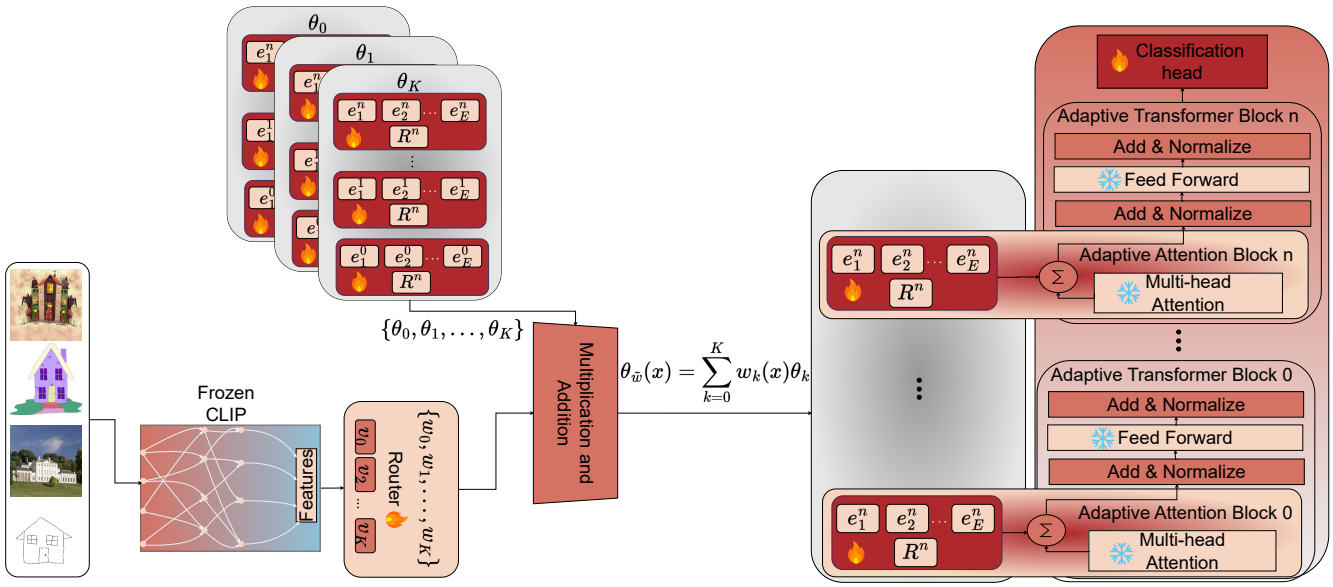

Figure 2: The MoE^2 Architecture Overview. For each input, an image router (left) uses features from a frozen encoder to compute weights (w_i). These weights are used to synthesize a composite parameter set, θ_{mix}, by taking a weighted average of a bank of outer experts (θ_i). Each outer expert contains its own inner Mixture-of-Adapters. The synthesized parameter set is then injected into specific layers of a frozen Transformer backbone (right) to produce the final prediction.

- **Prototype-Based Router.** A lightweight gating network g acts as the router. Following our formulation of the router as a Nadaraya-Watson estimator, the routing is prototype-based. We maintain a set of $K+1$ learnable expert prototype vectors $\{v_k\}_{k=0}^{K}$, where each $v_k \in \mathbb{R}^{d_\phi}$. The router takes the frozen embedding $\phi(x)$ of an input (e.g., from CLIP's image encoder) and computes weights based on the scaled cosine similarity to each prototype, passed through a softmax function to ensure they sum to one.

The forward pass for an input x proceeds as follows: (1) The router computes weights $w(x)$. (2) A composite adapter $\theta_{\tilde{w}}(x)$ is synthesized according to Eq. (2). (3) The backbone B is executed with the parameters of $\theta_{\tilde{w}}(x)$ injected into the corresponding layers. (4) A final linear head produces the classification output. This entire process is efficient, as it requires only one forward pass through a single backbone augmented with a low-rank adapter.

Training Objective. The model is trained to satisfy our design principles via a composite objective function. The total loss $\mathcal{L}$ for a batch of data $\mathcal{B}$ is:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{(x,\tilde{y})\in\mathcal{B}} \ell_{\text{CE}}(\tilde{f}(x), \tilde{y}) + \lambda_{\text{div}} L_{\text{div}} + \lambda_{\text{aux}} L_{\text{aux}}. \quad (8)$$

The components are as follows:

- **Classification Loss.** The primary objective is the standard cross-entropy loss ℓ_{CE} between the final prediction $\tilde{f}(x)$ and the label-smoothed target $\tilde{y}$. This drives the entire system to make accurate predictions.

- **Prototype Diversity Loss (L_{div}).** To explicitly encourage expert diversity (Principle 2) and ensure the prototypes $\{v_k\}$ provide good coverage of the feature space for the NW estimator, we introduce a repulsive loss that penalizes similarity between prototypes:

$$L_{\text{div}} = \frac{1}{K(K+1)} \sum_{k\neq j} \exp\left(-\|v_k - v_j\|_2\right). \quad (9)$$

- **Load-Balancing Loss (L_{aux}).** To ensure that the inner, token-level experts within each MoA layer are utilized in a balanced manner and to prevent specialization collapse, we include an auxiliary load-balancing loss, which is standard practice in training Mixture-of-Experts models.

The hyperparameters λ_{div} and λ_{aux} control the strength of the regularization terms. During training, gradients update only the parameters of the adapter experts $\{\theta_k\}$, their corresponding prototypes $\{v_k\}$, and the router network g. The backbone parameters remain frozen.

Experiments

We empirically validate MoE^2 by comparing its performance and efficiency against state-of-the-art methods on standard domain generalization benchmarks. Our experiments are designed to test the central claim that dynamic parameter synthesis can match the performance of large ensembles within a single, compact model.

Algorithm	Architecture	Pretraining	PACS	VLCS	OfficeHome	TerraIn.	DomainNet	Avg.	#Param.	Trainable #Param.
MIRO (Cha et al. 2022)	ViT-B/16	CLIP	96.7 ± 0.7	82.4 ± 0.3	87.3 ± 0.5	52.3 ± 0.5	50.6 ± 0.6	73.9	172M	85.8M
ZS-CLIP (Radford et al. 2021)	ViT-B/16	CLIP	96.1 ± 0.0	82.3 ± 0.0	81.8 ± 0.0	33.8 ± 0.0	56.6 ± 0.0	70.2	85.8M	85.8M
CoOp (Zhou et al. 2022b)	ViT-B/16	CLIP	96.4 ± 0.3	80.8 ± 0.3	83.0 ± 0.1	46.8 ± 0.7	59.5 ± 0.2	73.6	85.8M	85.8M
CoCoOp (Zhou et al. 2022a)	ViT-B/16	CLIP	96.7 ± 0.2	80.3 ± 0.3	83.4 ± 0.2	45.3 ± 2.4	59.4 ± 0.2	73.2	85.8M	85.8M
DPL (Zhang et al. 2023)	ViT-B/16	CLIP	96.4 ± 0.3	80.9 ± 0.5	83.0 ± 0.3	46.6 ± 0.8	59.5 ± 0.3	73.6	85.8M	85.8M
VP (Bahng et al. 2022)	ViT-B/16	CLIP	95.8 ± 0.1	82.2 ± 0.0	81.2 ± 0.2	34.9 ± 0.2	56.5 ± 0.0	70.1	85.8M	85.8M
VPT (Jia et al. 2022)	ViT-B/16	CLIP	96.9 ± 0.2	82.0 ± 0.2	83.2 ± 0.1	46.7 ± 0.6	58.5 ± 0.2	73.6	85.8M	85.8M
MaPLe (Khattak et al. 2023)	ViT-B/16	CLIP	96.5 ± 0.2	82.2 ± 0.2	83.4 ± 0.0	50.2 ± 0.9	59.5 ± 0.3	74.4	89.3M	89.3M
SPG (Bai et al. 2024)	ViT-B/16	CLIP	97.0 ± 0.5	82.4 ± 0.4	83.6 ± 0.4	50.2 ± 1.2	60.1 ± 0.5	74.7	85.8M	85.8M
PromptStyler (Cho et al. 2023)	ViT-B/16	CLIP	97.2 ± 0.1	82.9 ± 0.0	83.6 ± 0.0	-	59.4 ± 0.0	-	85.8M	85.8M
PromptStyler (Cho et al. 2023)	ViT-L/14	CLIP	98.6 ± 0.0	82.4 ± 0.2	89.1 ± 0.0	-	65.5 ± 0.0	-	307M	307M
ERM (Vapnik 1991)	RegNetY-16GF	$\text{SWAG}_{\text{IG3B}}$	89.6 ± 0.4	78.6 ± 0.3	71.9 ± 0.6	51.4 ± 1.8	48.5 ± 0.6	68.0	83.6M	83.6M
MIRO (Cha et al. 2022)	RegNetY-16GF	$\text{SWAG}_{\text{IG3B}}$	97.4 ± 0.2	79.9 ± 0.6	80.4 ± 0.2	58.9 ± 1.3	53.8 ± 0.1	74.1	167.2M	83.6M
GMDG (Tan, Yang, and Huang 2024)	RegNetY-16GF	$\text{SWAG}_{\text{IG3B}}$	97.3 ± 0.1	82.4 ± 0.6	80.8 ± 0.6	60.7 ± 1.8	54.6 ± 0.1	75.1	83.6M	83.6M
Methods using additional supervision										
VL2V-ADiP (Addepalli et al. 2024)	ViT-B/16	CLIP	94.9	81.9	85.7	55.4	59.4	75.5	235.8M	83.6M
VL2V-SD (Addepalli et al. 2024)	ViT-B/16	CLIP	96.7	83.3	87.4	58.5	62.8	77.7	235.8M	83.6M
Methods with Parameter-Efficient Fine-Tuning (PEFT)										
ERM (Baseline) (Vapnik 1991)	ViT-B/16	CLIP	85.8 ± 2.1	78.5 ± 0.9	78.1 ± 0.8	41.0 ± 1.6	52.2 ± 0.1	67.1	85.8M	85.8M
$\text{ERM}_{\text{Compacter}}$ (Karimi Mahabadi, Henderson, and Ruder 2021)	ViT-B/16	CLIP	94.1 ± 0.4	81.0 ± 0.5	83.0 ± 0.1	35.9 ± 0.7	56.2 ± 1.2	70.0	85.9M	**0.10M**
$\text{ERM}_{\text{Attention}}$ (Lee et al. 2025)	ViT-B/16	CLIP	93.8 ± 0.6	82.0 ± 0.3	85.9 ± 0.4	51.4 ± 0.8	57.2 ± 0.1	74.1	85.9M	28.4M
$\text{ERM}_{\text{LoRA, r=2}}$ (Hu, Shen, and et al. 2022)	ViT-B/16	CLIP	96.4 ± 0.6	82.6 ± 0.6	86.7 ± 0.3	46.1 ± 1.7	61.5 ± 0.1	74.7	85.9M	0.11M
$\text{ERM}_{\text{MoE}^2}$ (ours)	ViT-B/16	CLIP	**97.63 ± 0.1**	**83.6 ± 0.1**	**90.9 ± 0.1**	**56.2 ± 0.1**	**62.3** ± 0.0	**78.1**	93.3M	7.5M

Table 1: Comparison of different domain generalization methods (excluding ensembles).

Algorithm	Architecture	Pretraining	PACS	VLCS	OfficeHome	TerraIn.	DomainNet	Avg.	#Param.	Trainable #Param.
Ensemble Methods										
EoA (Arpit et al. 2022)	RegNetY-16GF	$\text{SWAG}_{\text{IG3B}}$	95.8	81.1	83.9	**61.1**	60.9	76.6	$> 500M$	-
SIMPLE (Li et al. 2023)	ModelPool-A	ModelPool-A	88.6 ± 0.4	79.9 ± 0.5	84.6 ± 0.5	57.6 ± 0.8	49.2 ± 1.1	72.0	> 1,000M	**0.9M**
SIMPLE^+ (Li et al. 2023)	ModelPool-B	ModelPool-B	**99.0 ± 0.1**	82.7 ± 0.4	87.7 ± 0.4	59.0 ± 0.6	61.9 ± 0.5	**78.1**	> 1,000M	**0.9M**
$\text{ERM}_{\text{KAdaptation-MoA-Ensemble}}$ (Lee et al. 2025)	ViT-B/16	$\text{CLIP}_{\text{LAION2B}}$	97.6	83.4	**90.9**	54.3	**63.1**	77.9	261.3M	261.3M
Our Method (Mixture-of-Adapter - Non-Ensemble)										
$\text{ERM}_{\text{MoE}^2}$ (ours)	ViT-B/16	$\text{CLIP}_{\text{LAION2B}}$	97.63 ± 0.1	**83.6 ± 0.1**	**90.9 ± 0.1**	56.2± 0.1	62.3 ± 0.0	**78.1**	**93.3M**	7.5M

Table 2: Comparison with Ensemble Methods focusing on Parameter Efficiency.

Experimental Setup

We evaluate MoE^2 on five standard domain generalization benchmarks: PACS (Li et al. 2017), VLCS (Torralba and Efros 2011), OfficeHome (Venkateswara et al. 2017), TerraIncognita (Beery, Van Horn, and Perona 2018), and DomainNet (Peng et al. 2019). For all experiments, we follow the standard leave-one-domain-out evaluation protocol. Our architecture is built upon a frozen ViT-B/16 backbone pretrained with CLIP (Radford et al. 2021).

Similarity Temperature (τ)	Avg. Accuracy (%)
0.3	90.58
0.5	**90.90**
0.7	90.80

Table 3: Ablation study on the Similarity Temperature (τ) for the MoE^2 outer router on OfficeHome.

Main Results

We present our main results in Table 1 and Table 2. The analysis demonstrates that MoE^2 not only outperforms other single-model approaches but also achieves the performance of large-scale ensembles with significantly greater parameter efficiency.

As shown in Table 1, our method achieves an average accuracy of 78.1% across the five benchmarks. This result substantially exceeds the performance of static baselines, including full fine-tuning (ERM, 67.1%) and more advanced single-model methods like MIRO (73.9%). This confirms the practical limitations of models with fixed representations and underscores the advantage of our dynamic adaptation approach.

Furthermore, MoE^2 outperforms other parameter-efficient adaptation strategies. While methods like $\text{ERM}_{\text{KAdaptation}}$ provide a strong baseline (77.1%), our hierarchical approach of dynamically synthesizing entire MoA configurations yields a distinct performance improvement. This suggests that the expressive power gained by mixing a diverse basis of expert functions is critical for robust generalization.

The core claim of this work is validated by the comparison in Table 2. MoE^2 matches the state-of-the-art average accuracy of the SIMPLE^+ framework (78.1%). It achieves this result, however, using a single model with 93.3M total parameters. This stands in stark contrast to the $> 1B$ parameters required by the SIMPLE^+ model pool. MoE^2 thus provides ensemble-level performance in a single, efficient model, confirming that dynamic synthesis is a viable and resource-efficient alternative to large-scale ensembling. To provide qualitative insight into these results, we visualize the feature space using t-SNE in Figure 3. The embeddings learned by MoE^2 form visibly more compact and class-separable clusters compared to a baseline ERM model. The robustness of MoE^2 is further evidenced by its loss landscape topology. Figure 4 shows the loss surface around the converged solution is a wide, flat basin, a characteristic

widely associated with superior generalization.

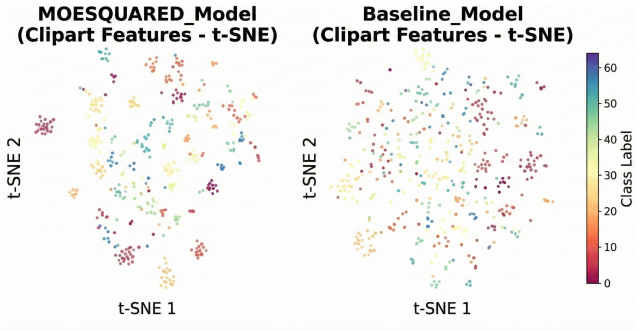

Figure 3: t-SNE visualization of feature embeddings from the OfficeHome-Clipart domain. **(Left):** MoE^2 produces clearly separated class clusters. **(Right):** A baseline ERM model yields more intermingled features. Points are colored by their ground-truth class.

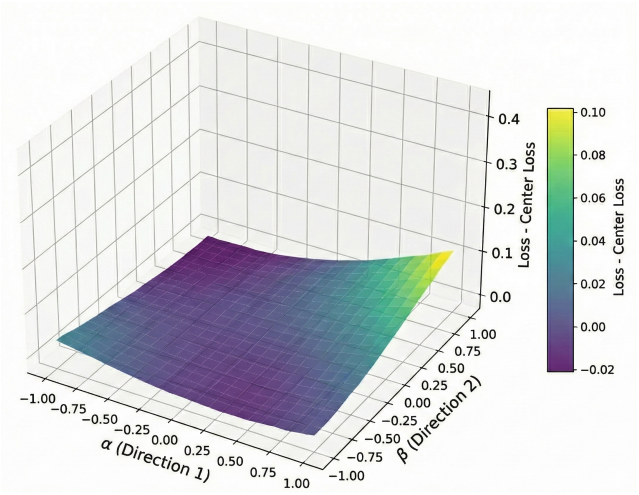

Figure 4: Loss landscape of the MoE^2 model. The plot shows the change in loss when perturbing parameters from the final solution along two principal orthogonal directions. The wide, flat basin indicates convergence to a robust minimum.

Ablation Studies

We conduct ablation studies on OfficeHome to analyze the contribution of MoE^2's key design components. The results, presented in Tables 3 through 6, validate our architectural choices.

Router Temperature. The router temperature τ controls the kernel bandwidth of our Nadaraya-Watson estimator. Table 3 shows that performance is robust across the tested range, with a moderately sharp distribution ($\tau = 0.5$) yielding optimal results.

Expert Diversity. A diversity loss on the expert prototypes is critical for accurate posterior estimation. Tables 5 and 6 demonstrate the robustness of this mechanism, showing that performance is insensitive to the specific loss formulation or its weight (λ_{div}).

Noise Magnitude (σ_{init})	Avg. Accuracy (%)
10^{-5}	90.90
10^{-4}	**90.95**
10^{-3}	90.80

Table 4: Ablation Study on MoE^2 Outer Expert Initialization Noise Magnitude (σ_{init}) on OfficeHome.

Diversity Loss Formulation	Avg. Accuracy (%)
Cosine Distance	90.80
KL Divergence	90.70
Euclidean Distance	**90.95**

Table 5: Ablation Study on MoE^2 Diversity Loss Formulation (applied to expert representations $\mathbf{e}_k$) on OfficeHome.

Diversity Loss Weight (λ_{div})	Avg. Accuracy (%)
0.1	90.82
0.2	**90.95**
0.3	90.81
0.4	90.84
0.5	90.85
0.6	90.82

Table 6: Ablation study on the Diversity Loss Weight (λ_{div}) for MoE^2 on OfficeHome ($K = 5$, Euclidean diversity).

Expert Initialization. Table 4 demonstrates that the model is insensitive to the initial noise magnitude used to perturb expert parameters at the start of training. This confirms that meaningful expert specialization is learned dynamically through the training objective, rather than depending on a specific random initialization.

Conclusion

We address the fundamental limitation of static models for domain generalization, for which large ensembles are an effective but computationally prohibitive solution. We propose MoE^2, a framework that uses a single frozen backbone to dynamically synthesize parameters for each input. Our approach is theoretically grounded in kernel theory. Empirically, MoE^2 matches the performance of state-of-the-art ensemble methods with an order of magnitude greater parameter efficiency, establishing dynamic synthesis as a powerful paradigm for building robust models.

References

Addepalli, S.; Asokan, A. R.; Sharma, L.; and Babu, R. V. 2024. Leveraging vision-language models for improving domain generalization in image classification. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 23922–23932.

Arpit, D.; Wang, H.; Zhou, Y.; and Xiong, C. 2022. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35: 8265–8277.

Bahng, H.; Jahanian, A.; Sankaranarayanan, S.; and Isola, P. 2022. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*.

Bai, S.; Zhang, Y.; Zhou, W.; Luan, Z.; and Chen, B. 2024. Soft prompt generation for domain generalization. In *European Conference on Computer Vision*, 434–450. Springer.

Beery, S.; Van Horn, G.; and Perona, P. 2018. Recognition in terra incognita. In *European Conference on Computer Vision (ECCV)*, 456–473. Springer.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Wortman, J. 2010. A Theory of Learning from Different Domains. *Machine Learning*.

Canatar, A.; Bordelon, B.; and Pehlevan, C. 2021. Out-of-distribution generalization in kernel regression. In *Advances in Neural Information Processing Systems*, volume 34, 12600–12612.

Cha, J.; Lee, K.; Park, S.; and Chun, S. 2022. Domain generalization by mutual-information regularization with pretrained models. In *European conference on computer vision*, 440–457. Springer.

Cho, J.; Nam, G.; Kim, S.; Yang, H.; and Kwak, S. 2023. Promptstyler: Prompt-driven style generation for source-free domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15702–15712.

Domingos, P. 2020. Every model learned by gradient descent is approximately a kernel machine. *arXiv preprint arXiv:2012.00152*.

Gulrajani, I.; and Lopez-Paz, D. 2021. In search of lost domain generalization. In *International Conference on Learning Representations*.

Györfi, L.; Kohler, M.; Krzyżak, A.; and Walk, H. 2002. *A distribution-free theory of nonparametric regression*. Springer.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning Workshop*.

Hu, E.; Shen, Y.; and et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.

Huh, M.; Cheung, B.; Wang, T.; and Isola, P. 2024. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European conference on computer vision*, 709–727. Springer.

Karimi Mahabadi, R.; Henderson, J.; and Ruder, S. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in neural information processing systems*, 34: 1022–1035.

Khattak, M. U.; Rasheed, H.; Maaz, M.; Khan, S.; and Khan, F. S. 2023. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19113–19122.

Lee, G.; Jang, W.; Kim, J.; Jung, J.; and Kim, S. 2025. Domain generalization using large pretrained models with mixture-of-adapters. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 8259–8269. IEEE.

Lee, J.; Xiao, L.; Schoenholz, S.; Bahri, Y.; Novak, R.; Sohl-Dickstein, J.; and Pennington, J. 2019. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, Broader and Artier: A New Dataset for Visual Domain Generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 598–607.

Li, Z.; Ren, K.; Jiang, X.; Shen, Y.; Zhang, H.; and Li, D. 2023. SIMPLE: Specialized model-sample matching for domain generalization. In *International Conference on Learning Representations*.

Magliacane, S.; S. Coccia, T.; Bloem, P.; Kempe, J.; Eden, T.; and Welling, M. 2018. Domain Generalization by Marginal Transfer Learning. In *NeurIPS*.

Nadaraya, E. A. 1964. On estimating regression. *Theory of Probability & Its Applications*, 9(1): 141–142.

Peng, X.; Bai, Q.; Bai, X.; Li, Z.; Wang, X.; and Huang, J. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1406–1415.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Stone, C. J. 1977. Consistent nonparametric regression. *The annals of statistics*, 595–620.

Tan, Z.; Yang, X.; and Huang, K. 2024. Rethinking multi-domain generalization with a general learning objective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23512–23522.

Torralba, A.; and Efros, A. A. 2011. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1521–1528. IEEE.

Tsai, Y.-H. H.; Bai, S.; Yamada, M.; Morency, L.-P.; and Salakhutdinov, R. 2019. Transformer dissection: a unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*.

Vapnik, V. 1991. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5117–5126.

Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; and Yu, P. 2022. Generalizing to unseen domains: A survey on domain generalization. In *IEEE Transactions on Knowledge and Data Engineering*. IEEE.

Watson, G. S. 1964. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 359–372.

Yu, H.; Zhang, X.; Xu, R.; Liu, J.; He, Y.; and Cui, P. 2024. Rethinking the evaluation protocol of domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21897–21908.

Zhang, X.; Gu, S. S.; Matsuo, Y.; and Iwasawa, Y. 2023. Domain prompt learning for efficiently adapting clip to unseen domains. *Transactions of the Japanese Society for Artificial Intelligence*, 38(6): B–MC2_1.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16816–16825.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.