LEARNING SUBSET-SHARED INVARIANCES FOR DOMAIN GENERALIZATION WITH MIXTURE-OF-EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization (DG) aims to learn a model from one or more source domains that generalizes to an unseen target domain without accessing target data during training. A common approach enforces invariance of representations across all source domains, assuming predictive structure is globally shared. However, exact conditional invariance can exclude predictive factors shared only within subsets of source domains. To address this limitation, we propose subset-shared invariance, where predictive structure is assumed stable only within domain subsets. We implement this principle with a mixture-of-experts architecture, where routing induces expert- and class-specific alignment strengths across source-domain pairs and softly composes the resulting expert representations for prediction. This creates routing-conditioned alignment that prioritizes domain relationships according to learned responsibilities rather than enforcing them uniformly. To facilitate effective decomposition, we develop training objectives that encourage selective alignment, confident and balanced routing, and diverse expert specialization. Experiments on DomainBed benchmarks demonstrate improved out-of-domain generalization and greater robustness under increasing domain heterogeneity. Our results suggest that DG should move beyond enforcing a single global invariance and instead model invariance through partially shared structure across domain subsets.

1 INTRODUCTION

Domain generalization (DG) refers to the challenge of training models that perform well on unseen domains, without having access to data from these target domains during training (Muandet et al., 2013; Gulrajani & Lopez-Paz, 2021; Koh et al., 2021). A widely used approach in DG is to develop domain-invariant representations by ensuring that the conditional distribution $P(Z|Y)$ remains consistent across different domains. This idea is central to many methods, including feature alignment (Sun & Saenko, 2016; Peng et al., 2019) and invariant learning frameworks such as invariant risk minimization (IRM) (Arjovsky et al., 2019) and its extensions (Rame et al., 2022; Shi et al., 2022). Nonetheless, applying invariance uniformly to all domains may be too restrictive (see Figure 1). In real-world settings, domains may only partially overlap in structure. For instance, specific features could be stable in some domains but vary in others (Piratla et al., 2020; Yao et al., 2024). Therefore, imposing global invariance can eliminate useful predictive information, especially when domain shifts are heterogeneous (Geirhos et al., 2021; Wang et al., 2024). This observation indicates that predictive structure is not universally shared, but instead distributed among subsets of domains. Similar patterns of subset-based heterogeneity are also found

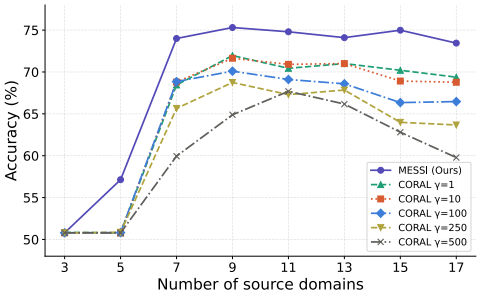


Figure 1: Test accuracy under fixed-budget source-domain growth. Performance initially improves as more source domains are added, but begins to decline as domain heterogeneity increases. This trend is consistent with global invariance becoming restrictive and discarding useful predictive structure. In contrast, MESSI remains stable by preserving predictive factors shared within domain subsets.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

in multimodal learning, where individual samples may use different modality combinations (Yun et al., 2024).

We study when global conditional-invariance constraints exclude partially shared predictive structure, motivating a routing-conditioned view of alignment that learns how strongly each source-domain relationship should be aligned rather than treating all domain pairs uniformly. Specifically, we introduce *subset-shared invariance*, where predictive structure may be shared more strongly among some source domains than others rather than uniformly across all domains. This motivates a structured decomposition in which different experts emphasize different domain relationships. In practice, we realize this principle through a soft relaxation, using input-dependent routing statistics to assign expert- and class-specific alignment strengths across source-domain pairs and to combine the resulting expert representations for prediction. To instantiate this principle, we leverage a mixture-of-experts (MoE) architecture (Jacobs et al., 1991; Shazeer et al., 2017) as a mechanism for learning such a decomposition. A routing function induces a soft partition over inputs, which in turn defines subsets of domains associated with each expert.

We softly weight invariance across source-domain pairs using routing-induced expert responsibilities. This results in a *routing-induced invariance*, where the model learns *where* invariance should hold, rather than enforcing it uniformly across all domains.

MoE models have previously been used in DG to improve capacity and specialization (Li et al., 2023; Chen et al., 2024; Dai et al., 2021). While prior work mainly treats experts as specialized feature modules, our method uses routing to shape the invariance structure itself, allowing different experts to capture relationships shared across different domain subsets. To realize this idea, we propose a subset-conditioned invariance objective that matches class-conditional expert-feature distributions with routing-dependent strengths. Expert responsibilities are aggregated within each class-domain cell to weight source-domain pairs, and entropic optimal transport (OT) is used as the primary discrepancy. We further encourage expert diversity together with confident and balanced routing.

- We identify a limitation of global invariance and show that enforcing invariance across more domain relationships can restrict predictive information. This motivates *subset-shared invariance*, where invariance is modeled over domain subsets rather than globally.
- We propose a routing-based MoE framework that learns expert-specific alignment strengths across class-domain pairs and composes the resulting representations for prediction.
- We demonstrate improved out-of-domain generalization on standard DG benchmarks and greater robustness as source-domain heterogeneity increases.

2 Related Work

Domain Generalization and Invariant Representation Learning. DG is commonly approached by learning representations that are invariant across domains, typically by enforcing the conditional distribution $P(Z|Y)$ to be consistent across environments (Muandet et al., 2013; Fang et al., 2013; Li et al., 2018b). This principle underlies a wide range of methods, including feature alignment approaches (Sun & Saenko, 2016; Peng et al., 2019; Li et al., 2018a) and invariant learning frameworks such as IRM and its variants (Arjovsky et al., 2019; Rame et al., 2022; Shi et al., 2022). Subsequent work has explored different mechanisms to approximate invariance, including data augmentation (Shankar et al., 2018; Zhou et al., 2021), episodic and meta-learning strategies (Balaji et al., 2018; Dou et al., 2019), and self-supervised learning Carlucci et al. (2019); Lv et al. (2022). More recent approaches refine invariance through causal reasoning (Mahajan et al., 2021; Lv et al., 2022; Yin et al., 2024; He et al., 2025; Li et al., 2025a), disentangled representations (Zhang et al., 2022; Cheng et al., 2024; Pan et al., 2025), and indirect alignment objectives (Wei et al., 2025). Despite their differences, these approaches largely share the assumption that a single representation can be made invariant across all domains. In contrast, we argue that invariant structure is often only partially shared, and should be modeled at the level of domain subsets.

Optimization and Structural Constraints. A complementary line of work studies DG from an optimization perspective, emphasizing that generalization depends not only on representation invariance but also on the solutions reached during training. Methods in this direction improve robustness through gradient regularization (Rame et al., 2022; Shi et al., 2022; Nguyen et al., 2025;

Do et al., 2025), geometry-aware optimization, and flatness constraints (Cha et al., 2021; Wang et al., 2023; Mansilla et al., 2021). More recent approaches explicitly shape training trajectories (Ballas & Diou, 2025; Cho et al., 2025b; Di Zhao et al., 2026) or constrain interference with pretrained knowledge (Hu et al., 2024; Yun et al., 2025; Javed et al., 2025; WU, 2025). While these methods improve generalization through optimization dynamics, they do not explicitly address how invariant structure should be distributed across domains. Our work is complementary, focusing on the structure of invariance itself rather than the optimization process.

Modular Experts and Parameter-Efficient Adaptation. Recent DG work has moved toward modular architectures instead of a single shared model. Expert-based methods, such as sparse experts (Li et al., 2023; Chen et al., 2026) and multi-expert distillation (Chen et al., 2024), use specialized modules to capture transferable patterns. Parameter-efficient approaches such as prompt-based Wen et al. (2025); Li et al. (2026; 2024), adapter-based (Ji et al., 2025), low-rank (Li et al., 2025b), multimodal (Xu et al., 2026), and regularization-based (Cha et al., 2022; Wei et al., 2024; Cho et al., 2025a; Ni et al., 2024) are focusing on targeted adaptation. While these methods improve robustness through specialization, they rarely model predictive factors shared only by subsets of domains or use routing for invariance. Recent MoE-based work, such as FlexMoE (Yun et al., 2024), addresses subset-shared structure in multimodal settings with predefined indices, but requires explicit subset labels that are usually unavailable in DG.

Difference from prior work. Unlike prior expert-based or parameter-efficient DG methods, which primarily exploit specialization or constrained adaptation, our approach by contrast, models which domains should be aligned. We introduce *subset-shared invariance* and use routing-conditioned expert alignment, so that experts capture predictive factors stable within only certain domain subsets. This avoids forcing all information into a single invariant representation. Unlike FlexMoE, which require predefined subset structures, our method discovers subset structure directly from data. The routing mechanism adaptively determines which domains to align, enabling data-driven decomposition of invariant structure without explicit labels for subset membership.

3 Rethinking Feature Invariance in Domain Generalization

We revisit feature invariance in DG and show that enforcing conditional invariance across more domains can progressively restrict the class of feasible representations, potentially removing predictive structure shared only across subsets of domains.

3.1 Problem Setup

We consider the multi-source DG setting with K source domains and no access to target-domain data during training. Formally, let $X \in \mathcal{X}$ denote the input, $Y \in \mathcal{Y}$ the label, and $D \in \{1, \dots, K\}$ the domain index, where $\{\mathcal{D}_k\}_{k=1}^{K}$ are the source-domain distributions over (X, Y).

A predictor is parameterized as $\hat{y} = f_\theta(z)$ with representation $z = g_\phi(x)$, where $g_\phi : \mathcal{X} \to \mathbb{R}^d$ is a feature extractor and f_θ is a classifier. Training minimizes the empirical risk across source domains (Muandet et al., 2013; Gulrajani & Lopez-Paz, 2021; Vapnik, 1991):

$$\mathcal{L}_{\text{cls}} = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{(x,y)\sim\mathcal{D}_k} \big[\ell(f_\theta(g_\phi(x)), y)\big]. \tag{1}$$

3.2 Monotonic Shrinkage under Global Conditional Invariance

We characterize exact class-conditional invariance over a source-domain set. For $\mathcal{K} \subseteq 1, \dots, K$, let $\mathcal{P}(\mathcal{K}) = (i, j) : i, j \in \mathcal{K}, , i < j$. We evaluate predictive information under a fixed source-only reference law Q with $Q(D = k) = 1/K$ and $Q(X, Y \mid D = k) = \mathcal{D}_k(X, Y)$. For each class c, comparisons are restricted to $\mathcal{K}_c = k : \mathcal{D}_k(Y = c) > 0$.

Proposition 3.1. *For any* $(i, j) \in \mathcal{P}(\mathcal{K})$,

$$I(Z, D \,|\, Y, D \in \{i, j\}) = 0 \tag{2}$$

if and only if

$$P(Z|Y, D = i) = P(Z|Y, D = j). \tag{3}$$

Theorem 3.2. *Suppose that*

$$I(Z, D \mid Y, D \in \{i, j\}) = 0, \qquad \forall (i, j) \in \mathcal{P}(\mathcal{K}). \tag{4}$$

Then

$$P(Z|Y, D = i) = P(Z|Y, D = j), \qquad \forall i, j \in \mathcal{K}, \tag{5}$$

and therefore $Z \perp D|Y$ over $D \in \mathcal{K}$.

We next define the feasible representation class satisfying these constraints as

$$\mathcal{F}_{\mathcal{K}} = \{\phi \in \Phi : I(g_\phi(X), D \mid Y, D \in \{i, j\}) = 0,\ \forall (i, j) \in \mathcal{P}(\mathcal{K})\}. \tag{6}$$

We measure the predictive capacity of this feasible class under the same reference distribution Q as

$$V_{\mathcal{K}}^{Q} = \sup_{\phi \in \mathcal{F}_{\mathcal{K}}} I_Q(g_\phi(X), Y). \tag{7}$$

Theorem 3.3. *If $\mathcal{K} \subseteq \mathcal{K}'$, then*

$$\mathcal{F}_{\mathcal{K}'} \subseteq \mathcal{F}_{\mathcal{K}}, \tag{8}$$

and consequently

$$V_{\mathcal{K}'}^{Q} \leq V_{\mathcal{K}}^{Q}. \tag{9}$$

Adding invariance constraints cannot enlarge the feasible representation class. The inclusion need not be strict: an informative globally invariant representation may remain feasible. This result characterizes exact invariance constraints, rather than finite-sample target accuracy, the corresponding empirical study is reported in Appendix D.3. This exact-constraint result is independent of whether a finite-weight empirical penalty is summed or averaged.

3.3 Class-Conditional Subset-Shared Invariance

To connect the theory to the method, let $Z^{(m)} = h_m(b_\phi(X))$, let $\pi(X)$ be the input-dependent routing distribution, and let $Z_\pi = \sum_{m=1}^{M} \pi_m(X) Z^{(m)}$ be the representation used for prediction. Define $\mu_{k,c}^{(m)} = \mathcal{L}_Q(Z^{(m)} \mid Y = c, D = k)$ as the class-domain feature law.

Definition 3.4 (Class-conditional subset-shared invariance)**.** *For expert m and class c, let $\mathcal{K}_{m,c} \subseteq \mathcal{K}_c$ be a domain support. Expert m is invariant on this support when*

$$\mu_{i,c}^{(m)} = \mu_{j,c}^{(m)}, \qquad \forall i, j \in \mathcal{K}_{m,c}. \tag{10}$$

Supports may overlap and vary across classes, only supports containing at least two domains impose cross-domain equality.

Lemma 3.5 (Conditional-independence characterization)**.** *For $|\mathcal{K}_{m,c}| \geq 2$, equation 10 holds if and only if*

$$I_Q\Big(Z^{(m)}, D \mid Y = c,\ D \in \mathcal{K}_{m,c}\Big) = 0. \tag{11}$$

This definition selects domain relationships rather than individual samples and matches the full-cell distributions aligned by the method. For $M = 1$ with complete supports it recovers global conditional invariance, for $M > 1$, expert-level and composed-representation invariance are distinct because routing is input-dependent.

Let $\mathcal{E} = \{\mathcal{E}_{m,c}\}_{m,c}$ denote the fixed unordered domain-pair sets induced by the supports, and let Θ collect the encoder, expert, and router parameters in a fixed model class containing constant expert maps. Define

$$\mathcal{F}_{\mathcal{E}} = \Big\{\Theta : \mu_{i,c}^{(m)} = \mu_{j,c}^{(m)} \text{ for all } (i, j) \in \mathcal{E}_{m,c},\ m, c\Big\}, \tag{12}$$

$$V_{\mathcal{E}}^{Q} = \sup_{\Theta \in \mathcal{F}_{\mathcal{E}}} I_Q(Z_\pi, Y). \tag{13}$$

Corollary 3.6 (Alignment-support monotonicity). *If $\mathcal{E}_{m,c} \subseteq \mathcal{E}'_{m,c}$ for every expert and class, then $\mathcal{F}_{\mathcal{E}'} \subseteq \mathcal{F}_{\mathcal{E}}$ and $V^Q_{\mathcal{E}'} \leq V^Q_{\mathcal{E}}$. The conclusion also holds when the same router is fixed in both families.*

Holding the source law, model class, and evaluated representation fixed, the structural choice is which distributional equalities are imposed. Since equality propagates along connected domain pairs within an expert, edge sparsity alone does not define distinct invariance supports.

Corollary 3.6 shows that relaxing alignment constraints cannot reduce predictive capacity. However, this does not imply a strict gain. The following proposition shows that subset-level alignment can preserve strictly more label information under the same model class and fixed router.

Proposition 3.7 (Strict separation under fixed routing). *Let $\mathcal{E}_{\mathrm{all}}$ denote complete expert alignment across all source domains and $\mathcal{E}_{\mathrm{sub}}$ denote subset-supported alignment. For a fixed router π, let $V^Q_{\mathcal{E}}(\pi)$ be the predictive capacity under alignment structure $\mathcal{E}$. There exist source distributions, a two-expert model class, and an input-dependent router π such that*

$$V^Q_{\mathcal{E}_{\mathrm{all}}}(\pi) = 0 < V^Q_{\mathcal{E}_{\mathrm{sub}}}(\pi) = I_Q(X, Y). \tag{14}$$

Moreover, linear experts sharing a Bayes-optimal classifier attain the subset-supported optimum.

Thus, subset-level alignment can preserve predictive information excluded by complete alignment, the proof is given in Appendix A.7. Consistently, the shared classifier supervises the composed representation Z_π, whose finite expected cross-entropy satisfies

$$I_Q(Z_\pi, Y) \geq H_Q(Y) - \mathcal{L}_{\mathrm{cls}}. \tag{15}$$

Hence, reducing classification loss increases a lower bound on the label information in Z_π, while expert-level alignment controls the class-conditional domain relationships.

Proposition 3.7 gives a strict separation at fixed routing, router learning is governed by the objective below. For shared-classifier probabilities $q_\theta(\cdot \mid Z_\pi)$, finite expected cross-entropy satisfies

$$\mathcal{L}_{\mathrm{cls}} = H_Q(Y) - I_Q(Z_\pi, Y) + \mathbb{E}_{Z_\pi} \mathrm{KL}(Q(Y \mid Z_\pi) \,\|\, q_\theta(Y \mid Z_\pi)). \tag{16}$$

Thus $H_Q(Y) - \mathcal{L}_{\mathrm{cls}}$ lower-bounds the predictive information in Z_π, with equality for a classifier realizing the conditional label law. Classification supervises prediction, while expert-level alignment controls class-conditional domain agreement.

4 Proposed Method

To model class-conditional domain relationships of different strengths, we use a routing-based MoE with expert-specific alignment and input-dependent prediction (Figure 2). Routing responsibilities determine alignment strength. Given an input x, a shared encoder produces $u = b_\phi(x)$, the router outputs a distribution over experts $\pi(x) = \mathrm{softmax}(g_\psi(u))$, and each expert produces an expert-specific representation $z^{(m)} = h_m(u)$. The final representation is formed by aggregating expert outputs according to the routing distribution $z(x) = \sum_{m=1}^{M} \pi_m(x) z^{(m)}$. The router receives neither class nor domain labels, they are used only to form training-time alignment statistics.

This design introduces three challenges: (1) assigning expert- and class-specific alignment strengths, (2) discouraging redundant experts, and (3) encouraging confident routing with balanced utilization. To address these challenges, we optimize the following objective:

$$\mathcal{L} = \mathcal{L}_{\texttt{cls}} + \lambda_{\texttt{ssi}} \mathcal{L}^{\mathrm{OT}}_{\texttt{ssi}} + \lambda_{\mathrm{sp}} \mathcal{L}_{\texttt{sp}} + \lambda_{\mathrm{bal}} \mathcal{L}_{\texttt{bal}} + \lambda_{\mathrm{div}} \mathcal{L}_{\texttt{div}}. \tag{17}$$

Here, $\mathcal{L}_{\mathrm{cls}}$ denotes the classification loss, $\mathcal{L}^{\mathrm{OT}}_{\texttt{ssi}}$ weights expert discrepancies by routing responsibility, $\mathcal{L}_{\texttt{sp}}$ encourages confident routing, $\mathcal{L}_{\texttt{bal}}$ balances expert utilization, and $\mathcal{L}_{\texttt{div}}$ encourages diverse expert representations. We describe each component below. Architectural details are provided in Appendix B.1. Full optimization details and hyperparameters are provided in Appendices F.2 and F.3.

4.1 Routing-Conditioned Subset Alignment

We connect instance-level routing to domain-level alignment through $\rho^{(m)}_{k,c} = \mathbb{E}_Q[\pi_m(X) \mid D = k, Y = c]$, estimated within each non-empty class-domain cell as

$$\widehat{\rho}^{(m)}_{k,c} = \frac{1}{|\mathcal{D}_{k,c}|} \sum_{(x,y) \in \mathcal{D}_k,\, y=c} \pi_m(x). \tag{18}$$

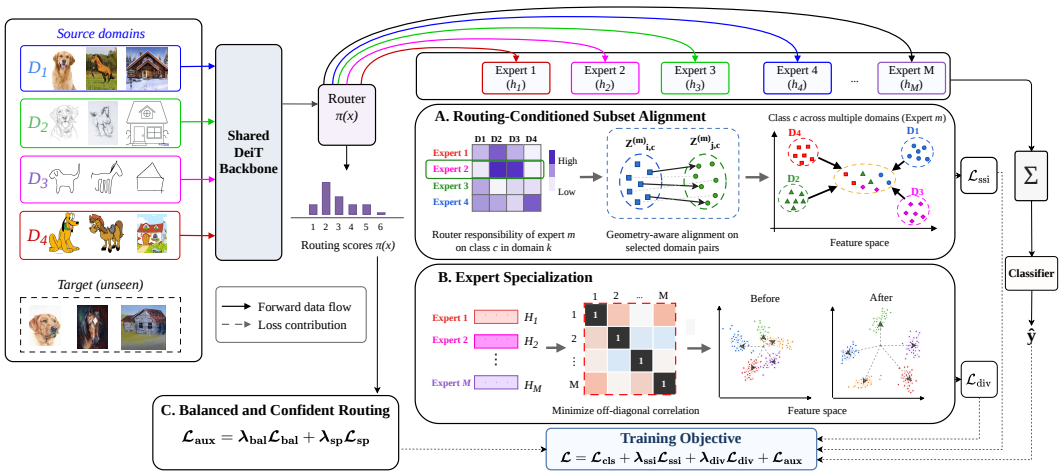


Figure 2: Overview of MESSI. An input is encoded by a shared backbone and routed to multiple experts, whose outputs are composed into the final representation for prediction. The training objective combines classification loss with routing-weighted class-conditional alignment, confident and balanced routing, and expert specialization terms.

The responsibilities form an expert- and class-specific domain affinity profile. Thresholding at τ gives $\mathcal{K}_{m,c}^{\tau} = \{k \in \mathcal{K}_c : \rho_{k,c}^{(m)} > \tau\}$, corresponding to the support in Definition 3.4, training instead uses continuous pair weights. Expert indices are exchangeable, so only the induced alignment structure matters. The empirical pair weight is

$$\widehat{a}_{ijc}^{(m)} = \sigma\left(\alpha\widehat{\rho}_{i,c}^{(m)}\right)\sigma\left(\alpha\widehat{\rho}_{j,c}^{(m)}\right), \qquad \alpha > 0. \tag{19}$$

Pairs receive larger weights when expert m has high responsibility in both domains, yielding soft alignment strengths rather than hard pair selection.

For each non-empty cell, let $\widehat{\mu}_{k,c}^{(m)} = |\mathcal{D}_{k,c}|^{-1}\sum_{(x,y)\in\mathcal{D}_{k,c}}\delta_{h_m(b_\phi(x))}$ be its empirical feature law, and let $\mathcal{I}_B$ contain valid minibatch tuples (m, i, j, c) with $i < j$. Our primary discrepancy is entropic optimal transport:

$$\mathcal{L}_{\mathtt{ssi}}^{\mathrm{OT}} = \sum_{(m,i,j,c)\in\mathcal{I}_B} \mathrm{sg}\left(\widehat{a}_{ijc}^{(m)}\right) W_\varepsilon\left(\widehat{\mu}_{i,c}^{(m)}, \widehat{\mu}_{j,c}^{(m)}\right), \tag{20}$$

$$W_\varepsilon(P,Q) = \min_{\gamma\in\Pi(P,Q)} \mathbb{E}\|z - \tilde{z}\|_2^2 + \varepsilon\,\mathrm{KL}(\gamma \,\|\, P\otimes Q).$$

Here sg stops gradients through the pair weights, preventing the router from reducing alignment loss by lowering its responsibilities. Alignment updates the experts and shared encoder, while classification and routing regularizers train the router, shared-encoder updates may still affect routing indirectly.

The theory targets distribution equality, entropic OT is an upper surrogate for squared Wasserstein discrepancy, not a mutual-information estimator. For empirical measures with n, n' atoms,

$$0 \le W_\varepsilon(P,Q) - W_2^2(P,Q) \le \varepsilon \log\min\{n, n'\}. \tag{21}$$

Thus $\mathcal{L}_{\mathtt{ssi}}^{\mathrm{OT}}$ upper-bounds the corresponding routing-weighted squared Wasserstein discrepancies up to the entropic excess. Since W_2^2 vanishes at distribution equality whereas the regularized cost may not, exact invariance and finite-weight regularization remain distinct, see Appendix A.

Replacing OT with MMD preserves the routing statistics and pair weights while changing the discrepancy, separately trained variants may learn different responsibilities. With a characteristic kernel, population MMD vanishes exactly at distribution equality. Empirically, both outperform same-backbone ERM, with OT achieving higher accuracy at greater training cost (Tables 1 and 5).

4.2 Expert Specialization

A final failure mode is redundancy across experts: even with well-behaved routing, different experts may still converge to similar outputs. To discourage this, let $H_m \in \mathbb{R}^{B\times r}$ be the batch output matrix of expert m, and let $\tilde{H}_m = H_m/(\|H_m\|_F + \varepsilon)$ be its normalized version. Specifically, we minimize

$$\mathcal{L}_{\texttt{div}} = \sum_{m\neq n} \left\| \frac{1}{B} \tilde{H}_m^\top \tilde{H}_n \right\|_F^2 . \tag{22}$$

This normalized cross-moment penalty discourages cross-expert feature dependence, while classification supervises the composed prediction. Recent works have explored enforcing orthogonality among experts to promote functional diversity (Hu et al., 2024; Hendawy et al., 2024; Oldfield et al., 2024; Feng et al., 2025; Guo et al., 2025a). MOORE (Hendawy et al., 2024) and OMOE (Feng et al., 2025) use explicit orthogonalization, while (Guo et al., 2025a) uses regularization. We retain a normalized penalty to encourage diversity without directly projecting the representations used for prediction.

4.3 Balanced and Confident Routing

Routing confidence and aggregate utilization are distinct: diffuse routing activates many experts per input, whereas concentrated aggregate load collapses usage to a few experts. We address these two failure modes by regularizing routing at both the instance and dataset levels:

$$\mathcal{L}_{\texttt{sp}} = -\mathbb{E}_x \left[\sum_{m=1}^{M} \pi_m(x) \log \pi_m(x) \right], \quad \mathcal{L}_{\texttt{bal}} = \sum_{m=1}^{M} \left(\mathbb{E}_x[\pi_m(x)] - \frac{1}{M} \right)^2 . \tag{23}$$

The two objectives regularize routing at complementary scales. The sparsity term $\mathcal{L}_{\texttt{sp}}$ acts locally on each routing distribution $\pi(x)$, encouraging low-entropy expert assignment for individual samples. The balancing term $\mathcal{L}_{\texttt{bal}}$ instead operates on the aggregated routing statistics $\mathbb{E}_x[\pi_m(x)]$, preventing marginal load from concentrating on a few experts without enforcing uniform routing across classes or domains.

5 Experiments

Our experiments are organized around three claims. $\mathcal{C}_1$: Subset-conditioned expert alignment improves standard DG performance. $\mathcal{C}_2$: MESSI is more robust than globally aligned methods as the number of source domains increases. $\mathcal{C}_3$: The gains arise from routing-induced subset-conditioned alignment rather than additional MoE capacity, global alignment, or random sparse alignment.

Table 1: OOD accuracy (%) under the DomainBed training-domain validation criterion, averaged over three seeds. **MESSI-Ti** and **MESSI-S** use DeiT-Ti/16 and DeiT-S/16 backbones, respectively. Best results are in **bold**, second-best results are underlined.

Method	Backbone	PACS ↑	OfficeHome ↑	TerraInc ↑	DomainNet ↑	Avg. ↑
ERM (Vapnik, 1991)	ResNet-50	83.8±0.8	66.6±0.4	47.2±0.8	41.6±0.2	59.8
CORAL (Sun & Saenko, 2016)	ResNet-50	83.5±0.4	66.0±0.5	43.8±0.9	39.2±0.3	58.1
Fishr (Rame et al., 2022)	ResNet-50	85.5±0.2	68.6±0.2	47.4±1.6	41.7±0.3	60.8
SAGM (Wang et al., 2023)	ResNet-50	86.4±1.2	69.4±0.2	48.8±1.0	43.2±0.4	61.9
LFME (Chen et al., 2024)	ResNet-50	84.9±0.4	68.5±0.2	**49.5±0.8**	38.6±0.2	60.4
ERM (Vapnik, 1991)	DeiT-S/16	86.2±0.1	72.2±0.4	42.0±0.8	47.3±0.2	61.9
DynMoE (Guo et al., 2025b)	DeiT-S/16	85.2±0.4	73.4±0.3	44.5±0.6	45.9±0.4	62.3
GMoE (Li et al., 2023)	DeiT-S/16	87.3±0.1	73.5±0.1	48.2±0.5	47.8±0.5	64.2
OMoE (Feng et al., 2025)	DeiT-S/16	87.1±0.3	73.2±0.1	46.0±0.5	46.1±0.2	63.1
MESSI-Ti-OT (Ours)	DeiT-Ti/16	86.0±0.3	69.6±0.2	42.5±1.1	43.3±0.4	60.4
MESSI-S-MMD (Ours)	DeiT-S/16	89.3±0.6	74.0±0.3	47.5±0.4	46.2±0.4	64.3
MESSI-S-OT (Ours)	DeiT-S/16	**90.9±0.7**	**76.1±0.2**	49.3±0.2	**48.6±0.3**	**66.2**

5.1 Experimental Setup

Benchmarks. We evaluate standard OOD generalization on DomainBed (Gulrajani & Lopez-Paz, 2021), covering PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), TerraIncognita (Beery et al., 2018), and DomainNet (Peng et al., 2019), and domain growth on Rotated-Colored MNIST, a controlled variant inspired by Rotated MNIST and Colored MNIST (Ghifary et al., 2015; Arjovsky et al., 2019) (Appendix C.2). We follow DomainBed's leave-one-domain-out protocol (Appendix C.1), while Rotated-Colored MNIST lets us vary source-domain number and heterogeneity in a controlled setting, serving as a diagnostic stress test for the proposed failure mode. Appendices C.2 and C.3 provide details on Rotated-Colored MNIST and the fixed-budget domain-growth protocol.

MESSI uses a pretrained DeiT (Touvron et al., 2021) encoder with an $M{=}6$ expert MoE head; each expert is a two-layer MLP, and the router computes soft weights from the final CLS feature. We evaluate DeiT-Ti/16 and DeiT-S/16 using the same input resolution, augmentations, optimization schedule, and model-selection criterion as the baselines. MESSI-specific hyperparameters are fixed across datasets without dataset-specific tuning. Appendix E.2 reports sensitivity to expert count and routing regularization weights. All experiments use NVIDIA H100 GPUs. Appendix F provides the baselines and parameter counts.

5.2 DomainBed Results

Table 1 evaluates $\mathcal{C}_1$ by reporting OOD accuracy on DomainBed benchmarks. Reported baselines provide broader DG context, while *MESSI-S* enables a same-backbone comparison to transformer-based methods. Additional backbone comparisons are presented in Appendix D.1. The same-backbone comparison in Table 1 and the model-size analysis in Appendix F.6 rule out the simple explanation that MESSI improves by using a larger MoE model. MESSI-S uses fewer method-specific parameters and lower inference cost than GMoE/O-MoE, while achieving higher average OOD accuracy. However, parameter count alone does not isolate the training mechanism. We therefore further compare against same-architecture controls in Appendix E.1, including a classification-only MoE and variants that remove subset-conditioned alignment, routing sparsity, load balancing, or diversity. These controls test whether the gain remains after model capacity, backbone, routing module, and training budget are fixed. Per-target-domain accuracies for PACS, OfficeHome, TerraIncognita, and DomainNet are reported in Appendix D.2.As a supporting real-world many-domain evaluation, we report a controlled WILDS-iWildCam comparison in Appendix D.4. This experiment uses the same DeiT-S/16 backbone and preprocessing across MoE methods and is not intended as a leaderboard-scale WILDS evaluation.

Table 2: Fixed-budget domain-growth results. Peak is the best test accuracy across source-domain budgets, and $\mathrm{Drop} = \mathrm{Peak} - \mathrm{Acc}(K_{\max})$. Best results are shown in **bold**. Full curves and setup details are provided in Appendix D.3.

Method	Peak ↑	$\mathrm{Acc}(K_{\max})$ ↑	Drop ↓	Avg ↑
CORAL (γ=1)	71.96	69.38	2.58	65.38
CORAL (γ=10)	71.64	68.77	2.87	65.20
CORAL (γ=100)	70.10	66.46	3.64	63.88
CORAL (γ=250)	68.75	63.67	5.08	62.36
CORAL (γ=500)	67.69	59.78	7.91	60.36
MESSI (Ours)	**75.31**	**73.45**	**1.86**	**69.32**

5.3 In-depth Analysis

Routing-conditioned subset alignment explains the gain. Table 4 evaluates $\mathcal{C}_3$ using same-architecture controls on PACS. Both global and random sparse alignment reduce accuracy relative to ERM-MoE, showing that simply adding alignment to an MoE architecture is insufficient. Domain-only routing also underperforms, indicating that routing mass alone is too coarse without class-domain responsibility. In contrast, the routing-conditioned class-domain selector avoids the degradation observed with global and random alignment, while the full objective achieves the best performance. These results support $\mathcal{C}_3$: the improvement depends

Table 3: $\mathcal{C}_3$ **diagnostic on PACS.**

Metric	Control	MESSI
Resp. corr. r	$+1.00$	-0.20
Slot overlap J	0.95	0.04
Discrepancy ↓	0.234 ± 0.020	$\mathbf{0.096 \pm 0.014}$

on where alignment is applied, whereas sparse and balanced routing make the subset decomposition more effective.

Table 4: Mechanism ablation under the same MoE architecture. The first group isolates the alignment selector. The last two rows separate routing-conditioned class-domain alignment from routing regularization. Best results are shown in **bold**.

Variant	Alignment selector	Class-aware	Routing-cond.	PACS ↑
ERM-MoE	none	–	–	89.1±0.3
Global-MoE	all class-domain slots	✓	–	87.1±0.2
Random-Subset-MoE	matched random slots	✓	–	87.0±0.1
Domain-Only Routing	routing domain mass	–	✓	88.2±0.2
MESSI w/o $\mathcal{L}_{\texttt{sp}}, \mathcal{L}_{\texttt{bal}}$	routing class-domain mass	✓	✓	89.0±0.2
MESSI (Ours)	routing class-domain mass	✓	✓	**90.9±0.7**

MESSI changes the alignment structure. We test whether MESSI learns a distinct subset structure rather than merely reweighting global alignment. Treating each class-domain-expert tuple (m, i, j, c) as an alignment slot, we compare selectors using Pearson correlation r between slot weights, Jaccard overlap J between selected slots, and class-conditional discrepancy on the MESSI-selected set S_{M}. Table 3 confirms that routing changes where class-conditional alignment is applied, rather than merely sparsifying a global alignment pattern.

Ablation test. Objective ablations are reported in Appendix E.1. Additional diagnostics on expert specialization, routing behavior, and routing-aware alignment controls are provided in Appendices E.3, E.4, and E.5. Appendix E.6 studies the pairwise-to-global invariance trade-off by sweeping the invariance weight and measuring pairwise discrepancy, domain predictability, and target accuracy.

Training Cost and Inference Efficiency Table 5 shows that MESSI introduces training-time overhead mainly through the subset-conditioned discrepancy. MESSI-S-MMD costs $4.24\times$ ERM and is close to GMoE+SAGM, while MESSI-S-OT costs $12.39\times$ ERM. The gap between the two MESSI-S variants isolates the cost of the entropic OT solver. MMD gives a non-iterative discrepancy, whereas OT runs Sinkhorn iterations over selected class-conditional domain pairs.

Table 5: Training and inference cost on PACS, measured on NVIDIA H100 GPUs. Train time is measured in wall-clock seconds per 1K optimization steps after a 100-step warm-up. Relative time is normalized by ERM. Inference is measured with all training-only losses disabled.

Method	Backbone	Train	Rel.	Infer.
ERM	ResNet-50	103.78±0.50	1.00×	32.42±0.01
SAGM	ResNet-50	211.37±1.26	2.04×	32.44±0.01
GMoE	DeiT-S/16	209.71±0.49	2.02×	74.22±0.11
GMoE+SAGM	DeiT-S/16	429.88±15.86	4.14×	74.17±0.08
MESSI-Ti-OT	DeiT-Ti/16	1200.86±22.38	11.57×	**23.50±0.01**
MESSI-S-MMD	DeiT-S/16	439.68±5.35	4.24×	61.40±0.01
MESSI-S-OT	DeiT-S/16	1286.26±28.57	12.39×	61.41±0.01

6 Conclusion

We proposed MESSI, a domain generalization framework based on subset-shared invariance. Our analysis shows that enforcing invariance across more domain relationships can restrict the feasible predictive representation and exclude factors shared only within domain subsets. MESSI addresses this with a routing-conditioned MoE, where routing determines expert-specific alignment strengths over class-domain pairs and the resulting expert representations are composed for prediction. Experiments on DomainBed and controlled source-domain expansion support modeling invariance at the subset level rather than always enforcing a single global invariant space.

Limitations. When source domains already admit a strong globally invariant representation, the additional routing decomposition in MESSI may introduce unnecessary complexity and reduce statistical efficiency relative to simpler global alignment methods.

AI USE STATEMENT

AI tools were used only to assist with language editing and improve the clarity of the manuscript. All scientific content, analyses, results, and conclusions were produced and verified by the authors.

ETHICS STATEMENT

This work does not involve new human-subject data collection or deployment decisions. We use existing research datasets and follow their established research-use protocols.

REPRODUCIBILITY STATEMENT

REFERENCES

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

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

Aristotelis Ballas and Christos Diou. Gradient-guided annealing for domain generalization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20558–20568, 2025.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Sara Beery, Arushi Agarwal, Elijah Cole, and Vighnesh Birodkar. The iwildcam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*, 2021.

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2229–2238, 2019.

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

Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. In *European conference on computer vision*, pp. 440–457, 2022.

Liang Chen, Yong Zhang, Yibing Song, Zhiqiang Shen, and Lingqiao Liu. Lfme: A simple framework for learning from multiple experts in domain generalization. *Advances in Neural Information Processing Systems*, 37:102919–102947, 2024.

Xuweiyi Chen, Wentao Zhou, Aruni RoyChowdhury, and Zezhou Cheng. Point-moe: Large-scale multi-dataset training with mixture-of-experts for 3d semantic segmentation. In *The Fourteenth International Conference on Learning Representations*, 2026.

De Cheng, Zhipeng Xu, Xinyang Jiang, Nannan Wang, Dongsheng Li, and Xinbo Gao. Disentangled prompt representation for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23595–23604, 2024.

Dong Kyu Cho, Inwoo Hwang, and Sanghack Lee. Peer pressure: Model-to-model regularization for single source domain generalization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15360–15370, 2025a.

Sumin Cho, Dongwon Kim, and Kwangsu Kim. One-step generalization ratio guided optimization for domain generalization. In *Forty-second International Conference on Machine Learning*, 2025b.

Yongxing Dai, Xiaotong Li, Jun Liu, Zekun Tong, and Ling-Yu Duan. Generalizable person re-identification with relevance-aware mixture of experts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16145–16154, 2021.

Jingfeng Zhang Di Zhao, Hongsheng Hu, Philippe Fournier-Viger, Gillian Dobbie, and Yun Sing Koh. Unlearning during training: Domain-specific gradient ascent for domain generalization. In *The Fourteenth International Conference on Learning Representations*, 2026.

Khoi Do, Nam-Khanh Le, Quoc-Viet Pham, Binh-Son Hua, Won-Joo Hwang, and Duong Nguyen. Domain generalization via pareto optimal gradient matching. In *28th European Conference on Artificial Intelligence, ECAI 2025*. IOS Press BV, 2025.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in neural information processing systems*, 32, 2019.

Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE international conference on computer vision*, pp. 1657–1664, 2013.

Jinyuan Feng, Zhiqiang Pu, Tianyi Hu, Dongmin Li, Xiaolin Ai, and Huimu Wang. Omoe: Diversifying mixture of low-rank adaptation by orthogonal finetuning. In *European Conference on Artificial Intelligence*, 2025.

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

Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. *Advances in Neural Information Processing Systems*, 34:23885–23899, 2021.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

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

Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Xinye Cao, Sicong Leng, Qimei Cui, et al. Advancing expert specialization for better moe. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. In *The Thirteenth International Conference on Learning Representations*, 2025b.

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

Zhuo He, Shuang Li, Wenze Song, Longhui Yuan, Jian Liang, Han Li, and Kun Gai. Learning time-aware causal representation for model generalization in evolving domains. In *Forty-second International Conference on Machine Learning*, 2025.

Ahmed Hendawy, Jan Peters, and Carlo D'Eramo. Multi-task reinforcement learning with mixture of orthogonal experts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=aZH1dM3GOX.

Jiajun Hu, Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Learn to preserve and diversify: Parameter-efficient group with orthogonal regularization for domain generalization. In *European Conference on Computer Vision*, pp. 198–216, 2024.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Saqib Javed, Hieu Le, and Mathieu Salzmann. Qt-dog: Quantization-aware training for domain generalization. In *Forty-second International Conference on Machine Learning*, 2025.

Yuyang Ji, Zeyi Huang, Haohan Wang, and Yong Jae Lee. Customizing domain adapters for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 934–944, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021.

Bo Li, Yifei Shen, Jingkang Yang, Yezhen Wang, Jiawei Ren, Tong Che, Jun Zhang, and Ziwei Liu. Sparse mixture-of-experts are domain generalizable learners. In *The Eleventh International Conference on Learning Representations*, 2023.

Chen Li, Huiying Xu, Changxin Gao, Zeyu Wang, Yun Liu, and Xinzhong Zhu. Towards single-source domain generalized object detection via causal visual prompts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Deng Li, Aming Wu, Yaowei Wang, and Yahong Han. Prompt-driven dynamic object-centric learning for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17606–17615, 2024.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018a.

Tao Li, Zhengbao He, Yujun Li, Yasheng Wang, Lifeng Shang, and Xiaolin Huang. Flat-lora: Low-rank adaptation over a flat loss landscape. In *International Conference on Machine Learning*, pp. 34549–34563. PMLR, 2025b.

Xinyao Li, Yinjie Min, Hongbo Chen, Zhekai Du, Fengling Li, and Jingjing Li. Generalizing vision-language models with dedicated prompt guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23239–23247, 2026.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 624–639, 2018b.

Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8046–8056, 2022.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International conference on machine learning*, pp. 7313–7324. PMLR, 2021.

Lucas Mansilla, Rodrigo Echeveste, Diego H Milone, and Enzo Ferrante. Domain generalization via gradient surgery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6630–6638, 2021.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pp. 10–18. PMLR, 2013.

Trong Binh Nguyen, Duong Minh Nguyen, Jinsun Park, Viet Quoc Pham, and Won-Joo Hwang. Federated domain generalization with data-free on-server matching gradient. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yao Ni, Shan Zhang, and Piotr Koniusz. Pace: Marrying generalization in parameter-efficient fine-tuning with consistency regularization. *Advances in Neural Information Processing Systems*, 37: 61238–61266, 2024.

James Oldfield, Markos Georgopoulos, Grigorios Chrysos, Christos Tzelepis, Yannis Panagakis, Mihalis Nicolaou, Jiankang Deng, and Ioannis Patras. Multilinear mixture of experts: Scalable expert specialization through factorization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=bIa03mAtxQ.

Tan Pan, Kaiyu Guo, Dongli Xu, Zhaorui Tan, Chen Jiang, Deshu Chen, Xin Guo, Brian C Lovell, LIMEI HAN, Yuan Cheng, et al. Minimal semantic sufficiency meets unsupervised domain generalization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International conference on machine learning*, pp. 7728–7738. PMLR, 2020.

Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pp. 18347–18377. PMLR, 2022.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2018.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *10th International Conference on Learning Representations, ICLR 2022*, pp. 1–28, 2022.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450, 2016.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

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

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3769–3778, 2023.

Yimu Wang, Yihan Wu, and Hongyang Zhang. Lost domain generalization is a natural consequence of lack of training domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15689–15697, 2024.

Wei Wei, Zixiong Li, Jing Yan, Mingwen Shao, and Lin Li. Indirect alignment and relationships preservation for domain generalization. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2054–2062, 2025.

Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. Stronger fewer & superior: Harnessing vision foundation models for domain generalized semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 28619–28630, 2024.

Changsong Wen, Zelin Peng, Yu Huang, Xiaokang Yang, and Wei Shen. Domain generalization in clip via learning with diverse text prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9559–9569, 2025.

Boqian WU. Dynamic sparse training versus dense training: The unexpected winner in image corruption robustness. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhipeng Xu, Zilong Wang, XINYANG JIANG, Dongsheng Li, De Cheng, and Nannan Wang. Reasoning-driven multimodal llm for domain generalization. In *The Fourteenth International Conference on Learning Representations*, 2026.

Huaxiu Yao, Xinyu Yang, Xinyi Pan, Shengchao Liu, Pang Wei Koh, and Chelsea Finn. Improving domain generalization with domain relations. In *The Twelfth International Conference on Learning Representations*, 2024.

Naiyu Yin, Hanjing Wang, Yue Yu, Tian Gao, Amit Dhurandhar, and Qiang Ji. Integrating markov blanket discovery into causal representation learning for domain generalization. In *European Conference on Computer Vision*, pp. 271–288, 2024.

Seokju Yun, Seunghye Chae, Dongheon Lee, and Youngmin Ro. Soma: Singular value decomposed minor components adaptation for domain generalizable representation learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25602–25612, 2025.

Sukwon Yun, Inyoung Choi, Jie Peng, Yangfan Wu, Jingxuan Bao, Qiyiwen Zhang, Jiayi Xin, Qi Long, and Tianlong Chen. Flex-moe: Modeling arbitrary modality combination via the flexible mixture-of-experts. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ihEHCbqZEx.

Hanlin Zhang, Yi-Fan Zhang, Weiyang Liu, Adrian Weller, Bernhard Schölkopf, and Eric P Xing. Towards principled disentanglement for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8024–8034, 2022.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021.

A PROOFS

A.1 PROBABILITY CONVENTIONS AND PAIRWISE OBJECTIVES

Unless stated otherwise, we use the fixed source-only law $Q(D = k) = 1/K$ and $Q(X, Y \mid D = k) = P_k(X, Y)$, where $P_k = \mathcal{D}_k$. All conditional laws below are induced by Q. We assume finite labels, measurable deterministic feature maps, and natural logarithms, and use the same Q when comparing alignment constraints. Proposition 3.7 assumes equal total mass for its two domain groups, satisfied by uniform weights when the groups have equal size.

For a source-domain set $\mathcal{K}$, we define the corresponding uniform source mixture over (X, Y, D) by

$$P_{\mathcal{K}}(D = k) = \frac{1}{|\mathcal{K}|}, \qquad P_{\mathcal{K}}(X, Y \mid D = k) = P_k(X, Y), \quad k \in \mathcal{K}. \tag{24}$$

Its marginal distribution over (X, Y) is therefore

$$P_{\mathcal{K}}(X, Y) = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} P_k(X, Y), \tag{25}$$

and we write

$$I_{\mathcal{K}}(Z, Y) = I_{P_{\mathcal{K}}}(g_\phi(X), Y). \tag{26}$$

A stylized pairwise information objective can then be written as

$$\mathcal{J}_{\mathcal{K}}(\phi) = I_{\mathcal{K}}(Z, Y) - \lambda \sum_{(i,j) \in \mathcal{P}(\mathcal{K})} I(Z, D \,|\, Y, D \in \{i, j\}) \,. \tag{27}$$

For $|\mathcal{K}| \geq 2$, an averaged alternative is

$$\bar{\mathcal{J}}_{\mathcal{K}}(\phi) = I_{\mathcal{K}}(Z, Y) - \frac{\lambda}{|\mathcal{P}(\mathcal{K})|} \sum_{(i,j) \in \mathcal{P}(\mathcal{K})} I(Z, D \,|\, Y, D \in \{i, j\}) \,. \tag{28}$$

For $|\mathcal{K}| = 1$, the pairwise penalty is zero.

These objectives are stylized motivations rather than equivalent reformulations of the empirical DG objective. In particular, changing $\mathcal{K}$ also changes $P_{\mathcal{K}}$ and hence $I_{\mathcal{K}}(Z, Y)$. Our monotonicity result therefore does not compare the finite-λ values of either objective. Instead, it concerns the representation classes satisfying the exact pairwise invariance constraints. Both summed and averaged penalties vanish exactly when all corresponding pairwise conditional mutual informations vanish.

A.2 PROOF OF PROPOSITION 3.1

Proof. All conditional quantities in this proof are evaluated under the two-domain distribution induced by conditioning on $D \in \{i, j\}$. Under this restricted distribution, D takes only two values. By the standard characterization of conditional mutual information,

$$I(Z, D \,|\, Y, D \in \{i, j\}) = 0 \tag{29}$$

if and only if

$$Z \perp D \mid Y, D \in \{i, j\}. \tag{30}$$

Therefore, for every label value y such that $P(Y = y \mid D = i) > 0$ and $P(Y = y \mid D = j) > 0$,

$$P(Z \mid Y = y, D = i) = P(Z \mid Y = y, D = j). \tag{31}$$

The converse follows directly from the same characterization of conditional independence. □

A.3 Proof of Theorem 3.2

Proof. By Proposition 3.1, for every $(i, j) \in \mathcal{P}(\mathcal{K})$,

$$P(Z \mid Y, D = i) = P(Z \mid Y, D = j). \tag{32}$$

Hence, all domains in $\mathcal{K}$ share the same conditional distribution of Z given Y. For a fixed label y, let $\mathcal{K}_y = \{k \in \mathcal{K} : P(Y = y, D = k) > 0\}$. By Proposition 3.1, all domains in $\mathcal{K}_y$ share the same conditional distribution of Z given $Y = y$. Denote this common distribution by ν_y.

$$P(Z \mid Y = y, D = i) = \nu_y, \qquad \forall i \in \mathcal{K}_y. \tag{33}$$

Conditioning on $D \in \mathcal{K}$ and marginalizing over the domains gives

$$P(Z \mid Y = y, D \in \mathcal{K}) = \sum_{i \in \mathcal{K}_y} P(Z \mid Y = y, D = i) P(D = i \mid Y = y, D \in \mathcal{K}) \tag{34}$$

$$= \nu_y. \tag{35}$$

Therefore, for every label y and every $i \in \mathcal{K}_y$,

$$P(Z \mid Y = y, D = i) = P(Z \mid Y = y, D \in \mathcal{K}), \tag{36}$$

which is precisely $Z \perp D \mid Y$ over $D \in \mathcal{K}$. □

A.4 Proof of Theorem 3.3

Proof. If $\mathcal{K} \subseteq \mathcal{K}'$, then

$$\mathcal{P}(\mathcal{K}) \subseteq \mathcal{P}(\mathcal{K}'). \tag{37}$$

Thus, every representation satisfying all pairwise invariance constraints for $\mathcal{K}'$ also satisfies those for $\mathcal{K}$, which implies

$$\mathcal{F}_{\mathcal{K}'} \subseteq \mathcal{F}_{\mathcal{K}}. \tag{38}$$

Since both feasible classes are evaluated under the same reference distribution Q, taking the supremum of $I_Q(g_\phi(X), Y)$ gives

$$V^Q_{\mathcal{K}'} = \sup_{\phi \in \mathcal{F}_{\mathcal{K}'}} I_Q(g_\phi(X), Y) \leq \sup_{\phi \in \mathcal{F}_{\mathcal{K}}} I_Q(g_\phi(X), Y) = V^Q_{\mathcal{K}}. \tag{39}$$

□

Remark A.1 (Scope of the monotonicity result)**.** *The reference distribution Q is fixed across domain sets and is used only to compare the predictive capacity of the nested feasible classes, it does not represent target-domain information available to the learning algorithm. The inclusion $\mathcal{F}_{\mathcal{K}'} \subseteq \mathcal{F}_{\mathcal{K}}$ may be an equality. Moreover, even when the inclusion is strict, the optimal predictive values may remain equal if a maximizer over $\mathcal{F}_{\mathcal{K}}$ also belongs to $\mathcal{F}_{\mathcal{K}'}$. Finally, the theorem concerns exact population constraints and does not assert that an optimizer of a finite-weight penalized objective satisfies exact invariance or that finite-sample target accuracy decreases monotonically as domains are added.*

A.5 Proof of Lemma 3.5

Proof. Fix expert m, class c, and $\mathcal{K}_{m,c} \subseteq \mathcal{K}_c$ with $|\mathcal{K}_{m,c}| \geq 2$, where $\mathcal{K}_c = \{k : P_k(Y = c) > 0\}$. Let $\mu^{(m)}_{k,c} = \mathcal{L}_Q(Z^{(m)} \mid Y = c, D = k)$ and define

$$w^{(m,c)}_k = Q(D = k \mid Y = c, D \in \mathcal{K}_{m,c}), \qquad k \in \mathcal{K}_{m,c}, \tag{40}$$

$$\bar{\mu}_{m,c} = \sum_{k \in \mathcal{K}_{m,c}} w^{(m,c)}_k \mu^{(m)}_{k,c}. \tag{41}$$

Conditional mutual information decomposes as

$$I_Q(Z^{(m)}, D \mid Y = c, D \in \mathcal{K}_{m,c}) = \sum_{k \in \mathcal{K}_{m,c}} w^{(m,c)}_k \, \mathrm{KL}\Big(\mu^{(m)}_{k,c} \,\|\, \bar{\mu}_{m,c}\Big). \tag{42}$$

Since all weights are positive and KL is nonnegative, the sum is zero iff $\mu^{(m)}_{k,c} = \bar{\mu}_{m,c}$ for every k, equivalently iff all feature laws in the support are equal. Singleton supports impose no cross-domain equality, empty supports are excluded from the conditional-information expression. □

A.6 Proof of Corollary 3.6

Proof. Let $\mathcal{E}_{m,c} \subseteq \mathcal{E}'_{m,c}$ for every expert and class. Any Θ satisfying all equalities in $\mathcal{E}'$ also satisfies those in $\mathcal{E}$, hence

$$\mathcal{F}_{\mathcal{E}'} \subseteq \mathcal{F}_{\mathcal{E}}. \tag{43}$$

Both families evaluate the same $Z_\pi = \sum_m \pi_m(X) Z^{(m)}$ under Q, so

$$\begin{aligned} V^Q_{\mathcal{E}'} &= \sup_{\Theta \in \mathcal{F}_{\mathcal{E}'}} I_Q(Z_\pi, Y) \\ &\leq \sup_{\Theta \in \mathcal{F}_{\mathcal{E}}} I_Q(Z_\pi, Y) = V^Q_{\mathcal{E}}. \end{aligned} \tag{44}$$

The same argument holds with a fixed router. The supports and model family are fixed comparison objects, not quantities assumed constant during training. □

Remark A.2 (Domain supports and input-dependent composition)**.** *A support specifies which full-cell distributions an expert aligns, while routing determines how expert outputs are composed. Within an expert and class, equality propagates along connected domain pairs, so sparse connected alignment can still imply all-domain equality. Different experts operate in different feature spaces. For $M = 1$ with unit routing, complete supports recover global conditional invariance, with multiple experts, expert-level and composed-representation invariance are distinct.*

A.7 Proof of Proposition 3.7

Proof. Let $\mathcal{G}_0$ and $\mathcal{G}_1$ be disjoint nonempty groups covering the source domains. Give every domain positive probability under Q, with $Q(D \in \mathcal{G}_0) = Q(D \in \mathcal{G}_1) = 1/2$. For $d \in \mathcal{G}_u$, set $U = u$ and define the conditional distribution by

$$Q(Y = y \mid D = d) = \tfrac{1}{2}, \qquad y \in \{-1, +1\}, \tag{45}$$

$$Q(E = -1 \mid Y = y, D = d) = \eta_u, \qquad 0 < \eta_0 < \eta_1 < \tfrac{1}{2}. \tag{46}$$

Here $E \in \{-1, +1\}$, so this specifies $E \perp Y \mid D$. Set $V = YE$ and $X = (U, V)$. Then $U \perp Y$ and $Q(X = (u, v)) = 1/4$ for every $u \in \{0, 1\}$ and $v \in \{-1, +1\}$. All logarithms are natural. Fix $1/2 < r < 1$ and $\kappa = \log(r/(1-r))$. The router is

$$\pi(X) = \mathrm{softmax}((1 - 2U)\kappa, 0) = \begin{cases} (r, 1-r), & U = 0, \\ (1-r, r), & U = 1. \end{cases} \tag{47}$$

Both constraint sets use the same deterministic expert class, containing constant maps and the linear maps below. Write $V^Q_{\mathcal{E}}(\pi)$ for the supremum of $I_Q(Z_\pi, Y)$ over experts satisfying $\mathcal{E}$ with this router fixed.

Complete alignment. Let g be any deterministic expert shared across domains. Alignment between a domain in $\mathcal{G}_0$ and one in $\mathcal{G}_1$ requires, at $Y = +1$,

$$(1 - \eta_0)\delta_{g(0,+1)} + \eta_0 \delta_{g(0,-1)} = (1 - \eta_1)\delta_{g(1,+1)} + \eta_1 \delta_{g(1,-1)}. \tag{48}$$

Suppose $g(0, +1) \neq g(0, -1)$. The left side has two distinct atoms with masses $1 - \eta_0$ and η_0. Equality requires the right side to have the same two atoms with the same masses. Its smaller mass is η_1, whereas the smaller mass on the left is η_0, contradicting $\eta_0 < \eta_1 < 1/2$. Thus the left side is a single point mass. Since both coefficients on the right are positive, all four expert values must coincide:

$$g(0, +1) = g(0, -1) = g(1, +1) = g(1, -1). \tag{49}$$

Every expert satisfying $\mathcal{E}_{\mathrm{all}}$ is therefore constant on the support of X. Writing these constants as c_1, c_2, we obtain $Z_\pi = \pi_1(U)c_1 + \pi_2(U)c_2$, a function of U. Since $U \perp Y$, every feasible pair satisfies $I_Q(Z_\pi, Y) = 0$. Constant experts are feasible, so

$$V^Q_{\mathcal{E}_{\mathrm{all}}}(\pi) = 0. \tag{50}$$

Subset-supported alignment. For each class, require expert 1 to align across domains in $\mathcal{G}_0$ and expert 2 across domains in $\mathcal{G}_1$. Within either group, the conditional law of X given Y is identical across domains. Applying a common deterministic expert preserves this equality. In particular, the following linear experts satisfy the subset constraints:

$$L_u = \log \frac{1-\eta_u}{\eta_u}, \qquad L_0 > L_1 > 0, \tag{51}$$

$$a_1 = \frac{rL_0 - (1-r)L_1}{2r-1}, \qquad a_2 = \frac{rL_1 - (1-r)L_0}{2r-1}, \tag{52}$$

$$Z^{(1)} = a_1 V, \qquad Z^{(2)} = a_2 V. \tag{53}$$

The denominator is nonzero. The identities $ra_1 + (1-r)a_2 = L_0$ and $(1-r)a_1 + ra_2 = L_1$ give

$$Z_\pi = \begin{cases} L_0 V, & U = 0, \\ L_1 V, & U = 1, \end{cases} = L_U V. \tag{54}$$

The four possible values $L_0, -L_0, L_1, -L_1$ are distinct. On this support, X is recovered by

$$U = \mathbf{1}\{|Z_\pi| = L_1\}, \qquad V = \operatorname{sign}(Z_\pi). \tag{55}$$

Consequently, for every $u \in \{0,1\}$ and $v, y \in \{-1,+1\}$,

$$Q(Y = y \mid Z_\pi = L_u v) = Q(Y = y \mid X = (u,v)). \tag{56}$$

Since V is uniform conditional on U, Bayes' rule gives

$$Q(Y = +1 \mid U = u, V = v) = \begin{cases} 1-\eta_u, & v = +1, \\ \eta_u, & v = -1, \end{cases} = \sigma(L_u v), \tag{57}$$

where $\sigma(z) = (1+e^{-z})^{-1}$. Together with the inverse above, this shows that a shared logistic classifier $q(Y = +1 \mid z) = \sigma(z)$ gives the exact posterior from Z_π. Under zero-one loss, its Bayes decision is $\widehat{Y} = \operatorname{sign}(Z_\pi) = V$, with risk

$$R_Q^* = Q(\widehat{Y} \neq Y) = Q(E = -1) = \frac{\eta_0 + \eta_1}{2}. \tag{58}$$

Let $h(p) = -p \log p - (1-p)\log(1-p)$. The posterior above gives $H_Q(Y \mid Z_\pi) = H_Q(Y \mid X) = [h(\eta_0) + h(\eta_1)]/2$, while $H_Q(Y) = \log 2$. Hence

$$\begin{aligned} I_Q(Z_\pi, Y) &= I_Q(X, Y) \\ &= \log 2 - \tfrac{1}{2}\big[h(\eta_0) + h(\eta_1)\big] > 0. \end{aligned} \tag{59}$$

The strict inequality follows from $h(\eta_u) < \log 2$ for $0 < \eta_u < 1/2$. For every deterministic representation Z of X, the chain rule gives $I_Q(X,Y) = I_Q(Z,Y) + I_Q(X,Y \mid Z) \geq I_Q(Z,Y)$. The constructed experts attain this upper bound, so

$$V_{\mathcal{E}_{\text{sub}}}^Q(\pi) = I_Q(X,Y). \tag{60}$$

Together,

$$V_{\mathcal{E}_{\text{all}}}^Q(\pi) = 0 < V_{\mathcal{E}_{\text{sub}}}^Q(\pi) = I_Q(X,Y). \tag{61}$$

□

Scope of the construction. Both families use the same source distribution and deterministic model class. The router is fixed to equation 47, the expert-level alignment constraints determine the two feasible families. The result concerns predictive information and Bayes risk under Q. Taking at least two domains per group makes each subset constraint a comparison across domains.

A.8 Predictive Information and Routing Gradients

Let Y have a finite label set and let $q_\theta(\cdot \mid Z_\pi)$ be the shared classifier's probability vector. All expectations use the fixed reference law Q. Assume $\mathcal{L}_{\mathrm{cls}} = \mathbb{E}_Q[-\log q_\theta(Y \mid Z_\pi)] < \infty$. Conditioning on Z_π and adding and subtracting the log posterior gives

$$\begin{aligned}\mathcal{L}_{\mathrm{cls}} &= \mathbb{E}_Q[-\log Q(Y \mid Z_\pi)] + \mathbb{E}_Q\left[\log \frac{Q(Y \mid Z_\pi)}{q_\theta(Y \mid Z_\pi)}\right] \\ &= H_Q(Y \mid Z_\pi) + \mathbb{E}_{z\sim Q_{Z_\pi}} \mathrm{KL}(Q(Y \mid Z_\pi = z) \,\|\, q_\theta(Y \mid z))\,. \end{aligned} \tag{62}$$

Here Q_{Z_π} is the distribution of Z_π under Q. Substituting $H_Q(Y \mid Z_\pi) = H_Q(Y) - I_Q(Z_\pi, Y)$ and using nonnegativity of KL yields

$$I_Q(Z_\pi, Y) \geq H_Q(Y) - \mathcal{L}_{\mathrm{cls}}. \tag{63}$$

Equality holds if and only if $q_\theta(\cdot \mid z) = Q(Y \in \cdot \mid Z_\pi = z)$ for Q_{Z_π}-almost every z.

For a fixed input and label, write $z = \sum_n \pi_n z^{(n)}$ with $\pi = \mathrm{softmax}(s)$. Hold the expert outputs fixed when differentiating with respect to s_m. At a point where the classification loss is differentiable,

$$\frac{\partial \pi_n}{\partial s_m} = \pi_n\big(\mathbf{1}\{n = m\} - \pi_m\big), \tag{64}$$

$$\frac{\partial z}{\partial s_m} = \sum_n z^{(n)} \frac{\partial \pi_n}{\partial s_m} = \pi_m\big(z^{(m)} - z\big). \tag{65}$$

The chain rule therefore gives the classification-path derivative

$$\frac{\partial \ell_{\mathrm{cls}}}{\partial s_m} = \pi_m \left\langle \nabla_z \ell_{\mathrm{cls}}, z^{(m)} - z \right\rangle. \tag{66}$$

For the alignment gradient, fix a minibatch and its valid pair set $\mathcal{I}_B$. Let ψ contain parameters exclusive to the router and let Θ_{feat} contain the expert and shared-encoder parameters. Each transport cost $W_{\varepsilon,t}$ depends on Θ_{feat} and has no additional dependence on ψ. At the current iterate $(\psi_0, \Theta_{\mathrm{feat},0})$, stop-gradient holds $\bar{a}_t = \widehat{a}_t(\psi_0, \Theta_{\mathrm{feat},0})$ constant during differentiation. The corresponding frozen-weight objective is

$$\widetilde{\mathcal{L}}^{\mathrm{OT}}_{\mathtt{ssi}}(\psi, \Theta_{\mathrm{feat}}, \bar{a}) = \sum_{t\in\mathcal{I}_B} \bar{a}_t W_{\varepsilon,t}(\Theta_{\mathrm{feat}}). \tag{67}$$

Its value at the current iterate equals the stated alignment loss. Wherever the feature costs are differentiable, the gradients used in that step satisfy

$$\nabla_\psi \widetilde{\mathcal{L}}^{\mathrm{OT}}_{\mathtt{ssi}} = 0, \tag{68}$$

$$\nabla_{\Theta_{\mathrm{feat}}} \widetilde{\mathcal{L}}^{\mathrm{OT}}_{\mathtt{ssi}} = \sum_{t\in\mathcal{I}_B} \bar{a}_t \nabla_{\Theta_{\mathrm{feat}}} W_{\varepsilon,t}. \tag{69}$$

Here $t = (m, i, j, c)$ and $\bar{a}_t$ is the fixed value of $\mathrm{sg}(\widehat{a}_t)$ in the differentiation step. Classification and routing regularizers provide gradients to the router's own parameters. Alignment can update the shared encoder through the feature costs, which can change subsequent routing outputs.

A.9 Entropic Transport and Distribution Matching

Let μ and ν be probability measures on $\mathbb{R}^d$ with finite second moments, and let $\varepsilon > 0$. For the set $\Pi(\mu, \nu)$ of couplings with these marginals, define

$$W_\varepsilon(\mu, \nu) = \inf_{\gamma\in\Pi(\mu,\nu)} \left\{ \int \|z - z'\|_2^2 \, d\gamma(z, z') + \varepsilon \,\mathrm{KL}(\gamma \,\|\, \mu \otimes \nu) \right\}. \tag{70}$$

Every feasible transport cost is at least $W_2^2(\mu, \nu)$ and the KL term is nonnegative. Therefore,

$$W_2^2(\mu, \nu) \leq W_\varepsilon(\mu, \nu). \tag{71}$$

Now suppose μ and ν are empirical measures with at most n and n' support points. Merge repeated atoms and let $p_i, q_j > 0$ denote the marginal masses on the resulting supports. The finite transport polytope is compact, so an optimal unregularized coupling γ_0 exists. Let (A, B) be its pair of atom indices. With $0 \log 0 = 0$,

$$\begin{aligned} \mathrm{KL}(\gamma_0 \,\|\, \mu \otimes \nu) &= \sum_{i,j} (\gamma_0)_{ij} \log \frac{(\gamma_0)_{ij}}{p_i q_j} = I(A, B) \\ &\le \min\{H(A), H(B)\} \le \log \min\{n, n'\}. \end{aligned} \tag{72}$$

Using γ_0 as a feasible coupling in the regularized problem gives

$$\begin{aligned} W_\varepsilon(\mu, \nu) &\le W_2^2(\mu, \nu) + \varepsilon \,\mathrm{KL}(\gamma_0 \,\|\, \mu \otimes \nu) \\ &\le W_2^2(\mu, \nu) + \varepsilon \log \min\{n, n'\}. \end{aligned} \tag{73}$$

For a fixed minibatch and parameter value, write $t = (m, i, j, c)$ and $\Delta_t = W_2^2(\widehat{\mu}_{i,c}^{(m)}, \widehat{\mu}_{j,c}^{(m)})$. Each cell is nonempty and its support size is at most its sample count $n_{i,c}$ or $n_{j,c}$. Since $\mathrm{sg}(\widehat{a}_t)$ has forward value $\widehat{a}_t \ge 0$, summing the pairwise bounds yields

$$\begin{aligned} \sum_{t \in \mathcal{I}_B} \widehat{a}_t \Delta_t &\le \mathcal{L}_{\texttt{ssi}}^{\mathrm{OT}} \\ &\le \sum_{t \in \mathcal{I}_B} \widehat{a}_t \Delta_t + \varepsilon \sum_{t \in \mathcal{I}_B} \widehat{a}_t \log \min\{n_{i,c}, n_{j,c}\}. \end{aligned} \tag{74}$$

These bounds apply to exact transport optima on the observed empirical measures. Sampling error and finite-iteration solver error require separate bounds.

Exact equality and finite regularization. For any finite collection of probability-measure pairs (μ_t, ν_t) with finite second moments and fixed weights $w_t \ge 0$,

$$\sum_t w_t W_2^2(\mu_t, \nu_t) = 0 \quad \Longleftrightarrow \quad \mu_t = \nu_t \text{ for every } t \text{ with } w_t > 0. \tag{75}$$

This follows from nonnegativity and the equality condition for the Wasserstein distance. A binary mask therefore enforces equality on its selected pairs at zero unregularized discrepancy. Positive weights on every valid pair enforce all corresponding equalities. Within one expert and class, equality also propagates along connected selected pairs. The statement applies to empirical or population laws under the stated moment assumptions.

For $\varepsilon > 0$, identical empirical measures with at least two distinct atoms satisfy $W_\varepsilon(\mu, \mu) > 0$. Indeed, the finite regularized problem attains its minimum, a zero value would require both zero transport cost and zero KL. Zero KL forces the coupling to be $\mu \otimes \mu$, whose transport cost is positive for such a measure. The separation proposition concerns exact population equalities. The training objective assigns finite alignment weights and uses the regularized empirical cost bounded above.

B Details of the MESSI MoE Architecture

B.1 MoE Architecture

MESSI keeps the pretrained DeiT encoder architecture unchanged and attaches a soft MoE head to the final CLS representation. Let $u = b_\phi(x)$ denote the encoder output. Each of the M experts produces

$$z^{(m)} = h_m(u), \tag{76}$$

while the router computes

$$\pi(x) = \mathrm{softmax}(g_\psi(u)). \tag{77}$$

The final representation is the weighted expert mixture

$$z(x) = \sum_{m=1}^{M} \pi_m(x) z^{(m)}, \tag{78}$$

followed by the classifier. In Appendix A, the corresponding population representation is $Z_\pi = z(X)$.

The routing-weighted class-conditional alignment loss is applied directly to the expert representations $\{z^{(m)}\}_{m=1}^{M}$. MESSI does not replace Transformer feed-forward layers with MoE blocks and does not use the block-level sparse top-k architecture of GMoE. GMoE is used only as a comparison baseline with its original architecture. All experts are evaluated per input, with routing weights controlling their mixture contributions. The router receives neither class nor domain labels.

B.2 Scalability of Subset-Conditioned OT

The subset-conditioned OT loss is computed on the current source minibatch. It does not use target-domain samples, memory banks, queues, cached features, cross-batch accumulation, or moving-average feature estimates. At each step, we concatenate the source-domain minibatches and compute labels y, source-domain indices d, routing probabilities $\pi_m(x)$, and per-expert features $h_m(b_\phi(x))$. Let $\mathcal{B}_{i,c} = \{n : y_n = c, d_n = i\}$ index class-domain cell (i, c) in the concatenated minibatch. For expert m, class c, and source domain i, define

$$\mathcal{Z}_{i,c}^{(m)} = \{h_m(b_\phi(x_n))\}_{n\in\mathcal{B}_{i,c}}, \qquad n_{i,c} = |\mathcal{B}_{i,c}|. \tag{79}$$

For $n_{i,c} > 0$, its uniform empirical feature distribution is

$$\widehat{\mu}_{i,c}^{(m)} = \frac{1}{n_{i,c}} \sum_{n\in\mathcal{B}_{i,c}} \delta_{h_m(b_\phi(x_n))}, \tag{80}$$

where δ_z is a point mass. Each cell uses all its samples uniformly, routing weights act between cells, not on individual samples.

Only active minibatch cells are used. A class c contributes only if it appears in at least two source domains in the concatenated minibatch. A domain pair (i, j) contributes only if $n_{i,c} > 0$ and $n_{j,c} > 0$. Missing class-domain cells are skipped. Singleton class-domain groups are allowed, since empirical OT is well-defined for non-empty empirical distributions.

For each active tuple (m, i, j, c), the OT term is weighted by the minibatch routing responsibility

$$\widehat{\rho}_{i,c}^{(m)} = \frac{1}{n_{i,c}} \sum_{n\in\mathcal{B}_{i,c}} \pi_m(x_n), \tag{81}$$

$$\widehat{a}_{ijc}^{(m)} = \sigma\left(\alpha\widehat{\rho}_{i,c}^{(m)}\right)\sigma\left(\alpha\widehat{\rho}_{j,c}^{(m)}\right), \qquad \alpha > 0. \tag{82}$$

The routing averages are computed from the current minibatch and detached before weighting the OT discrepancy. Thus routing sets pairwise alignment strength, while detaching the weights removes the direct incentive to lower the loss by reducing routing responsibility. Expert features and any trainable shared encoder still receive discrepancy gradients, see Appendix A.8.

For each active pair, we compute entropic OT with a differentiable log-space Sinkhorn solver, uniform empirical marginals, and a squared Euclidean cost matrix over expert features. The entropic regularization coefficient and the number of Sinkhorn iterations T are fixed. Let $\mathcal{C}_B$ be the set of active classes in the minibatch, and let

$$\mathcal{P}_B(c) = \{(i, j) : i < j,\ n_{i,c} > 0,\ n_{j,c} > 0\} \tag{83}$$

be the active source-domain pairs for class c. If d_z is the expert-feature dimension, the cost is

$$O\left(M(d_z + T)\sum_{c\in\mathcal{C}_B}\sum_{(i,j)\in\mathcal{P}_B(c)} n_{i,c}n_{j,c}\right). \tag{84}$$

Here d_z accounts for pairwise costs and T for Sinkhorn iterations. The cost therefore scales with the number and size of active class-domain pairs in the minibatch, not with the total number of dataset classes. This distinction matters for large-class datasets such as DomainNet, where most classes are absent from any single minibatch. The worst case remains expensive when many samples from the same class appear across multiple source domains, which explains the higher training-time cost of the OT variant in Table 5.

The OT loss is disabled at test time. Inference uses only the encoder, router, expert heads, and classifier, and therefore requires no Sinkhorn iterations or pairwise class-domain computations.

MMD variant for computational comparison. We also implement an MMD variant that keeps the same backbone, router, experts, active-pair selection rule, and auxiliary losses, replacing only the OT discrepancy. Using the same empirical measures and stopped pair weights,

$$\mathcal{L}_{\texttt{ssi}}^{\mathrm{MMD}} = \sum_{m=1}^{M} \sum_{c\in\mathcal{C}_B} \sum_{(i,j)\in\mathcal{P}_B(c)} \mathrm{sg}\Big(\widehat{a}_{ijc}^{(m)}\Big)\ \mathrm{MMD}^2\left(\widehat{\mu}_{i,c}^{(m)}, \widehat{\mu}_{j,c}^{(m)}\right), \tag{85}$$

where sg stops gradients without changing the forward value. With a Gaussian RBF kernel k,

$$\mathrm{MMD}^2(P,Q) = \mathbb{E}_{z,z'\sim P}k(z,z') + \mathbb{E}_{\tilde{z},\tilde{z}'\sim Q}k(\tilde{z},\tilde{z}') - 2\mathbb{E}_{z\sim P,\tilde{z}\sim Q}k(z,\tilde{z}). \tag{86}$$

For groups of size $n_{i,c}$ and $n_{j,c}$, MMD requires $O(n_{i,c}^2 + n_{j,c}^2 + n_{i,c}n_{j,c})$ kernel evaluations and no iterative solver. OT instead costs $O((d_z+T)n_{i,c}n_{j,c})$ per pair because it forms the cost matrix and runs Sinkhorn iterations. Both variants have the same inference-time architecture because the discrepancy loss is used only during training.

B.3 From Class-Domain Supports to Routing-Weighted Alignment

Appendix A.5 defines subset-shared invariance through class-conditional domain supports. Let $\mu_{k,c}^{(m)} = \mathcal{L}_Q(Z^{(m)} \mid Y=c, D=k)$ and $\mathcal{K}_c = \{k : P_k(Y=c) > 0\}$. A support $\mathcal{K}_{m,c} \subseteq \mathcal{K}_c$ requires

$$\mu_{i,c}^{(m)} = \mu_{j,c}^{(m)}, \qquad \forall i,j \in \mathcal{K}_{m,c}. \tag{87}$$

For $|\mathcal{K}_{m,c}| \geq 2$, Lemma 3.5 equivalently gives

$$I_Q\Big(Z^{(m)}, D \mid Y=c,\ D\in\mathcal{K}_{m,c}\Big) = 0. \tag{88}$$

This is a domain-level condition for each class, and equation 80 provides its minibatch empirical counterpart.

We connect these supports to routing through the population responsibility

$$\rho_{k,c}^{(m)} = \mathbb{E}_Q[\pi_m(X) \mid D=k, Y=c], \qquad k\in\mathcal{K}_c. \tag{89}$$

Its minibatch estimate $\widehat{\rho}_{k,c}^{(m)}$ determines the pair weights, giving

$$\mathcal{L}_{\texttt{ssi}}^{\mathrm{OT}} = \sum_{m=1}^{M} \sum_{c\in\mathcal{C}_B} \sum_{(i,j)\in\mathcal{P}_B(c)} \mathrm{sg}\Big(\widehat{a}_{ijc}^{(m)}\Big)\, W_\varepsilon\left(\widehat{\mu}_{i,c}^{(m)}, \widehat{\mu}_{j,c}^{(m)}\right). \tag{90}$$

Since $1/4 \leq \widehat{a}_{ijc}^{(m)} \leq \sigma(\alpha)^2$, routing assigns relative strengths to valid pairs rather than a binary support. Exact supports define the structural feasible-set comparison, the training objective uses continuous penalties.

Appendix A.9 shows that for empirical measures with at most n, n' atoms,

$$0 \leq W_\varepsilon(\mu,\nu) - W_2^2(\mu,\nu) \leq \varepsilon \log\min\{n,n'\}. \tag{91}$$

Thus entropic OT upper-bounds squared Wasserstein discrepancy up to an explicit excess term. Since $W_2^2 = 0$ characterizes distribution equality whereas W_ε need not vanish on identical distributions, OT serves as a surrogate for distribution matching rather than an estimator of mutual information.

Finally, classification supervises the composed representation Z_π, whereas alignment regularizes expert-level feature laws. Appendix A.8 connects the classification objective to $I_Q(Z_\pi, Y)$ without requiring per-expert classifiers. Expert indices and unordered domain-pair indices are permutation-invariant, so their numerical labels do not define the learned structure.

C Experimental Protocols and Datasets

This appendix specifies the datasets and evaluation protocols used in the main text. We first describe the DomainBed leave-one-domain-out protocol for standard OOD generalization. We then introduce Rotated-Colored MNIST, a controlled benchmark that disentangles shape-based prediction from rotation and color-label shortcut shifts, and define the fixed-budget domain-growth protocol used to vary source-domain heterogeneity without increasing the training budget. Finally, we describe the WILDS-iWildCam (Koh et al., 2021; Beery et al., 2021) protocol for the additional many-domain camera-trap evaluation.

C.1 DomainBed Protocol

We evaluate OOD generalization on DomainBed using the standard leave-one-domain-out protocol. For each dataset, one domain is selected as the target domain and all remaining domains are used for training. We follow the training-domain validation criterion: 20% of each source domain is reserved for validation, and the checkpoint with the highest average source-validation accuracy is selected. Target-domain data are not used for training, validation, hyperparameter tuning, or model selection.

All results are averaged over three random seeds. For each target split, all methods use the same source domains, target domain, and evaluation protocol. The dataset-level score is the mean accuracy over all target domains.

C.2 Rotated-Colored MNIST Construction

We construct Rotated-Colored MNIST as a controlled diagnostic for domain generalization. Each environment is defined by two nuisance factors: a rotation angle applied to the digit image and a color-label correlation applied after label noise. The prediction task is binary digit classification, so digit shape is the label-relevant factor, while rotation and color correlation define the environment shift.

Algorithm 1 summarizes the construction. We pool the MNIST training and test sets into $N = 70{,}000$ examples. The ten-way label is converted into a binary label by mapping digits $\{0, 1, 2, 3, 4\}$ to class 1 and digits $\{5, 6, 7, 8, 9\}$ to class 0. Let $\bar{y}$ denote this clean binary label. Following Colored MNIST, we corrupt it with label noise $\eta_y \sim \text{Bernoulli}(0.25)$:

$$y = \bar{y} \oplus \eta_y, \tag{92}$$

where $\oplus$ denotes XOR. This prevents the binary task from being perfectly deterministic.

Environment parameterization. We construct candidate environments indexed by $e \in \{0, \dots, E-1\}$. Each environment has a rotation angle θ_e and a color-flip probability p_e. The grid is controlled by a rotation step Δ_θ and a color step Δ_p. Let $C = \text{round}(1/\Delta_p)$ be the number of color levels per rotation. Under the row-major layout,

$$\theta_e = \Delta_\theta \left\lfloor \frac{e}{C} \right\rfloor, \qquad p_e = \Delta_p (e \bmod C). \tag{93}$$

Environments in the same row share the same rotation and differ only in color-label correlation, whereas moving across rows adds geometric variation.

Algorithm 1: Rotated-Colored MNIST construction.

```
# E: number of candidate environments
# Dth: rotation step
# Dp: color-correlation step

X, y10 = load_mnist()
X, y10 = shuffle(X, y10)

C = round(1.0 / Dp)
envs = []

for e in range(E):
    th = Dth * floor(e / C)
    p  = Dp  * (e % C)

    Xe, ye10 = split_env(X, y10, e)

    y0 = float(ye10 < 5)
    y  = xor(y0, Bernoulli(0.25))

    Xr = rotate(Xe, th)

    c = xor(y, Bernoulli(p))

    Xc = two_channel(Xr)
    Xc[arange(len(Xc)), 1 - c, :, :] = 0

    envs.append((Xc / 255.0, y))

return envs
```

In the fixed-budget domain-growth experiments, we use $\Delta_\theta = 45^\circ$ and $\Delta_p = 0.1$. Increasing the number of source environments therefore increases heterogeneity in both rotation and shortcut strength, without increasing the total training budget.

Rotation and colorization. For each environment e, images assigned to that environment are first rotated by θ_e using bilinear interpolation. The rotated grayscale image is then duplicated into two channels. The active color channel is determined by the noisy label y and the environment-specific flip probability p_e:

$$c = y \oplus \eta_c, \qquad \eta_c \sim \text{Bernoulli}(p_e). \tag{94}$$

The channel indexed by c is retained and the other channel is set to zero. The resulting input has shape $2 \times 28 \times 28$ and is normalized to $[0, 1]$.

The parameter p_e controls the color shortcut. When $p_e = 0$, color is perfectly correlated with the noisy label. When $p_e = 0.5$, color is independent of the label. When p_e approaches 1, the correlation is reversed.

Purpose of the benchmark. This benchmark is used only as a diagnostic. Because the rotation angle and color-label correlation are known by construction, it allows us to test whether a method preserves shape-based predictive structure as source-domain heterogeneity increases, or instead collapses toward an overly restrictive globally aligned representation.

C.3 Fixed-Budget Domain-Growth Protocol

We use Rotated-Colored MNIST to study how DG methods behave as the number of source domains increases. The construction above defines the candidate environment pool, this protocol defines how source domains are sampled under a fixed training budget.

Held-out target environment. For all domain-growth experiments, we fix the target environment to

$$\theta_{\text{test}} = 0^\circ, \qquad p_{\text{test}} = 0.5. \tag{95}$$

Under equation 93, this corresponds to $e_{\text{test}} = 5$ when $\Delta_p = 0.1$. Since $p_{\text{test}} = 0.5$, color is independent of the binary label. The target domain therefore tests whether a method learns shape-based prediction rather than relying on the color shortcut.

Fixed-budget source-domain expansion. We vary the number of source domains as

$$K_s \in \{3, 5, 7, 9, 11, 13, 15, 17\}. \tag{96}$$

For each K_s, source domains are sampled from the candidate pool excluding e_{test}. The total number of training examples is fixed to B, so each selected source domain contributes at most $\lfloor B/K_s \rfloor$ examples. This separates the effect of increasing source-domain heterogeneity from simply increasing the amount of training data.

Sampling, seeds, and model selection. For each K_s, we sample R independent source-domain subsets. Each method is trained with three random seeds per subset, and we report mean and standard deviation over both subset sampling and training seeds. Model selection uses source-validation data only, the target domain is used only for final evaluation.

Table 6: Fixed-budget domain-growth protocol on Rotated-Colored MNIST.

Setting	Value
Source domains	$K_s \in \{3, 5, 7, 9, 11, 13, 15, 17\}$
Total training examples	Fixed to B for all K_s
Max examples per source domain	$\lfloor B/K_s \rfloor$
Target environment	$\theta_{\text{test}} = 0^\circ$, $p_{\text{test}} = 0.5$
Subset samples	R per K_s
Training seeds	3 per subset
Model selection	Source-validation only
Target usage	Final evaluation only

C.4 WILDS-iWildCam Protocol

Benchmarks. We use three complementary evaluation settings. DomainBed evaluates standard OOD generalization under established DG protocols. Rotated-Colored MNIST provides a controlled setting in which the number and heterogeneity of source domains can be varied while keeping the target domain and total source budget fixed. WILDS-iWildCam provides an additional real-world many-domain comparison. Because our iWildCam setup uses 224×224 inputs and a lightweight preprocessing pipeline, we use it as a controlled comparison among MoE variants rather than a leaderboard-oriented WILDS evaluation.

Splits and model selection. We follow the official WILDS five-split protocol, summarized in Table 7. Models are trained on the *train* split. The *ID validation* and *ID test* splits contain held-out images from source camera locations, whereas the *OOD validation* and *OOD test* splits contain images from camera traps disjoint from the training cameras. We select checkpoints using OOD validation and report OOD test macro F1 as the primary generalization metric.

Table 7: Split structure of WILDS-iWildCam. The domain is the camera-trap location.

Split	Camera locations	Role
Train	Source camera traps	Optimization
ID Val	Source camera traps	In-distribution validation
ID Test	Source camera traps	In-distribution test
OOD Val	Unseen camera traps	OOD checkpoint selection
OOD Test	Unseen camera traps	Final OOD evaluation

Compared methods. We compare three MoE-based methods under the same DeiT-S/16 backbone, preprocessing pipeline, and six-expert configuration. GMoE (Li et al., 2023) is the original sparse MoE baseline. OMoE (Feng et al., 2025) augments the MoE baseline with Gram–Schmidt orthogonalization and the importance-CV^2 load-balancing objective. MESSI uses the same backbone family and is trained with the proposed subset-conditioned invariance objective.

Preprocessing. Images are resized to 224×224 and normalized with ImageNet statistics. During training, we apply random resized cropping, horizontal flipping, color jittering, and random grayscale augmentation. This setup is lighter than the official WILDS-iWildCam high-resolution pipeline. We therefore use this experiment as a controlled comparison among related MoE variants, not as a direct reproduction of leaderboard-scale WILDS results.

Many-domain batching. We treat each camera-trap location as a domain. Since the training split contains 243 source locations, using all domains in every minibatch is not tractable. We use stochastic domain-subset batching following the WILDS `n_groups_per_batch` convention (Koh et al., 2021; Beery et al., 2021). At each step, we sample $K = 4$ source locations uniformly without replacement and draw 8 images from each location, forming a balanced minibatch of 32 images. For MESSI, these $K = 4$ active domains define $K(K-1)/2 = 6$ candidate domain pairs for $\mathcal{L}_{\texttt{ssi}}$.

Optimization and metric. All methods are trained with Adam using learning rate 3×10^{-5} and zero weight decay. For MESSI, we set $\lambda_{\texttt{ssi}} = 0.01$, $\lambda_{\mathrm{sp}} = \lambda_{\mathrm{bal}} = 0.02$, $\lambda_{\mathrm{div}} = 0.02$, and routing-pair temperature $\alpha = 4.0$. For conditional MMD, we use a multi-scale RBF kernel with bandwidths $\{1, 2, 4, 8, 16\}$. We report per-class macro F1, the standard WILDS metric for iWildCam. Macro F1 gives equal weight to each species and is less dominated by frequent classes such as empty frames.

Training budget. All compared methods are trained for 150K steps. The longer schedule is used because the subset-conditioned invariance objective introduces an additional multi-domain alignment term that slows early optimization before stabilizing. We therefore interpret this experiment as a convergence-aware comparison under a shared backbone and preprocessing pipeline, rather than an equal-step compute benchmark.

D Extended Results

D.1 Extended Results on DomainBed

Table 8: OOD accuracy (%) under the DomainBed training-domain validation criterion, averaged over three seeds. **MESSI-Ti** and **MESSI-S** use DeiT-Ti/16 and DeiT-S/16 backbones, respectively. Best results are in **bold**, second-best results are underlined.

Method	Backbone	PACS ↑	OfficeHome ↑	TerraInc ↑	DomainNet ↑	Avg. ↑
ERM (Vapnik, 1991)	ResNet-50	83.8±0.8	66.6±0.4	47.2±0.8	41.6±0.2	59.8
IRM (Arjovsky et al., 2019)	ResNet-50	81.1±0.5	58.2±0.2	38.7±0.8	30.6±1.0	52.2
MMD (Li et al., 2018a)	ResNet-50	81.4±0.7	60.1±0.5	42.2±0.9	20.5±0.3	51.1
DANN (Ganin et al., 2016)	ResNet-50	79.4±0.3	59.6±0.4	38.1±0.8	31.6±0.1	52.2
MixStyle (Zhou et al., 2021)	ResNet-50	82.7±0.4	59.8±0.7	41.1±1.0	34.1±0.1	54.4
CORAL (Sun & Saenko, 2016)	ResNet-50	83.5±0.4	66.0±0.5	43.8±0.9	39.2±0.3	58.1
Fishr (Rame et al., 2022)	ResNet-50	85.5±0.2	68.6±0.2	47.4±1.6	41.7±0.3	60.8
SAGM (Wang et al., 2023)	ResNet-50	86.4±1.2	69.4±0.2	48.8±1.0	43.2±0.4	61.9
LFME (Chen et al., 2024)	ResNet-50	84.9±0.4	68.5±0.2	**49.5±0.8**	38.6±0.2	60.4
ERM (Vapnik, 1991)	DeiT-S/16	86.2±0.1	72.2±0.4	42.0±0.8	47.3±0.2	61.9
DynMoE (Guo et al., 2025b)	DeiT-S/16	85.2±0.4	73.4±0.3	44.5±0.6	45.9±0.4	62.3
GMoE (Li et al., 2023)	DeiT-S/16	87.3±0.1	73.5±0.1	48.2±0.5	47.8±0.5	64.2
OMoE (Feng et al., 2025)	DeiT-S/16	87.1±0.3	73.2±0.1	46.0±0.5	46.1±0.2	63.1
MESSI-Ti-OT (Ours)	DeiT-Ti/16	86.0±0.3	69.6±0.2	42.5±1.1	43.3±0.4	60.4
MESSI-S-MMD (Ours)	DeiT-S/16	89.3±0.6	74.0±0.3	47.5±0.4	46.2±0.4	64.3
MESSI-S-OT (Ours)	DeiT-S/16	**90.9±0.7**	**76.1±0.2**	49.3±0.2	**48.6±0.3**	**66.2**

D.2 DomainBed Per-Domain Results

Tables 9–12 report per-target-domain accuracy. Best results are shown in **bold**, and second-best results are underlined. The average column matches Table 1.

Table 9: Per-domain accuracy on PACS.

Method	Art ↑	Cartoon ↑	Photo ↑	Sketch ↑	Avg. ↑
ERM	86.2±0.8	76.0±0.7	95.9±0.4	77.1±0.9	83.8±0.8
IRM	83.0±0.6	74.5±0.4	95.8±0.3	71.1±0.7	81.1±0.5
MMD	84.0±0.8	74.8±0.6	95.7±0.3	71.1±1.0	81.4±0.7
CORAL	85.2±0.7	76.6±1.0	95.1±0.2	77.1±0.8	83.5±0.4
Fishr	87.4±0.3	78.4±0.4	97.7±0.1	78.5±0.5	85.5±0.2
SAGM	87.5±1.6	80.7±1.4	96.4±0.6	81.0±3.2	86.4±1.2
DynMoE	89.0±0.3	81.2±0.5	98.9±0.2	71.7±0.7	85.2±0.4
GMoE	89.9±0.4	83.7±0.5	98.9±0.1	76.7±0.6	87.3±0.1
OMoE	89.6±0.4	83.4±0.5	98.8±0.2	76.6±0.7	87.1±0.3
MESSI-Ti-OT (Ours)	88.3±0.9	78.1±1.1	98.1±0.5	79.3±2.5	86.0±0.3
MESSI-S-MMD (Ours)	90.4±0.6	86.7±0.8	98.7±0.4	81.4±0.6	89.3±0.6
MESSI-S-OT (Ours)	**91.6±0.3**	**88.7±0.9**	**99.1±0.3**	**84.1±1.4**	**90.9±0.7**

Table 10: Per-domain accuracy on OfficeHome.

Method	Art ↑	Clipart ↑	Product ↑	Real-World ↑	Avg. ↑
ERM	61.8±0.8	52.5±0.5	75.7±0.4	76.4±0.4	66.6±0.4
IRM	52.1±0.3	44.3±0.2	65.5±0.3	70.9±0.2	58.2±0.2
MMD	54.0±0.5	46.0±0.4	68.2±0.5	72.2±0.6	60.1±0.5
CORAL	61.8±0.4	52.7±0.6	74.2±0.5	75.3±0.5	66.0±0.5
Fishr	64.1±0.2	54.9±0.3	76.8±0.3	78.6±0.2	68.6±0.2
SAGM	64.8±0.7	55.3±0.6	78.1±0.7	79.4±0.6	69.4±0.2
DynMoE	71.6±0.3	59.5±0.4	79.6±0.2	82.7±0.3	73.4±0.3
GMoE	71.5±0.1	58.0±0.3	81.2±0.5	83.2±0.1	73.5±0.1
OMoE	71.1±0.2	57.8±0.3	80.9±0.4	83.0±0.2	73.2±0.1
MESSI-Ti-OT (Ours)	66.5±0.5	54.2±0.3	77.1±0.7	80.6±1.2	69.6±0.2
MESSI-S-MMD (Ours)	74.2±0.2	59.9±0.3	80.4±0.4	81.5±0.2	74.0±0.3
MESSI-S-OT (Ours)	**76.1±0.7**	**61.2±0.3**	**82.2±0.4**	**84.9±0.4**	**76.1±0.2**

Table 11: Per-domain accuracy on TerraIncognita.

Method	L100 ↑	L38 ↑	L43 ↑	L46 ↑	Avg. ↑
ERM	50.8±1.8	42.5±0.7	57.9±0.6	37.6±1.2	47.2±0.8
IRM	42.0±1.0	31.2±0.7	46.5±0.6	35.1±0.8	38.7±0.8
MMD	47.2±1.2	35.6±0.8	50.6±0.7	35.4±1.0	42.2±0.9
CORAL	46.1±1.0	39.6±3.2	53.3±0.7	36.2±0.8	43.8±0.9
Fishr	52.1±1.6	42.5±2.1	56.7±0.8	38.3±1.5	47.4±1.6
SAGM	52.7±3.2	**43.5±4.0**	**58.7±0.7**	40.3±1.4	48.8±1.0
DynMoE	54.8±0.8	36.4±0.7	49.0±0.5	37.8±0.9	44.5±0.6
GMoE	56.6±0.7	36.2±0.9	58.0±0.6	42.0±0.8	48.2±0.5
OMoE	55.5±0.8	34.2±1.4	55.3±0.6	39.0±0.7	46.0±0.5
MESSI-Ti-OT (Ours)	53.6±1.9	23.1±1.2	52.2±0.8	41.2±0.3	42.5±1.1
MESSI-S-MMD (Ours)	57.1±0.5	36.9±0.5	54.8±0.4	41.2±0.3	47.5±0.4
MESSI-S-OT (Ours)	**60.2±0.7**	38.3±0.5	56.7±0.3	**42.1±0.2**	**49.3±0.2**

Table 12: Per-domain accuracy on DomainNet.

Method	Real ↑	Clipart ↑	Painting ↑	Sketch ↑	Infograph ↑	Quickdraw ↑	Avg. ↑
ERM	61.2±0.4	58.8±0.3	46.7±0.4	50.1±0.7	19.4±0.3	13.4±0.2	41.6±0.2
IRM	48.0±0.8	42.5±0.7	34.2±0.8	36.8±1.0	12.1±0.5	10.0±0.6	30.6±1.0
MMD	33.8±0.5	29.3±0.4	23.0±0.4	24.8±0.5	6.7±0.2	5.4±0.2	20.5±0.3
CORAL	57.6±0.3	56.1±0.3	44.8±0.4	47.6±0.6	18.3±0.3	10.8±0.3	39.2±0.3
Fishr	60.0±0.2	58.4±0.1	47.1±0.2	50.1±0.2	20.6±0.1	14.0±0.1	41.7±0.0
SAGM	61.9±0.2	61.1±0.3	49.7±0.4	52.2±0.4	20.5±0.2	13.8±0.6	43.2±0.4
DynMoE	62.4±0.3	60.1±0.4	49.9±0.3	52.0±0.5	22.4±0.2	**28.6±0.4**	45.9±0.4
GMoE	68.8±0.3	66.1±0.4	55.2±0.4	55.1±0.5	24.7±0.5	16.9±0.4	47.8±0.5
OMoE	66.6±0.1	63.8±0.2	53.1±0.2	53.2±0.4	23.5±0.3	16.1±0.2	46.1±0.2
MESSI-Ti-OT (Ours)	62.4±0.5	60.1±0.3	49.1±0.3	49.5±0.3	22.1±0.4	16.4±0.3	43.3±0.4
MESSI-S-MMD (Ours)	67.4±0.3	64.9±0.3	53.4±0.5	53.7±0.3	24.1±0.5	13.7±0.3	46.2±0.4
MESSI-S-OT (Ours)	**69.9±0.2**	**66.9±0.2**	**55.7±0.4**	**55.9±0.2**	**25.6±0.5**	17.8±0.3	**48.6±0.3**

D.3 FIXED-BUDGET DOMAIN-GROWTH RESULTS

For the fixed-budget domain-growth experiment, we vary $K_s \in \{3, 5, 7, 9, 11, 13, 15, 17\}$ while keeping the total source training budget and the held-out target environment fixed. The target has $p_{\text{test}} = 0.5$, so color is independent of the binary label. Target accuracy therefore reflects whether a method learns shape-based prediction rather than relying on the color shortcut.

We summarize robustness using the peak target accuracy and the final accuracy at $K_{\max} = 17$. We define $\text{Drop} = \text{Peak} - \text{Acc}(K_{\max})$ and $\text{RelDrop} = \text{Drop}/\text{Peak}$. RelDrop prevents flat but low-performing methods from appearing robust simply because they never reach high accuracy.

Table 13: Fixed-budget domain-growth summary on Rotated-Colored MNIST. Drop is $\text{Peak} - \text{Acc}(K_{\max})$ with $K_{\max} = 17$, and RelDrop is Drop/Peak. Best results are shown in **bold**.

Method	Peak Acc. ↑	$\text{Acc}(K_{\max})$ ↑	Drop ↓	RelDrop ↓
CORAL (γ=1)	71.96	69.38	2.58	3.59%
CORAL (γ=10)	71.64	68.77	2.87	4.01%
CORAL (γ=100)	70.10	66.46	3.64	5.19%
CORAL (γ=250)	68.75	63.67	5.08	7.39%
CORAL (γ=500)	67.69	59.78	7.91	11.69%
MESSI (Ours)	**75.31**	**73.45**	**1.86**	**2.47%**

Table 14: Fixed-budget source-domain expansion on Rotated-Colored MNIST for $K_s \in \{3, 5, 7, 9\}$. The target environment and total source budget are fixed. Best results are shown in **bold**.

Method	$K_s = 3$ ↑	$K_s = 5$ ↑	$K_s = 7$ ↑	$K_s = 9$ ↑
CORAL (γ=1)	**50.84**	50.79	68.45	71.96
CORAL (γ=10)	50.79	50.79	68.80	71.64
CORAL (γ=100)	50.79	50.82	68.85	70.10
CORAL (γ=250)	50.79	50.86	65.65	68.75
CORAL (γ=500)	50.79	50.79	59.95	64.89
MESSI (Ours)	50.79	**57.14**	**74.00**	**75.31**

Table 15: Continuation of Table 14 for $K_s \in \{11, 13, 15, 17\}$. Best results are shown in **bold**.

Method	$K_s = 11$ ↑	$K_s = 13$ ↑	$K_s = 15$ ↑	$K_s = 17$ ↑
CORAL (γ=1)	70.44	71.00	70.20	69.38
CORAL (γ=10)	70.91	71.00	68.92	68.77
CORAL (γ=100)	69.10	68.59	66.35	66.46
CORAL (γ=250)	67.30	67.84	63.99	63.67
CORAL (γ=500)	67.69	66.17	62.81	59.78
MESSI (Ours)	**74.80**	**74.10**	**74.99**	**73.45**

Results. Table 13 shows that MESSI achieves the highest peak accuracy, the highest final accuracy at $K_{\max} = 17$, and the smallest relative drop. The full results in Tables 14 and 15 show the same pattern across individual source-domain budgets. At small K_s, all methods are close to chance because the source environments provide limited evidence for separating shape from color. Once source diversity becomes informative, MESSI improves sharply and remains stable as K_s increases, whereas stronger CORAL penalties become increasingly brittle. This supports the role of routing-conditioned subset alignment under increasing domain heterogeneity.

D.4 ADDITIONAL RESULTS ON WILDS-IWILDCAM

Table 16 reports additional results on WILDS-iWildCam (Koh et al., 2021; Beery et al., 2021) under the protocol in Appendix C.4. This experiment is a controlled many-domain comparison among closely related MoE variants, not a leaderboard-oriented WILDS evaluation.

Table 16: Additional results on WILDS-iWildCam. All methods use the same DeiT-S/16 backbone, 224×224 preprocessing pipeline, and training budget. We report macro F1 on OOD and ID splits. OOD test is the primary evaluation on unseen camera locations. Best results are shown in **bold**.

Method	Steps	OOD F1 ↑		ID F1 ↑	
		Val	Test	Val	Test
OMoE	150K	29.14	27.96	48.57	47.26
GMoE	150K	29.06	26.68	48.84	47.28
MESSI (Ours)	150K	**35.49**	**33.16**	**52.86**	**51.14**

Results. Table 16 shows that MESSI achieves the best macro F1 on all four evaluation splits. On the primary OOD test split, MESSI improves over OMoE by $+5.20$ F1 and over GMoE by $+6.48$ F1. The same pattern appears on the ID test split, where MESSI improves over OMoE by $+3.88$ F1 and over GMoE by $+3.86$ F1. These gains are obtained under the same backbone, preprocessing pipeline, number of experts, and training budget, indicating that the improvement is not explained by generic sparse MoE capacity alone.

Interpretation. The absolute scores are not intended to match heavily tuned WILDS leaderboard results, since we use lower-resolution inputs and a lightweight preprocessing pipeline. The purpose of this experiment is narrower: to test whether MESSI retains a relative advantage over closely related MoE baselines in a real many-domain setting. The consistent improvement on both OOD and ID splits supports the role of routing-conditioned subset alignment when domains are numerous, heterogeneous, and only partially share predictive structure.

E Ablations Isolate Routing-Conditioned Subset Alignment

E.1 Ablation on Training Objectives

The following ablations test whether MESSI's gains come from routing-induced subset-conditioned invariance rather than MoE capacity, generic feature alignment, or individual regularizers. Unless otherwise stated, all variants use the same backbone, number of experts, optimizer, augmentation pipeline, and model-selection criterion. We report OOD accuracy together with diagnostics tied to the intended role of each term: routing entropy for assignment confidence, expert-load standard deviation for usage imbalance, off-diagonal cosine similarity for expert redundancy, and class-conditional routing JS divergence for routing consistency. Diagnostics are computed on held-out source-validation splits, target-domain data are used only for final accuracy evaluation.

Table 17: Objective leave-one-out ablation on PACS, averaged over the four leave-one-domain-out runs. Diagnostic columns are rescaled as indicated in the headers. Best results are shown in **bold**.

Variant	PACS Acc. ↑	Routing Ent. ↓ ($\times 10^{-2}$)	Load Std. ↓ ($\times 10^{-1}$)	Offdiag Cos. ↓ ($\times 10^{-2}$)	Routing JS ↓ ($\times 10^{-4}$)
MESSI (Ours)	**90.90±0.70**	0.7±0.2	3.6±0.2	**-0.2±0.3**	**0.1±0.0**
w/o $\mathcal{L}_{\texttt{ssi}}$	86.89±0.75	1.2±0.3	3.5±0.2	-0.1±0.3	2.0±0.6
w/o $\mathcal{L}_{\texttt{sp}}$	87.06±0.90	53.2±5.5	3.0±0.3	0.0±0.4	1.0±0.4
w/o $\mathcal{L}_{\texttt{bal}}$	88.82±0.65	**0.3±0.1**	3.7±0.3	-0.1±0.3	**0.1±0.1**
w/o $\mathcal{L}_{\texttt{div}}$	88.79±0.70	5.8±1.2	1.2±0.2	35.6±3.0	33.0±4.0
w/o $\mathcal{L}_{\texttt{sp}}, \mathcal{L}_{\texttt{bal}}, \mathcal{L}_{\texttt{div}}$	88.59±0.80	39.7±4.5	**1.0±0.1**	38.9±3.5	30.0±3.5

Metric definitions. Routing entropy is $\mathbb{E}_x[-\sum_m \pi_m(x) \log \pi_m(x)]$. Load standard deviation is the standard deviation of average expert usage $\mathbb{E}_x[\pi_m(x)]$ across experts. Offdiag Cos. is the mean off-diagonal cosine similarity between expert representations. Routing JS is the average Jensen–Shannon divergence between domain-specific class routing prototypes and their class-wise template. Load standard deviation should be read together with routing entropy: a low load standard deviation can also arise from diffuse routing that uses all experts weakly.

Results. Table 17 shows that the full objective achieves the highest PACS accuracy. Removing $\mathcal{L}_{\texttt{ssi}}$ gives the largest drop, from 90.90% to 86.89%, indicating that routing-conditioned subset alignment is the main accuracy-critical component. Removing $\mathcal{L}_{\texttt{sp}}$ sharply increases routing entropy, so expert assignments become diffuse. Removing $\mathcal{L}_{\texttt{bal}}$ has a smaller effect but slightly increases load imbalance. Removing $\mathcal{L}_{\texttt{div}}$ strongly increases off-diagonal cosine similarity and routing JS, showing that experts become redundant and less class-consistent without explicit specialization. Removing all routing and specialization regularizers produces diffuse routing and high expert redundancy, with lower accuracy than the full model. Overall, $\mathcal{L}_{\texttt{ssi}}$ drives the OOD gain, while the auxiliary terms stabilize the sparse, balanced, and non-redundant expert decomposition.

E.2 Hyperparameter Sensitivity

We evaluate sensitivity to the number of experts and the routing regularization weights on PACS. All remaining hyperparameters are fixed to their default values.

Table 18: Sensitivity to the number of experts on PACS. We fix $\lambda_{\texttt{ssi}} = 0.01$ and $\lambda_{\texttt{sp}} = \lambda_{\texttt{bal}} = \lambda_{\texttt{div}} = 0.02$.

M	Acc. ↑	Params ↓	Routing Ent. ↓ ($\times 10^{-2}$)	Load Std. ↓ ($\times 10^{-1}$)
2	89.05 ± 0.68	24.0M	0.38 ± 0.71	4.95 ± 0.14
4	90.64 ± 0.21	26.4M	0.53 ± 0.21	4.33 ± 0.12
6	$\mathbf{90.90 \pm 0.70}$	28.8M	0.72 ± 0.23	3.62 ± 0.24
8	90.39 ± 0.58	31.1M	1.10 ± 0.40	3.30 ± 0.12
16	89.98 ± 0.46	40.6M	1.90 ± 0.52	2.42 ± 0.07

Performance varies by only 0.51 points for $M \in \{4, 6, 8\}$, while increasing to $M = 16$ adds parameters without improving accuracy. Thus, performance is not narrowly sensitive to the default choice $M = 6$.

Table 19: Sensitivity to $\lambda_{\texttt{sp}}$ on PACS with $M = 6$, $\lambda_{\texttt{bal}} = 0.02$, $\lambda_{\texttt{ssi}} = 0.01$, and $\lambda_{\texttt{div}} = 0.02$.

$\lambda_{\texttt{sp}}$	Acc. ↑	Routing Ent. ↓ ($\times 10^{-2}$)	Load Std. ↓ ($\times 10^{-1}$)
0.000	87.06 ± 0.90	53.20 ± 5.50	3.00 ± 0.30
0.005	89.37 ± 0.79	17.50 ± 3.50	3.20 ± 0.25
0.010	90.46 ± 0.73	4.40 ± 1.20	3.40 ± 0.20
0.020	$\mathbf{90.90 \pm 0.70}$	0.72 ± 0.23	3.62 ± 0.24
0.050	90.62 ± 0.74	0.25 ± 0.08	3.67 ± 0.08
0.100	89.77 ± 0.82	0.10 ± 0.04	3.70 ± 0.04
0.200	89.24 ± 0.80	0.05 ± 0.02	3.71 ± 0.02

Accuracy remains stable around the default for moderate sparsity weights, while removing $\mathcal{L}_{\texttt{sp}}$ produces diffuse routing and a clear performance drop.

Table 20: Sensitivity to $\lambda_{\texttt{bal}}$ on PACS with $M = 6$, $\lambda_{\texttt{sp}} = 0.02$, $\lambda_{\texttt{ssi}} = 0.01$, and $\lambda_{\texttt{div}} = 0.02$.

$\lambda_{\texttt{bal}}$	Acc. ↑	Routing Ent. ↓ ($\times 10^{-2}$)	Load Std. ↓ ($\times 10^{-1}$)
0.00	88.82 ± 0.65	0.30 ± 0.10	3.70 ± 0.02
0.02	$\mathbf{90.90 \pm 0.70}$	0.72 ± 0.23	3.62 ± 0.24
0.05	90.41 ± 0.62	0.97 ± 0.12	3.42 ± 0.10
0.10	89.73 ± 0.55	1.28 ± 0.18	1.69 ± 0.14
0.20	89.87 ± 0.48	1.35 ± 0.14	0.69 ± 0.11
0.50	89.88 ± 0.51	1.20 ± 0.16	0.44 ± 0.09
1.00	88.98 ± 0.66	1.18 ± 0.19	0.29 ± 0.06

Increasing $\lambda_{\texttt{bal}}$ progressively reduces expert-load imbalance while preserving competitive accuracy over a broad range. Performance decreases when balancing is removed or becomes overly dominant.

E.3 Expert Specialization

Figure 3 and Table 21 analyze expert specialization on OfficeHome. The goal of this diagnostic is to test whether MESSI learns distinct expert representations, rather than gaining accuracy only from increased MoE capacity.

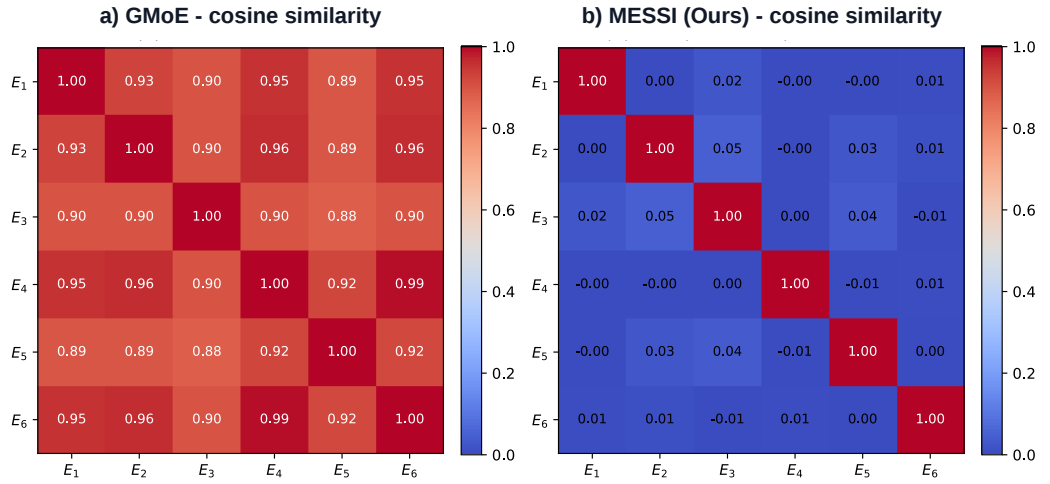


Figure 3: Expert specialization on OfficeHome. (a) Pairwise cosine similarity between expert representations in GMoE. (b) Pairwise cosine similarity between expert representations in MESSI. Lower off-diagonal similarity indicates stronger expert specialization.

Protocol. We follow the DomainBed leave-one-domain-out protocol on OfficeHome, using Art, Clipart, Product, and Real-World as the held-out domains in turn. All diagnostics are computed on the held-out target split, so the analysis reflects expert behavior on unseen domains. We use OfficeHome because its four visually diverse domains and fine-grained label space make expert redundancy easier to observe than on smaller benchmarks such as PACS.

Metrics. Both models use $M = 6$ experts. For each input x, we extract the output of each expert, denoted by $h_m(x)$. For MESSI, these are the outputs of the MoE head. For GMoE, we use the per-expert outputs from the final MoE layer before routing aggregation. We quantify specialization with the pairwise cosine matrix

$$C(x)_{m,n} = \cos(h_m(x), h_n(x)) \,, \tag{97}$$

averaged over samples and then over held-out domains. Lower off-diagonal values indicate less redundancy across experts.

Table 21 further reports three summary metrics. Mean Offdiag Cos. is the average off-diagonal entry of the cosine matrix. Effective rank is

$$\text{erank} = \exp\left(-\sum_i p_i \log p_i\right), \qquad p_i = \sigma_i \Big/ \sum_j \sigma_j, \tag{98}$$

where $\{\sigma_i\}$ are the singular values of the stacked expert-output matrix. Higher effective rank indicates that expert outputs span a more diverse subspace. Dead expert rate is the fraction of experts whose average routing mass falls below a small threshold.

Results. Figure 3 shows a clear qualitative difference. GMoE exhibits uniformly high off-diagonal cosine similarity, indicating that its experts produce highly redundant representations. In contrast, MESSI yields a near-diagonal matrix, showing that different experts encode substantially different features.

Table 21: Quantitative expert-specialization diagnostics on OfficeHome. Mean off-diagonal cosine measures expert redundancy. Effective rank measures the diversity of the expert-output subspace. Dead expert rate is the fraction of experts whose average routing mass falls below a small threshold. Best results are shown in **bold**.

Method	Mean Offdiag Cos. ↓	Effective Rank ↑	Dead Expert Rate ↓
OMoE	0.35 ± 0.04	3.95 ± 0.12	0.45 ± 0.07
GMoE	0.92 ± 0.03	3.64 ± 0.15	0.54 ± 0.06
MESSI w/o $\mathcal{L}_{\texttt{div}}$	0.50 ± 0.05	3.75 ± 0.14	0.40 ± 0.06
MESSI (Ours)	$\mathbf{0.01 \pm 0.01}$	$\mathbf{4.21 \pm 0.11}$	$\mathbf{0.38 \pm 0.05}$

Table 21 confirms this trend quantitatively. MESSI reduces mean off-diagonal cosine similarity from 0.92 in GMoE to 0.01, increases effective rank from 3.64 to 4.21, and lowers the dead expert rate from 0.54 to 0.38. Removing $\mathcal{L}_{\texttt{div}}$ weakens this effect: the off-diagonal cosine rises to 0.50 and the effective rank drops to 3.75. OMoE improves over GMoE, but remains substantially less specialized than the full MESSI model.

Takeaway. These results show that MESSI does not improve simply by adding MoE capacity. Instead, its routing and diversity objectives encourage experts to become both distinct and active, yielding a less redundant and more expressive decomposition.

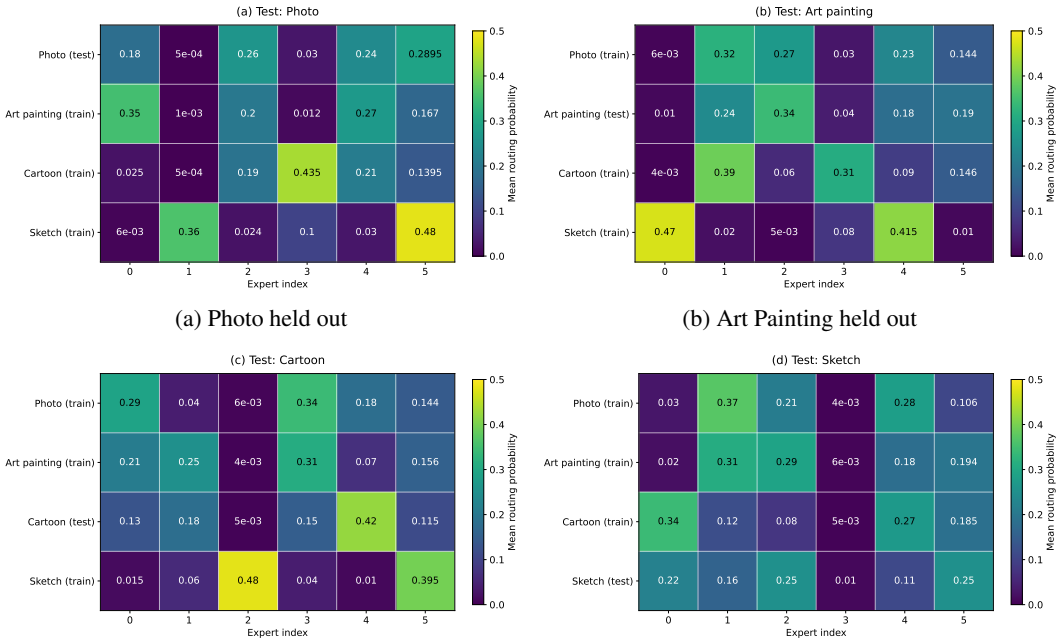
E.4 Routing Diagnostics

We analyze whether MESSI learns selective expert usage across PACS domains. For each leave-one-domain-out split, we average the routing probability of expert m over all samples from domain k,

$$\bar{\pi}_{k,m} = \frac{1}{|\mathcal{D}_k|} \sum_{x \in \mathcal{D}_k} \pi_m(x). \tag{99}$$

Each heatmap includes the three source domains and the held-out target domain. Since the router receives only the image representation, the target row measures routing behavior on an unseen domain without using its domain label.

Figure 4 shows that expert usage is selective and varies across domains. Different source domains place their largest routing mass on different expert subsets. For example, when Photo is held out, Cartoon primarily uses E_3, whereas Sketch favors E_1 and E_5. When Cartoon is held out, the unseen Cartoon domain places its largest mass on E_4, whereas Sketch favors E_2 and E_5. At the same time, several domains share active experts. In the Sketch split, for example, Photo and Art both assign substantial mass to E_1 and E_2. The held-out domains also distribute probability across multiple experts rather than following a one-expert-per-domain assignment. These patterns are consistent with routing-induced subset sharing, where expert usage varies across domains while individual experts can remain shared by multiple domains.



(a) Photo held out

(b) Art Painting held out

(c) Cartoon held out

(d) Sketch held out

Figure 4: Mean routing probabilities on PACS under leave-one-domain-out evaluation. Each panel uses a different held-out target domain. Rows show the three source domains and the held-out target domain, while columns show the six experts. Routing varies across domains while several domains share active experts, which is consistent with subset-level expert specialization.

E.5 Routing-Aware Alignment Controls

The subset-conditioned invariance objective makes routing determine where alignment is applied. We test this mechanism by comparing MESSI with two controls. **Global-MoE** uses the same MoE architecture and auxiliary regularizers, but aligns all valid class-conditional source-domain pairs uniformly. **Random-Subset-MoE** uses the same sparsity level as MESSI, but selects aligned slots at random instead of using routing responsibility. All diagnostics are computed on held-out source-validation splits only, target-domain samples are not used.

For expert m, source domain k, and class c, we compute the routing mass

$$\rho_{k,c}^{(m)} = \frac{1}{|\mathcal{D}_{k,c}|} \sum_{(x,y)\in\mathcal{D}_k,\, y=c} \pi_m(x). \tag{100}$$

This induces a pair responsibility for each class-conditional domain pair:

$$a_{ijc}^{(m)} = \sigma(\alpha\rho_{i,c}^{(m)})\sigma(\alpha\rho_{j,c}^{(m)}), \qquad i < j. \tag{101}$$

We call each tuple (m, i, j, c) a *slot*. A slot specifies which expert, domain pair, and class are considered for alignment.

Responsibility similarity. For each method A, we flatten the responsibilities $\{a_{ijc}^{(m,A)}\}$ over all common valid slots into a vector $\mathbf{a}_A$. We compare two methods using the Pearson correlation

$$r(A, B) = \mathrm{corr}(\mathbf{a}_A, \mathbf{a}_B). \tag{102}$$

We also compare the top-$q\%$ selected slots. Let S_A be the set of slots selected by method A. The Jaccard overlap is

$$J(A, B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}. \tag{103}$$

High $r(A, B)$ and $J(A, B)$ indicate that two methods select similar class-domain-expert slots.

Alignment on selected slots. A selected-versus-non-selected gap can be misleading because high-responsibility slots may already be easier to align. We therefore evaluate all methods on the same set of MESSI-selected slots. Let S_{M} be the top-$q\%$ slots selected by MESSI. For each method A, we compute

$$\Delta_A(S_{\mathrm{M}}) = \frac{1}{|S_{\mathrm{M}}|} \sum_{(m,i,j,c)\in S_{\mathrm{M}}} D\Big(\mathcal{Z}_{i,c}^{(m,A)},\, \mathcal{Z}_{j,c}^{(m,A)}\Big), \tag{104}$$

where $D(\cdot,\cdot)$ is the empirical energy distance between the two class-conditional expert-feature distributions. Lower values indicate better alignment on the slots selected by MESSI.

Table 22: Routing-aware alignment diagnostics. $r(A, B)$ is the Pearson correlation between pair-responsibility vectors, and $J(A, B)$ is the Jaccard overlap between top-$q\%$ selected-slot sets. $\Delta_A(S_{\mathrm{M}})$ is the feature discrepancy of method A evaluated on MESSI-selected slots, lower is better. Ctrl. Δ averages Global-MoE and Random-Subset-MoE. Diagnostics use held-out source-validation splits only.

Dataset	r(G,R)	J(G,R)	r(M,G)	J(M,G)	r(M,R)	J(M,R)	Ctrl. $\Delta\downarrow$	MESSI $\Delta\downarrow$
PACS	$+1.00$	0.95	-0.20	0.04	-0.20	0.04	0.234 ± 0.020	$\mathbf{0.096 \pm 0.014}$
OfficeHome	$+1.00$	0.93	-0.20	0.11	-0.20	0.11	0.289 ± 0.004	$\mathbf{0.217 \pm 0.006}$
TerraIncognita	$+1.00$	0.95	-0.20	0.13	-0.20	0.14	0.250 ± 0.011	$\mathbf{0.179 \pm 0.026}$
DomainNet	$+1.00$	0.95	-0.20	0.11	-0.20	0.10	0.252 ± 0.004	$\mathbf{0.143 \pm 0.032}$

Results. Table 22 shows two trends. First, Global-MoE and Random-Subset-MoE induce nearly identical responsibility patterns, with $r(\mathrm{G}, \mathrm{R}) = 1.00$ and high Jaccard overlap across datasets. In contrast, MESSI has weak or negative correlation with both controls and much lower selected-slot overlap. Thus, routing-aware alignment changes which expert-domain-class slots are selected for invariance, rather than simply sparsifying a global alignment objective.

Second, MESSI achieves lower discrepancy on the same MESSI-selected slots. The reduction is consistent across PACS, OfficeHome, TerraIncognita, and DomainNet. This indicates that the slots selected by MESSI are not only different from the controls, but are also better aligned after training. Together with the routing diagnostics above, these results show that the alignment weights depend jointly on class and domain rather than reflecting class specialization alone, supporting a structured class-domain decomposition learned through routing.

E.6 Pairwise-to-Global Invariance Sweep

We study how the invariance weight affects alignment, domain information, and target performance. On a diagnostic PACS split, we sweep $\lambda_{\text{inv}} \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ and report three quantities: pairwise class-conditional discrepancy, class-conditional domain-probe accuracy, and target-domain accuracy.

Figure 5 and Table 23 show three trends. First, increasing λ_{inv} steadily reduces pairwise discrepancy, indicating stronger class-conditional alignment across source-domain pairs. Second, the class-conditional domain-probe accuracy also decreases, approaching the source-domain chance level at $\lambda_{\text{inv}} = 10$. Thus, stronger invariance suppresses domain-predictive information in the learned representation. Third, target accuracy is not monotonic: it improves at moderate invariance strength, peaks at $\lambda_{\text{inv}} = 10^{-2}$, and then degrades sharply. At $\lambda_{\text{inv}} = 10$, target accuracy collapses even though pairwise discrepancy is nearly eliminated.

This sweep complements the theoretical result that enforcing pairwise conditional invariance over all domain pairs induces global conditional invariance. Empirically, however, stronger invariance is not always better. Moderate alignment improves generalization, whereas overly strong alignment removes label-relevant structure together with domain-specific variation. This supports the need for controlled, routing-conditioned alignment rather than uniformly enforcing global invariance.

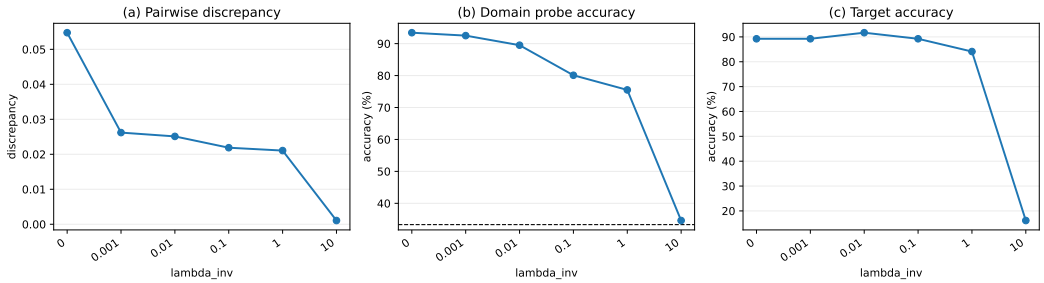


Figure 5: Pairwise-to-global invariance sweep on a PACS diagnostic split. We vary λ_{inv} and report pairwise class-conditional discrepancy, class-conditional domain-probe accuracy, and target-domain accuracy. Increasing λ_{inv} reduces pairwise discrepancy and domain predictability, but target accuracy peaks at moderate invariance strength and collapses under overly strong alignment. The dashed line in the domain-probe panel denotes the source-domain chance level.

Table 23: Numerical results for the pairwise-to-global invariance sweep on a PACS diagnostic split. Moderate invariance improves target accuracy, with the best result at $\lambda_{\text{inv}} = 10^{-2}$. In contrast, $\lambda_{\text{inv}} = 10$ nearly eliminates pairwise discrepancy and strongly suppresses domain predictability, but causes severe target-accuracy collapse.

λ_{inv}	Pairwise Disc. ↓	Domain Probe Acc. ↓	Target Acc. ↑
0	0.05475 ± 0.0030	93.42 ± 0.85	89.24 ± 0.90
10^{-3}	0.02620 ± 0.0022	92.51 ± 0.95	89.24 ± 0.85
10^{-2}	0.02511 ± 0.0019	89.51 ± 1.10	$\mathbf{91.69 \pm 0.31}$
10^{-1}	0.02188 ± 0.0017	80.09 ± 1.60	89.24 ± 1.05
1	0.02106 ± 0.0015	75.51 ± 1.85	84.11 ± 1.40
10	$\mathbf{0.00107 \pm 0.0004}$	$\mathbf{34.62 \pm 2.20}$	16.14 ± 2.80

F Implementation Details

F.1 Architecture

Backbone. MESSI is evaluated with DeiT-Ti/16 and DeiT-S/16 encoders initialized from ImageNet-1K pretrained checkpoints (Touvron et al., 2021). The DeiT-S/16 setting enables comparison with transformer-based and MoE-based baselines at the same encoder scale, whereas DeiT-Ti/16 tests whether subset-conditioned expert specialization remains effective with a lightweight encoder. In both settings, MESSI keeps the Transformer encoder unchanged and attaches a routing-based expert head to the final CLS representation. Thus, the backbone provides general visual features, while the expert head performs subset-conditioned specialization.

Expert head. MESSI uses $M = 6$ experts. Let $z = b_\phi(x)$ denote the final CLS representation produced by the encoder. Each expert maps z to a 384-dimensional representation with a two-layer MLP:

$$h_m(z) = W_{m,2}\,\sigma(W_{m,1}z), \tag{105}$$

where σ is GELU. The router maps the same CLS representation to an expert distribution:

$$\pi(x) = \mathrm{softmax}(W_{\mathrm{r}}z). \tag{106}$$

The final representation is the routed expert mixture

$$h(x) = \sum_{m=1}^{M} \pi_m(x)h_m(z), \tag{107}$$

and the classifier is applied to $h(x)$.

Router input and domain usage. The router receives only the image representation z. It does not receive the domain label. Domain labels are used only during training to compute source-domain alignment losses. At test time, MESSI requires neither domain labels nor target-domain data.

F.2 Training Details

Backbone. All MESSI-S variants use a DeiT-Small/16, while MESSI-Ti uses DeiT-Ti/16 encoder initialized from an ImageNet-1K pretrained checkpoint (Touvron et al., 2021). The encoder has model dimension 384. MESSI keeps the Transformer encoder unchanged and attaches a 6-expert mixture head to the final CLS representation, with expert output dimension 384.

Optimizer and schedule. We train with Adam (Kingma & Ba, 2014) and use a constant learning rate. The learning rate, weight decay, and number of training iterations are dataset-specific. These values are tuned once on source-validation splits following the DomainBed protocol and then fixed for all final runs:

Dataset	Learning rate	Weight decay	Iterations
PACS	3×10^{-5}	10^{-6}	5,000
OfficeHome	1×10^{-5}	10^{-6}	10,000
TerraIncognita	5×10^{-5}	10^{-4}	10,000
DomainNet	5×10^{-5}	0	15,000
iWildCam (WILDS)	3×10^{-5}	0	150,000

Mini-batch construction. For PACS, OfficeHome, TerraIncognita, and DomainNet, we follow the leave-one-domain-out protocol. Each minibatch contains 32 images per source domain, giving total batch sizes of 96 for PACS, OfficeHome, and TerraIncognita, and 160 for DomainNet. For iWildCam, we treat camera-trap locations as domains. At each step, we sample $K_{\mathrm{loc}} = 4$ source locations and draw $K_{\mathrm{img}} = 8$ images from each location, forming a minibatch of 32 images. These active locations define the domain pairs used by $\mathcal{L}_{\texttt{ssi}}$.

Augmentation. Training images use the following augmentation pipeline:

- `RandomResizedCrop(224, scale=(0.7,1.0)),`

- `RandomHorizontalFlip`,
- `ColorJitter(0.3, 0.3, 0.3, 0.3)`,
- `RandomGrayscale(p=0.1)`.

Images are then normalized with ImageNet statistics, $\mu = (0.485, 0.456, 0.406)$ and $\sigma = (0.229, 0.224, 0.225)$. Validation and test images are resized to 224×224 and normalized without augmentation.

Model selection. For PACS, OfficeHome, TerraIncognita, and DomainNet, we use the DomainBed training-domain validation criterion. A stratified 20% split of each source domain is reserved for validation, and we select the checkpoint with the highest mean source-validation accuracy. Target-domain data are not used for training, hyperparameter tuning, or model selection. For iWildCam, we follow the WILDS protocol and select checkpoints using the official OOD validation split, final performance is reported on the OOD test split.

F.3 Hyperparameters

All five datasets share the same MESSI objective weights and routing configuration. We denote the sub-invariance loss as $\mathcal{L}_{\texttt{ssi}}$ in the main method section, in the codebase the corresponding weight is called `lambda_inv`, i.e. $\lambda_{\texttt{ssi}} \equiv$ `lambda_inv`.

Hyperparameter	Value
$\lambda_{\texttt{ssi}}$	0.01
λ_{sp}	0.02
λ_{bal}	0.02
λ_{div}	0.02
Routing softmax temperature α	4.0
Number of experts M	6

F.4 Notation and Hyperparameter Summary

Table 24: Summary of notation used by MESSI. Domain indices refer to source domains in one leave-one-domain-out training run. Target-domain samples are not used when computing training losses.

Symbol	Meaning
K_s	Number of source domains in the current DG split
M	Number of experts
$c \in \mathcal{Y}$	Class index
$i, j \in \{1, \dots, K_s\}$	Source-domain indices
$b_\phi(x)$	Shared encoder representation
$h_m(b_\phi(x))$	Representation produced by expert m
$\pi_m(x)$	Router probability assigned to expert m
$\rho_{i,c}^{(m)}$	Mean routing mass for expert m, domain i, class c
$a_{ijc}^{(m)}$	Routing-induced weight for aligning domains i, j, class c, expert m
$\mathcal{Z}_{i,c}^{(m)}$	Minibatch expert features from domain i, class c
$\mathcal{C}_B$	Classes present in the current source minibatch
$\mathcal{P}_B(c)$	Source-domain pairs where class c appears in both domains
$\mathcal{L}_{\texttt{ssi}}$	Subset-conditioned invariance loss
$\mathcal{L}_{\texttt{sp}}$	Routing sparsity loss
$\mathcal{L}_{\texttt{bal}}$	Expert load-balancing loss
$\mathcal{L}_{\texttt{div}}$	Expert diversity loss

F.5 Baseline Implementations

We compare MESSI against three groups of baselines. The first group contains standard DomainBed baselines implemented with a ResNet-50 (He et al., 2016) backbone. ERM serves as the empirical

Table 25: Summary of implementation hyperparameters used by MESSI.

Hyperparameter	Role
M	Number of experts
α	Sharpness of routing-induced pair weights $a_{ijc}^{(m)}$
ϵ_{OT}	Entropic regularization coefficient in Sinkhorn OT
T_{OT}	Number of Sinkhorn iterations
$\lambda_{\mathtt{ssi}}$	Weight of subset-conditioned invariance loss
$\lambda_{\mathtt{sp}}$	Weight of routing sparsity loss
$\lambda_{\mathtt{bal}}$	Weight of load-balancing loss
$\lambda_{\mathtt{div}}$	Weight of expert diversity loss
β_{bal}	EMA coefficient for dataset-level routing load, if EMA balancing is used

risk minimization reference. IRM (Arjovsky et al., 2019) represents invariant risk minimization. MMD (Li et al., 2018a) and CORAL (Sun & Saenko, 2016) represent distribution-alignment methods, where MMD matches kernel mean embeddings and CORAL matches second-order feature statistics. DANN (Ganin et al., 2016) and CDANN (Li et al., 2018b) represent adversarial domain-alignment methods, with CDANN using class-conditional domain discrimination. MixStyle (Zhou et al., 2021) represents feature-statistics-based data augmentation. Fish (Shi et al., 2022) represents gradient-matching-based DG, and SWAD (Cha et al., 2021) represents flat-minima-based model selection and weight averaging. Together, these methods form the conventional DomainBed reference set across ERM, invariant learning, feature alignment, adversarial alignment, augmentation, gradient matching, and flat-minima optimization. Unless otherwise stated, we use the DomainBed protocol and the reported or reproduced DomainBed-compatible implementations under the same train-domain validation model-selection rule.

The second group contains recent optimization and architecture baselines. SAGM (Wang et al., 2023) represents sharpness-aware optimization for DG. GMoE (Li et al., 2023) represents sparse mixture-of-experts architectures for domain generalization, and GMoE+SAGM combines the GMoE backbone with the SAGM optimizer. LFME (Chen et al., 2024) is included as a recent expert-based DG method based on multi-expert learning. These baselines test whether the gains of MESSI can be explained by stronger optimization, larger model capacity, or generic expert specialization.

The third group contains controlled MoE variants used for mechanism analysis. These variants keep the MESSI backbone, number of experts, routing module, training budget, and model-selection criterion fixed, while changing only the training objective. This group includes classification-only MoE, globally aligned MoE, random-subset-aligned MoE, and leave-one-out variants that remove individual MESSI losses. These controls isolate the effect of routing-conditioned subset alignment from MoE capacity, global alignment, random sparse alignment, and auxiliary regularization.

F.6 Model Size

Table 26: Trainable parameter counts in millions, excluding the final dataset-specific classifier. Enc. denotes shared encoder parameters. MoE/Head denotes additional expert, router, and method-specific head parameters. Values marked by $\approx$ are computed from the implemented architecture or from reported model-size differences.

Method	Backbone	Enc.	MoE/Head	Total
ERM	ResNet-50	25.60	0	25.60
SAGM	ResNet-50	25.60	0	25.60
LFME	ResNet-50	25.60	0	25.60
GMoE	DeiT-S/16	21.70	$\approx$ 12.10	$\approx$ 33.80
GMoE + SAGM	DeiT-S/16	21.70	$\approx$ 12.10	$\approx$ 33.80
MESSI-S	DeiT-S/16	21.70	$\approx$ 7.09	$\approx$ 28.79
MESSI-Ti	DeiT-Ti/16	5.70	$\approx$ 1.78	$\approx$ 7.48

We exclude the final classifier because its size depends on the number of classes and is small relative to the encoder and expert modules. SAGM adds no trainable module, so it has the same parameter count as its base architecture, for the same reason, GMoE+SAGM has the same size as GMoE. LFME is listed with its ResNet-50 inference backbone for reference, although its training-time teacher/expert construction differs from a single MoE inference model. The GMoE MoE/Head count is estimated from the reported difference between GMoE-S/16 and ViT-S/16. MESSI head counts are computed from our implementation.

Model-size discussion. Table 26 shows that the gains of MESSI-S are not explained by a larger parameter budget. Under the DeiT-S/16 encoder scale, MESSI-S has approximately 28.79M trainable parameters, compared with approximately 33.80M for GMoE-S/16. Thus, MESSI-S uses about 15% fewer trainable parameters than GMoE-S/16 while still outperforming GMoE and OMoE in Table 1. Parameter count alone does not establish the mechanism, therefore, Appendix E.1 provides same-architecture ablations where the MoE capacity is fixed and only the objective terms are changed.

MESSI-Ti gives a smaller operating point. With a DeiT-Ti/16 encoder, the model has approximately 7.48M trainable parameters, making it about $3.85\times$ smaller than MESSI-S and $4.52\times$ smaller than GMoE-S/16. This variant is intended for settings where a lower parameter footprint is preferred, while retaining the same routing-conditioned subset-alignment design.