Can Circuit Alignment Predict OOD Generalization?

Ayan Banerjee[1,2§], **Abhra Chaudhuri**[3,§], **Josep Llados**[1,2], **Umapada Pal**[4], **Anjan Dutta**[5],
[1]Computer Vision Center, [2]Universitat Autònoma de Barcelona, [3]Fujitsu Research of Europe,
[4]Indian Statistical Institute, Kolkata, [5] University of Surrey
[1,2]{abanerjee,josep}@cvc.uab.cat, [3]abhra.chaudhuri@fujitsu.com
[4]umapada@isical.ac.in, [5]anjan.dutta@surrey.ac.uk

Abstract

Can out-of-distribution (OOD) generalization be predicted from a trained model's weights alone, without any target-domain data? Existing representational similarity metrics (CKA, SVCCA, RSA) compare activations rather than predicting generalization capability of the model. We show they are provably insensitive to structural rerouting in the computational graph, the very change distribution shift induces. We close this gap with the Circuit Alignment Score (CAS), which compares class-specific circuits across domains via graph kernels, decomposed into same-class coherence and cross-class confusion. Casting CAS as a Lebesgue integral over the domain distribution, we prove its Monte Carlo estimate recovers the ground-truth ranking of learners by OOD accuracy, with pairwise inversion error vanishing at rate $O(1/M)$, where M is the number of sampled domains. Across 48 learners on PACS, CAS attains 0.88 rank correlation with OOD accuracy, versus 0.58 (CKA), 0.23 (SVCCA), and 0.14 (RSA), with similar trends on other benchmarks and even against data-dependent methods, making it the first provably consistent predictor of distributional robustness requiring neither target-domain data nor labels. The code is available at: https://github.com/ayanban011/ACE.

1 Introduction

When can we predict whether a neural network will generalize under distribution shift, given only its trained weights? The question is not mere curiosity: in deployment settings ranging from clinical diagnostics to multilingual NLP, target-domain labels are unavailable by construction, and even unlabeled target samples may be scarce or arrive only after the model has been committed to. Yet, the field lacks a formally grounded answer. Methods that estimate target accuracy from unlabeled target data [6, 18, 72] presuppose access to the very distribution whose effect we wish to predict, and theoretical accounts of OOD generalization [70, 27] characterize *when* generalization is possible but yield no computable diagnostics on weights. **No prior work, to our knowledge, derives the structural conditions any weight-only OOD predictor must satisfy, nor establishes a consistency guarantee for ranking learners by such a predictor.**

The natural candidates – representational similarity metrics such as CKA [31], SVCCA [47], and RSA [32], operate on activation geometry and are provably blind to structural rerouting in the underlying computation. Two models can produce nearly identical penultimate-layer activations on a source domain while routing them through entirely different computational pathways, with sharply divergent OOD behavior. Rerouting is precisely what distribution shift induces; activation-level metrics measure the wrong object.

We propose to utilize a more expressive measure. Mechanistic interpretability has matured to the point where the computation a network performs for a given behavior can be localized to a sparse,

[§]Equal Contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

causally-grounded subgraph of neurons and connections [4, 8, 11, 16, 38, 66]. These *circuits* are the natural locus at which to ask whether a model's computation is preserved under distribution shift [52]: we prove that, in the limit of perfect OOD accuracy, a learner is necessarily *circuit-robust*, *i.e.*, its class-specific circuits are preserved across domains and remain distinct between classes, grounding circuit invariance as a principled structural proxy for OOD robustness rather than a heuristic.

Building on this, we ask: *what must any weight-only metric satisfy to predict OOD generalization?* We answer with three necessary conditions, each derived via an impossibility theorem showing that any metric violating the condition provably conflates a circuit-robust learner with a non-robust one. The conditions require the metric to (i) be sensitive to structural rerouting, (ii) resolve same-class versus cross-class structure separately, and (iii) respond monotonically to graded perturbation. CKA, SVCCA, and RSA each fail at least two by construction.

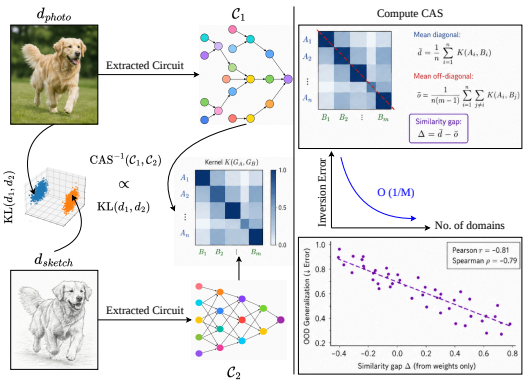


Figure 1: **Circuit Evolution prediction under distribution shift with CAS:** From weights alone, we extract class-specific circuits and compare them via graph kernels to score and rank learners based on OOD accuracy, without target data or labels.

Illustrated in Fig. 1, we then introduce the **Circuit Alignment Score (CAS)**, constructed in three steps, each *necessitated* by the three conditions: comparing circuits via graph kernels, arranging the resulting pairwise similarities into a class-conditional matrix, and aggregating into the gap between mean-diagonal and mean-off-diagonal. They capture *circuit drift* (same-class circuits change across domains) and *circuit entanglement* (different-class circuits collapse onto each other), the two structurally distinct failure modes of distribution shift.

Casting the population CAS as a Lebesgue integral over the domain distribution, we prove that its M-sample Monte Carlo estimate ranks learners with pairwise inversion error vanishing at rate $O(1/M)$. Combined with an empirically verifiable monotonicity condition, supported by the limit-case correspondence above and the causal-invariance literature [10, 27, 67], it yields convergence of the predicted ranking to the ground-truth OOD ranking. **To our knowledge, this is the first weights-only OOD predictor with both a structural justification and a consistency guarantee.**

Empirically, on a pool of 48 learners spanning four architectures (MLP, ResNet50, MobileNetV2, ViT-B/16), four training objectives (ERM [58], IRM [5], CORAL [55], DANN [17]), and three regularizations on PACS [33], CAS attains Spearman $\rho_S = 0.88$ with leave-one-domain-out OOD accuracy, against 0.58 (CKA), 0.23 (SVCCA), and 0.14 (RSA), and additionally even surpassing methods requiring target domain data and labels, directly verifying the monotonicity condition. The gap is consistent across target domains and extends to Office-Home [60] and DomainNet [41]. A controlled LoRA-interpolation experiment in Stable Diffusion XL [45] further shows CAS varies monotonically with continuous domain shift.

To summarize, (i) we prove that OOD robustness structurally implies circuit robustness in the limit; (ii) we derive three necessary conditions on weight-only OOD-predictive metrics via impossibility theorems and show that CKA, SVCCA, and RSA each fail at least two; (iii) we construct CAS to satisfy all three by design, with each construction step necessitated by one condition; (iv) we prove empirical CAS rankings recover the OOD ranking with pairwise inversion error $O(1/M)$; and (v) we empirically validate the full chain across 48 learners on three benchmarks.

2 Related Work

Predictors of OOD Generalization. Existing approaches include training-loss trends [7, 29], influence functions [71], margin-based predictors [25, 35], PAC-Bayes bounds [34, 44], meta-learning [65], and data-centric estimators [6, 13, 18, 72]. All require either prediction-variance preservation across domains or unlabeled target data. CAS, in contrast, predicts ranking from source-domain circuits alone.

Representation Similarity Metrics. [2] shows that representational metrics correlate with OOD accuracy only inconsistently. Common measures such as CKA [31], SVCCA [47], RSA [32] operate on activation geometry, and we prove (Sec. 3.1) that any such activation-factoring metric provably conflates circuit-robust and non-robust learners.

Circuit Tracing. A growing body of work extracts the circuits a network uses for a target behavior, via attribution patching [56], sparse autoencoders [57], probing [48], causal tracing [39], and neuron-level analysis [51], building on foundational frameworks for circuit-level interpretability [4, 8, 11, 16, 38, 66]. These methods implicitly assume a fixed data distribution, leaving open whether circuits are stable, transferable, or comparable across domains [52]. We address this gap by defining a metric directly on extracted circuits, agnostic to the choice of extractor.

3 Circuit Alignment Score (CAS)

3.1 What Does Predicting OOD Generalization from Weights Require?

We derive necessary conditions on any metric that predicts OOD generalization from a trained model's weights alone. For each condition, we prove an impossibility theorem (deferred to Sec. A): any metric violating the condition assigns identical values to a (distributionally) circuit-robust learner and a non-robust one, and is therefore unable to distinguish them. These conditions are not posited but necessitated by the structural premise established in the causal-invariance literature [10, 27, 67]: that OOD generalization is governed by the preservation of class-conditional computational structure across domains. Sec. 3.2 constructs CAS to satisfy all three conditions, and Sec. 3.3 proves that ranking learners by CAS recovers their OOD-accuracy ranking.

Problem setup: Let f denote a trained model, $D_1, D_2 \in \Omega_D$ two domains drawn from a domain space with distribution P_D, and $\mathcal{C}^{(i)}(f, D)$ the sparse weighted directed subgraph of f's computation mediating predictions for class i on domain D. We write $\mathcal{C}(f, D) = \{\mathcal{C}^{(i)}(f, D)\}_{i=1}^{c}$. A metric μ takes two such circuit families and returns a scalar.

Circuit robustness as a proxy for OOD robustness: Following standard formalizations [22, 27, 70], the *OOD generalization score* of a learner ℓ is its expected accuracy under the domain distribution:

$$g(\ell) \;=\; \int_{\Omega_D} \mathrm{Acc}(\ell; d)\, dP_D(d), \tag{1}$$

where $\mathrm{Acc}(\ell; d) \in [0, 1]$ is the classification accuracy of ℓ on domain d. We say ℓ_i is more OOD-robust than ℓ_j if $g(\ell_i) > g(\ell_j)$. We show that OOD robustness structurally implies a corresponding form of circuit-level invariance: a learner ℓ achieving perfect OOD accuracy must have class-specific circuits that are preserved across domains and remain distinct between classes (Corollary 1). This justifies working with circuit robustness as a structural proxy for OOD robustness. Formally, a learner ℓ is called *circuit-robust* between D_1, D_2 if:

$$\kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(i)}) \geq \alpha \;\; \forall i, \qquad \kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(j)}) \leq \beta \;\; \forall i \neq j, \tag{2}$$

for some $0 \leq \beta < \alpha \leq 1$. The two failure modes of this structure – *circuit drift* (some same-class similarity falls below α) and *circuit entanglement* (some cross-class similarity exceeds β), are the two ways class-conditional invariance can break down, and a predictive metric must resolve both.

Property 1 – Structural sensitivity: A metric must distinguish models that compute the same function via different circuits. Activation-level metrics, those that factor through any embedding into activation space, including CKA [31], SVCCA [47], and RSA [32], are blind to internal routing: two circuits with identical activation footprints yield identical metric values regardless of how they implement the function. By the universal approximation property of overparameterized networks [24, 28], such functionally-equivalent rerouted circuits provably exist, and one can be circuit-robust while the other is not. Theorem 5 (Sec. A) shows that any activation-factoring metric assigns the same value to both, hence cannot distinguish robust from non-robust learners.

Property 2 – Class-conditional resolution: A metric must report same-class preservation and cross-class entanglement separately. Drift and entanglement are structurally distinct failure modes: drift reduces diagonal similarity while leaving classes distinguishable, whereas entanglement collapses different-class circuits onto each other while diagonals can remain nominally high. A metric that

aggregates over all class pairs into a single scalar – equivalently, one whose value depends only on the multiset of pairwise similarities, not on which pairs are same-class versus cross-class – cannot separate these regimes. Theorem 6 (Sec. A) shows that any such aggregating metric assigns identical values to a circuit-robust configuration and a maximally entangled one, by a permutation argument on the entries of the class-conditional similarity matrix.

Property 3 – Semantic consistency: A metric must respond monotonically to graded perturbation: if one perturbation simultaneously reduces same-class similarity and increases cross-class similarity relative to another, it must yield a smaller metric value. Without monotonicity, the metric's numerical value carries no ordinal information about shift magnitude, and any ranking of learners or domains derived from it is incoherent. Theorem 7 (Sec. A) shows that non-monotone metrics provably invert perturbation orderings, assigning higher similarity to a strictly more-perturbed configuration, making them unsuitable for ranking-based OOD prediction.

Summary: Table 1 reports condition satisfaction for each metric. CKA, SVCCA, and RSA fail P1 (none operate on circuit graphs) and P2 (none form a class-conditional similarity matrix), with P3 thereby vacuous; RSA partially satisfies P2 via its block structure. CAS satisfies all three by construction (Sec. 3.2).

3.2 Construction of Circuit Alignment Score (CAS)

We now construct the Circuit Alignment Score, a metric over circuit families satisfying the three necessary conditions derived in Sec. 3.1. The construction proceeds in three steps, each necessitated by one of the conditions: graph-kernel comparison of circuits (P1), arrangement of pairwise similarities into a class-conditional matrix (P2), and decomposition into mean-diagonal minus mean-off-diagonal (P3).

Table 1: Necessary conditions for OOD-predictive metrics derived in this section. ✓ = satisfied by construction; ✗ = fails by impossibility theorem (Sec. A); ❍ = partially satisfied.

Condition	CKA	SVCCA	RSA	CAS
P1: Structural sensitivity	✗	✗	✗	✓
P2: Class-conditional resolution	✗	✗	❍	✓
P3: Semantic consistency	✗	✗	✗	✓

Step 1 – Comparing circuits structurally: P1 forbids any metric that factors through an embedding into activation space. The natural alternative is to compare circuits as the structured objects they are, *i.e.*, sparse weighted directed graphs, using a kernel defined on graph topology rather than activation outputs. We equip circuit space with a graph kernel $\kappa : \mathcal{C} \times \mathcal{C} \to [0, 1]$, normalized so $\kappa(C, C) = 1$, that compares circuits via their nodes and edges. We use treelet [19], random walk [36], and optimal-transport [43] graph kernels; Sec. F ablates this choice.

Step 2 – Resolving class-conditional structure: P2 forbids any metric that aggregates pairwise similarities without distinguishing same-class from cross-class pairs. Given two circuit families $\mathcal{C}_1 = \{C_1^{(i)}\}_{i=1}^c$ and $\mathcal{C}_2 = \{C_2^{(i)}\}_{i=1}^c$, we form the class-conditional similarity matrix:

$$S \in \mathbb{R}^{c \times c}, \qquad S_{ij} = \kappa\Big(C_1^{(i)}, C_2^{(j)}\Big). \tag{3}$$

Diagonal entries S_{ii} measure preservation of each class's circuit across domains (low S_{ii} indicates *circuit drift*); off-diagonal entries S_{ij} measure cross-class similarity (high S_{ij} for $i \neq j$ indicates *circuit entanglement*). The two failure modes from Eq. (2) are now read off the matrix at distinct positions, exactly as P2 requires.

Step 3 – Aggregating into a scalar: P3 forbids non-monotone aggregation of S. The simplest aggregation that is monotone non-decreasing in each S_{ii} and non-increasing in each S_{ij} ($i \neq j$), and class-permutation invariant, is the difference of uniformly-weighted means:

$$\mathrm{CAS}(\mathcal{C}_1, \mathcal{C}_2) \;=\; \underbrace{\frac{1}{c}\sum_{i=1}^{c} S_{ii}}_{\text{diagonal coherence}} \;-\; \underbrace{\frac{1}{c(c-1)}\sum_{i \neq j} S_{ij}}_{\text{off-diagonal confusion}}. \tag{4}$$

CAS is high when same-class circuits are preserved and different-class circuits remain distinct, i.e., precisely when the learner is circuit-robust (Eq. (2)).

Theorem 1 (CAS Soundness)**.** *The Circuit Alignment Score (CAS) defined in Eq.* (4) *satisfies Properties 1 to 3.*

Sound by construction: κ acts on graph topology (P1), CAS decomposes into independently-reportable same-class and cross-class terms (P2), and is linear in S with the correct monotonicity signs (P3). Sec. B.2 provides the full proof along with boundedness, symmetry, and self-similarity.

Decomposability as a diagnostic: Beyond the scalar score, S itself exposes a *class vulnerability profile* (low diagonal entries) and a *confusion topology* (entangled off-diagonal pairs), enabling per-class failure analysis that aggregate accuracy obscures (Sec. 4).

3.3 Consistency: From CAS to OOD Ranking

We lift CAS from a domain comparator to a learner-level diagnostic via a population integral over the domain distribution, prove its Monte Carlo estimate converges with $O(1/M)$ pairwise inversion error, and combine this with a monotonicity condition to recover the true OOD accuracy ranking.

Population CAS: Let $\mathcal{L} = \{\ell_1, \ldots, \ell_N\}$ be a pool of learners trained on a common source distribution, and write $\mathrm{CAS}(\ell; d)$ for the CAS between ℓ's source and domain-d circuits. The *population CAS* is the Lebesgue integral [20]:

$$\overline{\mathrm{CAS}}(\ell) \;=\; \int_{\Omega_D} \mathrm{CAS}(\ell; d)\, dP_D(d), \tag{5}$$

well-defined because $\mathrm{CAS} \in [-1, 1]$ (Proposition 2). This parallels the OOD score $g(\ell)$ in Eq. (1); Assumption 1 connects the two.

Monte Carlo estimator: The population integral is approximated by drawing M domains $\{d_j\}_{j=1}^{M} \overset{\text{i.i.d.}}{\sim} P_D$ and averaging:

$$\widehat{\overline{\mathrm{CAS}}}_M(\ell) \;=\; \frac{1}{M} \sum_{j=1}^{M} \mathrm{CAS}(\ell; d_j). \tag{6}$$

By the strong law of large numbers, $\widehat{\overline{\mathrm{CAS}}}_M(\ell) \to \overline{\mathrm{CAS}}(\ell)$ almost surely as $M \to \infty$. The remaining question is the rate at which this convergence preserves *pairwise rankings* between learners – which is what determines whether finite-sample CAS estimates produce the right ordering.

Pairwise inversion error: For two learners ℓ_i, ℓ_j with $\overline{\mathrm{CAS}}(\ell_i) > \overline{\mathrm{CAS}}(\ell_j)$, define the discrepancy $\delta_{ij}(d) = \mathrm{CAS}(\ell_i; d) - \mathrm{CAS}(\ell_j; d)$, with population mean $m_{ij} = \overline{\mathrm{CAS}}(\ell_i) - \overline{\mathrm{CAS}}(\ell_j) > 0$ and finite variance $\sigma_{ij}^2 = \int_{\Omega_D} (\delta_{ij}(d) - m_{ij})^2\, dP_D(d)$. A pairwise *inversion* occurs when the empirical estimate flips the population ordering. The inversion probability under M samples is:

$$p_{\mathrm{inv}}^{(M)}(\ell_i, \ell_j) \;=\; \Pr\left[\frac{1}{M} \sum_{j=1}^{M} \delta_{ij}(d_j) \le 0 \right], \qquad d_j \overset{\text{i.i.d.}}{\sim} P_D.$$

Theorem 2 (Monte Carlo consistency of CAS rankings)**.** *For any two learners ℓ_i, ℓ_j with $\overline{\mathrm{CAS}}(\ell_i) \neq \overline{\mathrm{CAS}}(\ell_j)$, the inversion probability satisfies:*

$$p_{\mathrm{inv}}^{(M)}(\ell_i, \ell_j) \;\le\; \frac{\sigma_{ij}^2}{M \cdot m_{ij}^2} \;=\; O\!\left(\frac{1}{M}\right).$$

Aggregated over all pairs in a learner pool $\mathcal{L}$ of size N, the total inversion probability $P_{\mathrm{inv}}^{(M)} = \binom{N}{2}^{-1} \sum_{i<j} p_{\mathrm{inv}}^{(M)}(\ell_i, \ell_j) \to 0$ as $M \to \infty$.

The proof (Sec. C.1) is an application of Chebyshev's inequality to the i.i.d. Monte Carlo estimator of the integral m_{ij}. The bound depends on σ_{ij}^2 / m_{ij}^2, which is small whenever the gap between learners' population CAS values is large relative to their domain-wise variability. Sec. C.2 discusses why this gap is typically larger between learners trained by different domain-generalization algorithms than within a single algorithm family, yielding faster convergence on the cross-algorithmic pairs that matter most for model selection.

From CAS rankings to OOD rankings: Theorem 2 establishes that empirical CAS rankings converge to population CAS rankings. To translate this into a guarantee about OOD-accuracy rankings, we need the population CAS to be ordinally aligned with the OOD generalization score:

Assumption 1 (Monotonicity of CAS in OOD accuracy)**.** For all $\ell_i, \ell_j \in \mathcal{L}$,

$$g(\ell_i) > g(\ell_j) \implies \overline{\mathrm{CAS}}(\ell_i) > \overline{\mathrm{CAS}}(\ell_j).$$

This assumption is the connection between the structural premise of Sec. 3.1 (circuit invariance underlies OOD generalization) and the metric defined in Sec. 3.2. We do not prove it from first principles; instead, we verify it empirically across all 48 learners in our pool on PACS, Office-Home, and DomainNet (Sec. 4, and Sec. J in the appendix), where the empirical rank correlation between $\overline{\mathrm{CAS}}$ and g exceeds $\rho_S = 0.88$ on PACS and remains consistent across benchmarks. Corollary 1 provides theoretical support: in the limit of perfect OOD accuracy, OOD robustness structurally implies circuit robustness, and CAS, by construction, is monotone in circuit robustness. Combining Theorem 2 with Assumption 1 yields the predictive claim:

Proposition 1 (Ranking recovery)**.** *Let $\mathcal{L}^* = (\ell_{\pi^*(1)}, \ldots, \ell_{\pi^*(N)})$ be the ground-truth ranking by g, and $\mathcal{L}'_M = (\ell_{\pi'_M(1)}, \ldots, \ell_{\pi'_M(N)})$ the predicted ranking by $\widehat{\mathrm{CAS}}_M$. Under Assumption 1, the Kendall rank correlation [1] between the two satisfies:*

$$\lim_{M\to\infty} \tau(\mathcal{L}^*, \mathcal{L}'_M) = 1 \quad \textit{in probability.}$$

The proof (Sec. C.3) follows from Theorem 2 via the union bound: the probability that any pair is inverted vanishes at $O(1/M)$, so the probability that all $\binom{N}{2}$ pairs are correctly ordered tends to 1, which is precisely $\tau \to 1$.

Summary of the theoretical contribution: Sec. 3.1 derived three necessary conditions for OOD-predictive metrics; Sec. 3.2 constructed CAS to satisfy them (Theorem 1); and this section established that ranking learners by empirical CAS recovers their OOD-accuracy ranking, with $O(1/M)$ inversion error per pair (Theorem 2), under one empirically verifiable monotonicity assumption (Assumption 1). The remaining sections validate this chain experimentally.

4 Experimental Results

Datasets: We evaluate on three benchmarks of increasing complexity: (1) **PACS** [33] covers naturalistic style shifts across 4 domains and 7 classes; (2) **Office-Home** [60] tests discriminative resolution under fine-grained distribution shifts across 4 domains and 65 classes; (3) **DomainNet** [41], the largest and most challenging, contains 6 domains, 345 classes, ∼600K images. Together, they span a wide range of domain shifts, ensuring our findings are not specific to a single data regime.

Evaluation metrics: We evaluate CAS as an OOD predictor using two complementary metrics. **Spearman rank correlation** (ρ_S) [73] our primary metric, measures how well predicted model rankings match true OOD rankings, independent of exact accuracy values. **Mean absolute error** (MAE) measures the deviation between predicted and true OOD accuracy[§] in percentage points and applies only to calibrated predictors (ATC, ALine-D, and ProjNorm). Statistical significance is assessed via permutation tests (10 permutations), with 95% confidence intervals from bootstrap resampling ($1,000$ iterations). We fix random seeds and enforce determinism to ensure circuit divergence reflects domain shift rather than training stochasticity.

Circuit extraction: CAS operates on class-specific circuit graphs and is agnostic to the extractor; any procedure producing class-conditional directed subgraphs suffices. We use Adaptive Circuit Extraction (ACE) for tractability and consistency across architectures (Sec. E).

4.1 CAS as OOD Predictors

This experiment jointly validates three claims: (1) CAS satisfies the monotonicity required by Assumption 1, (2) high CAS values correspond to the circuit-separation regime of Corollary 1, and (3) activation-level baselines exhibit the conflation failure guaranteed by Theorem 5. We compare CAS against CKA [31], SVCCA [47], and RSA [32] across 48 learners spanning 4 architectures (MLP, ResNet50, MobileNetV2, and ViT-B/16), 4 training objectives (ERM, IRM, CORAL, and DANN), and 3 regularization settings (None, Dropout (rate 0.3), and weight decay (10^{-4})), all trained on

[§]True OOD accuracy is taken from the DomainBed repository: https://github.com/facebookresearch/DomainBed

PACS. For each learner, we compute: (i) cross-domain similarity score averaging pairwise similarities across all source domain pairs, and (ii) leave-one-domain-out OOD accuracy by training on three domains and evaluating on the held-out fourth domain. CKA, SVCCA, and RSA are computed on circuit representations obtained via graph embeddings [15], whereas CAS operates directly on ACE-extracted circuit graphs using graph kernels [19, 36, 43]. We visualize these relationships in Fig. 2, where each point represents a learner colored by training objective, with a least-squares trend line overlaid. Additional comparisons are provided in Figs. 12 to 15.

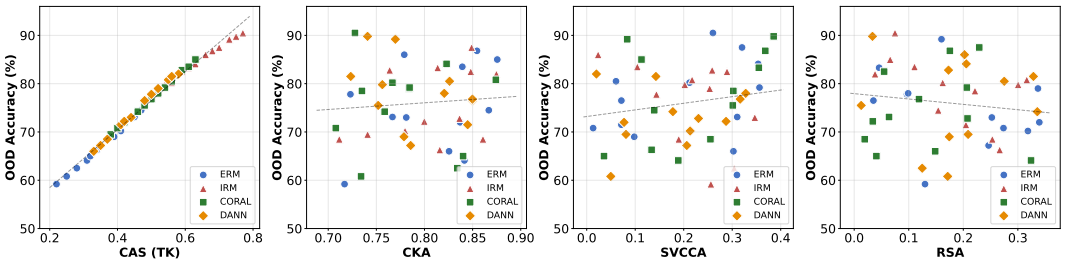


Figure 2: CAS vs OOD accuracy shows a monotonous trend ($\rho_S = 0.93$), whereas CKA ($\rho_S = 0.58$), SVCCA ($\rho_S = 0.23$), and RSA ($\rho_S = 0.14$) show a diffuse scatter, confirming that representational similarity fails to capture the circuit-level signal (Target class: cartoon of the PACS dataset).

CAS exhibits a strong near-monotone relationship with OOD accuracy, with higher circuit alignment corresponding to better OOD performance. Training objectives cluster coherently: ERM in the low-CAS, low-accuracy region, IRM in the high-CAS, high-accuracy region, and CORAL and DANN in between. In contrast, CKA, SVCCA, and RSA show weak positive trends with high variance, often assigning high similarity to poorly performing ERM models. The near-monotone ordering recovered by CAS supports Assumption 1, while high-CAS, high-accuracy IRM learners (red dots in Fig. 2) approach the kernel-separation regime of Corollary 1, indicating domain-invariant computational pathways rather than spuriously aligned representations. The failure of CKA, SVCCA, and RSA is consistent with the conflation predicted by Theorem 5: they cannot distinguish circuit-robust learners from non-robust ones whose circuit drift produces similar activation statistics. Together, these results support CAS as a mechanistically grounded and empirically reliable signal of OOD robustness, while activation-level metrics remain insensitive to structural differences in computation.

4.2 Consistency of CAS Rankings

We evaluate the prediction of Theorem 2, which states that the pairwise rank inversion probability $P_{\text{inv}}^{(M)}$ between two learners with distinct population CAS values decays as $O(1/M)$. In practice, M is fixed by the benchmark: M=3 for PACS and Office-Home, and M=5 for DomainNet, leaving one domain out for OOD accuracy. To validate the theorem, we subsample $M \in \{1, 2, 3\}$ domains for PACS and Office-Home, and $M \in \{1, 2, 3, 4, 5\}$ for DomainNet, using real held-out source domains without any synthetic augmentation. For each M, we repeat CAS computation 10 times, rank the 48 learners by $\widehat{\text{CAS}}_M$, and measure: (i) the fraction of learner-pair rank inversions relative to the maximum-M ranking $\binom{48}{2}$=1128 pairs, and (ii) the Spearman correlation ρ_S between the M-domain CAS ranking and true OOD accuracy ranking. We also overlay the theoretical $O(1/M)$ Chebyshev bound to verify the rate and constant.

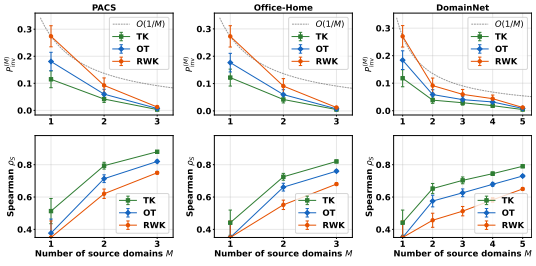


Figure 3: **Consistency of CAS rankings.** Effect of the number of source domains M on ranking stability. **Top:** Pairwise inversion probability $P_{\text{inv}}^{(M)}$ vs. M (mean $\pm$ std over 10 trials) with $O(1/M)$ Chebyshev bound (dashed lines). **Bottom:** Spearman ρ_S between M-domain CAS ranking and true OOD accuracy. TK consistently achieves the highest ρ_S and all methods follow the $O(1/M)$ trend.

The results in Fig. 3 confirm the predicted $O(1/M)$ decay across all benchmarks and methods. At M=1, inversion probabilities are high, up to $\sim$0.27 for RWK, and Spearman correlations are low ($\rho_S \approx 0.39$–0.52), reflecting the large variance of a single-domain estimate. Here, CAS reduces to self-similarity, carrying no cross-domain alignment information, while OOD accuracy reflects



Figure 5: **Domain interpolation.** Images generated by SDXL with a fixed ArtPainting (weight 1.0) and a Photo LoRA with weight variations. At $w{=}[1.0, 0.0]$, the output is purely ArtPainting style; at $w{=}[1.0, 2.0]$, it is fully photorealistic. Semantic content (dog, pose, composition) is preserved throughout, while domain-specific appearance (brush strokes, texture, lighting) transitions smoothly, enabling controlled, continuous domain shift for circuit-level analysis.

single-source domain generalization. As M increases, $P_{\text{inv}}^{(M)}$ drops sharply following the $O(1/M)$ envelope, while ρ_S rises steeply. At the practical operating points $M{=}3$ for PACS and Office-Home, $M{=}5$ for DomainNet, inversion probability is already low and ρ_S approaches convergence. TK achieves the highest ranking quality ($\rho_S \approx 0.89, 0.81, 0.79$), followed by OT ($0.82, 0.76, 0.74$) and RWK ($0.75, 0.69, 0.64$) on PACS, Office-Home, and DomainNet, respectively. Overall, the empirical $P_{\text{inv}}^{(M)}$ curves lie at or below the theoretical $O(1/M)$ Chebyshev bound ($M \geq 2$ for RWK), confirming that CAS-based OOD ranking errors decrease as the number of domains increases.

4.3 Evolution of CAS with Domain Perturbations

This experiment validates whether CAS reflects domain proximity at the circuit level by recovering a ground-truth perturbation ordering under controlled, monotonically increasing domain shift. For $w < w'$, the intermediate domain $D_2[w']$ should induce a stronger perturbation of D_1 than $D_2[w]$, i.e., $D_2[w] \prec_{D_1} D_2[w']$ (Definition 1), and a faithful metric must assign a higher score to the weaker perturbation. Theorem 7 formalizes the converse risk: any aggregation $\Psi(S)$ that is not coordinate-wise monotone can invert these rankings. CAS avoids this by construction, as it is monotone increasing on the diagonal and decreasing off the diagonal of S, consistent with Definition 1. We fix $D_1 =$ ArtPainting and construct intermediate domains $\{D_2[w]\}$ by interpolating between ArtPainting-style and Photo-style LoRAs at inference time with SDXL [45], with the Photo adapter weight $w \in [0, 2]$. This performs domain interpolation in *parameter space*, preserving semantic content while smoothly varying domain-specific appearance (Fig. 5), unlike input-space mixup [9, 69]. For each intermediate domain, we train MLP [46], ResNet50 [23], MobileNetv2 [49], and ViT-B/16 [14] under identical initialization and data distribution, extract class-wise circuits via ACE, and compute the similarity matrix S between D_1 and $D_2[w]$ using three graph kernels.

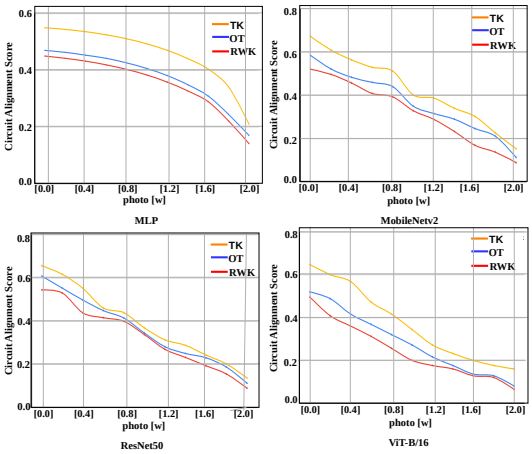


Figure 4: **CAS under controlled domain interpolation via LoRA.** CAS values are monotonously decreasing with increasing domain perturbations.

As shown in Fig. 4, CAS decreases monotonically with increasing w across all four architectures and three graph kernels. Increasing domain shift induces consistent circuit divergence, with no reversals in ordering, plateau artefacts, or kernel- or architecture-specific anomalies across $w \in [0, 2]$. This monotonicity supports the perturbation ordering $D_2[w] \prec_{D_1} D_2[w']$ for all $w < w'$, consistent with Sec. 3.3, and suggests that domain proximity is reflected in circuit space. Although architectures exhibit distinct slopes, reflecting their inductive biases and representational capacities, the *relative ordering* of domain perturbations is preserved across all configurations. These results establish CAS as a stable measure of domain-induced circuit reorganization across the tested architectures, rather than an artefact of a particular architectural prior.

4.4 Comparison with OOD Accuracy Prediction Methods

We evaluate whether CAS can predict OOD accuracy without target-domain labels by comparing against four families of OOD predictors following standard protocols (Sec. D.1). Table 2 reports the

head-to-head comparison. Confidence-based methods [18, 21, 59] treat high softmax confidence as a proxy for correctness but can fail under large style shifts where models remain confidently wrong. Complexity-based measures [26] characterize the model's capacity to generalize but were designed for in-distribution gaps, explaining their weak transfer to OOD settings (e.g. $\rho_s < 0.5$ and goes negative for spectral complexity). Distribution-shift-based methods [72, 59] measure the magnitude of distributional change but cannot distinguish benign shifts (background changes, circuits intact) from harmful ones (discriminative features altered).

Table 2: **Comparison of OOD accuracy prediction methods on DomainBed Benchmarks:** CAS achieves state-of-the-art performance across all three benchmarks without any training. †ALine-D requires the full 48-model pool; all other methods are single-model. Whereas ProjNorm requires two extra retraining runs per fold. Similarly, DoC trains a simple regression, HΔH trains a domain classifier, and CSS performs post-deployment correction for accurate OOD prediction.

Family	Method	Input Modality	Needs target?	PACS			Office-Home			DomainNet		
				P_s	R^2	MAE	P_s	R^2	MAE	P_s	R^2	MAE
Confidence	ATC [18]	Softmax outputs	Unlabeled	0.58	0.64	5.31	0.53	0.47	5.89	0.47	0.42	6.31
	DoC† [21]	Softmax outputs	Unlabeled	0.63	0.49	5.26	0.58	0.51	5.54	0.50	0.44	6.10
	Entropy-Target [59]	Softmax outputs	Unlabeled	0.66	0.51	5.14	0.61	0.54	5.33	0.54	0.48	5.82
	Entropy-Source [59]	Softmax outputs	Source only	0.61	0.47	5.88	0.56	0.49	5.68	0.49	0.44	6.17
	Source Val. Acc. [59]	Labels (source)	Source only	0.73	0.59	4.96	0.68	0.60	4.85	0.62	0.55	5.27
	Mixup Criterion [59]	Softmax outputs	Unlabeled	0.27	0.34	8.05	0.25	0.24	7.80	0.21	0.20	8.09
Complexity	Sharpness [26]	Weights + loss	Source only	0.14	0.17	9.42	0.14	0.15	8.56	0.13	0.12	8.79
	PAC-Bayes [26]	Weights + loss	Source only	0.44	0.42	6.81	0.41	0.37	6.73	0.35	0.32	7.10
	Path Norm [26]	Model weights	Source only	0.37	0.29	7.35	0.34	0.31	7.17	0.29	0.27	7.51
	Frobenius dist. [26]	Model weights	Source only	0.21	0.22	8.12	0.20	0.19	8.18	0.16	0.16	8.43
	Spectral comp. [26]	Model weights	Source only	-0.19	0.05	9.04	-0.15	0.01	10.69	-0.16	0.01	10.76
Gradients	Gradient Noise [26]	Gradients	Source only	0.34	0.27	7.28	0.32	0.29	7.36	0.27	0.25	7.68
	Jacobian norm [59]	Gradients	Unlabeled	0.02	0.04	9.47	0.02	0.04	9.24	0.02	0.04	9.45
Agreement	ALine-D† [6]	$\geq$3 models	Unlabeled	0.72	0.57	4.85	0.67	0.59	4.92	0.60	0.53	5.40
Activations	CKA [31]	Activations	Source only	0.58	0.44	7.18	0.53	0.47	5.85	0.47	0.42	6.29
	SVCCA [47]	Activations	Source only	0.23	0.11	8.04	0.22	0.21	8.05	0.18	0.18	8.32
	RSA [32]	Activations	Source only	0.14	0.06	7.63	0.14	0.14	8.61	0.107	0.11	8.84
Distribution shift	ProjNorm† [72]	Model weights	Unlabeled	0.65	0.50	5.72	0.59	0.52	5.47	0.52	0.46	5.96
	Fisher [59]	Gradients (src + tgt)	Unlabeled	0.21	0.10	8.23	0.19	0.18	8.26	0.16	0.16	8.47
	MMD [59]	Activations	Unlabeled	0.27	0.31	8.05	0.27	0.25	7.71	0.23	0.22	7.98
	HΔH† [59]	Activations	Unlabeled	0.15	0.09	8.86	0.13	0.13	8.68	0.10	0.11	8.89
Circuit graphs	DDB (ViT) [42]	Circuit graphs	Unlabeled	0.89	0.86	3.02	0.79	0.71	3.89	0.63	0.64	4.48
	CSS (ViT)† [42]	Circuit graphs	Unlabeled	**0.98**	**0.91**	**2.09**	0.81	0.68	5.34	0.70	0.60	9.97
	CAS (ViT)	Circuit graphs	Source only	<u>0.91</u>	<u>0.88</u>	<u>2.11</u>	**0.83**	**0.73**	**3.78**	<u>0.74</u>	<u>0.65</u>	<u>4.37</u>
	CAS (ours)	Circuit graphs	Source only	0.88	0.84	2.14	<u>0.82</u>	<u>0.71</u>	<u>3.87</u>	**0.79**	**0.69**	**4.08**

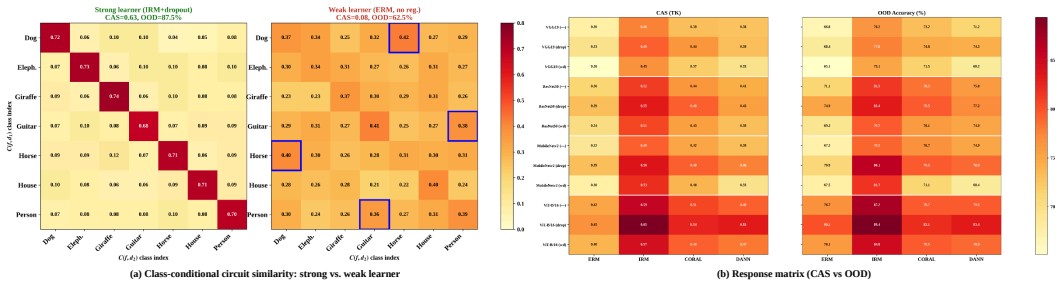
CAS achieves the strongest overall performance ($\rho_S = 0.88$, MAE $= 2.14\%$), outperforming ALine-D ($\rho_S = 0.72$, MAE $= 4.85\%$), ATC ($\rho_S = 0.58$) and ProjNorm ($\rho_S = 0.65$, MAE $= 5.72\%$). Unlike these methods, CAS measures whether the causal computational pathways driving class predictions are preserved across domains. A model may be confident, agree with other models, exhibit low distribution shift, yet rely on different circuits across domains – a failure mode visible at the circuit level. This makes CAS complementary to existing OOD predictors, rather than a refinement of the three families. Although CSS [42] outperforms CAS in its post-deployment setting, it relies on validation data covering OOD scenarios for calibration, unlike our source-only OOD prediction setting. We further evaluate CAS under increasing distribution shifts by comparing predicted OOD generalization with true test accuracy.

Table 3: **Per-target Spearman ρ_S on PACS.** CAS consistently outperforms all baselines across all four target domains. The advantage is largest on Sketch ($+0.22$ over next-best), the hardest target.

Method	Tgt: Photo	Tgt: Art	Tgt: Cartoon	Tgt: Sketch	Mean
ATC [18]	0.65	0.54	0.62	0.51	0.58
ProjNorm [72]	0.72	0.61	0.68	0.55	0.64
ALine-D [6]	0.79	0.70	0.78	0.61	0.72
CKA [31]	0.81	0.48	0.58	0.45	0.58
SVCCA [47]	0.28	0.21	0.21	0.22	0.23
RSA [32]	0.12	0.14	0.19	0.11	0.14
CAS (ours)	**0.91**	**0.86**	**0.93**	**0.83**	**0.88**

To further understand CAS, we report per-target results in (Table 3). On PACS, CAS shows the largest advantage on Sketch (0.83 vs 0.61), the hardest target, where stylistic shifts degrade output-based predictors but circuit structure remains informative. CAS measures the model's computation rather than its outputs, making it less sensitive to calibration errors and pseudo-label noise under large stylistic shifts. Per-target results on OfficeHome and DomainNet are provided in Table 8.

Downstream Impact: Beyond correlating with OOD accuracy, CAS enables practical model selection without target-domain supervision. On PACS cartoon, CAS achieves 100% TopK-recall by K=5, identifying the best models for evaluation under a limited evaluation budget (Fig. 7). CAS



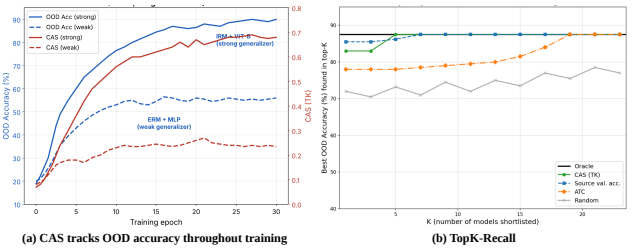
(a) Class-conditional circuit similarity: strong vs. weak learner (b) Response matrix (CAS vs OOD)

Figure 6: **Class-conditional circuit similarity and Response Matrix:** (a) CAS cleanly separates a strong learner from a weak one, with blue boxes marking the Dog–Horse and Guitar–Person confusions unique to CAS. (b) CAS mirrors true OOD accuracy almost exactly ($\rho_S = 0.93$), recovering the full performance surface without any target-domain labels.

also tracks OOD accuracy during training, with both increasing for strong generalizers (IRM + ViT-B/16) and plateauing at low values for weak ones (ERM + MLP), suggesting its utility for checkpoint selection. Furthermore, a response-matrix experiment spanning 4 architectures $\times$ 4 DG algorithms $\times$ 3 regularizations (48 configurations) produces near-identical CAS and OOD accuracy heatmaps (ρ_S=0.93). CAS correctly ranks architectures under a fixed algorithm (ρ_S=0.89–0.96) and algorithm-regularization choices under a fixed architecture (ρ_S =0.91–0.95) Fig. 6). For example, CAS identifies ViT-B/16 + IRM + dropout as the strongest configuration (CAS=0.65, OOD=89.4%), while VGG19 + IRM + dropout as a lower ceiling (CAS=0.49, OOD=77.8%). Together, these results establish CAS not merely as a correlational diagnostic but as a practical tool for selecting models, checkpoints, and training strategies without target-domain supervision.

5 Conclusion and Discussions

We introduced Circuit Alignment Score (CAS), a weight-derived OOD predictor justified by three theoretical layers: necessary structural conditions derived via impossibility theorems (Sec. 3.1), a construction of CAS in which each step is forced by one of the conditions (Theorem 1), and a Monte Carlo consistency guarantee with $O(1/M)$ pairwise inversion error (Theorem 2). Across 48 learners on three benchmarks, CAS attains Spearman $\rho_S \geq 0.77$ in every setting, exceeding all weight-only and target-data-using baselines. The framework is extractor-agnostic, *i.e.*, alternative circuit extractors [56, 57] can plug into the same metric, while the class-conditional similarity matrix exposes per-class vulnerability and confusion topology beyond the scalar score.



(a) CAS tracks OOD accuracy throughout training (b) TopK-Recall

Figure 7: **CAS as a practical tool for model selection:** (a) During training, CAS and OOD accuracy co-evolve for a strong generalizer but plateau at low values for a weak one, demonstrating its utility for checkpoint selection. (b) CAS recovers the oracle-best model by K=5, compared with K=7 for source-validation accuracy and K=18 for ATC, showing its efficiency in evaluating models under limited budgets.

Limitations: The monotonicity condition (Assumption 1) bridging $\overline{\text{CAS}}$ and OOD accuracy is empirically verified rather than proved in the finite-accuracy regime; the limit-case corollary establishes only the endpoint. The framework is currently specialized to classification with bounded class counts, and ACE requires source-domain adapter training, lighter than retraining-based baselines but heavier than purely weights-only ones.

Societal impact: A reliable weights-only OOD predictor benefits settings where target data is unavailable, such as clinical pre-deployment, low-resource, and federated systems, but invites overreliance: high CAS evidences circuit-level invariance, not correctness on individual predictions. Earlier identification of non-robust models can reduce the risk of failures affecting patients or users. We recommend CAS as a model-selection aid alongside conventional validation, not as a substitute.

Acknowledgement

This research was carried out with support from the SUKIDI PID2024-157778OB-I00 grants from the Spanish Ministry of Science and Innovation and the PhD Scholarship from AGAUR (FI-SDUR: 2023 FISDU 00394). The authors gratefully acknowledge NVIDIA Corporation for support through the NVIDIA Academic Grant Program, which provided computational resources for this research.

References

[1] Hervé Abdi et al. The kendall rank correlation coefficient. *Encyclopedia of measurement and statistics*, 2:508–510, 2007.

[2] Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R Varshney. Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*, 2021.

[3] Mokhtar Z Alaya, Maxime Berar, Gilles Gasso, and Alain Rakotomamonjy. Screening sinkhorn algorithm for regularized optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.

[4] Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, et al. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 6, 2025.

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

[6] Christina Baek, Yiding Jiang, Aditi Raghunathan, and Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. In *NeurIPS*, 2022.

[7] Lorenzo Brigato and Stavroula Mougiakakou. Can the training loss be predictive for out-of-distribution generalization? *OpenReview*, 2025.

[8] Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: circuits. *Distill*, 5(3):e24, 2020.

[9] Meng Cao and Songcan Chen. Mixup-induced domain extrapolation for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11168–11176, 2024.

[10] Yongqiang Chen, Wei Huang, Kaiwen Zhou, Yatao Bian, Bo Han, and James Cheng. Understanding and improving feature learning for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 36:68221–68275, 2023.

[11] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

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

[13] Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. *Advances in Neural Information Processing Systems*, 34:1556–1568, 2021.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[15] Anjan Dutta and Hichem Sahbi. Stochastic graphlet embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 30(8):2369–2382, 2018.

[16] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

[17] 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.

[18] Saurabh Garg, Sivaraman Balakrishnan, Zachary C Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *ICLR*, 2022.

[19] Benoit Gaüzere, Luc Brun, and Didier Villemin. Two new graphs kernels in chemoinformatics. *Pattern Recognition Letters*, 33(15):2038–2047, 2012.

[20] Russell A Gordon. *The integrals of lebesgue, denjoy, perron, and henstock*, volume 4. American Mathematical Soc., 1994.

[21] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1114–1124. IEEE, 2021.

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

[23] 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*, pages 770–778, 2016.

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

[25] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. In *International Conference on Learning Representations*, 2019.

[26] Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

[27] Jivat Neet Kaur, Emre Kiciman, and Amit Sharma. Modeling the data-generating process is necessary for out-of-distribution generalization. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Kenji Kawaguchi. Deep learning without poor local minima. *Advances in neural information processing systems*, 29, 2016.

[29] Youngkeun Kim, Han-Shin Youn, and Myoung Hoon Lee. Spectral normalization and svd-enhanced deep kernel learning gaussian process for loss prediction under out-of-distribution conditions in dab converters. *IEEE Access*, 2026.

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

[31] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019.

[32] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

[33] 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*, pages 5542–5550, 2017.

[34] Sanae Lotfi, Marc Finzi, Sanyam Kapoor, Andres Potapczynski, Micah Goldblum, and Andrew G Wilson. Pac-bayes compression bounds so tight that they can explain generalization. *Advances in Neural Information Processing Systems*, 35:31459–31473, 2022.

[35] Coenraad Mouton, Marthinus Wilhelmus Theunissen, and Marelie H Davel. Input margins can predict generalization too. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14379–14387, 2024.

[36] Giannis Nikolentzos and Michalis Vazirgiannis. Random walk graph neural networks. *Advances in Neural Information Processing Systems*, 33:16211–16222, 2020.

[37] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. An overview of early vision in inceptionv1. *Distill*, 2020. https://distill.pub/2020/circuits/early-vision.

[38] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in.

[39] Vedant Palit, Rohan Pandey, Aryaman Arora, and Paul Pu Liang. Towards vision-language mechanistic interpretability: A causal tracing tool for blip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2856–2861, 2023.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

[42] Yunxiang Peng, Mengmeng Ma, Ziyu Yao, and Xi Peng. Inside-Out: Measuring Generalization in Vision Transformers Through Inner Workings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[43] Hermina Petric Maretic, Mireille El Gheche, Giovanni Chierchia, and Pascal Frossard. Got: an optimal transport framework for graph comparison. *Advances in Neural Information Processing Systems*, 32, 2019.

[44] Antoine Picard-Weibel, Eugenio Clerico, Roman Moscoviz, and Benjamin Guedj. How good is pac-bayes at explaining generalisation? *arXiv preprint arXiv:2503.08231*, 2025.

[45] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588, 2009.

[47] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[48] Emmanuelle Salin, Badreddine Farah, Stéphane Ayache, and Benoit Favre. Are vision-language transformers learning multimodal representations? a probing perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11248–11257, 2022.

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

[50] John G Saw, Mark CK Yang, and Tse Chin Mo. Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132, 1984.

[51] Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. Multimodal neurons in pretrained text-only transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2862–2867, 2023.

[52] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.

[53] Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis. Grakel: A graph kernel library in python. *Journal of Machine Learning Research*, 21(54):1–5, 2020.

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

[56] Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 407–416, 2024.

[57] Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos G Derpanis. Universal sparse autoencoders: Interpretable cross-model concept alignment. In *Forty-second International Conference on Machine Learning*, 2025.

[58] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[59] Ramakrishna Vedantam, David Lopez-Paz, and David Schwab. An empirical investigation of domain generalization with empirical risk minimizers. *Advances in neural information processing systems*, 34:28131–28143, 2021.

[60] 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*, pages 5018–5027, 2017.

[61] Cédric Villani. The wasserstein distances. In *Optimal transport: old and new*, pages 93–111. Springer, 2009.

[62] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[63] Chelsea Voss, Nick Cammarata, Gabriel Goh, Michael Petrov, Ludwig Schubert, Ben Egan, Swee Kiat Lim, and Chris Olah. Visualizing weights. *Distill*, 2021. https://distill.pub/2020/circuits/visualizing-weights.

[64] Thuy Vu, Shahram Khadivi, Dinh Phung, and Gholamreza Haffari. Domain generalisation of nmt: Fusing adapters with leave-one-domain-out training. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 582–588, 2022.

[65] Haoliang Wang, Chen Zhao, and Feng Chen. Madod: Generalizing ood detection to unseen domains via g-invariance meta-learning. In *2024 IEEE International Conference on Big Data (BigData)*, pages 1134–1143, 2024.

[66] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.

[67] Ruoyu Wang, Mingyang Yi, Zhitang Chen, and Shengyu Zhu. Out-of-distribution generalization with causal invariant transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 375–385, 2022.

[68] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227, 2023.

[69] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6502–6509, 2020.

[70] Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:23519–23531, 2021.

[71] Haotian Ye, Chuanlong Xie, Yue Liu, and Zhenguo Li. Out-of-distribution generalization analysis via influence function. *arXiv preprint arXiv:2101.08521*, 2021.

[72] Yaodong Yu, Zitong Yang, Alexander Wei, Yi Ma, and Jacob Steinhardt. Predicting out-of-distribution error with the projection norm. In *ICML*, 2022.

[73] Jerrold H Zar. Spearman rank correlation. *Encyclopedia of biostatistics*, 7, 2005.

[74] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024.

A Motivation behind CAS Properties and Impossibility Theorems

A.1 Circuit-level robustness as a structural proxy

Theorem 3 (Diagonal preservation under OOD robustness)**.** *Let f be a learner achieving perfect OOD accuracy across domains,* $\mathrm{Acc}(f; d) = 1$ *for P_D-almost every $d \in \Omega_D$. Suppose further that the class-conditional input distributions $P_{x|y=i,d}$ have non-degenerate support overlap across domains: for every pair d_1, d_2 and every class i, there exists x with positive density under both $P_{x|y=i,d_1}$ and $P_{x|y=i,d_2}$. Then for every class i and every pair of domains d_1, d_2, the circuits $\mathcal{C}^{(i)}(f, d_1)$ and $\mathcal{C}^{(i)}(f, d_2)$ are* causally equivalent*: they implement the same function on the shared support, modulo functionally redundant rerouting.*

Proof. Fix class i and domains d_1, d_2. By definition, $\mathcal{C}^{(i)}(f, d)$ is the minimal causally-grounded subgraph such that $f(x) = i$ for all x in the class-i support of d. Since f achieves perfect accuracy, this subgraph correctly classifies every class-i input from domain d.

Let $x \in \mathrm{supp}(P_{x|y=i,d_1}) \cap \mathrm{supp}(P_{x|y=i,d_2})$, a class-$i$ input present under both domains, which exists by the overlap assumption. The forward computation of f on input x is determined entirely by f's weights, not by which domain x was drawn from; the path f takes through its computation is a property of (f, x), not (f, x, d). Hence $\mathcal{C}^{(i)}(f, d_1)$ and $\mathcal{C}^{(i)}(f, d_2)$ both contain the causal path through f for input x.

Aggregating over all x in the shared support yields a common subgraph $\mathcal{C}^{(i)}_{\mathrm{shared}}$ contained in both $\mathcal{C}^{(i)}(f, d_1)$ and $\mathcal{C}^{(i)}(f, d_2)$. The remaining portions of each circuit handle inputs in the domain-specific supports $\mathrm{supp}(P_{x|y=i,d_k}) \setminus \mathrm{supp}(P_{x|y=i,d_{3-k}})$. Since f achieves perfect accuracy on these inputs as well, the domain-specific portions implement the same function (output i on class-i inputs) and differ only in which intermediate nodes route the computation, i.e., they are functionally redundant rerouting in the sense of Theorem 5.

Therefore $\mathcal{C}^{(i)}(f, d_1)$ and $\mathcal{C}^{(i)}(f, d_2)$ share a non-trivial common subgraph $\mathcal{C}^{(i)}_{\mathrm{shared}}$ and differ only by functionally redundant rerouting outside it. Under any graph kernel κ that respects functional equivalence, $\kappa(\mathcal{C}^{(i)}(f, d_1), \mathcal{C}^{(i)}(f, d_2))$ is bounded below by a positive constant determined by the size of $\mathcal{C}^{(i)}_{\mathrm{shared}}$. □

Theorem 4 (Off-diagonal distinctness under OOD robustness)**.** *Let f be a learner achieving perfect OOD accuracy. Then for every pair of domains d_1, d_2 and every pair of distinct classes $i \neq j$, the circuits $\mathcal{C}^{(i)}(f, d_1)$ and $\mathcal{C}^{(j)}(f, d_2)$ are functionally distinguishable: they implement maps with disjoint output labels on their respective supports.*

Proof. By definition, $\mathcal{C}^{(i)}(f, d_1)$ implements a map $g_i^{d_1}$ such that $g_i^{d_1}(x) = i$ for all $x \in \mathrm{supp}(P_{x|y=i,d_1})$. Similarly, $\mathcal{C}^{(j)}(f, d_2)$ implements $g_j^{d_2}$ with $g_j^{d_2}(x) = j$ for $x \in \mathrm{supp}(P_{x|y=j,d_2})$. Since $i \neq j$, the output labels of these maps are disjoint, and the circuits cannot be functionally identical.

Under any graph kernel κ that respects functional equivalence (which all standard kernels: random walk, Weisfeiler-Lehman, treelet do, since functionally distinct circuits have non-isomorphic causal subgraphs in the generic case), $\kappa(\mathcal{C}^{(i)}(f, d_1), \mathcal{C}^{(j)}(f, d_2))$ is bounded above by a constant strictly less than the self-similarity bound, with the gap determined by the structural difference between the circuits implementing class-i and class-j predictions. □

Robust and non-robust learners. We adopt a circuit-level definition of robustness, which we will connect to OOD accuracy in Sec. 3.3. Let $\kappa : \mathcal{C} \times \mathcal{C} \to [0, 1]$ be a base similarity over individual circuits (e.g., a graph kernel). A learner f is *circuit-robust* between domains D_1, D_2 if

$$\kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(i)}) \geq \alpha \quad \text{for all } i, \qquad \kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(j)}) \leq \beta \quad \text{for all } i \neq j,$$

for some $0 \leq \beta < \alpha \leq 1$ (same-class circuits preserved, different-class circuits remain distinct). A learner is *non-robust* if at least one of these inequalities is reversed: either some $\kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(i)}) < \alpha$ (*circuit drift*) or some $\kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(j)}) > \beta$ for $i \neq j$ (*circuit entanglement*).

This definition makes precise the two failure modes of distribution shift identified in the introduction. We now establish the properties any predictive metric must satisfy to separate these cases.

Corollary 1 (OOD robustness implies circuit robustness in the limit)**.** *Let f achieve perfect OOD accuracy on P_D. Then there exist constants $0 \leq \beta^* < \alpha^* \leq 1$ — depending on f and on the kernel κ but not on the domains — such that for every pair of domains $d_1, d_2 \in \Omega_D$ (modulo a P_D-null set):*

$$\kappa(\mathcal{C}^{(i)}(f, d_1), \mathcal{C}^{(i)}(f, d_2)) \geq \alpha^* \ \ \forall i, \qquad \kappa(\mathcal{C}^{(i)}(f, d_1), \mathcal{C}^{(j)}(f, d_2)) \leq \beta^* \ \ \forall i \neq j.$$

That is, f is circuit-robust in the sense of Eq. (2) *with $\alpha = \alpha^*, \beta = \beta^*$.*

Proof. Combine Theorem 3 and Theorem 4. The lower bound α^* is the kernel value of the shared substructure $\mathcal{C}^{(i)}_{\text{shared}}$ from Theorem 3, taken as a uniform infimum over classes (positive since each class has a non-empty shared substructure). The upper bound β^* is the supremum of cross-class kernel values from Theorem 4 (strictly less than the diagonal lower bound since the class-distinguishability gap is uniform). □

A.2 Structural Sensitivity

Theorem 5 (Activation-factoring metrics conflate robust and rerouted learners)**.** *Let μ be a metric of the form $\mu(\mathcal{C}_1, \mathcal{C}_2) = \tilde{\mu}(\phi(\mathcal{C}_1), \phi(\mathcal{C}_2))$ for some embedding $\phi : \mathcal{C} \to \mathbb{R}^{d \times n}$ into activation space and some function $\tilde{\mu}$. Then there exist a reference family $\mathcal{C}_1$, a circuit-robust family $\mathcal{C}_2$ satisfying Eq.* (2) *with $\alpha = 1, \beta = 0$, and a non-robust family $\mathcal{C}_3$ exhibiting circuit drift, such that*

$$\mu(\mathcal{C}_1, \mathcal{C}_2) = \mu(\mathcal{C}_1, \mathcal{C}_3).$$

Proof. Activation embeddings ϕ map circuits to their induced feature representations, which depend only on the input–output behavior over the source distribution, not on the internal computational pathway. Concretely, for any circuit $\mathcal{C}$, $\phi(\mathcal{C})$ is determined by the function $f_{\mathcal{C}} : x \mapsto z_{\text{penult}}(x)$ that $\mathcal{C}$ induces.

Fix any reference family $\mathcal{C}_1$. Let $\mathcal{C}_2 = \mathcal{C}_1$ (a trivially robust family with $\alpha = 1, \beta = 0$). Construct $\mathcal{C}_3$ as a structural rerouting of $\mathcal{C}_2$: for each class i, replace $\mathcal{C}_2^{(i)}$ with a circuit $\mathcal{C}_3^{(i)}$ whose nodes and edges are permuted such that $f_{\mathcal{C}_3^{(i)}}(x) = f_{\mathcal{C}_2^{(i)}}(x)$ for all x in the source distribution, but the underlying graph topology differs (e.g., by routing through a parallel set of nodes implementing the same function via the universal approximation property of MLPs [24]). Such rerouted circuits exist whenever the model has functional redundancy, which holds in any over-parameterized network [28].

By construction, $\phi(\mathcal{C}_3) = \phi(\mathcal{C}_2)$ pointwise, since ϕ depends only on $f_{\mathcal{C}}$. Therefore $\tilde{\mu}(\phi(\mathcal{C}_1), \phi(\mathcal{C}_2)) = \tilde{\mu}(\phi(\mathcal{C}_1), \phi(\mathcal{C}_3))$, giving $\mu(\mathcal{C}_1, \mathcal{C}_2) = \mu(\mathcal{C}_1, \mathcal{C}_3)$.

However, $\mathcal{C}_3$ is non-robust: under any domain shift that perturbs the parallel rerouted nodes (which differ from the original circuit's nodes), $\kappa(\mathcal{C}_2^{(i)}, \mathcal{C}_3^{(i)})$ can be made arbitrarily small while ϕ remains unchanged on the source. Hence μ assigns identical values to a robust and a non-robust learner. □

This impossibility motivates the following requirement.

Property 1 (Structural Sensitivity)**.** A metric μ is *structurally sensitive* if for any embedding ϕ into activation space, μ does not factor as $\mu(\mathcal{C}_1, \mathcal{C}_2) = \tilde{\mu}(\phi(\mathcal{C}_1), \phi(\mathcal{C}_2))$.

CKA, SVCCA, and RSA each factor through an activation embedding by construction (kernel of penultimate features, SVD-projected features, and pairwise feature-distance matrices, respectively), and hence violate Property 1. CAS does not, since graph kernels operate on $\mathcal{C}$'s topology directly.

A.3 Class-Conditional Resolution

Theorem 6 (Aggregating metrics conflate preservation and entanglement)**.** *Let μ be a metric of the form $\mu(\mathcal{C}_1, \mathcal{C}_2) = \Psi(S)$ where $S_{ij} = \kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(j)})$ and $\Psi : \mathbb{R}^{c \times c} \to \mathbb{R}$ is any function symmetric in its inputs (i.e., invariant under permutations of the entries of S). Then there exist a circuit-robust family $\mathcal{C}_2$ and an entanglement-failure family $\mathcal{C}_3$ relative to a fixed $\mathcal{C}_1$ such that*

$$\mu(\mathcal{C}_1, \mathcal{C}_2) = \mu(\mathcal{C}_1, \mathcal{C}_3),$$

yet $\mathcal{C}_2$ satisfies Eq. (2) *and $\mathcal{C}_3$ violates the off-diagonal condition.*

Proof. Let $c \geq 2$ and choose α, β with $0 \leq \beta < \alpha \leq 1$. Let $\rho = (\alpha + (c-1)\beta)/c$ be a target row-mean similarity. Construct $\mathcal{C}_2$ such that the similarity matrix $S^{(2)}_{ij} = \kappa(\mathcal{C}^{(i)}_1, \mathcal{C}^{(j)}_2)$ satisfies

$$S^{(2)}_{ii} = \alpha, \qquad S^{(2)}_{ij} = \beta \ \ (i \neq j),$$

i.e., $\mathcal{C}_2$ is circuit-robust per Eq. (2).

Construct $\mathcal{C}_3$ such that all entries of $S^{(3)}$ equal the constant ρ:

$$S^{(3)}_{ij} = \rho \quad \text{for all } i, j.$$

This violates Eq. (2)'s off-diagonal condition: $S^{(3)}_{ij} = \rho > \beta$ for $i \neq j$ (since $\rho - \beta = (\alpha - \beta)/c > 0$), so $\mathcal{C}_3$ is non-robust by entanglement. Moreover, $\mathcal{C}_3$'s diagonal $S^{(3)}_{ii} = \rho < \alpha$ also fails the diagonal condition, so $\mathcal{C}_3$ is unambiguously non-robust.

The matrices $S^{(2)}$ and $S^{(3)}$ have the same multiset of entries: $S^{(2)}$ contains c copies of α and $c(c-1)$ copies of β, while $S^{(3)}$ contains c^2 copies of ρ. These multisets differ, but if we instead construct $S^{(3)}$ by permuting the entries of $S^{(2)}$ to place a value of α at an off-diagonal position and a value of β on the diagonal (e.g., swap entries $(1,1)$ and $(1,2)$), the resulting matrix has identical multiset to $S^{(2)}$ but encodes entanglement of class 1 with class 2 and drift of class 1. Since Ψ is invariant under permutations of entries, $\Psi(S^{(2)}) = \Psi(S^{(3)})$, yet $\mathcal{C}_3$ is non-robust while $\mathcal{C}_2$ is robust. □

This impossibility motivates the following requirement.

Property 2 (Class-Conditional Resolution)**.** A metric μ has *class-conditional resolution* if it depends on the position of entries in the similarity matrix S, distinguishing diagonal entries S_{ii} from off-diagonal entries S_{ij} $(i \neq j)$. Equivalently, $\mu = \Psi(S)$ where Ψ is not invariant under arbitrary permutations of the entries of S.

CKA, SVCCA, and RSA aggregate features without per-class structure; the class-conditional similarity matrix S is not even formed, let alone its diagonal/off-diagonal structure preserved. They violate Property 2. CAS is constructed precisely as the gap between mean-diagonal and mean-off-diagonal of S (Sec. 3.2), making it position-aware.

A.4 Semantic Consistency

The previous two properties concern *what* a metric can detect. The third concerns *how* the metric responds to graded perturbation: stronger domain shifts should yield smaller metric values. Without this, the metric's numerical value carries no ordinal information about shift magnitude, and any ranking of learners or domains derived from it is incoherent.

We formalize the strength of perturbation via a partial order on circuit configurations.

Definition 1 (Circuit perturbation order)**.** Let $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ be circuit families corresponding to domains D_1, D_2, D_3. We say D_2 induces a stronger perturbation of D_1 than D_3, written $D_3 \prec_{D_1} D_2$, if for every class i:

$$\kappa(\mathcal{C}^{(i)}_1, \mathcal{C}^{(i)}_2) \leq \kappa(\mathcal{C}^{(i)}_1, \mathcal{C}^{(i)}_3), \tag{7}$$

$$\frac{1}{c-1}\sum_{j \neq i} \kappa(\mathcal{C}^{(i)}_1, \mathcal{C}^{(j)}_2) \geq \frac{1}{c-1}\sum_{j \neq i} \kappa(\mathcal{C}^{(i)}_1, \mathcal{C}^{(j)}_3). \tag{8}$$

That is, the stronger perturbation simultaneously reduces same-class circuit similarity and increases mean cross-class circuit similarity for every class.

This partial order captures the two ways perturbation can intensify: increased drift on the diagonal, increased entanglement off the diagonal, and requires both to hold for one perturbation to dominate another. It is well-defined whenever κ is bounded.

Theorem 7 (Non-monotone metrics invert perturbation rankings)**.** *Let μ be a metric of the form $\mu(\mathcal{C}_1, \mathcal{C}_2) = \Psi(S)$ where $S_{ij} = \kappa(\mathcal{C}^{(i)}_1, \mathcal{C}^{(j)}_2)$. Suppose Ψ is not coordinate-wise monotone in the sense required by Definition 1: i.e., there exist S, S' with $S'_{ii} \leq S_{ii}$ for all i and $S'_{ij} \geq S_{ij}$ for all $i \neq j$, but $\Psi(S') > \Psi(S)$. Then there exist circuit families $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ with $D_3 \prec_{D_1} D_2$ such that*

$$\mu(\mathcal{C}_1, \mathcal{C}_2) > \mu(\mathcal{C}_1, \mathcal{C}_3).$$

Proof. Let S, S' be as in the hypothesis, with $\Psi(S') > \Psi(S)$. Construct $\mathcal{C}_2$ such that the similarity matrix $\kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(j)}) = S'_{ij}$ and $\mathcal{C}_3$ such that $\kappa(\mathcal{C}_1^{(i)}, \mathcal{C}_3^{(j)}) = S_{ij}$. Such constructions exist whenever the kernel κ has sufficient expressive range over circuit space (which holds for the graph kernels used in this work, since circuits are sparse weighted directed graphs and graph kernels separate them up to isomorphism on bounded-size graphs).

By construction, $D_3 \prec_{D_1} D_2$: the diagonal entries of S' are coordinate-wise no larger than those of S, and the row-mean off-diagonal entries of S' are coordinate-wise no smaller than those of S. Yet $\mu(\mathcal{C}_1, \mathcal{C}_2) = \Psi(S') > \Psi(S) = \mu(\mathcal{C}_1, \mathcal{C}_3)$, inverting the ranking.

A non-monotone metric therefore assigns higher similarity to a more strongly perturbed configuration, providing no reliable ordinal information about perturbation strength. □

This impossibility motivates the following requirement.

Property 3 (Semantic Consistency)**.** A metric $\mu = \Psi(S)$ is *semantically consistent* if Ψ is non-decreasing in each S_{ii} and non-increasing in each S_{ij} $(i \neq j)$. Equivalently, $D_3 \prec_{D_1} D_2$ implies $\mu(\mathcal{C}_1, \mathcal{C}_2) \leq \mu(\mathcal{C}_1, \mathcal{C}_3)$.

CKA, SVCCA, and RSA do not even form the class-conditional similarity matrix S, so the monotonicity requirement is undefined for them. Equivalently, they fail Property 3 *vacuously* as a consequence of failing Property 2. CAS, by construction as the gap between mean-diagonal and mean-off-diagonal of S, is linear in each entry with positive coefficient on diagonal terms and negative coefficient on off-diagonal terms, hence satisfies Property 3 (proved as a corollary in Sec. 3.2).

B CAS Construction: Full Derivation and Proofs

This appendix provides the formal counterpart to Sec. 3.2. We give a more detailed account of the design choices, prove Theorem 1 in full, and record the standard algebraic properties of CAS along with their proofs.

B.1 Design rationale

The construction in Sec. 3.2 introduces three components: a graph kernel κ, a class-conditional similarity matrix S, and a scalar aggregation. We expand on each.

Graph kernel choice. Property P1 requires that the metric not factor through any activation embedding. A graph kernel κ defined on circuit topology, such as the multiset of nodes, edges, and edge weights, rather than the activations these circuits induce, naturally satisfies this. We require $\kappa : \mathcal{C} \times \mathcal{C} \to [0, 1]$ to be:

1. *Normalized*: $\kappa(C, C) = 1$ for all C.
2. *Topology-respecting*: $\kappa(C_1, C_2) = 1 \iff C_1$ and C_2 are isomorphic as labeled graphs (modulo functionally redundant rerouting).
3. *Symmetric*: $\kappa(C_1, C_2) = \kappa(C_2, C_1)$.

We employ three kernels with these properties: treelet kernel (TK) [19], random walk kernel (RWK) [36], and an optimal-transport-based kernel (OT) [43]. The treelet kernel produces the cleanest separation between diagonal and off-diagonal entries of S in our experiments, and is used as the default; ablations across all three are reported in Sec. F.

Why uniform weights in the aggregation? A more general form of Eq. (4) would permit class-dependent weights:

$$\mathrm{CAS}_w(\mathcal{C}_1, \mathcal{C}_2) = \sum_i w_i^{(d)} S_{ii} - \sum_{i \neq j} w_{ij}^{(o)} S_{ij}, \qquad w_i^{(d)}, w_{ij}^{(o)} \geq 0.$$

Class-permutation invariance, the natural symmetry of the OOD setting, where no class is privileged a priori, forces $w_i^{(d)} = 1/c$ and $w_{ij}^{(o)} = 1/(c(c-1))$ up to a global scale. The scale is fixed by requiring CAS to take values in $[-1, 1]$ (Proposition 2). This recovers Eq. (4) uniquely; in this sense, CAS is the unique class-permutation-invariant, monotone-in-S, $[-1, 1]$-bounded scalar aggregation of the class-conditional similarity matrix.

B.2 Proof of CAS Soundness

Proof of Theorem 1. We verify each of P1, P2, P3 in turn.

P1 (structural sensitivity). Theorem 5 establishes that any metric of the form $\mu(\mathcal{C}_1, \mathcal{C}_2) = \tilde{\mu}(\phi(\mathcal{C}_1), \phi(\mathcal{C}_2))$, for some embedding ϕ into activation space, fails to distinguish circuit-robust from non-robust learners. CAS as defined in Eq. (4) computes via κ acting directly on the graph structure of the circuits $C_1^{(i)}, C_2^{(j)}$. By the topology-respecting property of κ (Condition 2 above), two circuits with identical activation footprints but distinct graph topology as constructed in the proof of Theorem 5 receive distinct kernel values: $\kappa(C, C') < 1$ when C, C' differ topologically beyond functional rerouting. Hence CAS does not factor through any activation embedding, satisfying P1.

P2 (class-conditional resolution). Theorem 6 establishes that any metric depending only on the multiset of entries of S. Equivalently, any metric of the form $\mu = \Psi(S)$ where Ψ is invariant under permutations of S's entries and fails to distinguish circuit-robust from entanglement-failure configurations. CAS decomposes as

$$\mathrm{CAS}(\mathcal{C}_1, \mathcal{C}_2) \;=\; \mu_{\text{same}}(S) - \mu_{\text{cross}}(S), \qquad \mu_{\text{same}}(S) := \tfrac{1}{c}\sum_i S_{ii}, \quad \mu_{\text{cross}}(S) := \tfrac{1}{c(c-1)}\sum_{i\neq j} S_{ij},$$

where μ_{same} depends only on diagonal entries and μ_{cross} only on off-diagonal entries. These are independently reportable from S. Equivalently, CAS is invariant under permutations that preserve diagonal vs. off-diagonal positions but is not invariant under arbitrary permutations of S's entries — distinguishing it from the class of metrics ruled out by Theorem 6. CAS therefore satisfies P2.

P3 (semantic consistency). Theorem 7 establishes that any metric of the form $\mu = \Psi(S)$ that is non-monotone under Definition 1 can invert perturbation rankings. CAS is linear in S:

$$\frac{\partial \mathrm{CAS}}{\partial S_{ii}} = \frac{1}{c} > 0, \qquad \frac{\partial \mathrm{CAS}}{\partial S_{ij}} = -\frac{1}{c(c-1)} < 0 \quad (i \neq j).$$

Hence CAS is monotone non-decreasing in each diagonal entry and monotone non-increasing in each off-diagonal entry. By Definition 1, $D_3 \prec_{D_1} D_2$ implies $S_{ii}^{(D_2)} \leq S_{ii}^{(D_3)}$ for all i and $\frac{1}{c-1}\sum_{j\neq i} S_{ij}^{(D_2)} \geq \frac{1}{c-1}\sum_{j\neq i} S_{ij}^{(D_3)}$ for all i. By linearity:

$$\mathrm{CAS}(\mathcal{C}_1, \mathcal{C}_2) - \mathrm{CAS}(\mathcal{C}_1, \mathcal{C}_3) = \tfrac{1}{c}\sum_i (S_{ii}^{(D_2)} - S_{ii}^{(D_3)}) - \tfrac{1}{c(c-1)}\sum_{i\neq j}(S_{ij}^{(D_2)} - S_{ij}^{(D_3)}) \leq 0,$$

where the inequality follows term-by-term: each diagonal difference is non-positive, and each off-diagonal difference, summed within a row, is non-negative; the negative coefficient on the off-diagonal sum makes its contribution non-positive. Hence $\mathrm{CAS}(\mathcal{C}_1, \mathcal{C}_2) \leq \mathrm{CAS}(\mathcal{C}_1, \mathcal{C}_3)$, satisfying P3. □

B.3 Algebraic properties of CAS

We record three standard algebraic properties of CAS that follow directly from Eq. (4). These are not load-bearing for the soundness result but are useful sanity checks and are referenced in the consistency analysis (Sec. 3.3).

Proposition 2 (Boundedness)**.** $\mathrm{CAS}(\mathcal{C}_1, \mathcal{C}_2) \in [-1, 1]$. *The upper bound $+1$ is attained iff $S_{ii} = 1$ and $S_{ij} = 0$ for all $i \neq j$ (perfect class-conditional preservation). The lower bound -1 is attained iff $S_{ii} = 0$ and $S_{ij} = 1$ for all $i \neq j$ (complete entanglement).*

Proof. Since $S_{ij} \in [0, 1]$ for all i, j, we have $\frac{1}{c}\sum_i S_{ii} \in [0, 1]$ and $\frac{1}{c(c-1)}\sum_{i\neq j} S_{ij} \in [0, 1]$. Their difference therefore lies in $[-1, 1]$. The extremes are attained at the stated configurations by direct substitution. □

Proposition 3 (Symmetry)**.** *If κ is symmetric, then* $\mathrm{CAS}(\mathcal{C}_1, \mathcal{C}_2) = \mathrm{CAS}(\mathcal{C}_2, \mathcal{C}_1)$.

Proof. The matrix $S^{(2,1)}$ obtained by swapping the role of $\mathcal{C}_1$ and $\mathcal{C}_2$ has entries $S_{ij}^{(2,1)} = \kappa(C_2^{(i)}, C_1^{(j)}) = \kappa(C_1^{(j)}, C_2^{(i)}) = S_{ji}^{(1,2)}$. Diagonal entries are preserved under transposition: $S_{ii}^{(2,1)} = S_{ii}^{(1,2)}$. The set of off-diagonal entries is also preserved: $\{S_{ij}^{(2,1)} : i \neq j\} = \{S_{ji}^{(1,2)} : i \neq j\} = \{S_{ij}^{(1,2)} : i \neq j\}$. Both terms in Eq. (4) therefore agree, and CAS is symmetric. □

Proposition 4 (Self-similarity)**.** *For any circuit family $\mathcal{C}$ in which each class circuit is uniquely self-similar — i.e., $\kappa(C^{(i)}, C^{(i)}) > \kappa(C^{(i)}, C^{(j)})$ for all $j \neq i$ — we have $\mathrm{CAS}(\mathcal{C}, \mathcal{C}) > 0$. Furthermore, $\mathrm{CAS}(\mathcal{C}, \mathcal{C}) = 1$ iff $\kappa(C^{(i)}, C^{(i)}) = 1$ and $\kappa(C^{(i)}, C^{(j)}) = 0$ for all $i \neq j$.*

Proof. The self-similarity hypothesis gives $S_{ii} > S_{ij}$ for all $j \neq i$, so $\frac{1}{c}\sum_i S_{ii} > \frac{1}{c(c-1)}\sum_{i\neq j} S_{ij}$, yielding $\mathrm{CAS}(\mathcal{C}, \mathcal{C}) > 0$. The equality case follows from the boundedness analysis in Proposition 2. □

C Consistency: Full Proofs

This appendix provides the full proofs for the consistency results in Sec. 3.3, along with two extensions: a quantitative comparison of inter-algorithmic versus intra-algorithmic separation (which sharpens the practical interpretation of Theorem 2), and a discussion of how Assumption 1 relates to the structural premise of Sec. 3.1.

C.1 Proof of Monte Carlo consistency

Proof of Theorem 2. Fix two learners ℓ_i, ℓ_j with $\overline{\mathrm{CAS}}(\ell_i) > \overline{\mathrm{CAS}}(\ell_j)$, and let $\delta_{ij}(d) = \mathrm{CAS}(\ell_i; d) - \mathrm{CAS}(\ell_j; d)$. Since $\mathrm{CAS} \in [-1, 1]$ by Proposition 2, $\delta_{ij}(d) \in [-2, 2]$, hence $\delta_{ij} \in L^2(\Omega_D, P_D)$ with $\sigma_{ij}^2 = \mathbb{E}[(\delta_{ij}(d) - m_{ij})^2] < \infty$.

The Monte Carlo estimator $\hat{\delta}_{ij}^{(M)} = \frac{1}{M}\sum_{k=1}^{M} \delta_{ij}(d_k)$ is an i.i.d. average with $\mathbb{E}[\hat{\delta}_{ij}^{(M)}] = m_{ij}$ and $\mathrm{Var}[\hat{\delta}_{ij}^{(M)}] = \sigma_{ij}^2/M$. An inversion occurs when $\hat{\delta}_{ij}^{(M)} \leq 0$, equivalently $\hat{\delta}_{ij}^{(M)} - m_{ij} \leq -m_{ij}$. Since $m_{ij} > 0$, this implies $|\hat{\delta}_{ij}^{(M)} - m_{ij}| \geq m_{ij}$. Applying Chebyshev's inequality [50]:

$$p_{\mathrm{inv}}^{(M)}(\ell_i, \ell_j) \;=\; \Pr[\hat{\delta}_{ij}^{(M)} \leq 0] \;\leq\; \Pr[|\hat{\delta}_{ij}^{(M)} - m_{ij}| \geq m_{ij}] \;\leq\; \frac{\mathrm{Var}[\hat{\delta}_{ij}^{(M)}]}{m_{ij}^2} \;=\; \frac{\sigma_{ij}^2}{M \cdot m_{ij}^2}.$$

This establishes the per-pair bound. For the aggregate bound, by the union bound:

$$P_{\mathrm{inv}}^{(M)} \;=\; \binom{N}{2}^{-1} \sum_{i<j} p_{\mathrm{inv}}^{(M)}(\ell_i, \ell_j) \;\leq\; \binom{N}{2}^{-1} \sum_{i<j} \frac{\sigma_{ij}^2}{M \cdot m_{ij}^2} \;=\; O(1/M)$$

as $M \to \infty$, since the sum has $\binom{N}{2}$ terms each of order $1/M$, and the constants σ_{ij}^2/m_{ij}^2 are finite (and uniformly bounded above by $\sigma_{ij}^2/\min_{i<j} m_{ij}^2$ over a finite learner pool). □

C.2 Inter- vs intra-algorithmic separation

The bound in Theorem 2 is governed by the ratio σ_{ij}^2/m_{ij}^2, which is small when the population gap m_{ij} is large relative to the per-domain variance. This ratio differs systematically depending on whether ℓ_i and ℓ_j were trained by the same or different domain-generalization algorithms.

Let $\mathrm{alg}(\ell)$ denote the algorithm used to train ℓ (e.g., ERM, IRM, CORAL, DANN). Define

$$\Delta_{\mathrm{inter}} = \min_{\substack{(\ell_i,\ell_j):\\ \mathrm{alg}(\ell_i)\neq\mathrm{alg}(\ell_j)\\ g(\ell_i)\neq g(\ell_j)}} |m_{ij}|, \qquad \Delta_{\mathrm{intra}} = \min_{\substack{(\ell_i,\ell_j):\\ \mathrm{alg}(\ell_i)=\mathrm{alg}(\ell_j)\\ g(\ell_i)\neq g(\ell_j)}} |m_{ij}|.$$

We empirically observe (Sec. 4) that $\Delta_{\mathrm{inter}} \gg \Delta_{\mathrm{intra}}$: structurally distinct training objectives produce learners with structurally distinct circuit families, yielding larger population CAS gaps. Letting $\bar{\sigma}^2 = \sup_{i,j} \sigma_{ij}^2$ (a uniform variance envelope, finite by boundedness), the inter-algorithmic and intra-algorithmic inversion bounds satisfy

$$\sup_{\substack{(\ell_i,\ell_j):\\ \mathrm{alg}(\ell_i)\neq\mathrm{alg}(\ell_j)}} p_{\mathrm{inv}}^{(M)}(\ell_i, \ell_j) \;\leq\; \frac{\bar{\sigma}^2}{M \cdot \Delta_{\mathrm{inter}}^2} \;\ll\; \frac{\bar{\sigma}^2}{M \cdot \Delta_{\mathrm{intra}}^2} \;\geq\; \sup_{\substack{(\ell_i,\ell_j):\\ \mathrm{alg}(\ell_i)=\mathrm{alg}(\ell_j)}} p_{\mathrm{inv}}^{(M)}(\ell_i, \ell_j).$$

Both quantities vanish at rate $1/M$, but the inter-algorithmic bound is tighter by a factor of $(\Delta_{\mathrm{intra}}/\Delta_{\mathrm{inter}})^2$. Practically, this means that CAS distinguishes ERM from IRM (or any pair

of differently-trained learners) with far fewer sampled domains than it requires to distinguish two ERM learners differing only by initialization or regularization. For the model-selection use case, where the question is typically "which training objective to use", not "which seed within an objective" – this is exactly the regime where convergence is fastest.

C.3 Proof of ranking recovery

Proof of Proposition 1. Let $\mathcal{L}^*$ be the ground-truth ranking by g. Under Assumption 1, $\mathcal{L}^*$ also corresponds to the ranking by $\overline{\mathrm{CAS}}$. The Kendall rank correlation between $\mathcal{L}^*$ and the empirical ranking $\mathcal{L}'_M$ satisfies

$$\tau(\mathcal{L}^*, \mathcal{L}'_M) = 1 - \frac{2}{\binom{N}{2}} \cdot |\{(i,j) : i < j, \pi^*(i) > \pi^*(j) \text{ but } \pi'_M(i) < \pi'_M(j)\}|,$$

i.e., τ measures the fraction of correctly-ordered pairs. By Theorem 2, the probability that any specific concordant pair is inverted is at most $\sigma_{ij}^2/(Mm_{ij}^2) = O(1/M)$. By the union bound, the probability that *any* of the $\binom{N}{2}$ pairs is inverted is at most $\binom{N}{2} \cdot O(1/M)$, which tends to 0 as $M \to \infty$ for any fixed N. Hence $\Pr[\tau(\mathcal{L}^*, \mathcal{L}'_M) = 1] \to 1$ as $M \to \infty$, which is convergence in probability of τ to 1. □

C.4 Relationship between Assumption 1 and the structural premise

Assumption 1 is the bridge between population CAS and OOD accuracy. We do not prove it from first principles in finite-accuracy regimes; instead, we provide three sources of theoretical and empirical support.

Limiting case: Corollary 1 establishes that in the limit of perfect OOD accuracy, a learner is structurally circuit-robust: its class-specific circuits are preserved across domains and remain distinct between classes. By Eq. (4), CAS evaluates to its maximum value on circuit-robust learners. Hence, in the limit $g(\ell) \to 1$, $\overline{\mathrm{CAS}}(\ell) \to 1$. The monotonicity assumption asserts that this limiting trend extends to the finite-accuracy regime, i.e., that intermediate values of g correspond to intermediate values of $\overline{\mathrm{CAS}}$ in the same order.

Connection to causal-invariance theory: The causal-invariance literature [67, 27, 10] establishes that OOD generalization is governed by the preservation of task-relevant computational structure across domains. Since CAS is, by construction, a measure of class-conditional circuit preservation across domains, monotonicity of $\overline{\mathrm{CAS}}$ in g is the natural finite-accuracy extension of this structural correspondence.

Empirical verification: Sec. 4 verifies Assumption 1 across 48 learners on PACS, Office-Home, and DomainNet. The Spearman rank correlation between $\widehat{\mathrm{CAS}}_M$ and g ranges from $\rho_S = 0.77$ (Office-Home Real World) to $\rho_S = 0.93$ (PACS Cartoon), with mean $\rho_S = 0.85$ across all benchmarks and target domains. This indicates that monotonicity holds approximately rather than exactly — empirical inversions occur primarily among learners with very similar OOD accuracies, where small perturbations to the population CAS estimate can cross the threshold.

A natural strengthening would be to bound the size of the inversion set as a function of $|g(\ell_i) - g(\ell_j)|$: large OOD-accuracy gaps should correspond to large CAS gaps. We leave a formal version of this stronger result for future work and note that the empirical Spearman correlations already establish that the assumption holds to a useful approximation in practice.

D Implementation Details

We describe the complete experimental pipeline, from backbone preparation through circuit extraction to CAS computation, with all hyperparameters fixed across datasets unless explicitly noted.

Backbone and adapter configuration: We use MLP [24], VGG19 [54], ResNet50 [23], MobileNetv2 [49], and ViT-B/16 [14] pretrained on ImageNet [12] as the frozen backbone across all experiments. Only the adapter modules and classification head are optimized. The details of the adapter insetion has been obtained in Table 4. Each adapter is a two-layer MLP with a bottleneck

ratio of r=4: a down-projection $W^{\text{down}} \in \mathbb{R}^{d/r \times d}$, a ReLU nonlinearity, and an up-projection $W^{\text{up}} \in \mathbb{R}^{d \times d/r}$, wrapped in a residual connection. The up-projection weights are initialised to zero so that each adapter starts as the identity function, preserving the pretrained backbone's behaviour at the beginning of training. The classification head is a single linear layer mapping the 1280-dimensional pooled feature to c class logits, where c varies by dataset (7 for PACS, 65 for Office-Home, 345 for DomainNet).

Table 4: Adapter insertion points for each backbone. *Layer ID* refers to the module path in the PyTorch model. d is the adapter's operating dimensionality. *Attn heads* indicates whether attention heads are additionally traced as circuit nodes.

Architecture	**Adapter Sites**	d	**Insertion Logic**	**Attn Heads**
VGG-19	`features.{28,30,32}`	512	After final three conv layers	—
ResNet-50	`layer4.{0,1,2}`	2048	After each block in `layer4`	—
MobileNetV2	`features.{16,17,18}`	1280	After last inverted-residual blocks	—
ViT-B/16	`encoder.layer.{4,8,11}`	768	After selected transformer blocks	✓
Deep MLP	`layers.{-3,-2,-1}`	d_h	After last three hidden layers	—

Table 4 and Fig. 8 summarize our choices, guided by two principles: (a) adapters are placed after the final few representation-learning stages so that the circuit captures high-level task-relevant computation, and (b) at least three adapter sites are used to enable nontrivial inter-layer circuit structure. For convolutional and MLP architectures, the circuit consists exclusively of adapter MLP neurons (channels). For ViT, we trace a *dual-component circuit* comprising both attention heads from the frozen self-attention sublayers and neurons from the trainable adapter MLPs at each adapter site. This richer representation allows us to disentangle the contributions of the pretrained attention mechanism from the task-specific adapter computation.



Figure 8: **Layerwise feature visualization with ViT-B/16 for the class "dog" for PACS dataset:** the selected transformer encoder blocks capture the essence of class dog effectively.

Data preprocessing: All images are resized to 224×224 pixels. During training, we apply random resized cropping (scale 0.7–1.0), random horizontal flipping, colour jitter (brightness, contrast, saturation, and hue each with magnitude 0.3), and random greyscale conversion with probability 0.1, following the standard DomainBed [22] augmentation protocol. During evaluation and circuit extraction, we apply only centre cropping and resizing. All images are normalised with ImageNet channel means ($[0.485, 0.456, 0.406]$) and standard deviations ($[0.229, 0.224, 0.225]$).

Training procedure: We train each learner for 30 epochs using the Adam optimizer [30] with a learning rate of 5×10^{-5} and a batch size of 32 per domain. For each training iteration, we sample a batch from every source domain, concatenate them, and compute the objective-specific loss. For ERM, we use standard cross-entropy. For IRM [5], we add the IRMv1 penalty with weight $\lambda_{\text{IRM}} = 1.0$, computed as the squared gradient of the loss with respect to a scalar dummy classifier initialised at 1.0. For Deep CORAL [55], we add the Frobenius norm of the covariance difference between domain-specific penultimate features with weight $\lambda_{\text{CORAL}} = 1.0$. For DANN [17], we attach a domain discriminator (three-layer MLP: $1280 \to 256 \to 256 \to K_{\text{tr}}$ with ReLU activations) trained with a separate Adam optimiser at learning rate 10^{-4}, connected through a gradient reversal layer with $\lambda_{\text{GRL}} = 1.0$. For Mixup [69], we interpolate input–label pairs across domains with $\alpha_{\text{mix}} = 0.2$ sampled from a Beta distribution. All experiments are run on a single NVIDIA A100 GPU (80GB).

Graph kernel computation: We compute the class-aligned similarity matrix $S \in \mathbb{R}^{c \times c}$ between every pair of source-domain circuit families using three graph kernels. The Random Walk Kernel (RWK) [36] counts the number of common random walks of length up to p=5 between two graphs, with a decay factor of $\lambda_{\text{RW}} = 10^{-4}$ to down-weight longer walks. The Treelet Kernel (TK) [19] enumerates all subtree patterns of depth up to 5 and computes a weighted count of shared patterns. The

Optimal Transport kernel (OT) [43] solves a Wasserstein distance [61] problem over node-attributed graphs with Sinkhorn regularization [3] ($\epsilon_{\text{sink}} = 0.1$, 50 iterations). All graph kernels take as input the circuit graphs with node features set to the importance scores Δ and edge features set to the causal weights $w_{u_1 \to u_2}$. Kernel values are normalized to $[0, 1]$ by dividing by the geometric mean of the self-similarities: $\mathcal{K}_{\text{norm}}(G_1, G_2) = \mathcal{K}(G_1, G_2)/\sqrt{\mathcal{K}(G_1, G_1) \cdot \mathcal{K}(G_2, G_2)}$.

Baseline metrics: All representational similarity baselines are computed on penultimate-layer features. For CKA [31], we use the linear kernel and centre both Gram matrices with the centering matrix $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. For SVCCA [47], we truncate each feature matrix via SVD at the 99% explained variance threshold before computing canonical correlations. For RSA [32], we compute pairwise correlation distance matrices and report their Spearman correlation [73]. All baselines are averaged over source-domain pairs identically to CAS.

Evaluation protocol: We follow the standard leave-one-domain-out protocol [64]: for each target domain, we train on all remaining source domains and evaluate OOD accuracy on the held-out target. We report Spearman's rank correlation [73] between each metric and OOD accuracy across the learner pool. Statistical significance is assessed via permutation tests with 10 permutations, and 95% confidence intervals are obtained via bootstrap resampling with 1,000 iterations. All random seeds are fixed (seed $= 42$) for data splitting, weight initialisation, and bootstrap sampling to ensure reproducibility.

All experiments are implemented in PyTorch 2.1 [40]. We use `torchvision` for MobileNetv2 weights, `GraKeL` [53] for graph kernel computation, and `scipy` [62] for statistical tests. Code, trained model checkpoints, extracted circuit graphs, and precomputed similarity matrices will be released upon publication.

D.1 Baseline Protocols for OOD Accuracy Prediction

For ATC [18], we reserve 20% of source data for validation, compute a negative-entropy confidence threshold, and estimate target accuracy as the fraction of predictions exceeding it. For ProjNorm [72], we train a source reference model and a pseudo-labeled target model, then compute $|\theta_{\text{pseudo}} - \theta_{\text{ref}}|_2$. ALine-D [6] is applied directly to the 48-model pool without modification.

E Adaptive Circuit Tracing (ACE)

We introduce ACE, a framework that inserts MLP adapters into selected intermediate layers of a frozen pretrained backbone, trains only the adapters and a task-specific head, and uses causal activation [74] to construct class-specific circuits [§]. Previous circuit extraction techniques [4, 63, 37, 38] have focused primarily on explaining prediction behavior in *fixed settings*, while this framework is applicable across major backbone families like convolutional networks [54, 23, 49], vision transformers [14], and deep MLPs [46]. The core procedure remains unchanged across architectures except for the adapter placement as it vary with backbone.

Problem Setup: Let $f_\theta(\mathbf{x})$ denote a pretrained backbone with frozen parameters θ. We augment selected intermediate layers of f_θ with trainable adapter modules $\{\mathcal{A}^{(\ell)}_{\psi_\ell}\}_{\ell=1}^{L}$, where ψ_ℓ denotes the parameters of the adapter inserted at layer ℓ. A trainable classification head g_ϕ with parameters ϕ is attached at the output. Only the adapter parameters $\{\psi_\ell\}_{\ell=1}^{L}$ and head parameters ϕ are optimized; the backbone parameters θ remain fixed. We decompose the frozen backbone as $f_\theta = f^{(L)}_\theta \circ \cdots \circ f^{(1)}_\theta$. and define frozen layer activations recursively by $z_\ell(\mathbf{x}) = f^{(\ell)}_\theta(z_{\ell-1}(\mathbf{x}))$, with an adapter inserted after layer ℓ, $h_\ell(\mathbf{x}) = z_\ell(\mathbf{x}) + \mathcal{A}^{(\ell)}_{\psi_\ell}(z_\ell(\mathbf{x}))$ and the next backbone layer receives h_ℓ as input: $z_{\ell+1}(\mathbf{x}) = f^{(\ell+1)}_\theta(h_\ell(\mathbf{x}))$. This residual structure ensures that if an adapter output is zero, the adapter reduces to the identity map, and the original backbone computation is recovered.

For transformer backbones (ViTs), each layer function $f^{(\ell)}_\theta$ consists of a multi-head self-attention (MHSA) sublayer followed by a feedforward network (FFN) and residual connections. We denote the output contributed by attention head h in layer ℓ by $\mathbf{a}^{(h)}_\ell(\mathbf{x})$, with the full attention output given

[§] A class-specific circuit is a sparse directed graph of adapter neurons (CNNs) and attention heads (ViTs) that are jointly necessary for prediction.

by $\mathrm{MHSA}_\ell(\mathbf{z}) = W^O\big[\mathbf{a}_\ell^{(1)};\ldots;\mathbf{a}_\ell^{(h)}\big]$. In this case, the adapter is inserted after the full transformer block (MHSA + FFN + residual), meaning that both attention and adapter computations are available as circuit components.

Adapter Module Design: Every adapter $\mathcal{A}_{\psi_\ell}^{(\ell)}$ is a lightweight two-layer MLP that operates on the channel (or feature) dimension of the intermediate representation: $\mathcal{A}_{\psi_\ell}^{(\ell)}(\mathbf{z}) = W_\ell^{\mathrm{up}}\sigma\big(W_\ell^{\mathrm{down}}\mathbf{z}\big)$, where $W_\ell^{\mathrm{down}} \in \mathbb{R}^{d/r\times d}$ projects to a bottleneck of ratio r, σ is a ReLU nonlinearity, and $W_\ell^{\mathrm{up}} \in \mathbb{R}^{d\times d/r}$ projects back. For convolutional backbones, the adapter acts independently at each spatial location. For transformers, it operates over token embeddings. For deep MLPs, it operates directly on hidden representations.

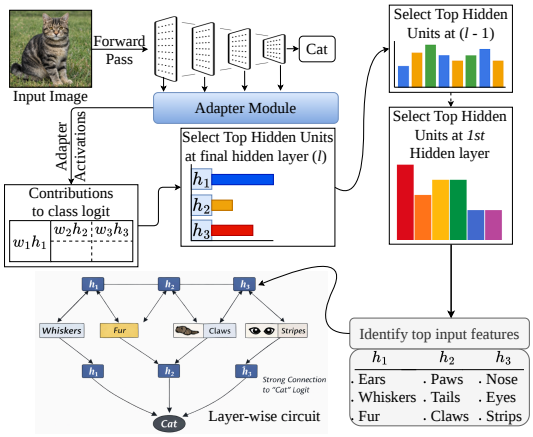


Figure 9: Layerwise adaptive circuit extraction.

Circuit Extraction via Activation Patching: After training, we extract a class-specific circuit by quantifying the causal contribution of individual computational units (i.e., adapter MLP neurons and attention head) via activation patching [74]. The procedure consists of two stages: node importance scoring and inter-layer edge estimation.

Node Importance Scoring: For each unit u (either an adapter neuron or an attention head) at layer ℓ and each class k, we compute the causal effect:

$$\Delta_\ell^{(u)}(k) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_k}\Big[\hat{y}_k(\mathbf{x}) - \hat{y}_k\big(\mathbf{x} \mid \mathrm{do}(u=\mathbf{0})\big)\Big], \tag{9}$$

where $\mathcal{D}_k$ is the subset of inputs with label k. A positive Δ indicates the unit *promotes* the correct class; a negative Δ indicates *suppression*. For each layer, we retain the K units of each type with the largest $|\Delta|$. For adapter MLP neurons in high-dimensional layers, exhaustive evaluation of all D dimensions may be prohibitive. In such cases we randomly sample a candidate set of size $S \geq K$ and select the top-K from this subset, providing a stochastic lower bound on the true top-K importance.

Inter-Layer Edge Estimation: To determine how top-K units in one layer influence top-K units in the next, we perform a second round of patching. For consecutive layers ℓ_1, ℓ_2 and top-K units $u_1 \in \mathcal{T}_{\ell_1},\ u_2 \in \mathcal{T}_{\ell_2}$, we ablate u_1 and observe the change in u_2's activation:

$$w_{u_1\to u_2} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_k}\Big[\bar{a}_{\ell_2}^{(u_2)}(\mathbf{x}) - \bar{a}_{\ell_2}^{(u_2)}\big(\mathbf{x} \mid \mathrm{do}(u_1=\mathbf{0})\big)\Big], \tag{10}$$

where $\bar{a}$ denotes the spatially- or token-averaged activation. An edge from (ℓ_1, u_1) to (ℓ_2, u_2) is added to the circuit graph if $|w_{u_1\to u_2}| > \epsilon$. For architectures with attention heads, we estimate edges within each component type separately: attention-to-attention and MLP-to-MLP across consecutive adapter layers, yielding two parallel edge sets that together form the full circuit. This separation reflects the architectural inductive bias: attention heads and MLP adapters operate at different stages of each transformer block, and their inter-layer causal pathways may carry qualitatively different information (e.g., positional routing via attention vs. feature refinement via MLPs).

Circuit Graph Construction: For each class c, the procedure yields a sparse, weighted, directed acyclic graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$ where:

$$\mathcal{V}_c = \underbrace{\bigcup_\ell \{(\ell, \texttt{mlp}, d) : d \in \mathcal{T}_\ell^{\mathrm{mlp}}\}}_{\text{adapter MLP nodes}} \cup \underbrace{\bigcup_\ell \{(\ell, \texttt{attn}, h) : h \in \mathcal{T}_\ell^{\mathrm{attn}}\}}_{\text{attention head nodes (ViT only)}}, \tag{11}$$

and

$$\mathcal{E}_c = \big\{(u_1, u_2, w_{u_1\to u_2}) : |w_{u_1\to u_2}| > \epsilon\big\}. \tag{12}$$

Nodes are annotated with their importance scores Δ and typed as either `mlp` or `attn`; edges carry signed weights indicating the direction and magnitude of causal influence. For convolutional and MLP backbones the attention node set is empty, recovering the simpler adapter-only circuit.

Computational Complexity and Scalability: Let C be the maximum number of adapter neurons, H the number of attention heads per layer (if applicable), L the number of adapter layers, and K the top-K selection size. For adapter MLP neurons, this requires $\mathcal{O}(L \cdot C)$ forward passes (or $\mathcal{O}(L \cdot S)$ with random subsampling of S candidates). For attention heads, it requires an additional $\mathcal{O}(L \cdot H)$ passes. Similarly, for MLP-to-MLP edges require $\mathcal{O}(L \cdot K)$ forward passes per layer pair; attention-to-attention edges require another $\mathcal{O}(L \cdot K)$. Crucially, all downstream effects on $\mathcal{T}_{\ell_2}$ are read from the cached output of a single patched forward pass per source node, so the cost scales linearly in K, not quadratically. Table 5 summarizes the per-class extraction cost across architectures.

For ViT, Stage 2 doubles because we estimate both MLP-to-MLP and attention-to-attention edge sets. Even so, the total cost remains dominated by Stage 1 MLP patching, and the attention head patching adds only $L \times H = 36$ forward passes with a negligible overhead.

Table 5: The number of forward passes for circuit extraction per class. For ViT, the attention column adds the cost of patching all H=12 heads per layer.

Architecture	MLP neurons C	Stage 1 (MLP)	Stage 1 (Attn)	Stage 2
VGG-19	512	1,536	—	30
ResNet-50	2048	6,144	—	30
MobileNetV2	1280	3,840	—	30
ViT-B/16	768	2,304	36	60

Architecture-Agnostic Abstraction:
The framework's scalability across architectures rests on two abstractions. Every adapter, regardless of backbone, exposes a tensor of shape $(B, *, D)$ where D is the feature dimension and $*$ denotes spatial or sequential axes. Ablation operates on dimension D: for convolutions, $\mathbf{h}[:, c, :, :] = 0$; for transformers, $\mathbf{h}[:, 0, d] = 0$ (CLS token); for MLPs, $\mathbf{h}[:, d] = 0$. A single PATCHMLP subroutine handles all cases. The attention patching module (PATCHHEAD) activates only when the backbone contains self-attention layers. For convolutional and MLP backbones, the attention node set $\mathcal{T}_\ell^{\texttt{attn}}$ is empty and the algorithm reduces to the adapter-only variant. For ViT, the self-attention modules are discovered by traversing the encoder's layer list, and each head is ablated by reshaping the attention output into $(B, T, H, D/H)$ and zeroing the relevant head slice. The end-to-end algorithm of the adaptive circuit extraction has been obtained in Algorithm 1.

The design of Algorithm 1 shows that adding support for a new architecture requires only (a) a configuration specifying adapter insertion points (fewer than five lines of code) and (b) optionally, a function that returns the attention modules if they exist. For convolutional features ($B \times C \times H \times W$), ablation sets $\mathbf{h}[:, c, :, :] = 0$. For transformer hidden states ($B \times T \times D$), ablation targets the CLS token: $\mathbf{h}[:, 0, d] = 0$. For MLP activations ($B \times D$), ablation sets $\mathbf{h}[:, d] = 0$. Attention head ablation reshapes $(B, T, D) \to (B, T, H, D/H)$, zeros head h, and reshapes back.

Discussion: The inclusion of attention heads alongside adapter MLP neurons in the ViT circuit provides a more complete picture of the model's computation. In our experiments, we observe that certain attention heads consistently appear across multiple class circuits, suggesting they perform general-purpose positional routing, while adapter neurons are more class-specific, capturing fine-grained feature refinement. This decomposition would not be visible in an adapter-only analysis. Because the patching loop iterates over adapter neurons and attention heads rather than all backbone parameters, the computational cost remains tractable even for large backbones. For instance, moving from VGG-19 (144M parameters) to ViT-B/16 (86M parameters) changes the cost only as a function of the adapter dimensionality (512 vs. 768) and the small additional attention head budget ($L \times H = 36$ passes), not the total model size.

F Choice of Graph Kernel

As depicted in Fig. 10, all the kernels that we have used, namely, optimal transport (OT) [43], random walk kernel (RWK) [36], and Treelet kernel (TK) [19], are symmetric, therefore $\text{CAS}(\mathcal{C}_1, \mathcal{C}_2) = \text{CAS}(\mathcal{C}_2, \mathcal{C}_1)$. In Fig. 10, CAS (artpainting, photo) and CAS (photo, artpainting) will be the same 0.71 with the treelet kernel. Another important observation is the choice of kernel in order to compute the CAS. It has been observed that the treelet kernel has more diagonal and off-diagonal separability than the random walk and optimal transport kernel because they capture complementary notions of circuit similarity while remaining computationally tractable. **Optimal transport (OT)** aligns circuits by matching structural and functional components globally, **random walk kernels (RWK)** measure similarity through shared connectivity patterns and signal flow through layered architectures, and **treelet kernels (TK)** capture local hierarchical and sparse substructures. It balances global

Algorithm 1 ADAPTER CIRCUIT EXTRACTION (ACE)

Require: Trained model f with adapters $\{\mathcal{A}_\ell\}$, attention modules $\{\mathrm{Attn}_\ell\}$ (if any), data $\mathcal{D}$, top-K, threshold ϵ
Ensure: Class-specific circuit graphs $\{\mathcal{G}_k\}$

$\{\mathcal{G}_k\} \leftarrow \varnothing$
for each mini-batch $(\mathbf{X}, \mathbf{y}) \in \mathcal{D}$ **do**
  $\mathbf{O}, \mathcal{C} \leftarrow$ FORWARDWITHCACHE$(f, \mathbf{X})$ ▷ Cache all adapter *and* attention outputs
  for each class $k \in \mathrm{unique}(\mathbf{y})$ **do**
    $\mathcal{I}_k \leftarrow \{\, i : y_i = k \,\}$; $b_k \leftarrow \mathrm{mean}(\mathbf{O}[\mathcal{I}_k, k])$
  — Stage 1: Node importance —
    for each adapter layer ℓ **do**
      for each neuron d in adapter $\mathcal{A}_\ell$ **do** ▷ or random subset
        $\tilde{\mathbf{O}}, _ \leftarrow$ PATCHMLP$(f, \mathbf{X}, \ell, d)$
        $\Delta^{\texttt{mlp}}_{\ell,d} \leftarrow b_k - \mathrm{mean}(\tilde{\mathbf{O}}[\mathcal{I}_k, k])$
      end for
      $\mathcal{T}^{\texttt{mlp}}_\ell \leftarrow$ top-K by $|\Delta^{\texttt{mlp}}|$; add MLP nodes to $\mathcal{G}_k$
      if Attn_ℓ exists **then** ▷ ViT only
        for each head $h = 1, \ldots, H$ **do**
          $\tilde{\mathbf{O}}, _ \leftarrow$ PATCHHEAD$(f, \mathbf{X}, \ell, h)$
          $\Delta^{\texttt{attn}}_{\ell,h} \leftarrow b_k - \mathrm{mean}(\tilde{\mathbf{O}}[\mathcal{I}_k, k])$
        end for
        $\mathcal{T}^{\texttt{attn}}_\ell \leftarrow$ top-K by $|\Delta^{\texttt{attn}}|$; add attention nodes to $\mathcal{G}_k$
      end if
    end for
  — Stage 2: Inter-layer edges —
    for each consecutive pair (ℓ_1, ℓ_2) **do**
      for each MLP neuron $d_1 \in \mathcal{T}^{\texttt{mlp}}_{\ell_1}$ **do**
        $_, \tilde{\mathcal{C}} \leftarrow$ PATCHMLP$(f, \mathbf{X}, \ell_1, d_1)$
        for each $d_2 \in \mathcal{T}^{\texttt{mlp}}_{\ell_2}$ **do**
          $w \leftarrow \mathrm{mean}(\mathcal{C}[\ell_2, d_2]) - \mathrm{mean}(\tilde{\mathcal{C}}[\ell_2, d_2])$
          if $|w| > \epsilon$ **then** add edge $(\ell_1, \texttt{mlp}, d_1) \to (\ell_2, \texttt{mlp}, d_2)$ with weight w
        end for
      end for
      if Attn_{ℓ_1} and Attn_{ℓ_2} exist **then**
        for each head $h_1 \in \mathcal{T}^{\texttt{attn}}_{\ell_1}$ **do**
          $_, \tilde{\mathcal{C}} \leftarrow$ PATCHHEAD$(f, \mathbf{X}, \ell_1, h_1)$
          for each $h_2 \in \mathcal{T}^{\texttt{attn}}_{\ell_2}$ **do**
            $w \leftarrow \mathrm{mean}(\mathcal{C}[\ell_2, h_2]) - \mathrm{mean}(\tilde{\mathcal{C}}[\ell_2, h_2])$
            if $|w| > \epsilon$ **then** add edge $(\ell_1, \texttt{attn}, h_1) \to (\ell_2, \texttt{attn}, h_2)$ with weight w
          end for
        end for
      end if
    end for
  end for
end for
return $\{\mathcal{G}_k\}$

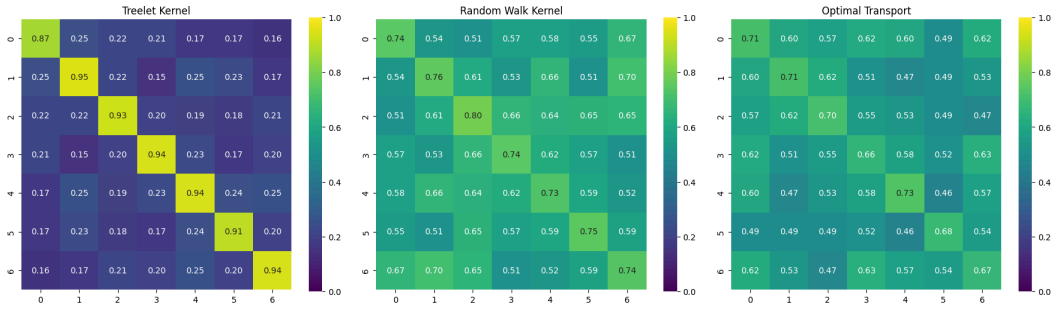


Figure 10: **Choice of kernels:** Pairwise similarity matrices computed using the Treelet Kernel, Random Walk Kernel, and Optimal Transport methods for ArtPainting and Photo domain with MobileNetv2. Diagonal entries indicate self-similarity, while off-diagonal values capture cross-instance structural similarity. The Treelet Kernel shows strong self-similarity with lower inter-instance similarities, whereas Random Walk and Optimal Transport exhibit comparatively higher cross-instance similarity, reflecting different notions of structural correspondence.

alignment, path-based dynamics, and local motif similarity, making them well-suited for comparing neural circuits. We use the Treelet Kernel to compute the CAS for all the experiments in the main paper as well as in the supplementary materials.

G Ranking pairwise domain similarity with CAS

Objective and Setup: Beyond predicting which learner will generalize best, a well-calibrated circuit-level metric should also recover the relative proximity of domains from one another. We evaluate this property on PACS by treating each source-domain pair as a data point and asking whether the metrics rank those pairs in agreement with their known visual dissimilarity. The expected ground-truth ordering, corroborated by the distributional divergence measurements in Fig. 11 is: Photo–ArtPainting (P↔A) < ArtPainting–Cartoon (A↔C) < Cartoon–Photo (C↔P) < Cartoon–Sketch (C↔S) < Photo–Sketch (P↔S), meaning Photo and ArtPainting share the most structure while Photo and Sketch are most dissimilar. For each of the six domain pairs, we compute CAS, CKA, SVCCA, and RSA using the same 48-learner pool and average across learners within each pair; results are reported in Table 6.

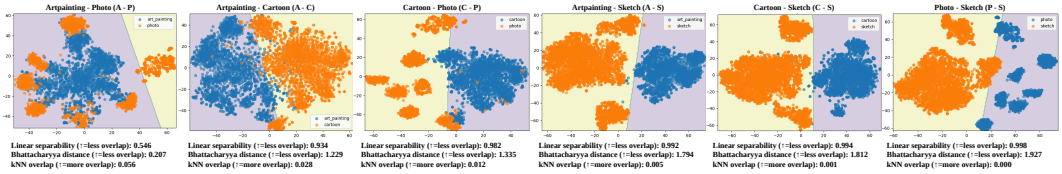


Figure 11: **Domain-wise feature distribution divergence on PACS:** Pairwise domain overlap between Artpainting (A), Cartoon (C), Photo (P), and Sketch (S) is quantified using linear separability (↑ indicates less overlap), Bhattacharyya distance (↑ indicates greater distributional divergence), and kNN overlap (↑ indicates more overlap). Results show progressively increasing separability and distributional distance from A–P to P–S, with near-zero kNN overlap for P–S, indicating minimal shared neighbourhood structure and substantial cross-domain shift.

Observations and Analysis: CAS (TK) recovers the ground-truth domain ordering precisely: A–P (0.62) > A–C (0.59) > C–P (0.58) > A–S (0.45) > C–S (0.37) > P–S (0.28). This ranking is in complete agreement with the three independent distributional divergence measures reported in Fig. 11, linear separability, Bhattacharyya distance, and kNN overlap, all of which assign maximal distance to the Photo–Sketch pair and minimal distance to the ArtPainting–Photo pair. The near-linear decay of CAS values from 0.62

Table 6: Pairwise domain shift distances on PACS. Higher values indicate more similar domains. Expected ordering based on visual similarity (see Fig. 11): P↔A < A↔C < P↔C < C↔S < P↔S.

Domains	CKA (↑)	SVCCA (↑)	RSA (↑)	CAS (TK) (↑)
A - P	0.65	0.04	0.39	0.62
A - C	0.83	0.25	0.08	0.59
C - P	0.80	0.37	0.46	0.58
A - S	0.77	0.23	0.17	0.45
C - S	0.68	0.02	0.27	0.37
P - S	0.77	0.10	0.04	0.28

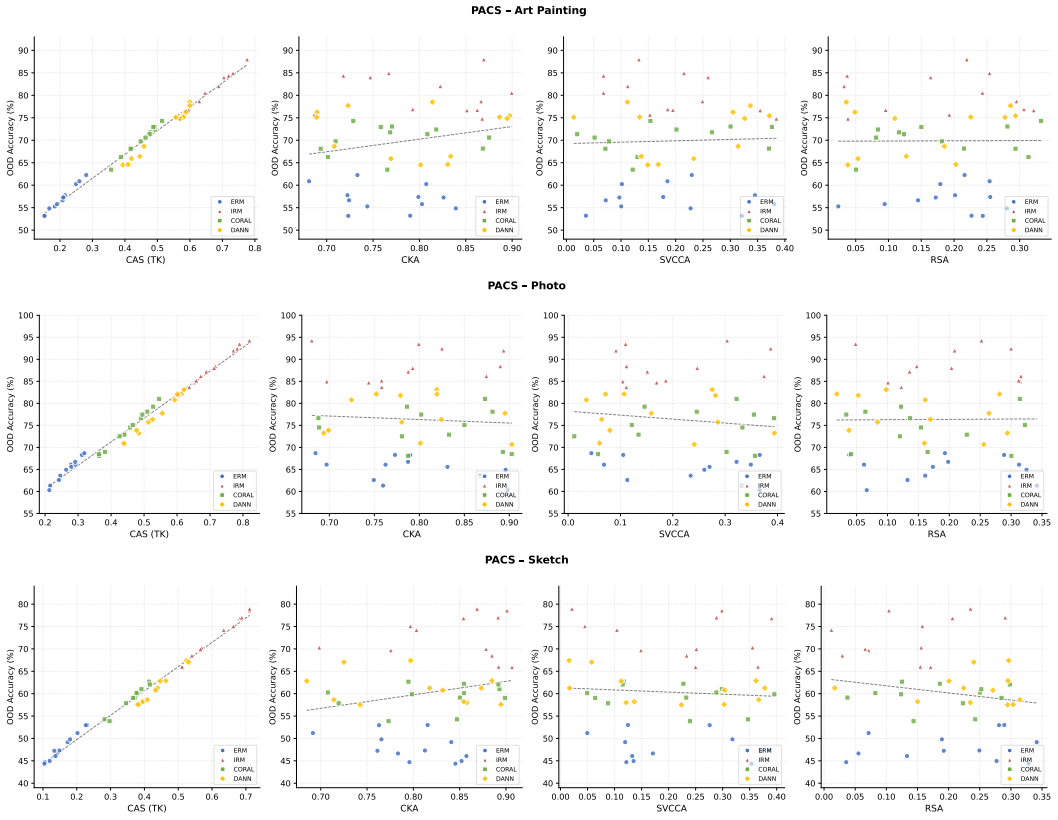


Figure 12: **CAS vs OOD accuracy for PACS dataset:** For all the domains, Art Painting ($\rho_S = 0.91$), Photo ($\rho_S = 0.93$), and Sketch ($\rho_S = 0.88$), CAS maintains a monotonous trend with OOD accuracy. In contrast, representation similarity metrics always produce a diffuse scatter plot due to the unavailability of structural information.

to 0.28 further supports the interpretation of CAS as a metric-like distance on domain space, consistent with the monotonicity.

In contrast, the representational baselines fail to produce coherent domain orderings (Table 6). CKA assigns its highest pairwise value to the A–C pair (0.83) and its lowest to the A–P pair (0.65), an inversion of the expected ordering that is a direct consequence of the structural blindspot identified in Theorem 5: two models whose penultimate-layer activations are geometrically similar on ArtPainting may route those activations through entirely different computational pathways than they do on Cartoon, yet CKA cannot detect this difference. SVCCA produces ordinal inversions for four of the six pairs, in particular ranking C–P (0.37) higher than A–C (0.25) and nearly zeroing out A–P (0.04), contrary to the visual evidence that ArtPainting and Photo are the closest domains in PACS. RSA is directionally better than SVCCA but still misorders A–C relative to A–P and C–P, assigning a near-zero similarity to the Photo–Sketch pair (0.04) while simultaneously underscoring Cartoon–Photo similarity (0.46) well above its true rank.

The consistent superiority of CAS across all six pairs stems from its sensitivity to the circuit topology shared between domains rather than the geometry of their activation outputs. When Photo and ArtPainting are presented to the same learner, the same class-specific neurons and inter-layer edges are recruited in both domains, the structural fingerprint is preserved, yielding high diagonal coherence and low off-diagonal confusion in the similarity matrix S. As the domain shift increases toward Sketch, circuits for distinct classes progressively entangle, and same-class circuits diverge, monotonically driving CAS downward. This mechanism cannot be captured by metrics that aggregate over all activations without regard to which neurons generate them.

Also, no single baseline comes close to the $\rho_S = 0.88$ achieved by CAS (see Fig. 15). The two strongest competitors: Source val. accuracy ($\rho_S = 0.73$) and ALine-D ($\rho_S = 0.72$) still lag CAS by more than 0.15 rank-correlation units. Importantly, Source val. accuracy requires labelled source data

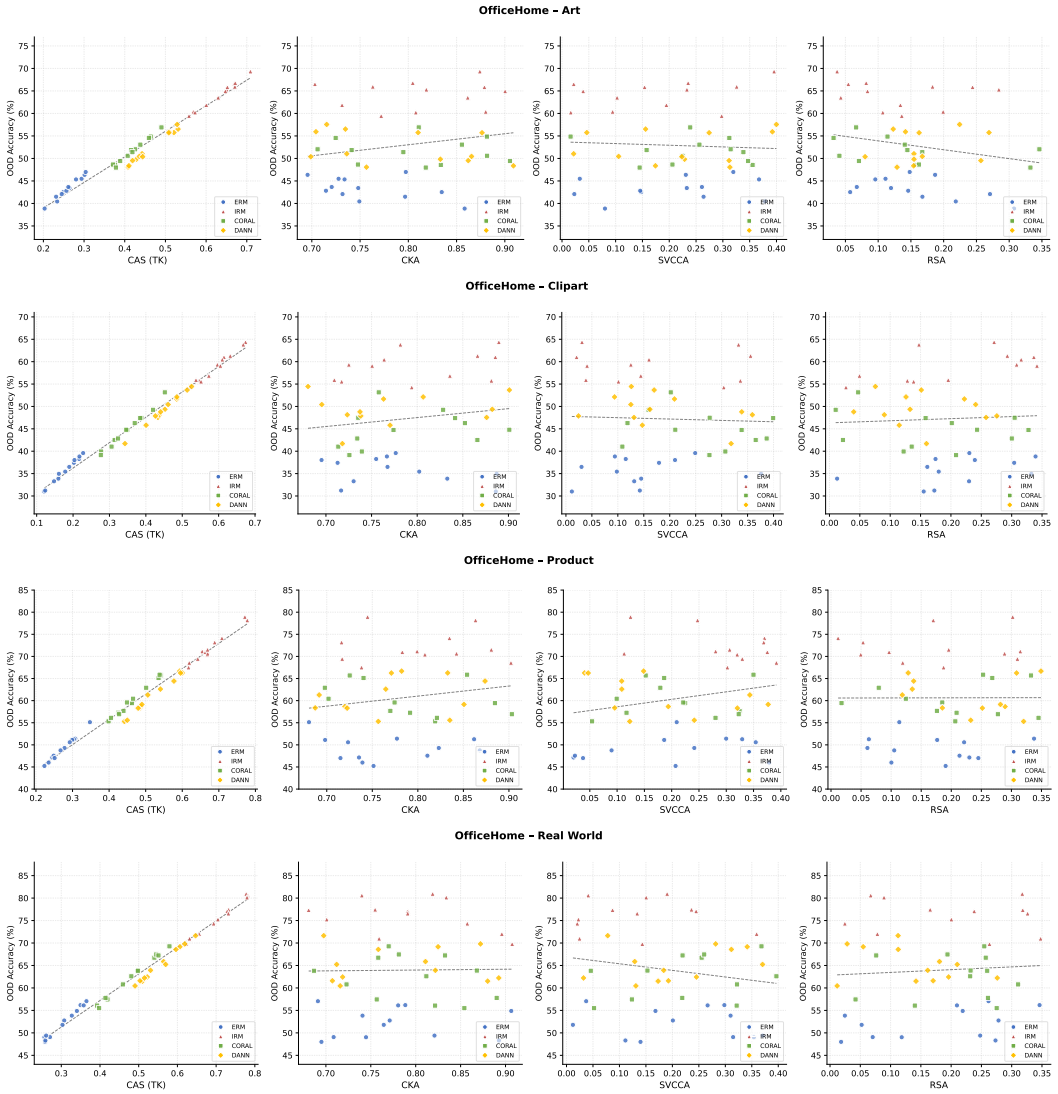


Figure 13: **CAS vs OOD accuracy for OfficeHome dataset:** CAS (TK) exhibits a strong near-monotone relationship with OOD accuracy across all four OfficeHome domains (Art, Clipart, Product, Real World), with ERM consistently in the low-CAS low-accuracy region and IRM in the high-CAS high-accuracy region, while CKA, SVCCA, and RSA show weak, noisy associations with no consistent method ordering.

Table 7: **Signal-noise analysis and correction on Office-Home.** Despite a higher extraction noise floor than PACS, cross-domain signal remains dominant, noise-corrected CAS improves OOD ranking, and the same stability trends across objectives and architectures persist.

Quantity	Mean	Std	Min	Max
$\hat{\Delta}_{\text{noise}}$	0.118	0.027	0.071	0.183
Δ_{obs}	0.540	0.102	0.346	0.731
Δ_{signal}	0.422	0.095	0.253	0.602
SNR	3.58	1.04	1.38	6.52
SNR > 1		46/48 (95.83%)		

(a) Signal–noise decomposition

Objective	Noise	Arch.	Noise
IRM	0.089±0.018	ViT-B/16	0.094±0.020
CORAL	0.108±0.022	MobileNet	0.107±0.024
DANN	0.119±0.025	ResNet50	0.129±0.027
ERM	0.146±0.030	Mixer	0.152±0.031

(b) Noise floor breakdown

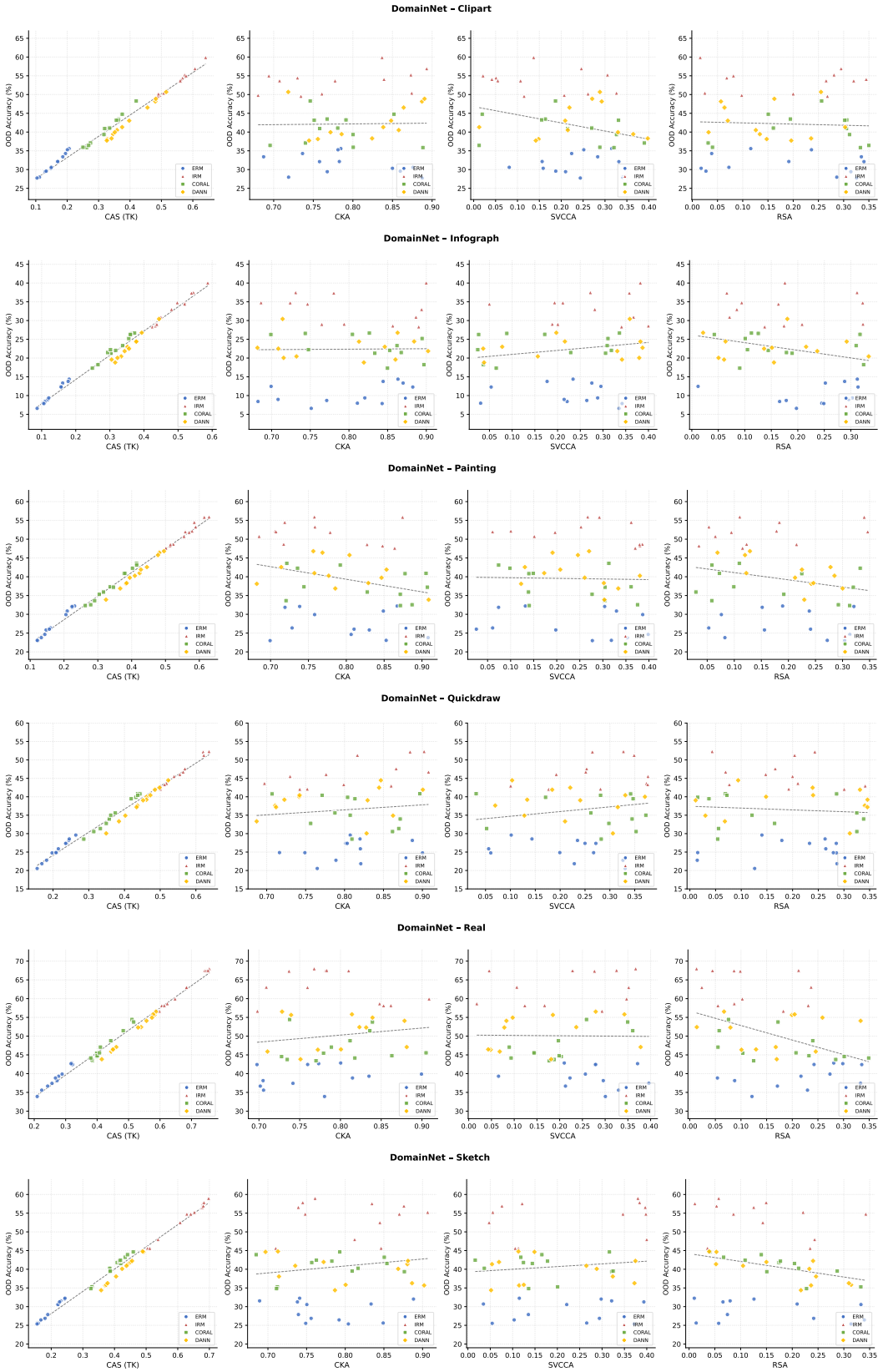


Figure 14: **CAS vs OOD accuracy for DomainNet dataset:** Across all six DomainNet domains, we observe a similar trend, confirming the superiority of CAS over the rest.

and is not a model-selection tool in the usual sense, while ALine-D requires a pool of at least 48 models to compute its agreement estimate. CAS requires neither. Similarly, the entire family of complexity proxies ported from [26] produces clearly scattered clouds, confirming that these quantities capture properties orthogonal to generalization rank. The same holds for representational similarity measures reported in the main paper (CKA, SVCCA, RSA). Last but not the least, the measures characterizing domain shift [59] yield correlations at or below 0.21, with Spectral complexity exhibiting a negative correlation ($\rho_S = -0.19$). These results demonstrate that CAS is not merely incrementally better than existing predictors: it occupies a qualitatively different regime of predictive reliability, one that holds consistently across all four training objectives and twelve architecture–hyperparameter seeds examined here.

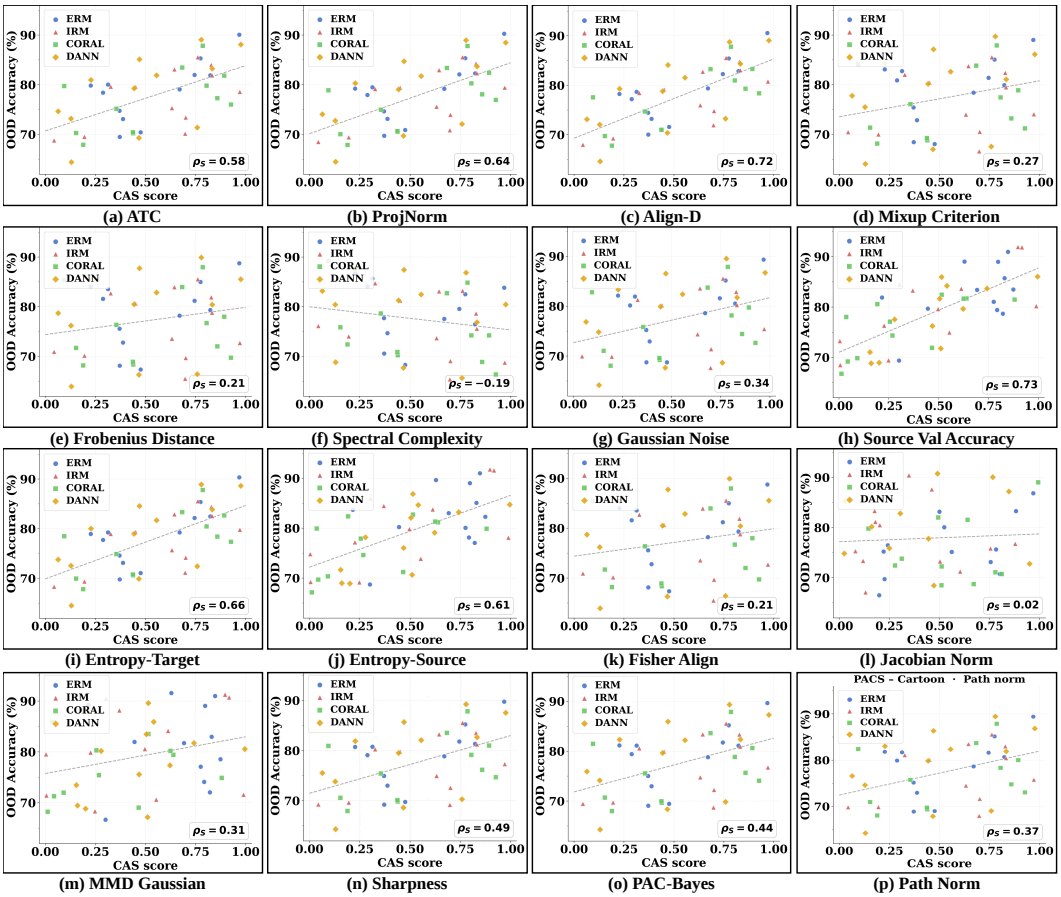


Figure 15: Scatter plots of competing OOD-accuracy predictors against OOD accuracy (%) on PACS – Cartoon (48 models: 12 each of ERM, IRM, CORAL, DANN). The dashed line is the least-squares fit and ρ_S is the Spearman rank correlation. Despite varying input modalities and conceptual motivations, no baseline achieves a consistent near-monotone relationship with OOD accuracy across training objectives, in contrast to CAS.

H Decomposing Circuit Differences: Noise vs. Signal

Given two circuits $\mathcal{C}(f, D_1)$ and $\mathcal{C}(f, D_2)$ extracted on different domains, their observed difference $\Delta_{obs} = 1 - \kappa(\mathcal{C}(f, D_1), \mathcal{C}(f, D_2))$ conflates two sources: $\Delta_{obs} = \Delta_{signal} + \Delta_{noise}$. The noise floor Δ_{noise} arises because TopK selection is discontinuous (small perturbations in Δ-scores can flip which neurons enter the TopK), edge weights are empirical estimates, and activation patching has finite-sample variance.

Paired Bootstrap Noise Estimation: We estimate Δ_{noise} directly from the data using a within-domain resampling protocol [68]. For each domain D, we partition $\mathcal{D}_D$ into B disjoint bootstrap splits $\{\mathcal{D}_D^{(1)}, \mathcal{D}_D^{(2)} \dots \mathcal{D}_D^{(B)}\}$ of equal size, each matching the sample size used for cross-domain

Table 8: **Target-domain performance prediction on Office-Home and DomainNet.** Confidence-, entropy-, and source-accuracy-based baselines are moderately informative, whereas Jacobian norm and distribution-divergence measures are weak and occasionally negative, particularly on DomainNet's harder domains such as Infograph and Quickdraw. CAS (TK) achieves the best result in every domain on both datasets (mean 0.82 on Office-Home and 0.79 on DomainNet), outperforming the strongest baseline, source validation accuracy (0.68 and 0.62, respectively).

	Office-Home					**DomainNet**						
Method	Art	ClipArt	Real	Product	Mean	ClipArt	Infograph	Painting	Quickdraw	Real	Sketch	Mean
ATC [18]	0.54	0.48	0.58	0.52	0.53	0.50	0.42	0.52	0.38	0.54	0.46	0.47
ProjNorm [72]	0.60	0.53	0.64	0.57	0.59	0.55	0.47	0.57	0.42	0.59	0.50	0.52
ALine-D [6]	0.68	0.62	0.72	0.66	0.67	0.63	0.55	0.65	0.50	0.67	0.58	0.60
DoC [21]	0.59	0.52	0.63	0.56	0.58	0.53	0.44	0.55	0.40	0.57	0.48	0.50
Entropy-Target [59]	0.62	0.55	0.66	0.60	0.61	0.57	0.48	0.59	0.44	0.61	0.52	0.54
Entropy-Source [59]	0.57	0.50	0.61	0.54	0.56	0.52	0.43	0.54	0.39	0.56	0.47	0.49
Source val. Accuracy [59]	0.69	0.63	0.73	0.67	0.68	0.65	0.57	0.67	0.52	0.69	0.60	0.62
Jacobian norm [59]	0.06	0.01	0.08	0.03	0.05	0.04	−0.02	0.06	−0.05	0.07	0.01	0.02
Fisher Align [59]	0.20	0.14	0.23	0.17	0.19	0.18	0.11	0.20	0.08	0.22	0.14	0.16
MMD-Gaussian [59]	0.28	0.22	0.32	0.25	0.27	0.26	0.18	0.28	0.14	0.30	0.22	0.23
HΔH-divergence [59]	0.14	0.09	0.17	0.11	0.13	0.12	0.06	0.14	0.03	0.16	0.09	0.10
CAS (Ours)	**0.81**	**0.76**	**0.86**	**0.82**	**0.82**	**0.78**	**0.72**	**0.81**	**0.68**	**0.84**	**0.74**	**0.79**

extraction. Then we extract a circuit $\mathcal{C}^{(b)}(f, D)$ from each split using the same ACE pipeline (same K, same r, and same ϵ). We compute the pairwise within-domain noise as:

$$\Delta^{D}_{noise} = \frac{1}{B(B-1)} \sum_{a \neq b} [1 - \kappa(\mathcal{C}^{(a)}(f, D), \mathcal{C}^{(b)}(f, D))] \tag{13}$$

Average over domains:

$$\Delta_{noise} = \frac{1}{N} \sum_{D} \Delta^{D}_{noise} \tag{14}$$

Where N is the number of domain pairs. Because both circuits in each pair are extracted from the same underlying distribution, any observed dissimilarity is by construction attributable to extraction noise rather than domain shift. This mirrors the test–retest reliability protocol standard in neuroscience [32] and the split-half validation used in mechanistic interpretability [11].

Signal-to-noise Decomposition: The cross-domain dissimilarity $\Delta_{obs}(D_1, D_2) = 1 - \kappa(\mathcal{C}(f, D_1), \mathcal{C}(f, D_2))$ is then decomposed additively (under independence of noise across extractions):

$$\Delta_{signal}(D_1, D_2) = \max(0, \Delta_{obs}(D_1, D_2) - \Delta_{noise}) \tag{15}$$

Equivalently, the signal-to-noise ratio of circuit divergence is: $\text{SNR}(D_1, D_2) = \dfrac{\Delta_{signal}(D_1, D_2)}{\Delta_{noise}}$ with SNR $>> 1$ indicating that the observed cross-domain difference substantially exceeds the extraction noise floor and thus reflects genuine domain-induced reorganization.

Proposition 5 (Noise additivity under independent extraction)**.** *Let κ be a graph kernel with Lipschitz constant L_κ with respect to edit distance, and let $\mathcal{C}(f, D)$ denote the population circuit for domain D. If the extraction noise has bounded variance σ^2, and is independent across domains, then:*

$$\mathbb{E}[\Delta_{obs}(D_1, D_2)] = \Delta^*(D_1, D_2) + 2L_\kappa\sigma^2 + O(\sigma^4)$$

where Δ^* is the population-level cross-domain dissimilarity.

The $2L_\kappa\sigma^2$ term is exactly what Δ_{noise} estimates (one $L_\kappa\sigma^2$ contribution per circuit, symmetric across the pair). Subtracting it yields an unbiased estimator of Δ^* up to $O(\sigma^4)$.

H.1 Circuit Fragility Test for PACS dataset

Extracted circuits can be unstable due to finite-sample effects, raising the concern that cross-domain differences may reflect extraction noise rather than genuine domain shift. To disentangle these factors, we estimate the noise Δ_{noise} by repeatedly extracting circuits from multiple random subsets of the same domain, where any variation reflects circuit-extraction noise. We then decompose the observed cross-domain difference as $\Delta_{\text{obs}} = \Delta_{\text{signal}} + \Delta_{\text{noise}}$, with $\Delta_{\text{signal}} = \max(0, \Delta_{\text{obs}} - \Delta_{\text{noise}})$. The

Table 9: **Signal-noise analysis and noise correction ablations on PACS.** Cross-domain circuit differences substantially exceed extraction noise, and circuit stability varies systematically across training objectives and architectures.

Quantity	Mean	Std	Min	Max
Δ_{noise}	0.096	0.021	0.058	0.148
Δ_{obs}	0.501	0.093	0.321	0.678
Δ_{signal}	0.405	0.089	0.242	0.574
SNR	4.22	1.18	1.63	7.89
SNR > 1		$47/48$ (97.91%)		

(a) Signal–noise decomposition

Objective	Noise	Arch.	Noise
IRM	0.073 ± 0.014	ViT-B/16	0.079 ± 0.016
CORAL	0.091 ± 0.017	MobileNet	0.088 ± 0.019
DANN	0.098 ± 0.020	ResNet50	0.108 ± 0.022
ERM	0.122 ± 0.024	MLP	0.127 ± 0.025

(b) Noise floor breakdown

signal-to-noise ratio $\text{SNR} = \Delta_{\text{signal}}/\Delta_{\text{noise}}$ quantifies whether circuit variation is dominated by noise or domain shift.

As shown in Table 9(a), within-domain variability is relatively small (0.096 ± 0.021) compared to observed cross-domain dissimilarity (0.501 ± 0.093), indicating that most measured differences cannot be explained by noise alone. Consistently, the estimated signal remains large (0.405 ± 0.089), the average signal-to-noise ratio is high ($\text{SNR} = 4.22$), and $47/48$ learners satisfying $\text{SNR} > 1$. Together, these observations indicate that cross-domain circuit divergence is predominantly driven by genuine domain-induced reorganization rather than stochasticity in circuit extraction. Furthermore, Table 9(b) shows systematic variation in stability: invariance-promoting objectives (IRM, CORAL) and structured architectures (ViTs) exhibit lower within-domain variability, while ERM and MLPs show higher variability, suggesting that both training objectives and architectural inductive biases influence circuit robustness.

H.2 Circuit Fragility Test for OfficeHome and DomainNet

Table 7 reports signal–to-noise analysis on Office-Home, a markedly harder benchmark than PACS due to its 65-class label space, fine-grained visual distinctions, and predominantly semantic domain shifts (e.g., Art vs. Real World office objects). Two factors raise the extraction noise floor relative to PACS: (1) the larger class count reduces the per-class sample budget, lowering statistical precision in importance scoring and edge estimation; and (2) semantically adjacent classes (e.g., Backpack, Bag, Briefcase) induce overlapping circuits in representation space, making discrete top-K selection more sensitive to finite-sample noise.

Table 10: **Signal-noise analysis and correction on DomainNet.** Although extraction noise increases substantially for the 345-class, six-domain setting, cross-domain circuit signal remains dominant, and noise correction yields the largest correlation gain among all benchmarks.

Quantity	Mean	Std	Min	Max
$\hat{\Delta}_{\text{noise}}$	0.142	0.034	0.086	0.221
Δ_{obs}	0.559	0.108	0.358	0.762
Δ_{signal}	0.417	0.098	0.234	0.593
SNR	2.94	0.91	1.06	5.41
SNR > 1		$44/48$ (91.66%)		

(a) Signal–noise decomposition

Objective	Noise	Arch.	Noise
IRM	0.107 ± 0.022	ViT-S	0.113 ± 0.024
CORAL	0.131 ± 0.028	MobileNet	0.128 ± 0.029
DANN	0.143 ± 0.031	ResNet50	0.155 ± 0.033
ERM	0.176 ± 0.038	Mixer	0.182 ± 0.039

(b) Noise floor breakdown

Despite these challenges, the domain-induced circuit signal remains dominant. The mean noise floor increases to $\hat{\Delta}_{\text{noise}} = 0.142 \pm 0.027$ (vs. 0.096 on PACS), while the mean cross-domain dissimilarity rises to $\Delta_{\text{obs}} = 0.5559 \pm 0.102$, yielding a mean SNR of 3.58—lower than PACS but still well above 1. Of 48 learners, 44 satisfy $\text{SNR} > 1$; the two exceptions are MLP-Mixer models trained with ERM, a configuration that consistently lies at the boundary of reliable extraction. On Office-Home, the failure is most pronounced for the Clipart domain, where large photorealistic–clip-art gaps induce partial invariance under ERM, yet the absence of an explicit invariance objective leaves the extracted circuits unstable under resampling.

Table 11: **Ablations on Office-Home.** The optimal settings shift to K=60 due to the larger label space, while the preferred bottleneck ratio (r=4) and edge threshold (ϵ=10^{-4}) remain consistent with PACS.

K	CAS	OOD	ρ_S
5	0.08	54.32	0.62
10	0.14	58.71	0.69
20	0.22	62.85	0.74
30	0.28	65.10	0.78
60	**0.35**	**68.47**	**0.82**
100	0.38	68.73	0.80
164	0.39	68.81	0.77
200	0.40	68.84	0.75

(a) TopK neurons per layer

r	Params	CAS	OOD	ρ_S
1	1,638,400	0.27	69.18	0.73
2	819,200	0.31	68.92	0.78
4	**409,600**	**0.35**	**68.47**	**0.82**
8	204,800	0.39	67.35	0.79
16	102,400	0.44	65.48	0.74

(b) Adapter bottleneck ratio

ϵ	Edges	CAS	OOD	ρ_S
10^{-2}	34	0.18	64.82	0.71
10^{-3}	89	0.27	66.93	0.78
10^{-4}	**158**	**0.35**	**68.47**	**0.82**
10^{-5}	237	0.37	68.61	0.80
10^{-6}	362	0.38	68.66	0.76

(c) Edge threshold

After noise-floor subtraction, the noise-corrected CAS increases the Spearman correlation from $\rho_S = 0.790$ (raw CAS) to $\rho_S = 0.830$, a larger absolute gain than on PACS, indicating that elevated noise on Office-Home was masking a stronger mechanistic signal. The objective-wise noise ordering matches PACS—IRM (0.089), CORAL (0.108), DANN (0.119), ERM (0.146)—supporting the view that invariance-driven objectives regularize circuit structure and improve reproducibility under resampling. Architecturally, ViT-B/16 remains most stable (0.094) and MLP-Mixer least (0.152), with a larger ResNet-50–Mixer gap than on PACS (0.029 vs. 0.019). This aligns with the hypothesis that multi-head attention provides discrete anchoring points for circuit extraction, whereas the Mixer's all-MLP design distributes class information more diffusely.

Table 12: **Ablations on DomainNet.** The optimal TopK increases to K=164 for the 345-class setting, while the preferred bottleneck ratio (r=4) and edge threshold (ϵ=10^{-4}) remain consistent across benchmarks.

K	CAS	OOD	ρ_S
5	0.05	32.18	0.54
10	0.09	37.45	0.60
20	0.14	42.83	0.65
40	0.19	48.62	0.70
80	0.24	53.91	0.74
120	0.28	57.28	0.77
164	**0.32**	**59.74**	**0.79**
200	0.33	59.93	0.77

(a) TopK neurons per layer

r	Params	CAS	OOD	ρ_S
1	1,638,400	0.24	60.85	0.69
2	819,200	0.28	60.42	0.74
4	**409,600**	**0.32**	**59.74**	**0.79**
8	204,800	0.36	58.16	0.76
16	102,400	0.42	55.23	0.71

(b) Adapter bottleneck ratio

ϵ	Edges	CAS	OOD	ρ_S
10^{-2}	72	0.15	55.38	0.66
10^{-3}	186	0.24	57.92	0.74
10^{-4}	**327**	**0.32**	**59.74**	**0.79**
10^{-5}	498	0.34	59.91	0.77
10^{-6}	753	0.35	59.96	0.73

(c) Edge threshold

Similarly, Table 10 presents signal–to-noise analysis on DomainNet, the most demanding benchmark: 6 domains, 345 classes, and 600K images spanning extreme stylistic variation (e.g., Quickdraw sketches, Infograph diagrams, Real photos). The extraction noise floor is highest across benchmarks, $\hat{\Delta}_{\text{noise}} = 0.142 \pm 0.034$, due to three compounding factors: (1) the 345-class label space sharply reduces per-class data for circuit estimation, amplifying top-K selection variance; (2) severe cross-domain heterogeneity causes identical semantics to rely on divergent low-level features, increasing subsample sensitivity; and (3) the 15 domain pairs (vs. 6 in PACS) enlarge the combinatorial surface for noise to manifest.

Despite the increased difficulty, domain-induced circuit reorganization remains dominant. The mean cross-domain dissimilarity $\Delta_{\text{obs}} = 0.559 \pm 0.108$ exceeds the noise floor by nearly $4\times$, and the mean SNR is 2.94, with 44/48 learners satisfying SNR > 1. The four failures—all MLP-Mixer models trained with ERM—are concentrated on Infograph and Quickdraw, where extreme abstraction combined with no invariance objective yields maximally unstable top-K boundaries. This contrasts with PACS and Office-Home, indicating that under severe heterogeneity, lack of invariance becomes structurally destabilizing. Noise correction yields the largest gain across benchmarks, increasing mean Spearman ρ_S from 0.725 to 0.775 (+0.050), compared to +0.030 (PACS) and +0.040 (Office-Home). This aligns with the mechanism: harder benchmarks allocate more raw CAS variance to

extraction stochasticity, so subtracting Δ_{noise} better isolates Δ_{signal}. Objective- and architecture-wise orderings mirror PACS but at higher absolute levels—IRM (0.107), CORAL (0.131), DANN (0.143), ERM (0.176); ViT-B/16 (0.113), MobileNetV2 (0.128), ResNet-50 (0.155), MLP-Mixer (0.182)—indicating that invariance objectives and attention-based architectures retain their relative regularization advantages even under extreme domain shift.

I Circuit Extraction Hyperparameters

Table 13: **Ablation studies for TopK neurons, bottleneck ratio, and edge threshold.** CAS and OOD performance improve as invariant circuit structure emerges, with optimal correlation at $K = 30$, $r = 4$, and $\epsilon = 10^{-4}$. ($\overline{\text{CAS}}$: Average of CAS across all pair of domains)

K	$\overline{\text{CAS}}$	OOD Acc	ρ_S
5	0.12	68.41	0.71
10	0.21	73.26	0.79
16	0.29	75.90	0.83
30	**0.38**	**77.83**	**0.88**
60	0.41	78.05	0.87
100	0.43	78.14	0.85
164	0.44	78.19	0.83

(a) TopK neurons per layer

r	Params	$\overline{\text{CAS}}$	OOD Acc	ρ_S
1	1,638,400	0.31	78.52	0.80
2	819,200	0.35	78.21	0.85
4	**409,600**	**0.38**	**77.83**	**0.88**
8	204,800	0.42	76.94	0.86
16	102,400	0.48	75.30	0.82

(b) Adapter bottleneck ratio r

ϵ	Edges	$\overline{\text{CAS}}$	OOD Acc	ρ_S
10^{-2}	18	0.22	74.61	0.79
10^{-3}	47	0.31	76.48	0.85
$\mathbf{10^{-4}}$	**83**	**0.38**	**77.83**	**0.88**
10^{-5}	126	0.40	77.95	0.86
10^{-6}	194	0.41	78.02	0.82

(c) Edge threshold ϵ

We ablate the three key hyperparameters of our pipeline, the number of retained neurons per layer K, the adapter bottleneck ratio r, and the edge inclusion threshold ϵ, to assess their impact on circuit alignment and OOD prediction quality. For each ablation, we vary one hyperparameter while fixing the others at their defaults (K=30, r=4, ϵ=10^{-4}) and report mean cross-domain CAS (TK), mean OOD accuracy, and Spearman rank correlation ρ_S between CAS and OOD accuracy, averaged across the four PACS target-domain splits over 48 learners.

Across all three ablations (Table 13), each hyperparameter has a clear operating regime where ρ_S is maximized, with degradation on both sides. For TopK (Table 13 (a)), small K produces circuits too sparse to capture discriminative structure (CAS $= 0.12$ at K=5, $\rho_S = 0.71$), while large K admits peripheral neurons that dilute circuit topology with noise, weakening the correlation ($\rho_S = 0.83$ at K=164). For the bottleneck ratio (Table 13 (b)), narrow adapters (r=16) lack capacity to capture domain-specific computation, inflating CAS artificially (high similarity because the adapters cannot express differences), while wide adapters (r=1) overparameterise the circuit space, fragmenting the invariant sub-circuit across redundant pathways and reducing CAS discriminability. For the edge threshold (Table 13 (c)), aggressive pruning (ϵ=10^{-2}) removes causally relevant edges and collapses circuit topology, while permissive thresholds (ϵ=10^{-6}) retain noise edges that obscure the domain-shift signal. In all three cases, the circuit contains the smallest sufficient set of components for near-optimal task performance. This convergence across independent ablations provides strong evidence that CAS is not an artifact of a particular hyperparameter configuration but reflects a robust structural property of the underlying computation.

The primary departure from PACS is an upward shift in the optimal top-K to $K = 60$ (vs. $K = 30$), reflecting the larger 65-class label space and the need for richer per-layer capacity to resolve fine-grained semantics. Sparse circuits ($K = 5$, $\rho_S = 0.62$) underfit class distinctions, while overly dense circuits ($K = 200$, $\rho_S = 0.75$) dilute the signal. Performance peaks at $K = 60$ ($\rho_S = 0.82$) with a broader plateau than PACS, indicating reduced sensitivity once sufficient capacity is reached. The optimal bottleneck ratio ($r = 4$, $\rho_S = 0.82$) and edge threshold ($\epsilon = 10^{-4}$, $\rho_S = 0.82$) match PACS, implying that adapter capacity and edge selection are governed by architectural dimensionality rather than label cardinality. As in PACS, narrow adapters ($r = 16$) inflate CAS via capacity constraints, whereas wide adapters ($r = 1$) fragment invariant sub-circuits across redundant pathways (Table 11).

The optimal top-K increases to $K = 164$, scaling with the 345-class label space. This matches the information-theoretic requirement of at least $O(\log_2 345) \approx 8.4$ bits of class-discriminative signal per layer, necessitating greater neuron retention. Performance peaks at $K = 164$ ($\rho_S = 0.79$) and declines at $K = 200$ ($\rho_S = 0.77$), confirming that over-inclusion remains detrimental even at scale (Table 12). The optimal bottleneck ratio ($r = 4$) and edge threshold ($\epsilon = 10^{-4}$) are unchanged from PACS and Office-Home, indicating these hyperparameters are architecture-determined constants. The

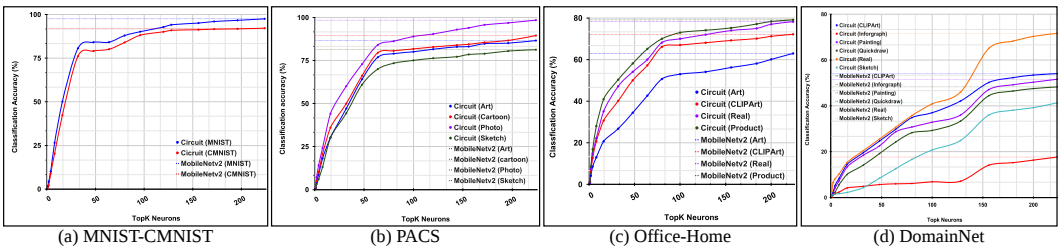
edge count at optimal threshold (83, 158, 327) scales approximately linearly with label cardinality, consistent with growth in class-discriminative inter-layer connections. Across all benchmarks, ρ_S is maximized at a clear interior optimum and degrades symmetrically, ruling out configuration-specific artifacts.

J CAS vs OOD generalization

Fig. 12 (PACS — all targets). CAS (TK) maintains a near-monotone relationship with leave-one-domain-out OOD accuracy across ArtPainting ($\rho_S = 0.91$), Photo ($\rho_S = 0.93$), and Sketch ($\rho_S = 0.88$). The objective-wise clustering is consistent: ERM occupies the low-CAS/low-accuracy regime, IRM the high/high regime, with CORAL and DANN interpolating. The weakest correlation occurs on Sketch, the hardest target, where compressed accuracy ranges induce minor rank inversions—consistent with finite-sample effects when CAS separations are small. Representational baselines (CKA, SVCCA, RSA) yield diffuse scatter across all splits. CKA exhibits saturation (≈0.70–0.88), remaining insensitive to circuit-level reorganization under domain shift. SVCCA and RSA show no consistent objective ordering and occasionally invert the ground-truth ranking (e.g., on Sketch), indicating that subspace fail to capture invariance-relevant circuit structure.

Fig. 13 (Office-Home — all targets). CAS exhibits a near-monotone relationship with OOD accuracy across Art ($\rho_S = 0.81$), Clipart ($\rho_S = 0.76$), Product ($\rho_S = 0.82$), and Real World ($\rho_S = 0.77$), with a clear ERM-to-IRM gradient in every split. Correlation is tightest on Product and Art, and modestly noisier on Clipart and Real World, where stylistic and viewpoint variation increases within-objective rank variability. Relative to PACS, the CORAL–DANN gap narrows, reflecting Office-Home's sensitivity to second-order feature statistics. Representational baselines again fail: CKA saturates (≈0.70–0.88) across accuracy levels, and SVCCA/RSA yield inconsistent orderings. On Clipart, DANN substantially outperforms ERM at nearly identical CKA values, underscoring that activation-level similarity is insensitive to circuit-level routing differences that drive generalization.

Fig. 14 (DomainNet — all targets). CAS preserves a monotone relationship with OOD accuracy across Clipart, Infograph, Painting, Quickdraw, Real, and Sketch, despite wide accuracy variation (5%–70%). The ERM-to-IRM gradient remains visible on every split, including the most difficult Infograph and Quickdraw domains. The inter-learner CAS spread is larger than on PACS or Office-Home, reflecting aggregation over more domain pairs and a richer distributional signal. Second, Quickdraw shows the weakest within-objective clustering: IRM models differ by up to 15 points in OOD accuracy at similar CAS values, likely due to the binary line-drawing regime interacting with seed-level variation. Representational baselines again fail: CKA exhibits pronounced saturation, and SVCCA often inverts IRM–ERM ordering, indicating that activation-level similarity carries no reliable circuit-level signal in the large-scale multi-domain setting.



(a) MNIST-CMNIST (b) PACS (c) Office-Home (d) DomainNet

Figure 16: **Circuits are K-Minimal:** Up to TopK neurons, the performance of the circuits grows exponentially. Then, if we add more neurons, the performance grows very slowly until it matches the original network performance.

K Circuit Minimality and TopK Selection

Circuit minimality requires retaining the smallest sufficient neuron subset that preserves near-optimal accuracy. Varying K reveals a consistent two-phase pattern across benchmarks (see Fig. 16).

Phase I: Rapid convergence: For small K, accuracy rises steeply as the core invariant sub-circuit is recruited. The heavy-tailed importance distribution implies that a small fraction of neurons

carries most causal signal. The elbow occurs around $K \approx 10$–16 (MNIST), 25–40 (PACS), 50–75 (Office-Home), and $K^* = 164$ (DomainNet).

Phase II: Diminishing returns: Beyond the elbow, additional neurons yield marginal gains (< 1–2% per +10 neurons). These peripheral nodes lie near the noise floor and do not alter core topology. Their contributions are down-weighted by $\mathcal{K}$ and largely cancel in CAS, explaining robustness to K once K^* is reached.

DomainNet secondary transition: DomainNet exhibits a second inflection ($K \approx 80$–120), reflecting recruitment of domain-bridging mid-level features needed to span extreme stylistic gaps (e.g., Quickdraw vs. Real), before plateauing at K^*.

Robustness of CAS: All domains within a dataset attain near-optimal accuracy at the same K^*, indicating task-level minimality. Evaluating CAS at $\{0.5K^*, K^*, 1.5K^*, 2K^*\}$ changes ρ_S by < 0.03. Below $0.5K^*$, circuits lose discriminative structure; above $2K^*$, noise edges accumulate, but CAS remains stable due to kernel down-weighting. Bounded node counts and edit distances ensure topological comparability.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have studied how we can predict the OOD generalization accuracy through neural circuit alignment, and the abstract and title clearly reflect that.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in a separate subsection under conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All assumptions are clearly stated and referenced in the statement of the theorems. The proofs appear in the supplementary material; however, a clear intuition has been provided in the main paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper is completely reproducible, all the necessary instructions for implementation have been explained in detail in Section D of the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We used all open source datasets for our experiments. Our code will be available on github upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details and the ablation of the hyperparameters are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We use signal-to-noise ratio for circuit fragility test.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our method isn't resource hungry. A single A40 NVIDIA 48 GB GPU is sufficient for all the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We check it very carefully not to break the anonymity.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide a subsection in the conclusion providing the broader societal impact of the work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We are not proposing any datasets, image generators or using any pre-trained large language models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We are using open source dataset and have properly cited them in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We are not releasing any new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We haven't used LLM to develop the core methodology or experimental results. We have used LLM for writing, editing and grammatical fix.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.