Inside-Out: Measuring Generalization in Vision Transformers Through Inner Workings

Supplementary Material

Table 1. Overview over datasets in pre-deployment setting and their associated domains, along with a shift type for each dataset

Dataset	Shift type	Domain
PACS [9]	style shift	art_painting cartoon photo sketch
Camelyon17 [7]	institution shift	ID Hospitals hospital1 hospital2
Terra Incognita [1]	geographic shift	location 38 location 43 location 46 location 100

A. Details on Datasets

Pre-deployment setting. We use 11 domains collected from three datasets: PACS [9], Camelyon17 from WILDS [7], and the Terra Incognita [1] datasets, see domain and shift type details in Table 1. For PACS and Terra Incognita, we consider all possible in-distribution to out-of-distribution (ID $\rightarrow$ OOD) domain pairs, i.e., we train on one domain(ID domain) and evaluate the model on all the others (OOD domains). For the Camelyon17 dataset we train on the official ID split provided by WILDS, and group the OOD split by hospital id, results in two OOD domains. Then we evaluate on both OOD domains.

Post-deployment setting. In Section 4, we drawn domains from four datasets, FMoW from WILDS [7], PACS [9], Camelyon17 from WILDS [7] with a slightly different setting, and ImageNet [2]; see domain and shift type details in Table 2. For the FMoW dataset, the official data split creates (train, id_val, val, test) by the year the images were taken. We train on the train split and use the id_val split for ID evaluation set. For OOD domains, we split the val (time1) and test (time2) sets by the regions where the images were taken; this results in 10 domains in total. For PACS, we use *Sketch* as the ID domain and treat the remaining three as OOD, because this is the most challenging distribution shift. To expand the number of OOD domains, we randomly split each OOD domain into three subsets, expanding the OOD domains to 9. For Camelyon17, the dataset can be split into 5 hospitals and each hospital contains 10 slides of digitized Whole Slide Images (WSIs).

Table 2. Overview over datasets in post-deployment setting and their associated ID and OOD domains, along with a shift type for each dataset. The hospital id and slide id in Camelyon17 domains as well as the region id in FMoW domains follows the indexing in original dataset metadata. The time id correspondence in FMoW dataset is: time 1 $\rightarrow$ val split and time 2 $\rightarrow$ test split in the original dataset split

Dataset	ID domain	OOD domain
PACS [9]	Sketch	art_painting subset 1-3 cartoon subset 1-3 photo subset 1-3
Camelyon17 [7]	hospital 0 slide 0-7 hospital 1 slide 10-17	hospital 0 slide 8-9 hospital 1 slide 18-19 hospital 2 slide 20-29 hospital 3 slide 30-39 hospital 4 slide 40-49
FMoW [7]	Official ID split	time 1 region 0-4 time 2 region 0-4
ImageNet [2]	ImageNet Validation	ImageNet-Sketch ImageNet-v2 motion blur 0-4 defocus blur 0-4 zoom blur 0-4 snow 0-4 frost 0-4

This results in 50 slides each originating from a specific patient at a specific hospital. We use the first 8 slides from hospital 0 and hospital 1 for training, leaving all other slides for OOD evaluation; this results in 34 OOD domains. For ImageNet, we use the validation set as the ID domain and collect 27 OOD domains from ImageNet-C [6], ImageNet-v2 [14] and ImageNet-Sketch [19].

B. Details of Model Zoo Construction and Model Selection

Pre-deployment setting. To obtain a diverse set of models, we train/fine-tune different pretrained ViTs listed in Table 3. To balance model diversity with computational efficiency, we adopt a two-stage hyperparameter selection strategy. First, we perform an extensive hyperparameter sweep on a representative subset of each dataset. Specifically, for PACS we conduct the sweep on the *photo* domain; for Camelyon17, on a subset of the official in-distribution (ID) dataset; and for Terra Incognita, on location 38. The initial search is performed over an expanded

Table 3. The list of ViTs used in our study. Pretrained weights are sourced from the PyTorch Image Models (timm) library [21], using the model names listed

Model	Timm model name
random init ViT	N/A
OPENAI CLIP	vit_base_patch16_clip_224.openai
LAION2b CLIP	vit_base_patch16_clip_224.laion2b
ImageNet 21k	vit_base_patch16_224_in21k
ImageNet 1k	vit_base_patch16_224.orig_in21k_ft_in1k
MAE	vit_base_patch16_224.mae

Table 4. Hyperparameter sweep grid for constructing the model zoo for the PACS and Terra Incognita pre-deployment experiment

Learning Rate	Batch Size	Weight Decay	Fine-tune
3e-2 1e-2 3e-3	128 256	0 1e-2	Only head Whole model

grid consisting of learning rates $(10^{-1}, 3\times10^{-2}, 10^{-2}, 3\times10^{-3}, 10^{-3})$, batch sizes $(128, 256)$, and weight decays $(0, 10^{-1}, 10^{-2}, 10^{-3})$.Based on the results of this sweep, we construct a reduced hyperparameter grid by selecting configurations that achieve strong performance across all pretraining types. In particular, we ensure that for each pretraining type, at least one configuration within the reduced grid attains near-optimal performance. This pre-selection strategy follows prior practice in [20]. As a result, for PACS and Terra Incognita, we adopt a $3\times2\times2$ grid over learning rate, batch size, and weight decay. Due to computational constraints, we further reduce the grid for Camelyon17 to $3\times2\times1$. The final hyperparameter configurations are summarized in Table 4 and Table 5, respectively.

Post-deployment setting. In the post-deployment experiments, we are focusing on a single model for each dataset and the model selection is performed from the pre-constructed model zoo based on the $\mathrm{DDB}_{\mathrm{out}}$ criterion, or by directly adopting pretrained models when appropriate for the dataset. For PACS, we directly select models from the model zoo obtained in the pre-deployment stage. For Camelyon17 and FMoW, since the in-distribution (ID) and out-of-distribution (OOD) settings differ from those used during pre-deployment, we conduct an additional lightweight hyperparameter sweep using a reduced $2\times2\times1$ grid over learning rate $(10^{-2}, 3\times10^{-3})$, batch size (256), and weight decay $(0, 10^{-2})$. For ImageNet, we directly adopt ImageNet-1K pretrained models from the Timm library without further fine-tuning.

Table 5. Hyperparameter sweep grid for constructing the model zoo for the Camelyon17 pre-deployment experiment

Learning Rate	Batch Size	Weight Decay	Fine-tune
3e-2 1e-2 3e-3	128 256	0	Only head Whole model

C. Model Performances Across Different Pretraining

We report the average ID and OOD performance for each pretraining strategy, aggregated over all models in the corresponding model zoo and across all generalization tasks. For each model, ID performance is evaluated on the ID test set, while OOD performance is computed as the mean accuracy across all remaining domains. We then average these ID and OOD metrics over all models sharing the same pretraining strategy. Finally, for each dataset, we report the mean ID accuracy, mean OOD accuracy, and the corresponding ID–OOD performance gap in Table 6.

D. Detailed Definitions of Distance Metrics for the Circuit Shift Score

Here we provide the formal definitions of distance metrics for the graph-based $\mathrm{CSS}_{(g,\cdot)}$ and vector-based $\mathrm{CSS}_{(v,\cdot)}$ variants.

Graph-based distance metrics: Let $C_1 = (\mathcal{V}, \mathcal{E}, W_1)$ and $C_2 = (\mathcal{V}, \mathcal{E}, W_2)$ be two circuit graphs of the same model with respect to two different input distributions.

- laplacian Spectrum Distance [18]: Let $\{\lambda_i(C)\}_{i=1}^{|\mathcal{V}|}$ be the ordered eigenvalues of circuit graph C's Laplacian matrix. The distance is the Euclidean distance between the eigenvalue vectors of the two graphs:

$$d_{\text{Laplacian}}(C_1, C_2) = \sqrt{\sum_{i=1}^{|V|} \left(\lambda_i(C_1) - \lambda_i(C_2)\right)^2}$$

- NetLSD (Net Laplacian Spectral Descriptor) [17]: The NetLSD signature is a vector derived from the solution to the heat equation on a graph. This results in a graph size agnostic feature extraction function. Consequently, given C_1 and C_2, we prune the circuit graph as $C_1^k = (\mathcal{V}_1, \mathcal{E}_1, W_1)$ and $C_2^k = (\mathcal{V}_2, \mathcal{E}_2, W_2)$ by retaining the top-k edges following Hanna et al. [5] and extract the NetLSD signature vectors from both circuits. The distance is the L2 distance between these signature vectors. For a full definition, we refer the reader to [17].
- Jacarrd Similarity: We first derive the pruned circuit graphs C_1^k and C_2^k. This metric measures the overlap of

Table 6. Model generalization comparison. Columns show averaged ID, OOD accuracy and OOD-ID Gap of each model over all domains within the dataset.

Model	Benchmark	ID accuracy	OOD accuracy	OOD-ID Gap
random init	PACS	0.477 ± 0.018	0.201 ± 0.010	0.276 ± 0.156
	Camelyon17	0.934 ± 0.009	0.686 ± 0.014	0.248 ± 0.006
	Terra Incognita	0.565 ± 0.014	0.221 ± 0.011	0.344 ± 0.023
	FMoW	0.547 ± 0.020	0.497 ± 0.009	0.051 ± 0.003
ViT-B MAE pretrained	PACS	0.580 ± 0.017	0.242 ± 0.012	0.338 ± 0.011
	Camelyon17	0.912 ± 0.021	0.836 ± 0.018	0.076 ± 0.005
	Terra Incognita	0.633 ± 0.017	0.220 ± 0.008	0.413 ± 0.023
	FMoW	0.569 ± 0.007	0.512 ± 0.013	0.057 ± 0.011
ViT-B openai CLIP	PACS	0.921 ± 0.010	0.695 ± 0.021	0.226 ± 0.014
	Camelyon17	0.955 ± 0.010	0.881 ± 0.010	0.075 ± 0.012
	Terra Incognita	0.772 ± 0.014	0.334 ± 0.012	0.438 ± 0.018
	FMoW	0.639 ± 0.015	0.578 ± 0.008	0.060 ± 0.005
ViT-B laion2b CLIP	PACS	0.913 ± 0.011	0.693 ± 0.024	0.219 ± 0.016
	Camelyon17	0.962 ± 0.008	0.887 ± 0.010	0.075 ± 0.010
	Terra Incognita	0.750 ± 0.020	0.338 ± 0.012	0.412 ± 0.028
	FMoW	0.647 ± 0.021	0.591 ± 0.009	0.055 ± 0.017
ViT-B ImageNet 21k	PACS	0.966 ± 0.003	0.677 ± 0.013	0.289 ± 0.014
	Camelyon17	0.979 ± 0.004	0.917 ± 0.002	0.062 ± 0.005
	Terra Incognita	0.807 ± 0.012	0.344 ± 0.008	0.463 ± 0.016
	FMoW	0.618 ± 0.012	0.575 ± 0.020	0.043 ± 0.019
ViT-B ImageNet 1k	PACS	0.922 ± 0.005	0.658 ± 0.012	0.264 ± 0.008
	Camelyon17	0.964 ± 0.007	0.908 ± 0.008	0.056 ± 0.002
	Terra Incognita	0.771 ± 0.012	0.286 ± 0.011	0.485 ± 0.017
	FMoW	0.602 ± 0.015	0.547 ± 0.010	0.055 ± 0.006

the edge sets over the two pruned circuits. Given the edge sets in the two circuits, denoted as $\mathcal{E}_1$ and $\mathcal{E}_2$, the Jaccard distance is defined as:

$$d_{\text{Jaccard}}(\mathcal{E}_1, \mathcal{E}_2) = 1 - \frac{|\mathcal{E}_1 \cap \mathcal{E}_2|}{|\mathcal{E}_1 \cup \mathcal{E}_2|}$$

Vector-based distance metrics: Let $e_1, e_2 \in \mathbb{R}^E$ be the two vectors of edge weights from two circuits, defined over the full edge set $\mathcal{E}$ of the model architecture.

- Cosine Similarity: compute the cosine similarity between the two vectors.

$$d_{\text{Cosine}}(e_1, e_2) = 1 - \frac{e_1 \cdot e_2}{\|e_1\|\|e_2\|}$$

- Spearman Ranking Correlation Coefficient (SRCC): Measures the rank correlation. Let $rg(e)$ be the rank vector of e.

$$\text{SRCC}(e_1, e_2) = \rho(\text{rg}(e_1), \text{rg}(e_2))$$

- ℓ^2 Distance (Euclidean Distance): The standard Euclidean distance between the two vectors.

$$d_{\ell^2}(e_1, e_2) = \|e_1 - e_2\|_2 = \sqrt{\sum_{i=1}^{|E|} (e_{1,i} - e_{2,i})^2}$$

E. Calibration Set Construction Detail

In the *post-deployment* setting, our goal is to monitor potential performance degradation and identify "silent failures" of the model. To support reliable threshold calibration for our circuit-based metric, we construct a diverse corruption set that simulates realistic distribution shifts. The corruption set used in our experiments includes: (1) 9 types of Stylization, cartoon, contour, edge, edge-enhance, pallete, posterize, solarize and emboss. These corruptions introduce texture, edge, and color-style distortions, capturing a wide range of appearance changes that real-world data may undergo. (2) `fog`, `frost`, `gaussian noise`, `shot noise`, `defocus blur`, and `snow`, each applied with severity levels 1–5. We adopt these corruptions because they are widely used to benchmark robustness and model

degradation under natural image perturbations.

F. Detailed Scatter Plots

Pre-deployment setting. we evaluate all metrics across the full collection of 34 ID → OOD tasks. Figure 1, 2 and 3 presents the corresponding scatter plots in PACS, Camelyon17 and Terra Incognita, illustrating the relationship between each metric and GT OOD performance.

Post-deployment setting. Figure 4 displays the complete set of scatter plots for every metric across datasets, enabling a comprehensive comparison of their predictive behaviors.

G. Circuit Discovery Method Benchmark

We benchmark five existing circuit discovery methods on vision tasks, following the standardized evaluation protocol introduced by Mueller et al. [13]. Our goal is to assess the faithfulness and efficiency of each method in identifying circuits that reliably capture the causal mechanisms underlying model predictions.

Experimental setup. We evaluate circuit discovery across three vision benchmarks: Color-MNIST [8], Waterbirds [15], and ImageNet [2]. Due to the large size of ImageNet, we randomly sample 10000 samples from the validation set for evaluation. The evaluated methods include: (1) Edge Activation Patching (EActP) [12], (2) Edge Attribution Patching (EAP) [16], (3) EAP with Integrated Gradients (EAP-IG), with two variants: EAP-IG-inputs [5] and EAP-IG-activation [11], following [5], we set gradient integration steps to 5, (4) Information Flow Route (IFR) [4], and (5) Uniform Gradient Sampling (UGS) [10].

Table 7. CMD(↓) scores across circuit discovery methods. All evaluations were performed using mean ablations. We **bold** and underline the best and second-best methods per column, respectively.

Method	Color-MNIST		Waterbirds	ImageNet
	small-ViT	ViT-B/16	ViT-B/16	ViT-B/16
Random	0.555	0.748	0.754	0.732
EActP	0.466	0.095	0.271	0.360
EAP	0.332	0.103	0.299	0.376
EAP-IG-inp	0.567	**0.063**	0.242	0.325
EAP-IG-act	0.452	0.076	0.327	0.381
IFR	0.724	0.565	0.590	0.585
UGS	**0.300**	0.114	**0.053**	**0.102**

Faithfulness metrics. To quantify how faithfully an extracted circuit C^k captures the model's causal structure, we adopt the faithfulness definition from Mueller et al. [13].

Table 8. CPR(↑) scores across circuit discovery methods. All evaluations were performed using mean ablations. We **bold** and underline the best and second-best methods per column, respectively.

Method	Color-MNIST		Waterbirds	ImageNet
	small-ViT	ViT-B/16	ViT-B/16	ViT-B/16
Random	0.263	0.274	0.260	0.299
EActP	1.679	0.732	0.698	0.804
EAP	1.658	0.712	0.570	0.655
EAP-IG-inp	**2.033**	**0.902**	0.706	0.813
EAP-IG-act	1.658	0.858	0.656	0.810
IFR	1.025	0.499	0.409	0.410
UGS	1.231	0.893	**0.946**	**0.897**

Given a full model $\mathcal{M}$ and its circuit subgraph C^k (retaining activations on the top-k attributed edges), the faithfulness score is defined as:

$$f(C^k, \mathcal{M}; \mathrm{KL}) = \frac{\mathrm{KL}(y \,\|\, y'_{C^k}) - \mathrm{KL}(y \,\|\, y'_{\emptyset})}{1 - \mathrm{KL}(y \,\|\, y'_{\emptyset})}, \quad (1)$$

where y and y'_{C^k} denote the clean output of the model without ablating any activation, and the counterfactual output of the model, with all edges outside of C^k is ablated. $\mathrm{KL}(\cdot \,\|\, \cdot)$ denotes the Kullback–Leibler divergence. As mentioned in Section 2, we ablate edges with mean ablation, i.e., the output of an edge is neutralized by replacing its corresponding activations with their pre-computed mean over the input dataset. Here, $y'_{\emptyset}$ denote the outputs from the empty circuit, effectively ablating all edges. This formulation measures the proportion of the model's explanatory power preserved by the circuit, normalized between the trivial (empty) and complete models. Following Mueller et al. [13], we evaluate each method using two aggregate metrics, the integrated circuit performance ratio (CPR) and the integrated circuit-model distance (CMD), Instead of selecting a single circuit threshold (which would make evaluation highly sensitive to hyperparameter choices), both metrics aggregate faithfulness continuously over all circuit sizes k:

$$\mathrm{CPR} = \int_0^1 f(C^k)\, dk, \quad \mathrm{CMD} = \int_0^1 |1 - f(C^k)|\, dk, \quad (2)$$

where $f(C^k)$ is the faithfulness at fraction k of retained edges. CPR captures how much of the model's behavior is positively preserved across circuit scales, with higher CPR indicates that a method consistently identifies components that support the model's predictions. CMD instead measures the overall deviation from perfect fidelity, with lower CMD indicates that a method successfully identifies components with any strong effect on the model's computation,

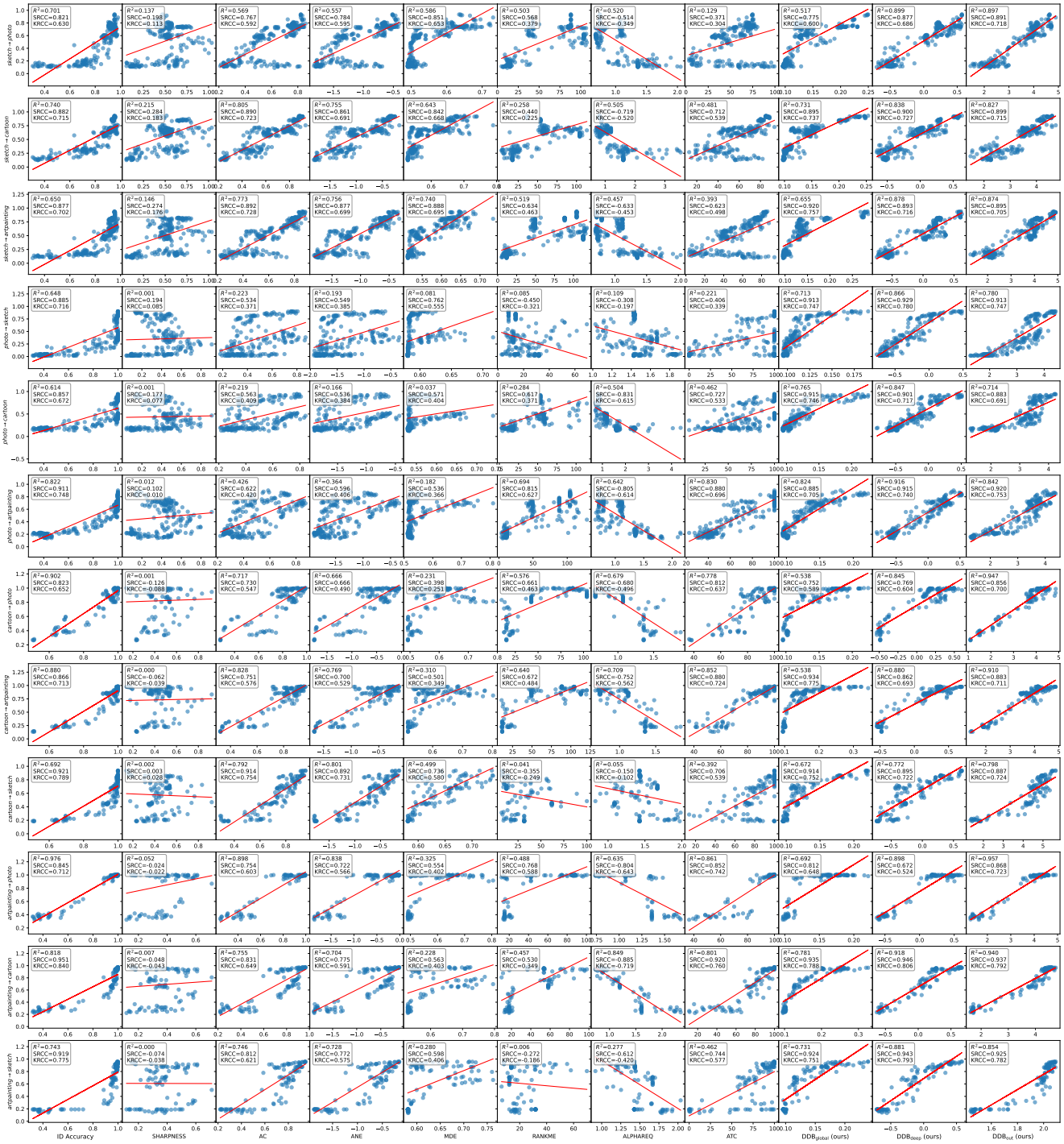

Figure 1. All pre-deployment metrics' scatter plots for all 12 (ID→OOD) generalization tasks in PACS. Each blue dot represents one trained network. Each row represent one generalization task, and the name is shown in the y-axis label. Each column represent one pre-deployment metric. Y-axis shows models' performance on the OOD domain, and x-axis shows value of the corresponding metric. All raws share the same y-axis.

making it better suited for uncovering the full underlying algorithm. In practice, these integrals are approximated using discrete samples of k, following the implementation protocol of Mueller et al. [13].

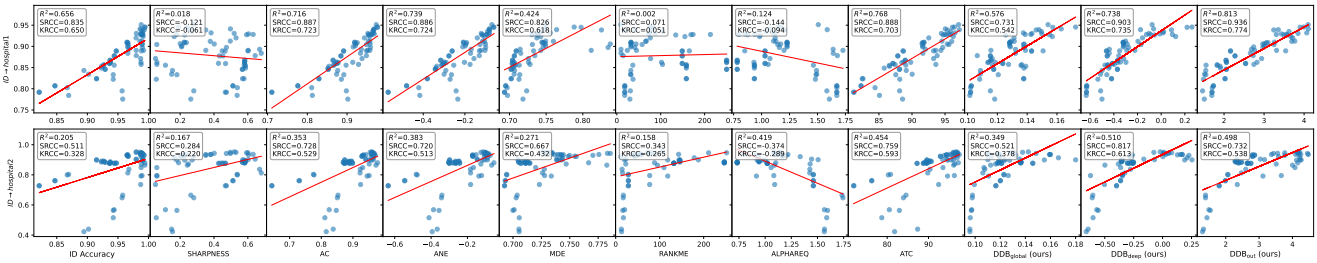

Figure 2. All pre-deployment metrics' scatter plots for the 2 (ID→OOD) generalization tasks in the Camelyon17 dataset.

Results and analysis. Tables 8 and 7 report CPR and CMD across datasets and methods. We observe that Uniform Gradient Sampling (UGS) achieves the highest overall faithfulness, followed closely by EAP-IG-inputs. However, UGS incurs prohibitive computational cost due to repeated gradient sampling, making it impractical for large-scale analyses. In contrast, EAP-IG-inputs achieves comparable faithfulness with significantly lower computational overhead, offering a practical balance between interpretability fidelity and efficiency. Consequently, we adopt EAP-IG-inputs as the primary circuit discovery method in the remainder of our work.

H. Training Dynamic of the DDB Metric

We present the training dynamics of the Dependency Depth Bias (DDB) metric alongside the corresponding OOD performance for all three pre-deployment datasets in Figure 5. Across all datasets, DDB closely follows the trajectory of OOD accuracy throughout training, confirming that it captures the evolving generalization behavior of the model.

I. Ablation on τ for DDB Metrics

To better understand the sensitivity of the Dependency Depth Bias (DDB) metric to its hyperparameter τ, we conduct an extensive ablation across all three DDB variants. Recall that $\tau \in (0, 0.5]$ controls the partitioning of shallow versus deep layers, influencing how the metric weighs shallow- versus deep-layer circuit contributions. We have shown the ablation results for $\mathrm{DDB}_{\mathrm{out}}$ in Section 3. Here we report the results for $\mathrm{DDB}_{\mathrm{out}}$ and $\mathrm{DDB}_{\mathrm{out}}$ in Table 9 and Table 10, respectively. While DDB values vary noticeably with different choices of τ, the results reveal a consistent and optimal value that yields the strongest correlation with OOD performance. These findings indicate that, although DDB is sensitive to τ, selecting τ with the optimal value leads to consistently strong predictive performance.

J. Generalization Motifs

Here, we visualize the extracted *Generalization Motifs* obtained via CCA analysis for all *pre-deployment* generalization tasks (Figure 6). Each motif is shown as a heatmap that highlights the pro- and anti-generalization inter-layer connections of a given task T. Although the motifs differ across tasks, they also exhibit consistent global patterns. In particular, we observe a strong contrast between the correlation strengths of shallow versus deep layers, a recurring phenomenon that directly motivates the design of our DDB metric.

Table 9. Ablation on $\mathrm{DDB}_{\mathrm{deep}}$'s hyperparameter τ. The results consistently show that $\tau = 0.3$ yields the strongest correlation scores.

Score	τ				
	0.1	0.2	0.3	0.4	0.5
R^2	0.441	0.467	**0.750**	0.491	0.433
SRCC	0.780	0.788	**0.853**	0.743	0.630
KRCC	0.592	0.609	**0.681**	0.565	0.478

Table 10. Ablation on $\mathrm{DDB}_{\mathrm{global}}$'s hyperparameter τ. The results consistently show that $\tau = 0.1$ yields the strongest correlation scores.

Score	τ				
	0.1	0.2	0.3	0.4	0.5
R^2	**0.683**	0.516	0.541	0.500	0.447
SRCC	**0.786**	0.653	0.655	0.606	0.553
KRCC	**0.620**	0.514	0.519	0.478	0.433

K. Overhead Analysis

To understand the practical feasibility of using circuit metrics for model evaluation and selection, we analyze the computational overhead introduced by circuit discovery and circuit metric calculation.

Circuit discovery is the major overhead. Confidence-based metrics require only a single forward pass to obtain logits. This operation is highly efficient and scales linearly with the number of input samples n. Empirically, on an NVIDIA A6000 GPU, a forward pass with a batch size of

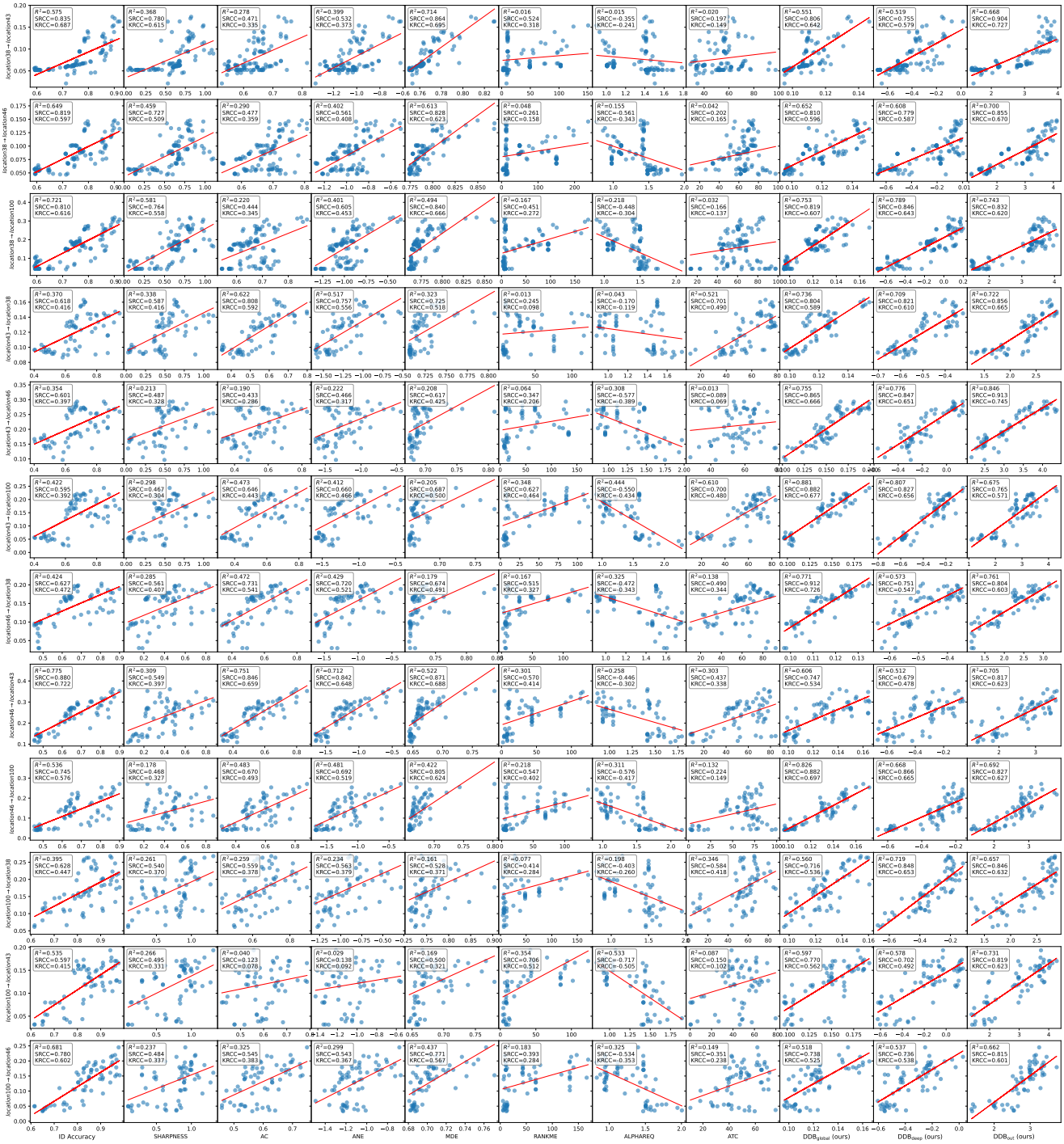

Figure 3. All pre-deployment metrics' scatter plots for all 12 (ID→OOD) generalization tasks in the Terra Incognita dataset.

32 through a ViT-B/16 model takes approximately 123 ms. Circuit discovery, in contrast, requires gradient-based estimation of edge-level contributions. The EAP-IG [5] method used in our experiments performs one forward pass followed by a fixed number of backward passes; following Hanna et al. [5], we set this number to 5. Under identical hardware and batch size, full circuit discovery requires approximately 1585 ms per batch, which is the major bottleneck.

Figure 7 further break down the computation overhead

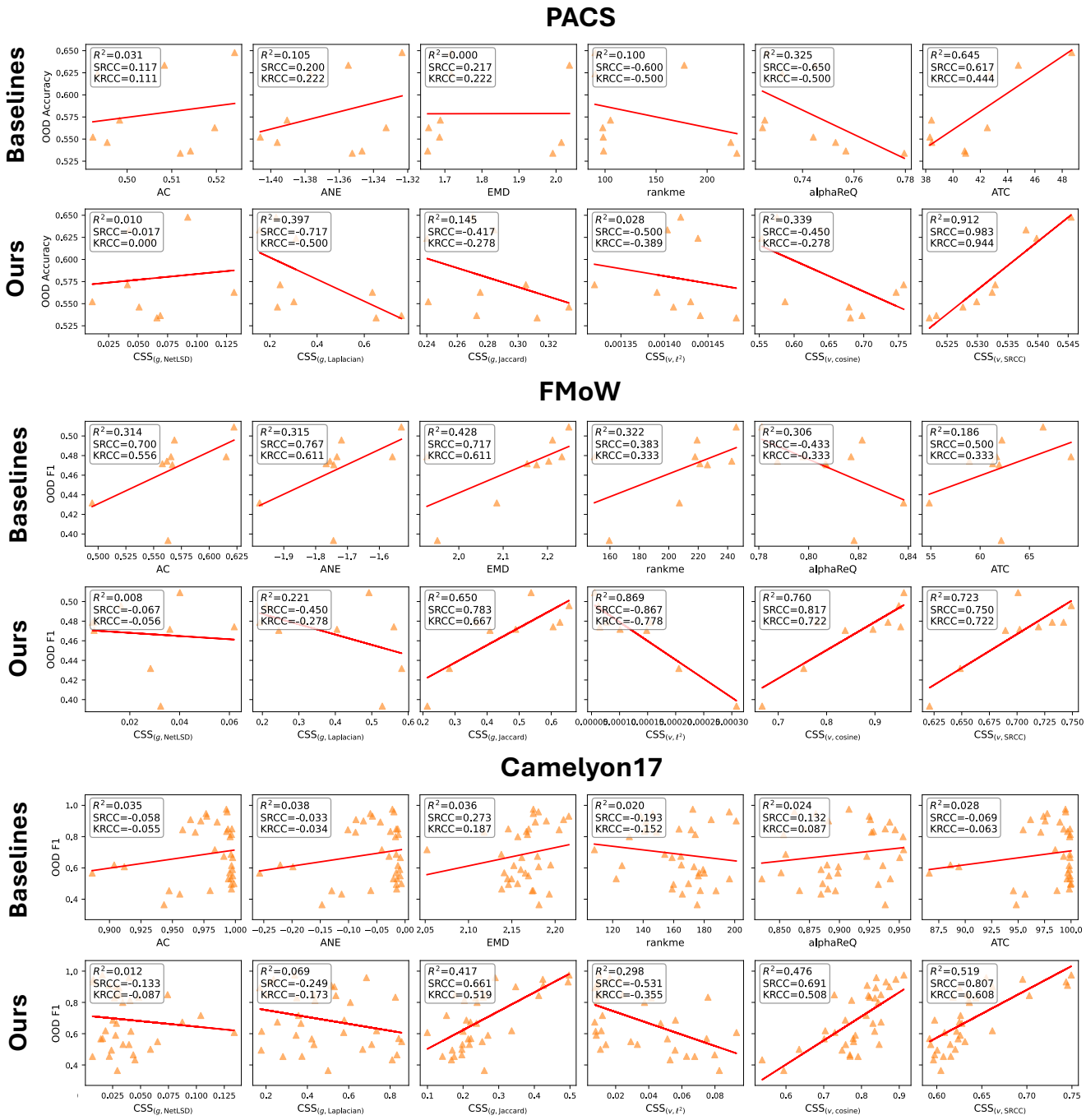

Figure 4. All post-deployment metrics' scatter plot. In each plot, the yellow triangles represent OOD domains in the dataset, Y-axis shows models' GT performance on the OOD domain, and x-axis shows value of the corresponding metric. All rows share the same y-axis.

in circuit discovery, showing that the backward pass is the primary computational bottleneck. Hence we propose two solutions to accelerate circuit discovery. (1) In this work we adopt EAP-IG for circuit discovery, which requires multiple rounds of forward and backward passes due to Integrated Gradients (IG). Using EAP instead can eliminate multiple IG passes. Profiling results show that this achieves approximately a 5× speedup, which means the integration steps in the IG method could be reduced to directly optimize runtime. (2) Backward passes can be further approximated with zeroth-order gradient approximation [3], further improving efficiency while reducing memory usage and en-

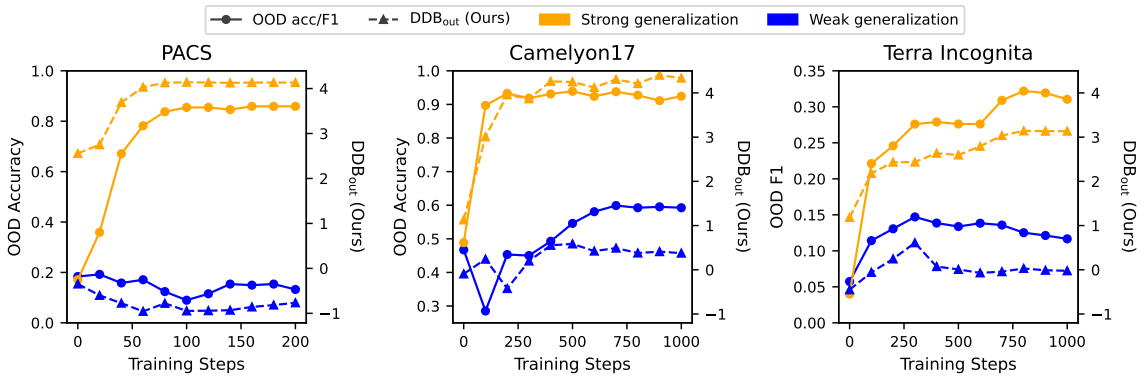

Figure 5. The training dynamics of OOD performance (left y-axis) vs. our $\mathrm{DDB}_{\mathrm{out}}$ metric (right y-axis) on PACS, Camelyon17 and Terra Incognita. Across all datasets, DDB closely follows the trajectory of OOD accuracy throughout training, confirming that it captures the evolving generalization behavior of the model.

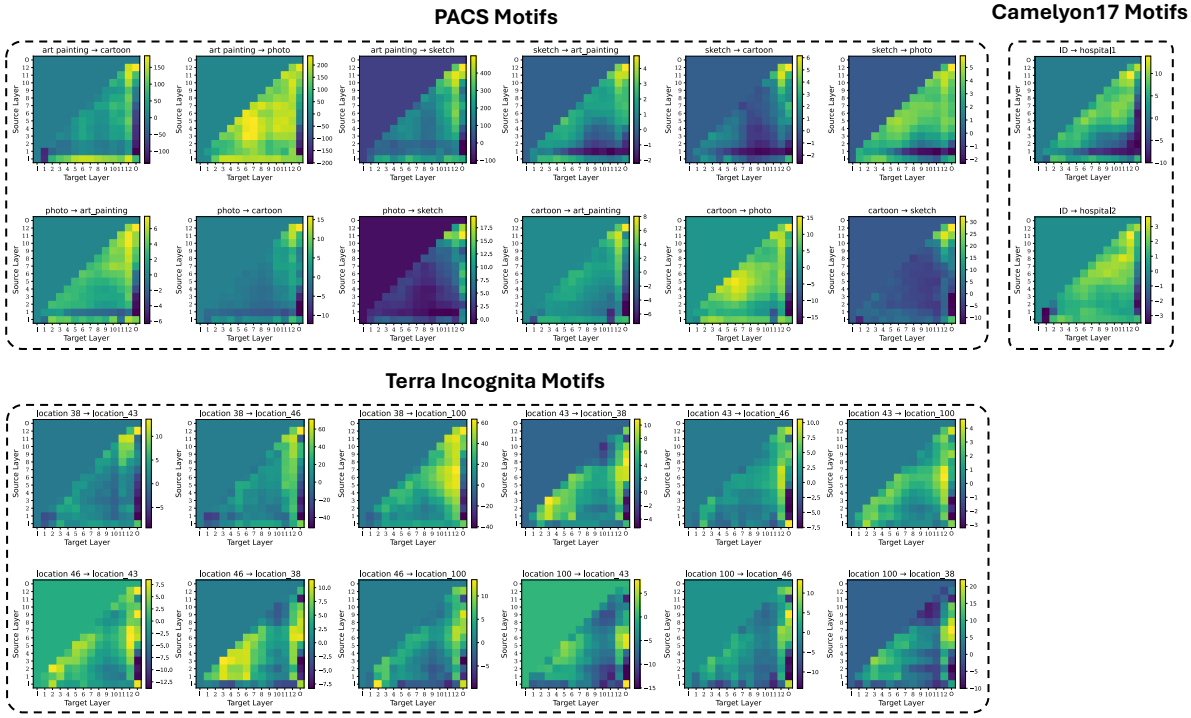

Figure 6. Pre-deployment *Generalization Motifs* $\mathbf{v}_T$ (Eq.3) of all tasks. Brighter regions indicate the inter-layer dependencies positively correlated with OOD generalization; darker regions indicate negative correlations.

abling larger batch parallelism.

Metric calculation overhead is negligible. After circuits are discovered, the computation of circuit metrics (e.g., DDB, CSS) involves only graph-level operations on the induced circuit structure. These operations scale as $\mathcal{O}(L^2)$, where L is the number of Transformer layers. Importantly, this does not scale with number of input samples. As a result, each circuit graph needs to be processed only once.In practice, metric computation takes approximately 52,ms per circuit, which is negligible compared to the cost of circuit discovery. Furthermore, circuits can be aggregated across multiple batches prior to metric evaluation, amortizing this overhead even further.

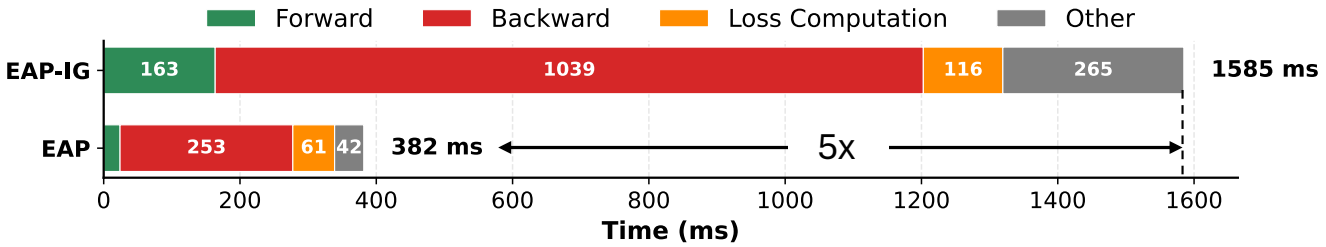

Figure 7. Circuit discovery runtime profile for a single batch (size 32). Backward pass is the major bottleneck, and replacing EAP-IG with EAP yields approximately 5× speedup.

References

[1] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 1

[2] 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. 1, 4

[3] Malladi et al. Fine-tuning language models with just forward passes. *NeurIPS*, 2023. 8

[4] Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale. *arXiv preprint arXiv:2403.00824*, 2024. 4

[5] Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. *arXiv preprint arXiv:2403.17806*, 2024. 2, 4, 7

[6] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1

[7] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021. 1

[8] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 4

[9] 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. 1

[10] Maximilian Li and Lucas Janson. Optimal ablation for interpretability. *Advances in Neural Information Processing Systems*, 37:109233–109282, 2024. 4

[11] Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024. 4

[12] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022. 4

[13] Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. Mib: A mechanistic interpretability benchmark, 2025. 4, 5

[14] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 1

[15] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 4

[16] Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*, 2023. 4

[17] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. Netlsd: hearing the shape of a graph. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2347–2356, 2018. 2

[18] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 2

[19] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019. 1

[20] Florian Wenzel, Andrea Dittadi, Peter Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, et al. Assaying out-of-distribution generalization in transfer learning. *Advances in Neural Information Processing Systems*, 35:7181–7198, 2022. 2

[21] Ross Wightman. PyTorch Image Models. 2