Cross-Model Circuit Discovery

Harrish Thasarathan [1 2] **Matthew Kowal** [3] **Thomas Fel** [3] **Konstantinos G. Derpanis** [1 2]

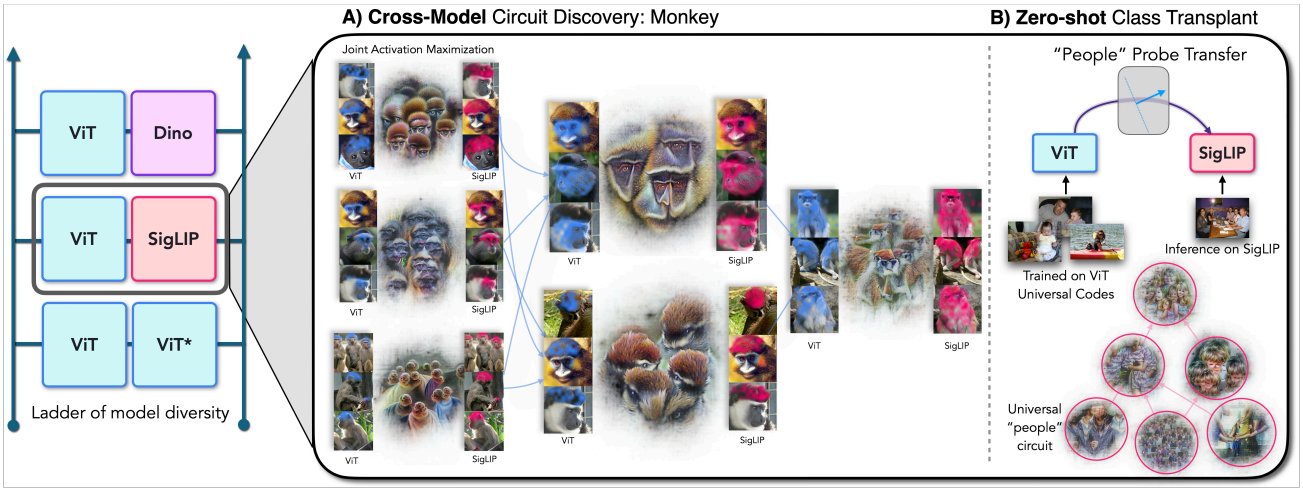

Figure 1. Overview of Cross-Model Circuit Discovery. (A) We introduce a learning based framework for discovering concept-based circuits that exist in common between separately trained deep neural networks. We achieve this by learning to align a per-layer concept basis, allowing us to construct comparable circuits from a shared pool of interpretable concepts across layers. (B) We demonstrate that cross-model circuits are functionally load-bearing through zero-shot transplant of a concept-space probe trained on one model to the other and visualize the corresponding cross-model circuit.

Abstract

Consider two large vision models. They process the same image and both correctly predict the class "rabbit." How much of the circuit computation along the way was shared? Model diffing offers a natural lens on this question. So far, however, it has largely operated on a single layer and at the level of representations rather than circuits. In this work, we introduce Universal Circuits (UCs), enabling model diffing at the circuit level. Specifically, we extend CLTs across both layers and models, with losses that encourage sparsity for interpretability and output fidelity for faithfulness. We train UCs between pairs of standard large vision models. We find that a compact cross-model intersection of pruned class circuits, typically a few hundred universal (shared) features per pair, produces 89-98% of full-circuit classification accuracy in both models, while a complementary set of universal features (hundreds per pair) is kept by only one model's circuit, reflecting how each model weights shared concepts differently in its own representations. To demonstrate a downstream use of UC, we perform model *surgery*: a class is successfully transferred from one model to another with no gradient steps taken in the new model. More broadly, our work demonstrates that large vision models leverage shared multi-layer algorithms for downstream performance, and that these algorithms can be discovered, compared, and reused across models.

[*]Equal contribution [1]York University, Toronto, Canada [2]Vector Institute, Toronto, Canada [3]Goodfire AI. Correspondence to: Harrish Thasarathan <harryt@yorku.ca>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

A growing body of evidence suggests that deep neural networks, despite differences in training data and objectives, independently develop (partially) shared representations of the world (Moschella et al., 2022; Huh et al., 2024; Sucholutsky et al., 2023; Muttenthaler et al., 2023; Hermann

& Lampinen, 2020; Chen et al., 2024) Recent work on cross-model concept discovery has given this hypothesis an interpretable footing: by jointly training sparse dictionaries across models, one can recover shared *features* that fire on the same inputs across architectures and objectives (Thasarathan et al., 2025; Bhalla et al., 2024; Kowal et al., 2024b; Parekh et al., 2024; Dhimoila et al., 2026; Papadimitriou et al., 2025; Pach et al., 2025) Because these features are aligned across models, they also make model *diffing* possible: by measuring which features are shared and which are model-specific, we can study how training data, objectives, and architectures can impact important model attributions (i.e., generalization), or audit safety-relevant behaviours (e.g., diagnosing failure modes) (Geirhos et al., 2020; D'Amour et al., 2020; Raji & Buolamwini, 2019; Buolamwini & Gebru, 2018; Anwar et al., 2024). Yet a feature, however semantically aligned, is not how models make decisions. Instead, they composes concepts across layers into a *circuit*: a structured sequence of computations that turns inputs into decisions (Olah et al., 2020b; Cammarata et al., 2020; Olah et al., 2020a; Bricken et al., 2023; Templeton et al., 2024; Ameisen et al., 2025; Marks et al., 2025; Conmy et al., 2023; Wang et al., 2023; Meng et al., 2022; Kowal et al., 2024c).

Robust methods for discovering shared and unique circuits across different models remains unexplored. Existing cross-model alignment methods (Thasarathan et al., 2025; Nasiri-Sarvi et al., 2025; Jiralerspong & Bricken, 2026; Lindsey et al., 2024; Bhalla et al., 2024; Kowal et al., 2024b; Parekh et al., 2024; Papadimitriou et al., 2025; Dhimoila et al., 2026) operate at the level of individual layers, but a layer provides only a snapshot of computation. Consequently, sharing a feature at one layer says little about whether two models actually share the algorithm that produced it or the computation it supports downstream (Makelov et al., 2023; Paulo & Belrose, 2025; Fel et al., 2025a; Gresele et al., 2020). Conversely, various interpretability tools have emerged for tracing multi-layer circuits in both vision (Kowal et al., 2024c; Rajaram et al., 2024; Joseph, 2023; Joseph et al., 2025b; Fel et al., 2023c; Lepori et al., 2023) and language models (Ameisen et al., 2025; Marks et al., 2025; Meng et al., 2022; Wang et al., 2023; Huben et al., 2024; Conmy et al., 2023), but they consider single-models and offer no mechanism for comparing circuits across networks. A method for cross-model circuit analysis that combines both multi-layer circuit-tracing properties with the cross-model alignment properties of universal concept dictionaries, would enable a more holistic view of how models share computation.

To this end, we introduce Universal Circuits (UCs), a generalization of Cross-Layer Transcoders (CLTs) trained jointly across multiple models to expose both shared and model-specific circuits throughout their full classification stacks. Whereas CLTs model circuits within a single network, UCs do so both within and across networks simultaneously, combining hard concept alignment, model-specific capacity, hybrid cross-prediction, and classifier-in-the-loop training so that the shared feature space is aligned for both activation reconstruction and downstream classification. To probe how this functional alignment behaves under increasing representational divergence, we organize evaluation around pairs of standard vision transformers spanning a ladder of architectural and training differences (ViT (Dosovitskiy et al., 2020), DINO (Caron et al., 2021), and SigLIP (Zhai et al., 2023)). We find that vision transformers converge on a small set of shared circuits accounting for a disproportionately large fraction of each model's classification capability, mirroring the single-layer findings of Thasarathan et al. (2025): the *intersection circuit*—the shared circuit both models' pruners independently keep—constitutes only $\sim$30% of the union of the two pruned circuits, yet recovers $\sim$90% of full-model classification. Features outside this intersection play a smaller functional role, with model-specific features becoming more important as divergence grows, giving a graded rather than binary picture of circuit-level convergence. Finally, we demonstrate UCs as a tool for zero-shot classification on unseen classes: given a concept-space linear probe trained on one model's aligned features for a held-out class, we transplant it into another model's residual stream via closed-form ridge regression, enabling class-level circuit extraction and achieving strong cross-model classification compared to naive residual transplantation ($\sim$0.94 vs. $\sim$0.50 test AUROC).

To recap, our main contributions are as follows. We introduce UC, the first method for discovering shared circuits across both models and layers, and apply them to large vision models. To make this possible, we adapt cross-layer transcoders (Ameisen et al., 2025) from language to vision with a softmax-KL objective on class logits and class-level circuits for classification, and introduce a *hybrid cross-reconstruction* objective that combines hard index binding with separate shared and model-specific capacity. Applying UC to a *diversity ladder* of vision-transformer pairs (ViT–ViT, DINOv1–ViT, SigLIP–ViT), we show several findings: (i) aligned models converge on a small number of shared circuits that account for a disproportionate share of classification ability, (ii) a long tail of less functionally important features running from aligned to model-specific, and (iii) early layers have a larger portion of shared circuitry than deeper layers in absolute count, with per-layer agreement decaying with model divergence. We further show that this shared feature space supports zero-shot probe transplantation for held-out classes via a closed-form residual-stream readout, enabling class-level per-model circuit comparison for novel classes and demonstrating a practical use of the framework.

2. Related Work

Vision interpretability. Early vision interpretability used gradients or masking-based approaches to determine the importance of individual pixels to outputs (Simonyan et al., 2013; Selvaraju et al., 2017; Bach et al., 2015; Petsiuk et al., 2018; Fong & Vedaldi, 2017; Fel et al., 2021). These methods show *where* the model is looking but not *what* it is looking at (Hase & Bansal, 2020; Hsieh et al., 2021; Nguyen et al., 2021; Colin et al., 2022; Kim et al., 2022; Sixt et al., 2020; Adebayo et al., 2018). Concept-based interpretability addresses this issue by probing or clustering a models activations into a set of supervised or unsupervised localized features (Kim et al., 2018; Bau et al., 2017; Ghorbani et al., 2019; Fel et al., 2023b; Oikarinen & Weng, 2023; Rao et al., 2024). This approach has seen success in extracting human-interpretable concepts from both images (Zhang et al., 2021; Fel et al., 2023c;b; Graziani et al., 2023; Vielhaben et al., 2023) and videos (Kowal et al., 2024a). Sparse Autoencoders (SAEs) (Cunningham et al., 2023; Bricken et al., 2023; Rajamanoharan et al., 2024; Gao et al., 2025a; Surkov et al., 2024; Gorton, 2024; Bhalla et al., 2024), a dictionary learning method (Tošić & Frossard, 2011; Rubinstein et al., 2010; Elad, 2010; Mairal et al., 2014; Dumitrescu & Irofti, 2018) building on early work in sparse coding (Olshausen & Field, 1996; 1997; Lee et al., 2006; Foldiak & Endres, 2008; Rentzeperis et al., 2023), have emerged as the tool of choice for unsupervised concept learning. First popularized in language interpretability, they have since been applied to a range of frontier vision models (Joseph et al., 2025a;b; Fel et al., 2025a; Rao et al., 2024) Despite SAEs producing causally relevant features, single-layer decompositions are not unique: many sparse codes can fit the same activations, and there is no guarantee that any one of them recovers the underlying generating factors. Recent geometric analyses of SAE features (Fel et al., 2025b) further suggest that single-layer structure alone may be insufficient, as features at a given layer appear to be composed from those at earlier layers.

Circuit-based interpretability. *Circuits* discover concepts as nodes at multiple layers in a model and compute attributions between nodes. One line of work aim to discover circuits in models where neurons represent concepts (Cammarata et al., 2020; Olah et al., 2020a; Conmy et al., 2023; Bau et al., 2017; Olah et al., 2020a; Shaham et al., 2024; Marks et al., 2025), while later work extends this to distributed representations (Kowal et al., 2024a). The leading approach, cross layer transcoders (CLTs), train a SAE with a dictionary shared across multiple layers (Ameisen et al., 2025). Another recent work trains a model from scratch using sparsity regularization to produce an intrinsically interpretable and sparse model (Gao et al., 2025b). While these approaches learn concepts composed across layers, there are not approaches that have been extended to the cross-model setting.

Feature Universality. A growing body of work studies what information is shared across different DNNs. Representational alignment methods measure the similarity or mutual information between representation spaces, either across different models or between biological and artificial systems (Kriegeskorte et al., 2008; Sucholutsky et al., 2023). These approaches often require paired data, such as image-text pairs, in order to compare how different systems encode the same inputs across modalities. Recent work has argued based on these approaches that foundation models may converge toward a shared, *Platonic* representation of the world despite differences in training modality (Huh et al., 2024), though this claim remains debated (Gröger et al., 2026). A related line of work looks for shared features across models trained on different tasks by comparing their activations post hoc, including Rosetta *Neurons* (Dravid et al., 2023) and Rosetta *Concepts* (Kowal et al., 2024b). More recent methods train sparse autoencoder variants jointly across models in order to *diff* their learned concepts, as in Universal SAEs (Thasarathan et al., 2025) and crosscoders (Lindsey et al., 2024; Jiralerspong & Bricken, 2026). These methods make it possible to identify shared and model-specific concepts at a single layer. In contrast, we introduce a method for discovering shared circuits across both models and layers.

3. Method

In this section, we first review the cross-layer transcoder (CLT) (Ameisen et al., 2025) (§3.1), originally developed for tracing computational circuits in language models (see §A.3 for the full self-contained recap of the CLT architecture and the per-image circuit discovery pipeline we adopt for each model individually). Its defining property, and the reason we use it over single-layer sparse autoencoders, is that a CLT learns how features write across layers, enabling the tracing of cross-layer circuits representing model computation. We then describe our proposed extension across models, the Universal Circuits (§ 3.2)

Notation. Vectors and matrices are bold ($\boldsymbol{z}, \boldsymbol{D}$); scalars and indices are plain. Lowercase denotes individual vectors ($\boldsymbol{x}_{i,\ell}^{M}$); capitals denote matrices formed by stacking those vectors row-wise across a batch ($\boldsymbol{X}_{\ell}^{M}$). Indices: i image, ℓ CLT layer, f feature, c class. Model identifier $M \in \{A, B\}$ is always a superscript. A class subscript c is suppressed in subsections that fix a single class.

3.1. Background: Cross-Layer Transcoders (CLTs)

The CLT (Ameisen et al., 2025) replaces a transformer's MLPs with a sparse dictionary that traces feature composition across the depth of a single model. A per-layer encoder $\boldsymbol{E}_{\ell}$ maps the residual at layer ℓ to a sparse code $\boldsymbol{z}_{\ell} \in \mathbb{R}^{d_{\text{feat}}}$

via batch-top-k. For a source layer s and write layer w with $w \geq s$, a decoder bank $\{\boldsymbol{D}_{s,w} \in \mathbb{R}^{d_{\text{feat}} \times d_{\text{model}}}\}_{s \leq w}$ writes those features back into the residual stream: each row of $\boldsymbol{D}_{s,w}$ is one feature's write direction from s into w. Stacking codes across a batch as $\boldsymbol{Z}_s \in \mathbb{R}^{n \times d\text{feat}}$, the write-layer reconstruction sums contributions across all preceding sources,

$$\widehat{\boldsymbol{Y}}_w^M = \sum_{s \leq w} \boldsymbol{Z}_s^M \boldsymbol{D}_{s,w}^M, \qquad \widehat{\boldsymbol{Y}}_w^M \in \mathbb{R}^{n \times d_{\text{model}}}. \quad (1)$$

This per-(s, w) structure induces a feature-to-feature write graph across depth, enabling circuit-tracing rather than only concept-recovering. The encoder–decoder *bank* (i.e., multi-layer collection) is trained against the original MLP outputs by a self-reconstruction objective,

$$\mathcal{L}_{\text{self}} = \sum_w \left\| \widehat{\boldsymbol{Y}}_w - \boldsymbol{Y}_w \right\|_F^2 + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}, \quad (2)$$

with $\mathcal{L}_{\text{aux}}$ the top-k auxiliary loss (Lindsey et al., 2024) that revives features ranking highly in pre-activation but outside the active set. An end-to-end objective (Braun et al., 2024; Ameisen et al., 2025) further constrains the CLT-substituted forward pass to remain faithful to the frozen base classifier through a softmax-KL loss on class logits. Circuits are extracted per-input as pruned attribution graphs over the CLT-substituted forward pass; full architectural mechanics, attribution-graph construction, and pruning details are in §A.3.

3.2. Universal Circuits

3.2.1. Model Architecture

For each frozen pretrained vision transformer M, we replace the native head with a randomly initialized linear classifier over mean-pooled patch tokens. Each classifier is trained from scratch with the backbone frozen, ensuring a consistent head architecture across models and making upstream training the only variable across the model pairs under analysis. The rationale for mean-pool over alternatives like SigLIP's attention head, and how this shapes CLT features is provided in §A.5.

A Universal Circuit operates jointly over a pair of models (Fig. 2, left), by instantiating one CLT bank per model. For a model pair (A, B), the bank associated with model $M_{\in\{A,B\}}$ consistes of a per-(model, layer) encoder $\boldsymbol{E}_\ell^M$ and a per-(s, w) decoder bank $\{\boldsymbol{D}_{s,w}^M\}$, each trained against the corresponding model's MLP outputs via $\mathcal{L}_{\text{self}}^M$ (Eq. 2). What makes the architecture *universal* is a shared partition of the feature axis: each sparse code splits as

$$\boldsymbol{z}_{i,\ell}^M = [\, \boldsymbol{u}_{i,\ell}^M ;\, \boldsymbol{s}_{i,\ell}^M \,], \qquad d_{\text{feat}} = d_{\text{univ}} + d_{\text{spec}}, \quad (3)$$

where the universal slice $\boldsymbol{u}_{i,\ell}^M \in \mathbb{R}^{d_{\text{univ}}}$ has indices bound across models by training, and the specific slice $\boldsymbol{s}_{i,\ell}^M \in \mathbb{R}^{d_{\text{spec}}}$ remains private to each model, providing dedicated capacity for content the other model need not represent. Unlike cross-coders (Jiralerspong & Bricken, 2026), which separate universal and specific content into distinct decoder streams, we keep both within the same decoder to avoid doubling the bank size. The methods that encourage universality of these aligned indices stem from two sources: in their *selection* at sparsification time, where joint TopK is applied at the encoder level to align which indices fire, and in the *concept* each decoder represents at those indices, where the hybrid cross-prediction objective pushes the decoder vectors to encode the same thing across models.

Joint TopK: aligning index selection. At each CLT layer, we stack the universal pre-activations of both models over a batch as $\widetilde{\boldsymbol{U}}_\ell^M \in \mathbb{R}^{n \times d_{\text{univ}}}$ (one row per patch token, one column per universal index). We adopt *hard alignment* (Nasiri-Sarvi et al., 2025), applying batch-TopK to the summed activations,

$$\boldsymbol{m}_\ell = \text{batch-TopK}\left(\sum_M \widetilde{\boldsymbol{U}}_\ell^M \right) \in \{0, 1\}^{n \times d_{\text{univ}}}. \quad (4)$$

A single binary mask $\boldsymbol{m}_\ell$ then gates both models via $\boldsymbol{U}_\ell^M = \widetilde{\boldsymbol{U}}_\ell^M \odot \boldsymbol{m}_\ell$, keeping the same (patch, index) positions active in each while preserving native magnitudes, while the specific slice $\boldsymbol{S}_\ell^M$ is sparsified independently per model with its own batch-TopK.

3.2.2. UC Training

Hybrid cross-prediction: Aligning what universal indices represent. Joint TopK aligns index *selection* but not index *meaning*: $\boldsymbol{D}^A$ row f and $\boldsymbol{D}^B$ row f are still free to write unrelated directions into their respective residual streams. To guide the two decoder banks into learning consistent readouts of each universal index (Fig. 2, right), for each ordered pair $A \to B$ we form a hybrid code from the source's universal slice and the target's specific slice, and train against the target's MLP output through the target decoder bank:

$$\mathcal{L}_{\text{cross}}^{A \to B} = \sum_w \left\| \widehat{\boldsymbol{Y}}_w^{A \to B} - \boldsymbol{Y}_w^B \right\|_F^2 + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}^{A \to B}, \quad (5)$$

where $\widehat{\boldsymbol{Y}}_w^{A \to B}$ is the reconstruction of $\boldsymbol{Y}_w^B$ from $[\, \boldsymbol{U}_s^A ;\, \boldsymbol{S}_s^B \,]$ through $\{\boldsymbol{D}_{s,w}^B\}$, with the $B \to A$ direction symmetric. Only the universal slice is shared across models, so cross-model alignment acts only on universal features, while the target-specific slice captures residual variation unique to model B. Alternatively, a naive variant feeding the *full* sparse code through the target's bank would implicitly force specific indices to cross-align, undoing the partition. To stop

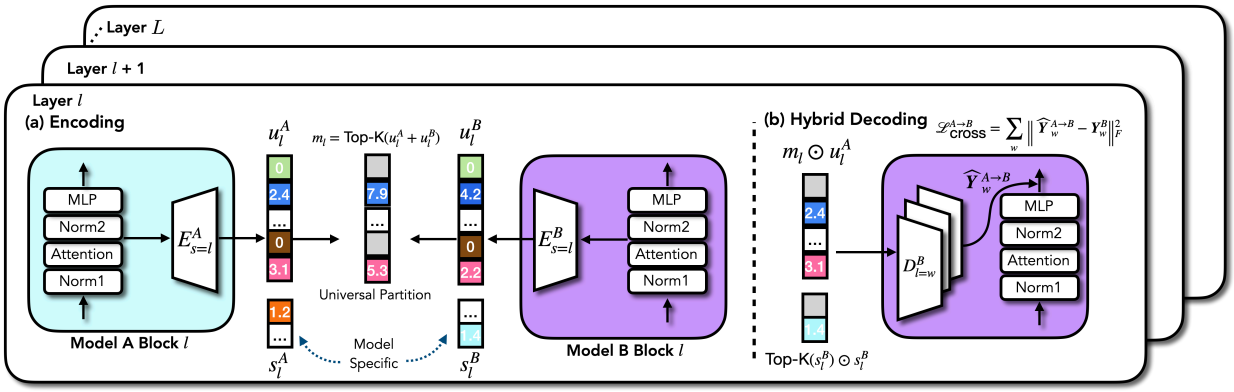

Figure 2. **Cross-Model, Cross-Layer Alignment.** To discover cross-model circuits, we introduce an objective to first align the per-layer concept-basis between distinct models. For each model under consideration, an encoder E_l^M is assigned with corresponding decoder bank $D_{s,w}^M$, representing the set of concept dictionaries from preceding source layer s which contribute to write layer $w = l$. We partition the encoded concept activation z_l from each per-layer encoder into u_l^M and s_l^M and bind the universal partition across models by applying the $\text{TopK}_{batch-k}(\cdot)$ operator across $\sum_M u_l^M$ forming a sparse mask over our universal representation. To encourage alignment, we reconstruct $\hat{\boldsymbol{Y}}_w^{A\to B}$, the post-mlp activations of model B by decoding the post sparsified hybrid activations of $[u_l^A, s_l^B]$ through B's decoder bank.

target-specific from self-absorbing the alignment gradient meant for universal, we anneal a per-layer Bernoulli dropout on $\boldsymbol{S}_s^B$ during training (full schedule in §A.5).

End-to-end faithfulness and total objective. Self-reconstruction and cross-prediction both match per-layer MLP outputs; following Braun et al. (2024); Ameisen et al. (2025), we add an end-to-end term that keeps the full CLT-substituted forward pass faithful to the frozen base classifier,

$$\mathcal{L}_{\text{e2e}}^M = \text{KL}\big(\boldsymbol{p}_{\text{base}}^M \,\|\, \boldsymbol{p}_{\text{e2e}}^M\big), \tag{6}$$

with $\boldsymbol{p}_{\text{base}}^M, \boldsymbol{p}_{\text{e2e}}^M$ the softmax distributions from the frozen ViT and CLT-in-the-loop passes; the shared head architecture makes this a single softmax-KL form across the diversity ladder. The total objective is given by,

$$\mathcal{L} = \sum_M \mathcal{L}_{\text{self}}^M + \lambda_{\text{cross}} \sum_{A\neq B} \mathcal{L}_{\text{cross}}^{A\to B} + \lambda_{\text{e2e}} \sum_M \mathcal{L}_{\text{e2e}}^M, \tag{7}$$

where λ_{cross} and λ_{e2e} are scalar hyperparameters.

4. Experimental Results

Across the diversity ladder, we ask whether models share circuitry (§4.1), whether it matters for downstream performance (§4.2), and whether shared slots compute the same thing across models (§4.3). The first two address *circuit-level* convergence; the third, the *feature-level* alignment that explains why both pruners independently land on the same circuits.

Implementation details. We evaluate Universal Circuits on three pairs of vision transformers, each anchored on a supervised ViT (Dosovitskiy et al., 2020) and spanning a successively wider training axis: ViT–ViT (same architecture and recipe, different run), DinoV1–ViT (Caron et al., 2021) (same architecture and ImageNet data, self-supervised vs supervised), and SigLIP–ViT (Zhai et al., 2023) (same architecture, image–text contrastive vs supervised, different data). All backbones come from `timm` (Wightman, 2019). The three pairs share a single UC configuration (Table 1), replacing MLPs in the last six blocks. Training and evaluation use STL10 (Coates et al., 2011) (96 $\to$ 224 bicubic); we measure classification accuracy of CLT-built per-class circuits on the 8000-image test set. In each model, we extract per-image attribution circuits via Neumann-series indirect-influence pruning (Ameisen et al., 2025), then union them over 100 images per class to form a class-level circuit (§A.4). The cross-model circuit for a pair is the intersection of the two class-level circuits.

4.1. How much circuitry do models share?

Cross-model circuits are visualized two ways: per-concept activation heatmaps for encoder alignment, and joint activation-maximization images exciting paired decoder entries across models for decoder alignment (§A.6). Figures 4 show qualitative UC results for SigLIP–ViT, revealing hierarchically composed cross-model circuits: low-level shape and texture detectors at early layers compose into class- and sub-class-discriminating concepts at later layers, such as military-aircraft sub-types.

We compute overlap via feature Jaccard between each pair's class circuits on the universal partition as $J_c = |F_c^A \cap F_c^B| / |F_c^A \cup F_c^B|$. Figure 3c shows each pair's two pruned-circuit sizes as twin bars, with the cross-model intersection highlighted at the base. Feature Jaccard decreases across the diversity ladder (0.43, 0.32, 0.22). The cross-model intersection contains 857 universal features on ViT–ViT

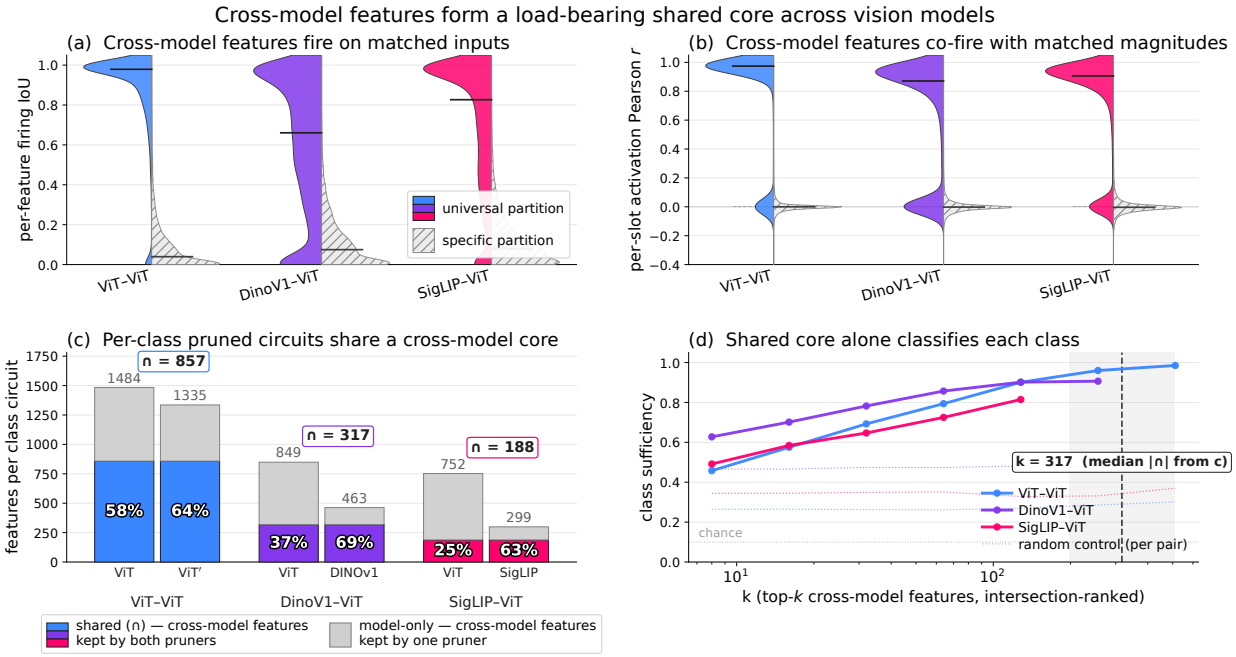

Figure 3. **Cross-model, cross-layer concept alignment and circuit overlap across the diversity ladder.** (a) Per-slot IoU of activation co-firing on the universal partition (top half of each violin) and specific partition (bottom half), across the three pairs. (b) Per-slot Pearson correlation of image-pooled universal-slice activations across models. (c) Per-pair pruned class-circuit sizes (twin bars) with the cross-model intersection highlighted at the base; intersection size $|\cap|$ shown above each pair. (d) Top-k classification accuracy from the cross-model intersection ranked by combined attribution magnitude, against random-control baselines.

and 188 on SigLIP–ViT, with an asymmetric intersection share on SigLIP–ViT: 63% of the SigLIP-side circuit lies inside ViT's, while only 25% of ViT's lies inside SigLIP's. Examining the layer distribution of these intersections, we find early layers consistently capture a larger portion of the shared core; when controlling for per-layer concept count, agreement stays roughly uniform for ViT–ViT but decays monotonically with depth as model pairs diverge, with the steepest decay at late layers for SigLIP–ViT (see §A.8).

Crucially, each pruner ranks features using its model's own per-token attribution magnitudes—uncorrelated across models at the token scale, decaying from $r = 0.40$ on ViT–ViT to -0.09 on SigLIP–ViT—so the intersection reflects *convergent, independent* judgment rather than overlap forced by joint top-k. We test next whether this independently-converged core suffices for classification.

4.2. Is shared circuitry functional?

To test whether intersection features are *sufficient* for classification, we restrict each pruned class circuit to its top-k intersection features (ranked by combined cross-model attribution), ablate the rest in a masked forward pass, and measure per-class STL10 test accuracy. Figure 3 (d) shows the resulting top-k accuracy curves. Overall, we see that the *universal circuit alone recovers model performance*: At $k \approx 317$ (the median intersection size), top-k accuracy reaches 89–98% of the full per-class circuit accuracy on every pair (mean 95%, intersection-ranked sufficiency 0.91), far outperforming baseline random universal subsets.

The intersection size decreases across the diversity ladder ($857 \rightarrow 317 \rightarrow 188$ universal features), but per-class sufficiency does not degrade in proportion. SigLIP–ViT's 188-feature intersection (a quarter of the ViT circuit, two-thirds of the SigLIP circuit) reproduces 89% of per-class sufficiency from the ViT side and 95% from the SigLIP side. The intersection is therefore both small and functionally load-bearing in both models. In Section A.7, we further analyze aligned concepts that are present in M^A class-level circuit but not M^B and find that they are functionally indistinguishable from randomly selected aligned concepts in M^B. The functional relevance of model-specific concepts varies along the diversity ladder, becoming on par with the cross-model intersection when the divergence between model pairs is highest (SigLIP–ViT).

Each model's pruner therefore independently converges on a small shared core that suffices for classification. Whether the universal slots both pruners select are tracking the fire on the same images with correlated magnitude is what we test directly next.

4.3. Do shared features fire and scale together across models?

The intersection result above only makes sense if the universal indices both pruners select track the same thing in

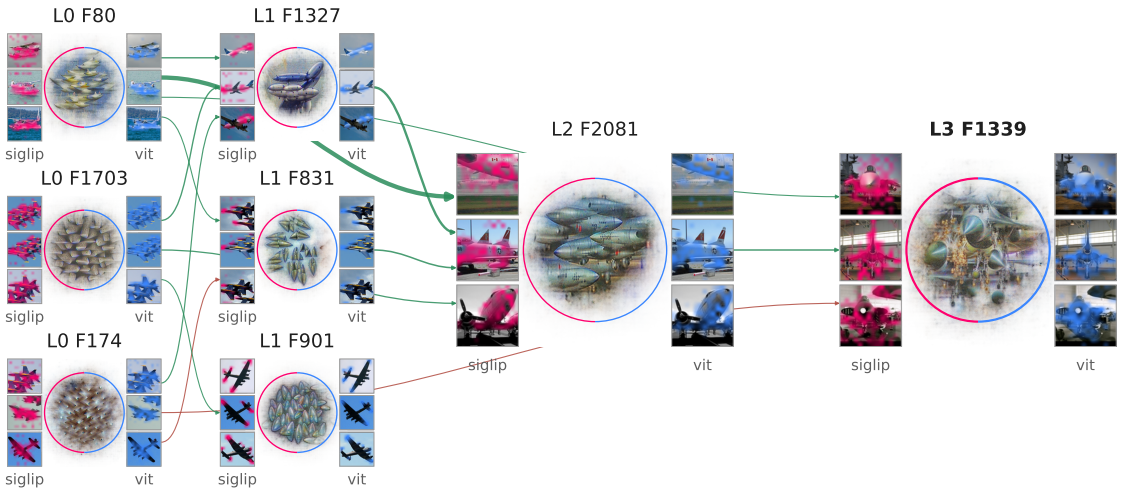

Figure 4. **Qualitative results of cross-model circuits.** We discover functional, class-specific circuits shared between pairs of vision models: SigLIP (pink) versus ViT (blue) for the *plane* class. Because both models were trained on distributions more granular than STL10, the circuit contains sub-class concepts (e.g., military aircraft) built from lower-level shape and texture features at earlier layers.

each model. We test this with two metrics: per-slot firing IoU (Fig. 3a) and per-slot Pearson correlation of image-pooled magnitudes (Fig. 3b) — the latter a strictly stronger condition that probes magnitude as well as firing support.

For each universal slot (ℓ, f) we compute the IoU between the sets of test images on which the two models fire that slot, $\mathrm{IoU}_{\ell,f} = |\mathcal{F}^A_{\ell,f} \cap \mathcal{F}^B_{\ell,f}|/|\mathcal{F}^A_{\ell,f} \cup \mathcal{F}^B_{\ell,f}|$, where $\mathcal{F}^M_{\ell,f} = \{i \in \mathcal{V} : \max_p z^M_{i,\ell,p,f} > \theta\}$, with threshold $\theta = 0.01$. Figure 3a shows the distribution of $\mathrm{IoU}_{\ell,f}$ across slots, separated by partition. Median per-slot IoU on the universal partition is 0.97 on ViT–ViT, 0.65 on DinoV1–ViT, and 0.82 on SigLIP–ViT, with permutation nulls below 0.07 on every pair. The specific partition is much lower on every pair (means 0.08–0.12, near the permutation null). The gap between the universal and specific partitions is large on every pair, indicating that joint TopK binds firing support as intended. We also compute, for each universal slot (ℓ, f), the Pearson correlation $r_{\ell,f}$ between the two models' per-image mean-pooled activations across the 8000 test images. Figure 3b shows the resulting distribution: mean per-slot Pearson on the universal partition is 0.63–0.77 across the three pairs (medians 0.87–0.97), with under 1.3% of slots negatively correlated. The specific partition sits at the permutation null on every pair, giving a universal–specific gap of +0.63 to +0.77. Universal slots therefore fire at aligned image-level magnitudes, not only on aligned images.

The mean–median gap on the universal partition (means 0.63–0.77 vs. medians 0.87–0.97) reveals a left-skewed distribution: most universal slots are tightly aligned, but a tail is not. This tail is consistent with the pool-membership analysis of §A.7: universal slots present in one model's pruned circuit but absent from the other's are functionally indistinguishable from random when ablated in the model that does not keep them. A fraction of universal slots fire together across both models but only do classification work in one of them.

Together, these two checks establish that the universal pool is aligned at the feature level: the same indices fire on the same inputs and carry comparable magnitudes per image. This is what makes a single concept-space probe trained on one model's universal slots meaningful when read off the other's, the basis of the held-out application we turn to next.

5. Application: Probe transplantation

Concept-space probe. Let $\bar{\boldsymbol{u}}^M_i \in \mathbb{R}^{d_{\text{univ}}}$ denote the mean-pooled universal slice from the deepest CLT layer $\ell^\star$. Given an image build set τ with binary labels $y_i \in \{0, 1\}$ for the held-out class, we train a linear probe $\boldsymbol{\alpha} \in \mathbb{R}^{d_{\text{univ}}}$ on source model A by minimizing class-balanced binary cross-entropy on logits $(\bar{\boldsymbol{u}}^A_i)^\top \boldsymbol{\alpha}$. Transplantation then reduces to applying the same probe $\boldsymbol{\alpha}$ to $\bar{\boldsymbol{u}}^B_i$, since joint TopK binds each coordinate of $\boldsymbol{\alpha}$ to the same universal feature index across models, while cross-prediction encourages those shared coordinates to encode aligned per-image semantics.

Results. A linear probe $\boldsymbol{\alpha} \in \mathbb{R}^{d_{\text{univ}}}$ trained on one model's image-pooled universal features for the `person` class from VOC 2012 (Everingham et al., 2010), a concept never seen during CLT training, transplants to the other model's universal features at AUROC 0.97–0.99 on a clean diagnostic subset and 0.87–0.92 on a hard subset that blocks an STL10-content shortcut (Fig. 5, leftmost bars; details in §A.2). Permutation nulls (probe coefficients shuffled before transplant) fall to chance (0.48–0.53). Probes trained on the specific partition collapse to chance (0.43–0.55) when transplanted, confirming that the transferable concept signal

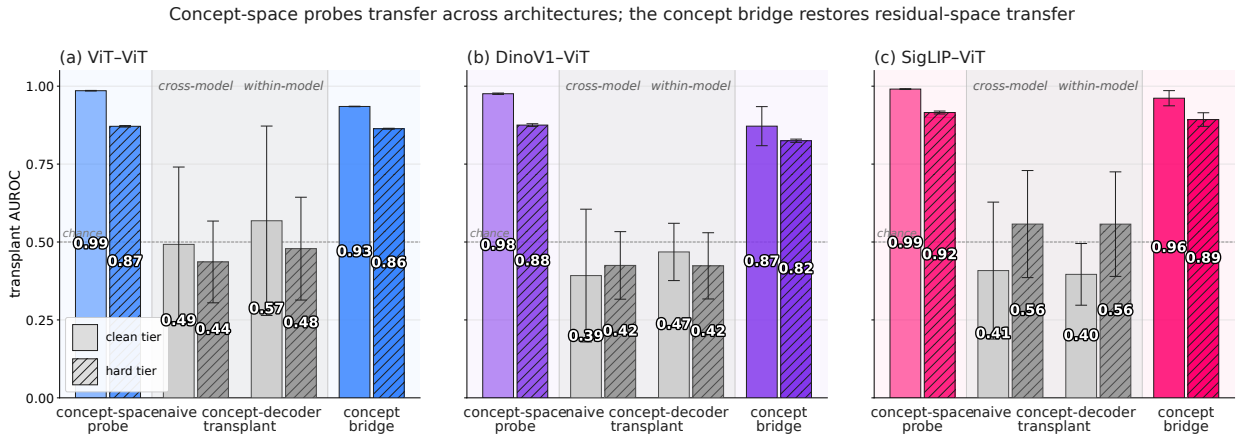

Figure 5. **Concept-space probe transplantation across the diversity ladder.** A linear probe on one model's image-pooled universal features for VOC `person` (left) transplants directly to the other model's universal features. A naive residual-stream readout (middle), obtained by composing the probe with the universal-slice decoder columns, fails on every pair. The bridge readout (right) ridge-fits the residual direction tracking the concept score in each model and matches direct concept-space AUROC. Bars are the mean over both transplant directions; whiskers span the per-direction range. Light bars show a clean diagnostic subset; hatched bars show a hard subset forcing STL10-overlap classes on both positives and negatives, blocking a "no-STL10-content = person" shortcut available in the clean subset (full construction in §A.2).

is in the universal pool. The transplant retains 98–100% of the AUROC obtained when the same probe is trained directly on the target model's features. Extracting a circuit for the transplanted concept in either model requires a residual-stream readout: a head row that the standard attribution-graph circuit-extraction pipeline of §A.3 can take gradients through. Composing the probe with the universal-slice decoder columns is the natural construction. It fails on almost every pair, even when source and target are the same model: within-model AUROC is 0.40–0.57 (Fig. 5, middle bars) against 0.97–0.99 clean / 0.87–0.92 hard for the same probe applied in concept space (Fig. 5, leftmost bars). Multiple structural properties likely drive the cross-model failure.

The post-final-LN residual at the head mixes pre-MLP content, cross-layer reconstructions, and unmodeled error—none of which decoder composition controls. The bridge readout (§A.2) sidesteps this by fitting, via ridge regression on the build images, the residual direction that tracks the concept score in each model. Bridge AUROC closes the gap to direct concept-space AUROC (0.87–0.96 clean, 0.82–0.89 hard cross-pair; Fig. 5, right), yielding a residual-stream readout that downstream circuit extraction consumes unchanged. Appendix Fig. 8 visualizes the cross-model circuit obtained by transplanting the *person* probe from ViT to SigLIP: both models build up the concept similarly, with early-layer face detectors composing into hair and clothing features that converge on person.

6. Conclusion

We introduced UC, a framework for discovering shared and model-specific multi-layer circuits between pairs of trained models. Our experiments revealed three findings: (i) across pairs of ViT, DINOv1, and SigLIP, a few hundred universal features per pair (188–857) recover 89–98% of full-circuit classification accuracy in both models; (ii) the shared core is graded, with most universal slots aligning tightly while a smaller tail diverges in functional role; and (iii) the aligned slot space supports zero-shot probe transplantation, where a probe trained on one model for an unseen class transfers at 0.87–0.99 AUROC. Altogether, UC provides a practical tool for circuit-level cross-model comparison, showing that vision transformers across distinct training recipes converge on a small load-bearing core of multi-layer circuitry that can be measured and partially reused across models.

Limitations. Our evaluation focuses on vision transformers and image classification; language, multimodal, and dense-prediction settings remains to be validated. Each model's decoder bank grows as $O(L^2)$ in CLT layers, and cross-prediction across M aligned models adds further compute and memory cost. Finally, since UC uses CLTs as its circuit-finding backbone, it inherits their limitations (Ameisen et al., 2025): CLTs replace only MLP modules, leaving attention-mediated computation outside the extracted circuit graph, and the sparse-code reconstruction of MLP outputs is necessarily imperfect, introducing a fidelity gap between the CLT-substituted forward pass and the base model.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 2018.

Ameisen, E., Lindsey, J., Pearce, A., Gurnee, W., Turner, N. L., Chen, B., Citro, C., Abrahams, D., Carter, S., Hosmer, B., Marcus, J., Sklar, M., Templeton, A., Bricken, T., McDougall, C., Cunningham, H., Henighan, T., Jermyn, A., Jones, A., Persic, A., Qi, Z., Thompson, T. B., Zimmerman, S., Rivoire, K., Conerly, T., Olah, C., and Batson, J. Circuit tracing: Revealing computational graphs in language models. 2025. https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Anwar, U., Saparov, A., Rando, J., Paleka, D., Turpin, M., Hase, P., Lubana, E. S., Jenner, E., Casper, S., Sourbut, O., et al. Foundational challenges in assuring alignment and safety of large language models. *ArXiv e-print*, 2024.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *Public Library of Science (PloS One)*, 2015.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., Unterthiner, T., Keysers, D., Koppula, S., Liu, F., Grycner, A., Gritsenko, A., Houlsby, N., Kumar, M., Rong, K., Eisenschlos, J., Kabra, R., Bauer, M., Bošnjak, M., Chen, X., Minderer, M., Voigtlaender, P., Bica, I., Balazevic, I., Puigcerver, J., Papalampidi, P., Henaff, O., Xiong, X., Soricut, R., Harmsen, J., and Zhai, X. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Bhalla, U., Oesterling, A., Srinivas, S., Calmon, F. P., and Lakkaraju, H. Interpreting clip with sparse linear concept embeddings (splice). *ArXiv e-print*, 2024.

Braun, D., Taylor, J., Goldowsky-Dill, N., and Sharkey, L. Identifying functionally important features with end-to-end sparse dictionary learning. *ArXiv e-print*, 2024.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Buolamwini, J. and Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. *Conference on fairness, accountability and transparency*, 2018.

Cammarata, N., Carter, S., Goh, G., Olah, C., Petrov, M., Schubert, L., Voss, C., Egan, B., and Lim, S. K. Thread: Circuits. *Distill*, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Chen, Y., Zhou, Z., and Yan, J. Going beyond neural network feature similarity: The network feature complexity and its interpretation using category theory. *Proceedings of the International Conference on Learning Representations*, 2024.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. PMLR, 2011.

Colin, J., Fel, T., Cadène, R., and Serre, T. What I cannot predict, I do not understand: A human-centered evaluation framework for explainability methods. *Advances in Neural Information Processing Systems*, 35:2832–2845, 2022.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, 2023.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *ArXiv e-print*, 2023.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. Underspecification presents challenges for credibility in modern machine learning. *The Journal of Machine Learning Research*, 2020.

Dhimoila, G., Fel, T., Boutin, V., and Picard, A. Cross-modal redundancy and the geometry of vision-language embeddings. *Proceedings of the International Conference on Learning Representations*, 2026.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth

16x16 words: Transformers for image recognition at scale. *Proceedings of the International Conference on Learning Representations*, 2020.

Dravid, A., Gandelsman, Y., Efros, A. A., and Shocher, A. Rosetta neurons: Mining the common units in a model zoo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Dumitrescu, B. and Irofti, P. Dictionary learning algorithms and applications. 2018.

Elad, M. Sparse and redundant representations: from theory to applications in signal and image processing. *Springer International Publishing*, 2010.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010.

Fel, T., Cadene, R., Chalvidal, M., Cord, M., Vigouroux, D., and Serre, T. Look at the variance! efficient black-box explanations with sobol-based sensitivity analysis. *Advances in Neural Information Processing Systems*, 2021.

Fel, T., Boissin, T., Boutin, V., Picard, A., Novello, P., Colin, J., Linsley, D., Rousseau, T., Cadène, R., Gardes, L., and Serre, T. Unlocking feature visualization for deeper networks with magnitude constrained optimization. *Advances in Neural Information Processing Systems*, 2023a.

Fel, T., Boutin, V., Moayeri, M., Cadene, R., Bethune, L., Chalvidal, M., and Serre, T. A holistic approach to unifying automatic concept extraction and concept importance estimation. *Advances in Neural Information Processing Systems*, 2023b.

Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., and Serre, T. Craft: Concept recursive activation factorization for explainability. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023c.

Fel, T., Lubana, E. S., Prince, J. S., Kowal, M., Boutin, V., Papadimitriou, I., Wang, B., Wattenberg, M., Ba, D., and Konkle, T. Archetypal sae: Adaptive and stable dictionary learning for concept extraction in large vision models. *Proceedings of the International Conference on Machine Learning*, 2025a.

Fel, T., Wang, B., Lepori, M. A., Kowal, M., Lee, A., Balestriero, R., Joseph, S., Lubana, E. S., Konkle, T., Ba, D., et al. Into the rabbit hull: From task-relevant concepts in dino to minkowski geometry. *ArXiv e-print*, 2025b.

Foldiak, P. and Endres, D. M. Sparse coding. *ArXiv e-print*, 2008.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. *Proceedings of the International Conference on Learning Representations*, 2025a.

Gao, L., Rajaram, A., Coxon, J., Govande, S. V., Baker, B., and Mossing, D. Weight-sparse transformers have interpretable circuits. *arXiv preprint arXiv:2511.13653*, 2025b.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.

Ghorbani, A., Wexler, J., Zou, J. Y., and Kim, B. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 2019.

Gorton, L. The missing curve detectors of inceptionv1: Applying sparse autoencoders to inceptionv1 early vision. *ArXiv e-print*, 2024.

Graziani, M., Nguyen, A.-p., O'Mahony, L., Müller, H., and Andrearczyk, V. Concept discovery and dataset exploration with singular value decomposition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

Gresele, L., Rubenstein, P. K., Mehrjou, A., Locatello, F., and Scholkopf, B. The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica. *Uncertainty in Artificial Intelligence*, 2020.

Gröger, F., Wen, S., and Brbić, M. Revisiting the platonic representation hypothesis: An aristotelian view. *arXiv preprint arXiv:2602.14486*, 2026.

Hase, P. and Bansal, M. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020.

Hermann, K. and Lampinen, A. What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, 2020.

Hsieh, C.-Y., Yeh, C.-K., Liu, X., Ravikumar, P., Kim, S., Kumar, S., and Hsieh, C.-J. Evaluations and methods for explanation through robustness analysis. *Proceedings of the International Conference on Learning Representations*, 2021.

Huben, R., Cunningham, H., Smith, L. R., Ewart, A., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. In *Proceedings of the International Conference on Learning Representations*, 2024.

Huh, M., Cheung, B., Wang, T., and Isola, P. Position: The Platonic Representation Hypothesis. In *Proceedings of the International Conference on Machine Learning*, 2024.

Jiralerspong, T. and Bricken, T. Cross-architecture model diffing with crosscoders: Unsupervised discovery of differences between llms. *arXiv preprint arXiv:2602.11729*, 2026.

Joseph, S. Vit prisma: A mechanistic interpretability library for vision transformers. https://github.com/soniajoseph/vit-prisma, 2023.

Joseph, S., Suresh, P., Hufe, L., Stevinson, E., Graham, R., Vadi, Y., Bzdok, D., Lapuschkin, S., Sharkey, L., and Richards, B. A. Prisma: An open source toolkit for mechanistic interpretability in vision and video. *ArXiv e-print*, 2025a.

Joseph, S., Suresh, P., Hufe, L., Stevinson, E., Graham, R., Vadi, Y., Bzdok, D., Lapuschkin, S., Sharkey, L., and Richards, B. A. Prisma: An open source toolkit for mechanistic interpretability in vision and video, 2025b. URL https://arxiv.org/abs/2504.19475.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *Proceedings of the International Conference on Machine Learning*, 2018.

Kim, S. S. Y., Meister, N., Ramaswamy, V. V., Fong, R., and Russakovsky, O. HIVE: Evaluating the human interpretability of visual explanations. *Proceedings of the IEEE European Conference on Computer Vision*, 2022.

Kowal, M., Dave, A., Ambrus, R., Gaidon, A., Derpanis, K. G., and Tokmakov, P. Understanding video transformers via universal concept discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024a.

Kowal, M., Dave, A., Ambrus, R., Gaidon, A., Derpanis, K. G., and Tokmakov, P. Understanding video transformers via universal concept discovery. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024b.

Kowal, M., Wildes, R. P., and Derpanis, K. G. Visual concept connectome (vcc): Open world concept discovery and their interlayer connections in deep models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024c.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

Lee, H., Battle, A., Raina, R., and Ng, A. Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems*, 2006.

Lepori, M. A., Pavlick, E., and Serre, T. Neurosurgeon: A toolkit for subnetwork analysis. *ArXiv e-print*, 2023.

Lindsey, J., Templeton, A., Marcus, J., Conerly, T., Batson, J., and Olah, C. Sparse crosscoders for cross-layer features and model diffing. 2024. https://transformer-circuits.pub/2024/crosscoders/index.html.

Mairal, J., Bach, F., and Ponce, J. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 2014.

Makelov, A., Lange, G., and Nanda, N. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. *ArXiv e-print*, 2023.

Marks, S., Rager, C., Michaud, E. J., Belinkov, Y., Bau, D., and Mueller, A. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 2022.

Moschella, L., Maiorca, V., Fumero, M., Norelli, A., Locatello, F., and Rodolà, E. Relative representations enable zero-shot latent space communication. *Proceedings of the International Conference on Learning Representations*, 2022.

Muttenthaler, L., Dippel, J., Linhardt, L., Vandermeulen, R. A., and Kornblith, S. Human alignment of neural network representations. *Proceedings of the International Conference on Learning Representations*, 2023.

Nasiri-Sarvi, A., Rivaz, H., and Hosseini, M. S. Sparc: Concept-aligned sparse autoencoders for cross-model and cross-modal interpretability. *arXiv preprint arXiv:2507.06265*, 2025.

Nguyen, G., Kim, D., and Nguyen, A. The effectiveness of feature attribution methods and its correlation with automatic evaluation scores. *Advances in Neural Information Processing Systems*, 2021.

Oikarinen, T. and Weng, T.-W. Clip-dissect: Automatic description of neuron representations in deep vision networks. *Proceedings of the International Conference on Learning Representations*, 2023.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. An overview of early vision in inceptionv1. *Distill*, 2020a.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 2020b. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Olshausen, B. A. and Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 1996.

Olshausen, B. A. and Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 1997.

Pach, M., Karthik, S., Bouniot, Q., Belongie, S., and Akata, Z. Sparse autoencoders learn monosemantic features in vision-language models. *ArXiv e-print*, 2025.

Papadimitriou, I., Su, H., Fel, T., Kakade, S., and Gil, S. Interpreting the linear structure of vision-language model embedding spaces. *Proceedings of the Conference on Language Modeling (COLM)*, 2025.

Parekh, J., Khayatan, P., Shukor, M., Newson, A., and Cord, M. A concept-based explainability framework for large multimodal models. *ArXiv e-print*, 2024.

Paulo, G. and Belrose, N. Sparse autoencoders trained on the same data learn different features. *ArXiv e-print*, 2025.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. *Proceedings of the British Machine Vision Conference*, 2018.

Rajamanoharan, S., Lieberum, T., Sonnerat, N., Conmy, A., Varma, V., Kramar, J., and Nanda, N. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *ArXiv e-print*, 2024.

Rajaram, A., Chowdhury, N., Torralba, A., Andreas, J., and Schwettmann, S. Automatic discovery of visual circuits. *arXiv preprint arXiv:2404.14349*, 2024.

Raji, I. D. and Buolamwini, J. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019.

Rao, S., Mahajan, S., Böhle, M., and Schiele, B. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. *Proceedings of the IEEE European Conference on Computer Vision*, 2024.

Rentzeperis, I., Calatroni, L., Perrinet, L. U., and Prandi, D. Beyond l1 sparse coding in v1. *PLoS Computational Biology*, 2023.

Rubinstein, R., Bruckstein, A. M., and Elad, M. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 2010.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

Shaham, T. R., Schwettmann, S., Wang, F., Rajaram, A., Hernandez, E., Andreas, J., and Torralba, A. A multimodal automated interpretability agent. *Proceedings of the International Conference on Machine Learning*, 2024.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Proceedings of the International Conference on Learning Representations*, 2013.

Sixt, L., Granz, M., and Landgraf, T. When explanations lie: Why many modified bp attributions fail. *Proceedings of the International Conference on Machine Learning*, 2020.

Sucholutsky, I., Muttenthaler, L., Weller, A., Peng, A., Bobu, A., Kim, B., Love, B. C., Grant, E., Achterberg, J., Tenenbaum, J. B., et al. Getting aligned on representational alignment. *ArXiv e-print*, 2023.

Surkov, V., Wendler, C., Terekhov, M., Deschenaux, J., West, R., and Gulcehre, C. Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders. *ArXiv e-print*, 2024.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

Thasarathan, H., Forsyth, J., Fel, T., Kowal, M., and Derpanis, K. Universal sparse autoencoders: Interpretable cross-model concept alignment. *ArXiv e-print*, 2025.

Tošić, I. and Frossard, P. Dictionary learning. *IEEE Signal Processing Magazine*, 2011.

Vielhaben, J., Blücher, S., and Strodthoff, N. Multi-dimensional concept discovery (mcd): A unifying framework with completeness guarantees. *The Journal of Transactions on Machine Learning Research*, 2023.

Wang, K. R., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *Proceedings of the International Conference on Learning Representations*, 2023.

Wightman, R. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models, 2019.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. *Proceedings of the IEEE International Conference on Computer Vision*, 2023.

Zhang, R., Madumal, P., Miller, T., Ehinger, K. A., and Rubinstein, B. I. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Zhou, C., Loy, C. C., and Dai, B. Extract free dense labels from clip. In *European Conference on Computer Vision (ECCV)*, 2022.

A. Additional Details

A.1. Additional Qualitative Results

In Figure 6 and 7 we present additional qualitative visualizations of cross-model intersection sub-circuits between Dino–ViT and SigLIP–ViT for the classes dog, monkey, and ship. In each class sub-circuit we find a clear low-level shape detectors to high-level feature abstractions starting with class specific eye, face, and ear detectors for animal classes that compose into progressively larger and more complicated features that compose the entire class entity in the image. For ease of visualization, we prune these sub-circuits to only show the max three attributing nodes per-layer to the final node presented. Note that all qualitative results use relative indexing for feature labels, with layer 0 corresponding to layer 6 in the base model and layer 5 corresponding to layer 12.

A.2. Application Details

Each VOC-person image is tagged with its non-person object labels; the clean and hard tiers split images by whether those tags overlap any of seven STL10-confound classes (aeroplane, bird, car, cat, dog, horse, boat). Clean keeps person-only positives + no-STL10-overlap negatives; hard keeps person+overlap positives + overlap-only negatives, so STL10 content is present on both sides and the probe cannot win on a "no STL10 $\rightarrow$ person" shortcut. After tier selection we balance each tier by random undersampling, drawing $\min(n_{\text{pos}}, n_{\text{neg}})$ images from whichever class is in surplus, applied separately to the train and eval slices so the probe sees an exact 50/50 person/non-person split in both.

Bridge readout. The natural way to materialise $\boldsymbol{\alpha}$ as a residual-stream direction in model M is to compose it with the cumulative decoder column of each universal feature, $\boldsymbol{w}^M_{\text{naive}} = \sum_f \alpha_f \sum_{w \geq \ell^\star} \boldsymbol{D}^M_{\ell^\star,w}[f,:]^\top$. We find this composition does not preserve the probe's direction in practice (§5), and adopt a basis-invariant construction. We read the transplanted probe as a per-image scalar

$$t_i^M \;=\; (\bar{\boldsymbol{u}}_i^M)^\top \boldsymbol{\alpha},$$

where the concept score it assigns image i in model M, with $t_i^A \approx t_i^B$ by the same alignment that lets $\boldsymbol{\alpha}$ transplant. We then fit, per target M, the residual-stream direction whose head logit best tracks this scalar on the build set,

$$\boldsymbol{w}^M_{\text{bridge}} \;=\; \arg\min_{\boldsymbol{w}} \sum_{i \in \mathcal{T}} \big((\boldsymbol{r}_i^M)^\top \boldsymbol{w} - t_i^M\big)^2 \;+\; \lambda \|\boldsymbol{w}\|_2^2, \tag{8}$$

where $\boldsymbol{r}_i^M \in \mathbb{R}^{d_{\text{model}}}$ is the mean-pooled post-final-LN residual at M's head and $\lambda = 1$. The fit operates entirely within model M, only the regression target t_i^M depends on the source's $\boldsymbol{\alpha}$. The resulting $\boldsymbol{w}^M_{\text{bridge}}$ substitutes for a head row in the standard circuit-extraction pipeline without further modification.

Naive concept-decoder readout. The natural way to express the source-trained probe $\boldsymbol{\alpha}$ as a residual-stream direction in target model M is to compose it with M's decoder columns. For each universal feature f, we take its cumulative residual-stream contribution at the read-out layer $\ell^\star$ — summed over the write layers $w \geq \ell^\star$ at which it writes back — weight by the probe coefficient α_f, and sum across features:

$$\boldsymbol{w}^M_{\text{naive}}; =; \sum_f \alpha_f \sum_{w \geq \ell^\star} \boldsymbol{D}^M_{\ell^\star,w}[f,:]^\top; \in; \mathbb{R}^{d_{\text{model}}}. \tag{9}$$

This vector substitutes directly for a head row in M, with no fitting in M: it is the construction one obtains by reading the CLT forward map and plugging in $\boldsymbol{\alpha}$.

A.3. Cross-Layer Transcoders and Attribution-Graph Circuits

We follow the notation of §3, with patch position written as p. The pipeline has two stages. The first trains a CLT to reconstruct each MLP block's output from a sparse, cross-layer feature code. The second runs the trained CLT on a single image and reads off a directed graph of which features drove the prediction, then prunes that graph to a circuit (Local replacement, Edge attribution, Pruning). The remaining paragraphs describe how per-image circuits aggregate into class-level circuits and how we evaluate them.

Architecture. A standard sparse autoencoder reconstructs one layer's activation from a sparse code at that same layer. A cross-layer transcoder (Ameisen et al., 2025) generalises this construction to a contiguous block range $[L_s, L_e)$ of model M, with the additional property that any feature encoded at layer s contributes to the reconstruction at every downstream layer $w \geq s$. This per-source-to-write structure lets the dictionary represent how features compose across layers, and is what circuit extraction operates on later. At every layer $\ell \in [L_s, L_e)$, an encoder $\boldsymbol{E}_\ell^M$ maps the pre-MLP residual into a sparse code $\boldsymbol{z}_\ell^M \in \mathbb{R}^{d_{\text{feat}}}$, sparsified by batch-top-k (the k largest activations per token across a batch are kept and the rest are zeroed). For each pair (s, w) with $s \leq w$, a decoder $\boldsymbol{D}_{s,w}^M \in \mathbb{R}^{d_{\text{feat}} \times d_{\text{model}}}$ writes the source-layer code into the reconstruction of the write-layer's MLP output,

$$\widehat{\boldsymbol{Y}}_w^M \;=\; \sum_{s \leq w} \boldsymbol{Z}_s^M \, \boldsymbol{D}_{s,w}^M,$$

where $\boldsymbol{Z}_s^M$ stacks codes across the batch and $\widehat{\boldsymbol{Y}}_w^M$ is the CLT's estimate of the original MLP output $\boldsymbol{Y}_w^M$. Each row of $\boldsymbol{D}_{s,w}^M$ is the write direction of one feature from layer s into layer w, and decoder rows are L_2-normalised. Training minimises $\sum_w \|\widehat{\boldsymbol{Y}}_w^M - \boldsymbol{Y}_w^M\|_F^2$ together with the top-k auxiliary term $\mathcal{L}_{\text{aux}}^M$ that revives features whose pre-activation ranks high but falls outside the active set.

Vision adaptation. Two details follow the vision setting rather than the language setting the CLT was developed in. First, only patch tokens are processed by the CLT. The CLS token bypasses the CLT and runs through the original MLP unchanged, and patch normalisation statistics are computed separately from CLS. Second, the encoder reads from the residual immediately before the second LayerNorm of each block (the input to the original MLP) after a per-(model, layer) standardisation by precomputed mean and variance. The decoder writes back into the post-MLP residual after de-standardisation.

Local replacement model. A trained CLT alone does not give us a circuit. To extract one, we want to know, on a single input, how much each upstream feature contributed to each downstream feature and to the final logit. These contributions are not well-defined a priori, since attention, LayerNorm, and the MLP couple residual-stream coordinates nonlinearly. We sidestep the nonlinearities by replacing them with their values on the image of interest, leaving a linear forward pass on top of which attribution becomes a single inner product.

Concretely, attribution operates on a frozen replacement of model M in which (i) attention patterns and LayerNorm scales are recorded on a clean forward pass and held fixed, (ii) the CLT replaces the MLPs in the CLT span, and (iii) the MLP outputs (now CLT reconstructions) are detached from the gradient graph. The resulting forward pass is linear in the residual stream at the observed activation. Gradients flow through residual additions and through frozen-pattern attention, but not through the MLP nonlinearity. We write $\boldsymbol{r}_p^{M,\ell} \in \mathbb{R}^{d_{\text{model}}}$ for the residual at layer ℓ and patch position p produced by this replacement model.

Edge attribution. An edge $A_{s \to t}^M$ in the attribution graph measures, for the current image, how much source feature s contributes to the activation of target feature t. Because the replacement model is linear, this contribution has a closed form. A target node t is identified by its CLT layer ℓ_t, patch position p_t, and feature index f_t, with encoder direction $\boldsymbol{e}_t^M = \boldsymbol{E}_{\ell_t}^M[f_t, :]^\top$. Its activation is the inner product between this direction and the residual at (ℓ_t, p_t),

$$a_t^M \;=\; (\boldsymbol{e}_t^M)^\top \, \boldsymbol{r}_{p_t}^{M,\ell_t}.$$

A source feature s at (ℓ_s, p_s, f_s) with activation z_s^M writes the vector $z_s^M \, \boldsymbol{D}_{s,w}^M[f_s, :]$ into each downstream layer w between ℓ_s and ℓ_t. Pushing this write through the linear replacement model to (ℓ_t, p_t) and projecting onto $\boldsymbol{e}_t^M$ gives

$$A_{s \to t}^M \;=\; z_s^M \sum_{\ell_s \leq w \leq \ell_t} \boldsymbol{D}_{s,w}^M[f_s, :] \, \boldsymbol{J}_{w \to \ell_t}^{M,\bullet} \, \boldsymbol{e}_t^M, \tag{10}$$

where $\boldsymbol{D}_{s,w}^M[f_s, :]$ is the f_s-th row of $\boldsymbol{D}_{s,w}^M$ and $\boldsymbol{J}_{w \to \ell_t}^{M,\bullet}$ is the Jacobian of $\boldsymbol{r}_{p_t}^{M,\ell_t}$ with respect to the residual at (w, p_s) through the replacement model. The bullet indicates that this Jacobian is taken with the CLT outputs detached, so it captures only paths through residual additions and frozen attention. We never form $\boldsymbol{J}^{M,\bullet}$ explicitly. A single backward pass from a_t^M produces the gradient at every write-layer residual, and gradient hooks contract those gradients against the active decoder rows (Ameisen et al., 2025). Embedding nodes (the pre-CLT residual at the input boundary of the CLT span) and error

nodes (the per-layer reconstruction residual $\boldsymbol{Y}_w^M - \widehat{\boldsymbol{Y}}_w^M$) enter as additional source vectors of the same form, accounting for content the CLT does not yet feature-decompose. Stacking all source-target attributions gives a weighted directed graph $\boldsymbol{A}^M$ with node ordering [features | errors | inputs | logit].

Indirect-influence pruning. The full graph is too dense to interpret directly. A single image typically activates thousands of features and the goal of pruning is to keep only those that materially affect the prediction. The direct edge weight $A^M_{s\to t}$ measures one-hop contribution, which undercounts a node's importance when its influence on the logit flows through chains of intermediate features. We therefore rank nodes by their *indirect influence*, the sum of contributions over all such chains.

To compute it, take the absolute adjacency $|\boldsymbol{A}^M|$, normalise its rows to sum to one to get $\widetilde{\boldsymbol{A}}^M$, place a one-hot weight $\boldsymbol{w}$ on the logit, and form

$$(\boldsymbol{\phi}^M)^\top \;=\; \sum_{k\geq 1} \boldsymbol{w}^\top \big(\widetilde{\boldsymbol{A}}^M\big)^k \;=\; \boldsymbol{w}^\top \widetilde{\boldsymbol{A}}^M \big(\mathbf{I} - \widetilde{\boldsymbol{A}}^M\big)^{-1}, \tag{11}$$

where ϕ_v^M is the indirect influence of node v on the logit, and the closed form on the right is the geometric (Neumann) sum of the series on the left. In practice we compute Eq. 11 by power iteration. Operationally, the logit's unit weight is propagated backward to its parents in proportion to normalised edge magnitudes, then redistributed further upstream, until the propagated mass has decayed.

Pruning then proceeds in two passes. Nodes are ranked by ϕ_v^M and the smallest set whose cumulative influence reaches a threshold τ_n (default 0.8) is kept. Edges are then pruned by an analogous influence-weighted score against τ_e (default 0.98). Inputs and the logit are always retained. A final cleanup iteratively drops feature nodes that have no outgoing edge to a kept downstream node, or that have no incoming edge except at the first CLT layer. Such nodes cannot lie on any path from input to logit and so cannot affect the prediction. The resulting subgraph is the per-image circuit for model M.

A.4. From Per-Image to Class-Level Circuits

The standard granularity for circuit extraction in the CLT literature is per-input. The natural functional unit of a vision classifier is the class, so we work at that granularity. For class c, we define a build set $\mathcal{I}_c$ of images with which to compute the class level circuit. The per-image circuit $C_{c,i}^M$ is extracted on model M via the standard attribution-graph pipeline (§A.3), the class-level circuit is the union of circuits constructed over the build images,

$$C_c^M \;=\; \bigcup_{i\in\mathcal{I}_c} C_{c,i}^M.$$

A feature or edge is kept iff it appears in at least one image. Recording the per-image frequency $\nu_c^M(x) = |\{i : x \in C_{c,i}^M\}|/|\mathcal{I}_c|$ for each circuit element x lets us tighten the merge post hoc to $C_c^M(\tau) = \{x : \nu_c^M(x) \geq \tau\}$ without re-extracting graphs; $\tau = 0$ recovers the union, $\tau = 1$ the intersection.

Figure 9 sweeps τ and the build-set size $|\mathcal{I}_c|$. Per-image circuits ($\tau \approx 1$) are pruned too aggressively to recover classification on their own; the union over $|\mathcal{I}_c| = 100$ images sits in the regime where sufficiency and faithfulness both saturate, which we adopt as the default. Cross-model circuit metrics in §4 are defined on these merged class-level circuits, with feature-index correspondence given by the universal partition (§3.2).

A.5. Training Details and Hyperparameters

This subsection collects the hyperparameters, schedules, and implementation choices used to train the UCs reported in the main paper. The values are summarised in Table 1.

Architecture sizing. The CLT replaces the MLPs of blocks $[L_s, L_e) = [6, 12)$ in each model (a 6-layer CLT span starting two-thirds of the way through the 12-block ViT-Base backbone). Per-model encoders and decoders operate at $d_{\text{model}} = 768$; the feature space has $d_{\text{univ}} = 3072$ universal dimensions ($4d_{\text{model}}$) and $d_{\text{spec}} = 1536$ specific dimensions per model ($2d_{\text{model}}$), for $d_{\text{feat}} = 4608$ total. Encoders are MLPs with batch normalisation; decoders are linear layers with L_2-normalised rows.

Models and classification head. Given two frozen pretrained vision transformers M_A, M_B that may differ in training objective and data, we put both models on a common evaluation footing by discarding each backbone's native head and

Table 1. Universal Circuit training hyperparameters (STL10).

Hyperparameter	**Value**
CLT span $[L_s, L_e)$	$[6, 12)$
d_{model}	768
d_{univ}, d_{spec}	3072, 1536
Top-k universal (initial $\rightarrow$ final)	$256 \rightarrow 32$
Top-k specific (initial $\rightarrow$ final)	$64 \rightarrow 16$
λ_{cross}	1.0
λ_{e2e} (target)	3.0
λ_{aux}	0.1
Optimiser	AdamW
Learning rate (peak)	3×10^{-2}
Final learning rate	10^{-6}
Weight decay	10^{-5}
Gradient clipping (max norm)	1.0
batch size	32
Epochs	150
LR warmup (epochs)	5
Top-k warmup, anneal (epochs)	20, 40
λ_{e2e} ramp start, duration (epochs)	40, 20
Specific-dropout p (start $\rightarrow$ final)	$1.0 \rightarrow 0.5$
Specific-dropout hold, ramp (epochs)	40, 40

attaching a randomly initialized linear classifier over mean-pooled patch tokens, trained from scratch on the target dataset with the backbone frozen. Mean-pooling weights every patch equally, ensuring the CLT features trained against this objective must be spatially consistent to contribute to the logit. Attention-pool readouts such as SigLIP's, by contrast, are optimised for image–text alignment and known to discard spatial detail (Zhou et al., 2022; Beyer et al., 2024). Holding the head architecture constant across the diversity ladder leaves only the upstream training objective and data as the variables we sweep, and gives each model a softmax over the same target classes against which the CLT's end-to-end objective is defined.

Compute and Wall-Clock All UC training runs were completed on a single workstation with $2\times$ NVIDIA RTX 6000 GPUs (frozen ViT pair on one device, trainable UC encoders and decoders on the other; §A.5). Each model pair trained for 150 epochs on STL10 in approximately **24 hours** of wall-clock time. Per-class circuit extraction is GPU-bound, the frozen forward pass, batched gradient-hook backward passes for edge attributions, and the Neumann-series power iteration all run as CUDA operations, with only threshold sorting on CPU and fits comfortably on a single RTX 6000 (peak $\sim$6 GB). Constructing a class-level circuit from 100 per-image attribution graphs takes approximately **1 hour per class**. The three pairs and ten classes reported in the main paper account for roughly $3 \times 48 + 30 \times 1 \approx 175$ GPU-hours. Across the full research project, including architecture exploration and ablations not reported here, we trained approximately **50 model pairs** and extracted class-level circuits for ten classes per pair, for a total of approximately $50 \times 48 + 500 \times 1 \approx \mathbf{2{,}900}$ **GPU-hours** ($\sim$17$\times$ the reported figure).

Curricula. Three quantities are annealed during training. Top-k on each slice follows a piecewise-linear schedule: hold at the initial value during a warmup window, linearly interpolate to the final value over the anneal window, hold at the final value thereafter. Starting dense and ending sparse helps the encoder bank populate features before the budget tightens. The end-to-end weight λ_{e2e} is held at 0 during a reconstruction-only window then ramped linearly to its target. The specific-dropout schedule applies a per-layer Bernoulli with probability p to the target's specific slice in the cross-prediction hybrid: with probability p, the slice $\boldsymbol{S}_s^B$ is zeroed before the hybrid code is decoded through the target's bank. We hold $p = 1.0$ for an initial window, ramp linearly to $p = 0.5$, and hold at 0.5 for the rest of training.

Auxiliary loss. The auxiliary term $\mathcal{L}_{\text{aux}}^M$ is the top-k auxiliary loss (Ameisen et al., 2025) computed against the residual reconstruction error: features whose pre-activations rank in the top-k_{aux} but did not make the top-k active set are decoded

against the residual $\boldsymbol{Y}_w^M - \widehat{\boldsymbol{Y}}_w^M$, providing a gradient signal that revives features which would otherwise stay dead. Cross-prediction uses the same auxiliary form computed through the target model's decoder bank, so dead-feature recovery in cross-prediction is local to each model's own bank.

Per-model preprocessing. Each model wrapper carries its native preprocessing transform. For STL10, raw 96×96 images are upsampled bicubically to each model's native input resolution: 224 for the patch16 models (ViT, SigLIP, DINOv1).

Normalization statistics. Pre-norm2 residual activations are standardised per (model, layer) using mean $\boldsymbol{\mu}_\ell^M$ and standard deviation $\boldsymbol{\sigma}_\ell^M$, computed once on the training set's patch tokens. The CLT operates in this standardised space and decoder outputs are denormalised before being added back into the residual stream. CLS tokens are excluded from the statistics computation since the CLT processes patches only.

A.6. Joint Activation Maximization

We visualize Universal Circuit features by Activation Maximization, adapting the Coordinated Activation Maximization (CAM) objective of Thasarathan et al. (2025). CAM was defined for a universal sparse autoencoder trained on the final-layer activations of each model: given a shared concept index k, one optimizes a separate input $\boldsymbol{x}^{M,\star}$ *per* model so that each model's activations align (in cosine) with that model's decoder column for concept k, the standard SAE feature direction targeted by MACO, regularized by the MACO Fourier-phase prior $\mathcal{R}$ of Fel et al. (2023a). We modify CAM in two ways for our cross-layer per-feature setting. First, we augment the decoder-direction term with a *dual* encoder term, so the score asks both whether the decoder write direction shows up in the residual stream *and* whether the encoder also fires the feature. Second, we optimize a *single* shared image across all K models rather than K separate inputs, so one picture lights up the same universal feature in every contributing CLT.

Per-model dual score. Fix a CLT layer ℓ and a universal feature index f. For an image $\boldsymbol{x}$ presented to model M (with M's native input preprocessing and the per-(model, layer) standardization from §A.5), define two scalar quantities:

- the *decoder alignment* of feature f – CAM's term, evaluated at the CLT layer rather than the final layer – the patch-averaged cosine between M's standardized mlp-output and the L_2-normalized self-decoder direction $\boldsymbol{d}_f^M = \boldsymbol{D}_{\ell,\ell}^M[f,:]$, clamped to $[0,1]$ before averaging: $c_f^M(\boldsymbol{x})$;
- the *encoder firing* of feature f, mean-pooled over patches: $z_f^M(\boldsymbol{x}) = [\boldsymbol{E}_\ell^M(\boldsymbol{x})]_f$.

The per-model score is their sum,

$$\mathcal{S}_f^M(\boldsymbol{x}) \;=\; c_f^M(\boldsymbol{x}) \;+\; \lambda_z\, z_f^M(\boldsymbol{x}), \tag{12}$$

with $\lambda_z = 1$ throughout. The encoder term z_f^M is included as an empirical regularizer that we found produced more structurally coherent MACO images and strongly activated the decoder direction $\boldsymbol{d}_f^M$ when fed back through M. The cosine is clamped because the unbounded objective of Fel et al. (2023a); Thasarathan et al. (2025) can otherwise be satisfied by inflating residual magnitudes along $\boldsymbol{d}_f^M$ rather than aligning patch directions.

Joint objective across models. Universal feature indices are bound across models by joint-TopK (§3.2), so the slot (ℓ, f) refers to the same concept in every model $M_{\in(A,B)}$. We exploit this by optimizing a *single* shared image $\boldsymbol{x}^\star$ that maximizes the average per-model score,

$$\boldsymbol{x}^\star \;=\; \arg\max_{\boldsymbol{x}} \; \frac{1}{K}\sum_{k=1}^{K} \mathcal{S}_f^{M_k}(\boldsymbol{x}) \;-\; \lambda\,\mathcal{R}(\boldsymbol{x}), \tag{13}$$

where $\mathcal{R}$ is the MACO regularizer of Fel et al. (2023a). Whereas CAM produces K separate optimized inputs, Eq. 13 returns a single image that visualizes how the same universal feature manifests across all models on identical pixels, the natural visual counterpart of a feature index that is shared by construction.

A.7. Pool-Membership Sufficiency

Figure 10 measures classification accuracy when a target model $M \in M_A, M_B$ is restricted to one of four feature subsets: features kept by *both* M_A's and M_B's per-class circuits (the intersection), features specific to M, features kept only by the

partner, and a random-feature control. Results are pooled across the two choices of target M. Across all three pairs, only the feature subsets that M itself retains classify well — both the intersection and M-specific features sit near ceiling. Features kept only by the partner fall to the random-feature baseline of $\sim 37\%$, despite living in the cross-model universal pool by construction. The operational meaning of "universal across models" is therefore mutual attestation: a feature transfers across the $M_A! \leftrightarrow! M_B$ boundary only when both circuits independently retain it. Pool membership alone does not suffice.

A.8. Cross-model circuit agreement by layer

Figure 11 plots, for each model pair $M_{\in A,B}$ and each CLT layer L, the number of features kept by both models in their per-class circuits — the per-layer intersection $|\mathcal{I}[L]|$. Panel (a) shows raw counts; panel (b) shows the same trajectories with each class normalized by its own total $|\mathcal{I}|$, isolating shape; panel (c) shows the per-layer Jaccard $|\mathcal{I}[L]|/|A[L] \cup B[L]|$, which further controls for the per-layer circuit size of each pair. Two findings. First, $|\mathcal{I}[L]|$ decays monotonically with depth in every pair, and absolute counts shrink as the M_A–M_B objective gap grows (panel a). Second, after normalizing out size (panel b), the more diverse pairs concentrate proportionally more of their intersection in early layers than ViT–ViT does. Panel (c) disentangles the two contributions: ViT–ViT's per-layer agreement rate is roughly depth-uniform near 50%, so its early-layer concentration in panels (a–b) is largely a circuit-size effect; for DinoV1–ViT and SigLIP–ViT, agreement itself decays with depth. Early-layer features are shared regardless of training objective; late-layer features diverge as objectives diverge.

A.9. Downstream Layer Importance

Why per-class circuits drop the last CLT layer. Figure 12 measures, for each CLT layer L, two quantities per (pair, model): the share of feature-to-logit attribution at L (panels a–c, what the pruner sees), and the change in classification KL when layer L's CLT features are zeroed end-to-end (panels d–f, what the model loses). Across all six (pair, model) combinations the last layer L_5 shows essentially no causal effect: ablation KL stays below 0.012 nats. For ViT and DinoV1 the attribution share at L_5 is also near zero, so the pruner's habit of dropping L_5 from per-class circuits is correct — those features are simply unused for classification. SigLIP is the partial exception: it retains roughly 13% attribution share at L_5, and its class circuits include some L_5 features accordingly, but the causal metric still measures negligible effect when those features are ablated, suggesting they are redundant with earlier layers under SigLIP's attention-pooled read-out. As a secondary observation, DinoV1 and SigLIP concentrate causal weight extremely early (L_0 ablation costs ≈ 0.4 and 0.7 nats), while ViT models distribute it more evenly across mid-layers.

A.10. Societal Impacts

UC is a diagnostic interpretability tool: it analyses already-released vision encoders to expose the multi-layer circuits they share, without introducing new generative, classification, or surveillance capabilities. The intended positive impact is to support transparency, auditing, and debugging of vision models — enabling practitioners to identify whether two systems rely on the same internal computation, to localise where their behaviour diverges, and to transfer concept-level probes across architectures for tasks such as bias diagnosis or failure-mode analysis. We do not foresee a direct path to harmful applications: the framework operates on small frozen encoders, releases no scraped data or generative model, and does not improve the capability of the analysed models. A speculative dual-use concern is that improved circuit-level understanding could in principle assist reverse-engineering of proprietary systems; we view this as low-risk in the present setting (publicly released backbones, STL10-scale evaluation) and substantially outweighed by the benefits of open interpretability tooling for safety and accountability.

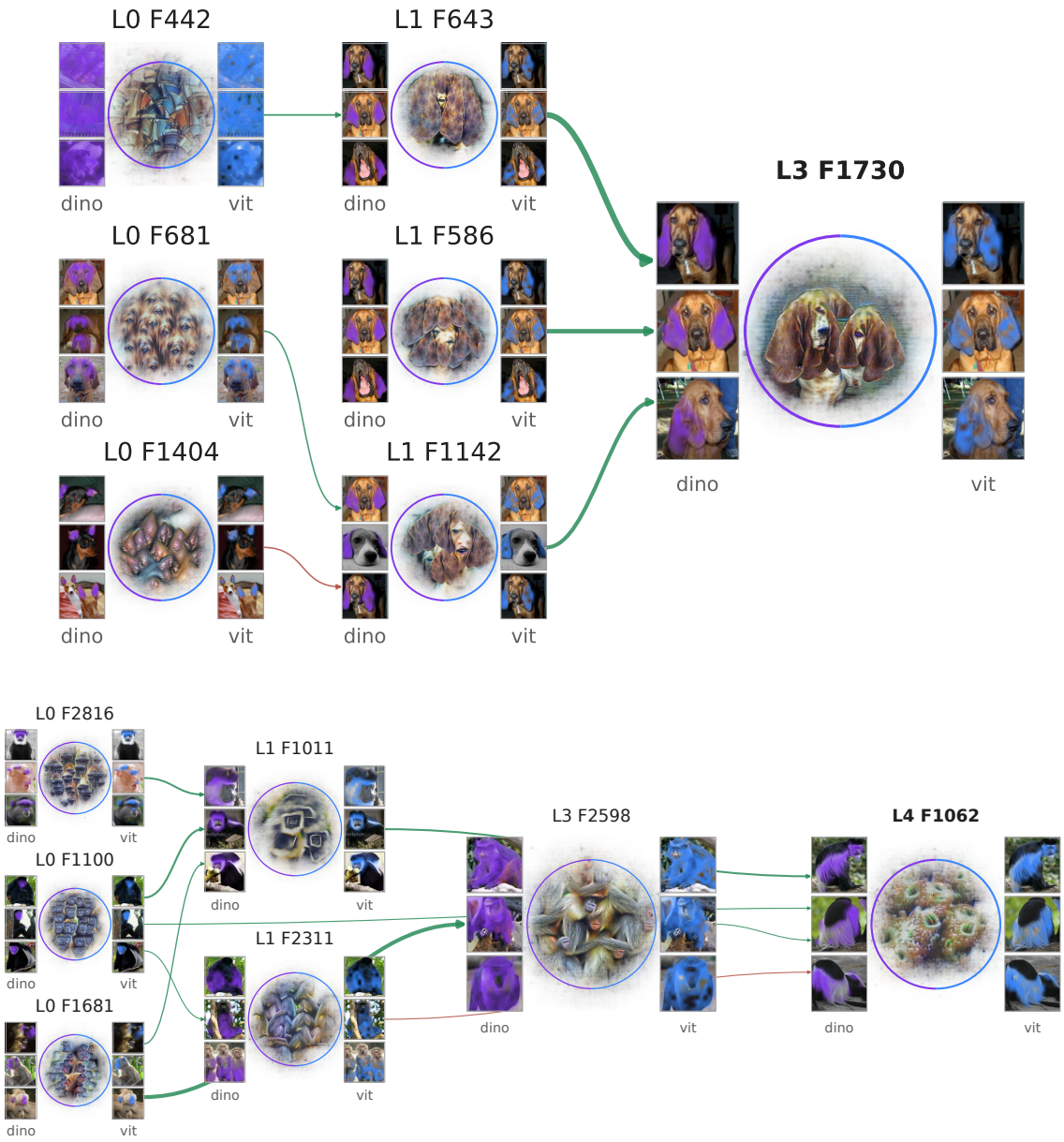

Figure 6. **Qualitative results of cross-model class circuits for Dino–ViT.** We visualize a monkey and dog sub-circuit present in both Dino and ViT. In the dog sub-circuit, the models isolate floppy ears for a particular breed of dog. This is evident in the way node L0 F1404, a feature for pointy ears actively supresses a downstream feature for floppy ear L1 F1142. In the monkey sub-circuit we find a rare optimization artifact for the decoder direction in L4 F1062 which fails to recover coherent structure that reflects the associated heatmaps.

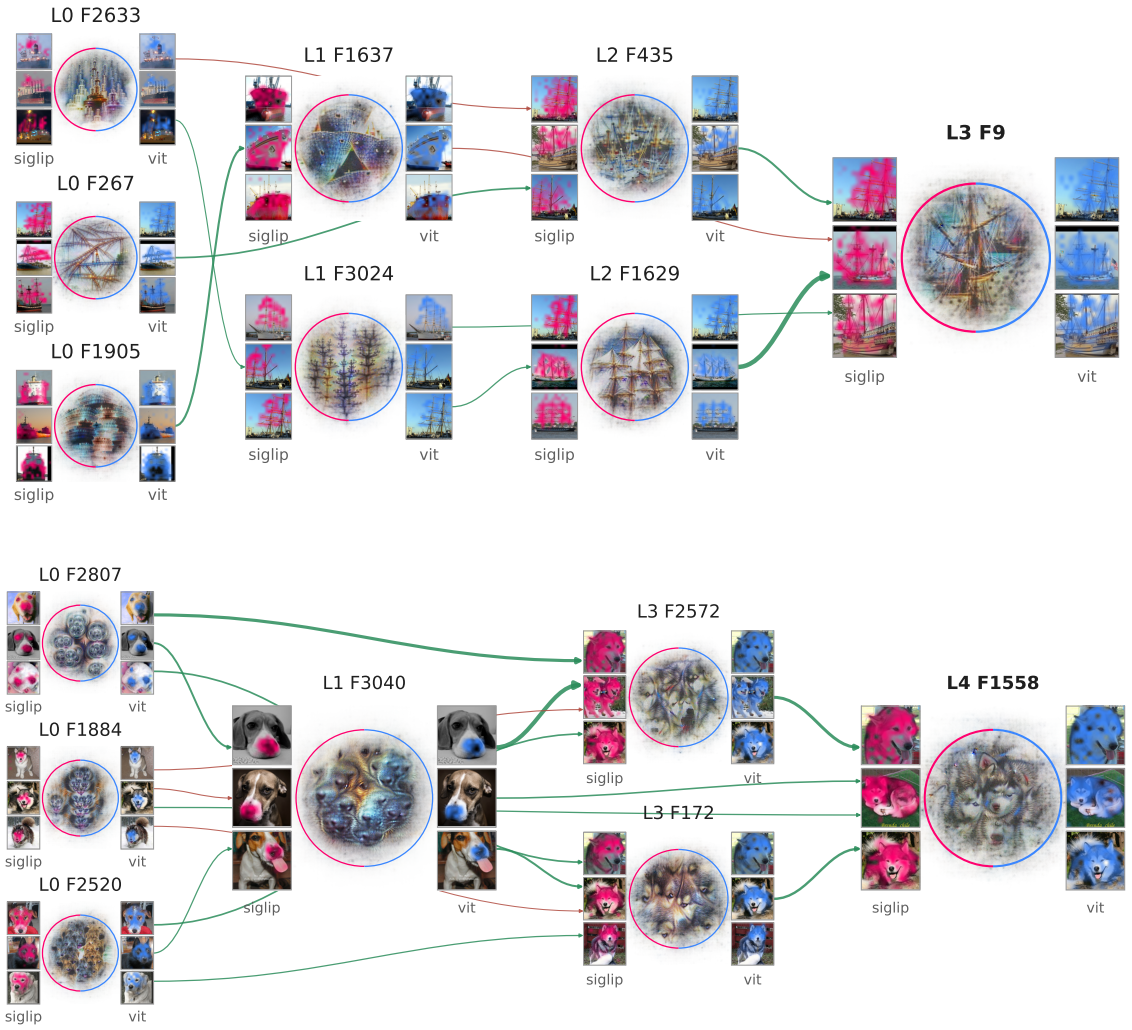

Figure 7. **Qualitative results of cross-model class circuits for SigLIP.** We show examples of class-level cross-model sub-circuits for the class ship (top) and dog (bottom). In the ship circuit, early nodes attempt to differentiate particular classes of ships, identifying modern exhaust pipes and cargo hulls which negatively attribute (red arrow) along side positively attributing masts which compose to identify ships with open sails. In the dog circuit, early features identify facial attributes, beginning with features for round shapes like eyes and nose which are used to compose huskies in later layers.

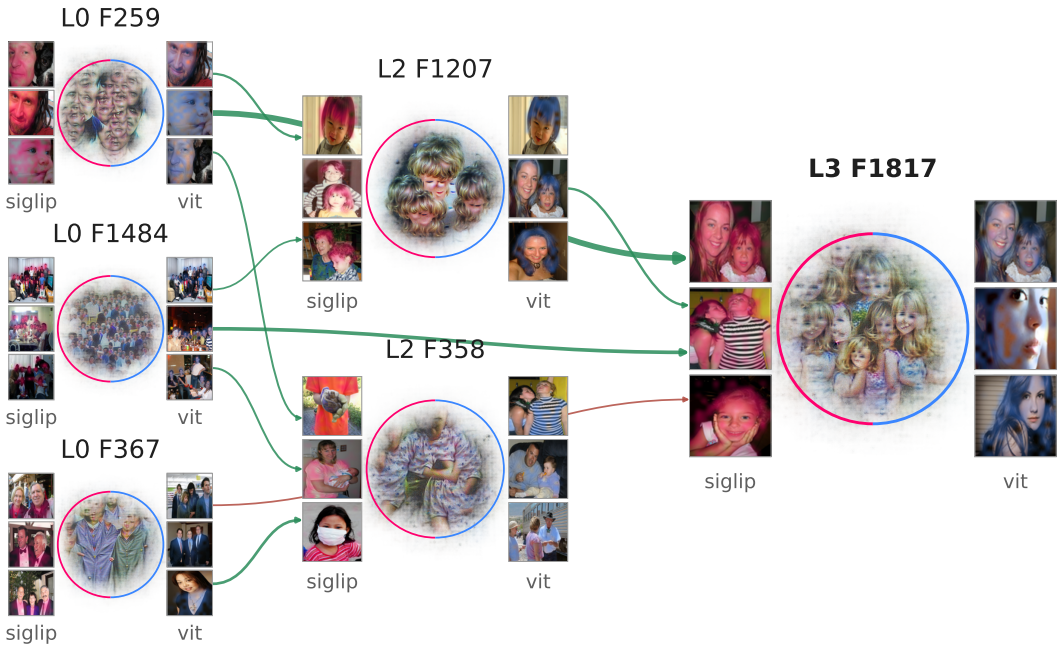

Figure 8. **Qualitative visualization of a transplanted cross-model circuit.** We train a concept-space probe representing VOC `person` on ViT, transfer this probe to SigLIP, and extract a cross-model circuit by fitting a residual stream vector following the bridge readout(§A.2). Both ViT and SigLIP compose a sub-circuit for representing people which involve identifying faces at different depths (e.g., groups at a distant versus close ups), and necklines at early layers which later compose into specific hair-types, clothes, and finally groups of people.

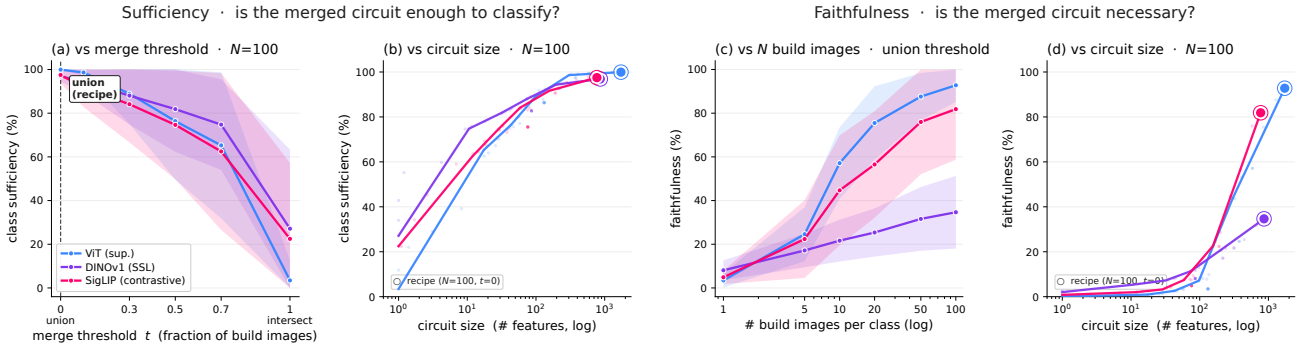

Figure 9. **Class-level circuit construction.** Sufficiency and faithfulness of the merged class-level circuit as we vary (a) the merge threshold τ from union ($\tau = 0$) to intersection ($\tau = 1$), (b,d) the resulting circuit size, and (c) the number of build images $|\mathcal{I}_c|$ contributing to the union. The default recipe ($|\mathcal{I}_c| = 100$, $\tau = 0$) sits in the regime where both sufficiency and faithfulness saturate, supporting the class-level union as the natural granularity for our universality questions.

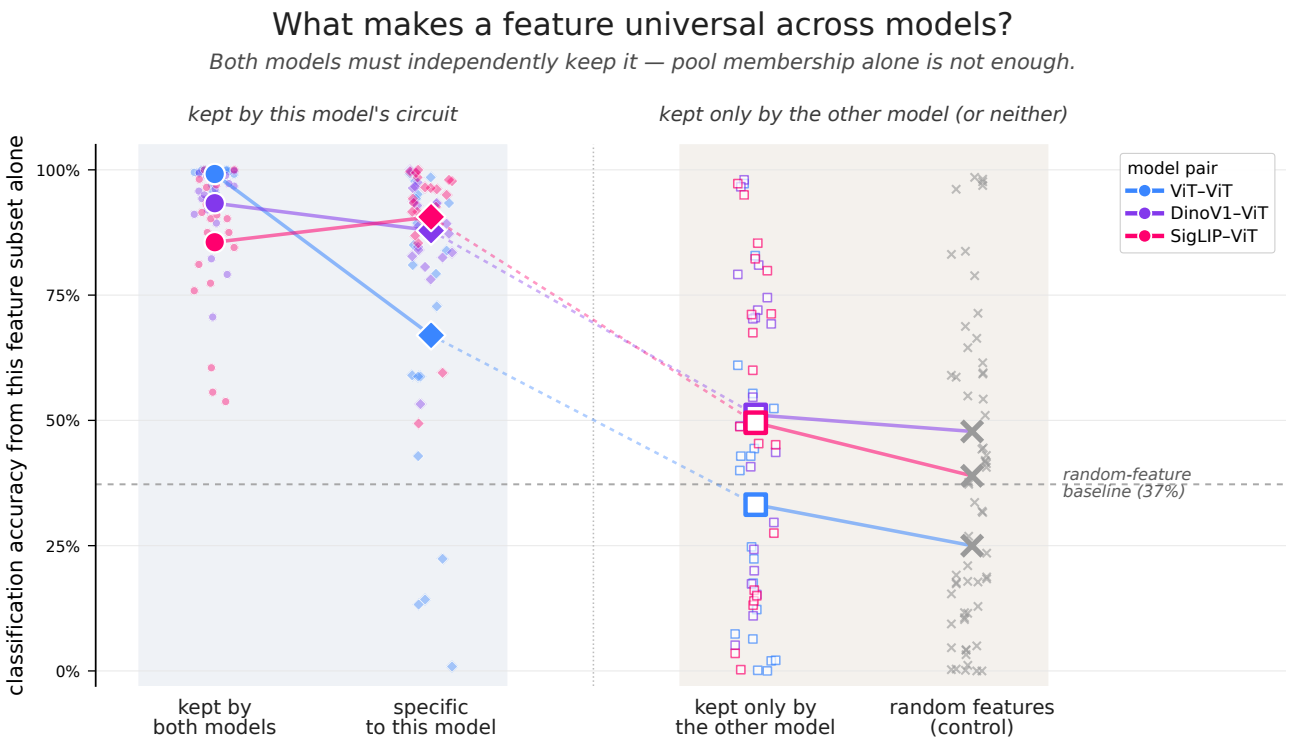

Figure 10. Universality is mutual attestation, not pool membership. Each colored line is one model pair $M_{\in A,B}$ — ViT–ViT blue, DinoV1–ViT purple, SigLIP–ViT pink. At each x-position the model is given access to only one subset of features and asked to classify STL10 images. Reading left to right: features kept by both models' per-class circuits (the shared core); features in this model's specific pool that this model kept; features the other model kept but this model did not; and a random-feature control. Results are pooled across both directions (each model takes the role of "this model" in turn). The y-axis is the resulting classification accuracy. Each faint dot is one (class, direction) measurement; the bold marker is the pair mean. The shaded zones group subsets kept by this model (left) versus those not kept (right); the dashed gray line marks the random-feature baseline ($\approx 37\%$). The slope-line is dotted across the divide to emphasize the gap. All three pairs show the same pattern: the left two subsets sit near ceiling, the right two on the baseline.

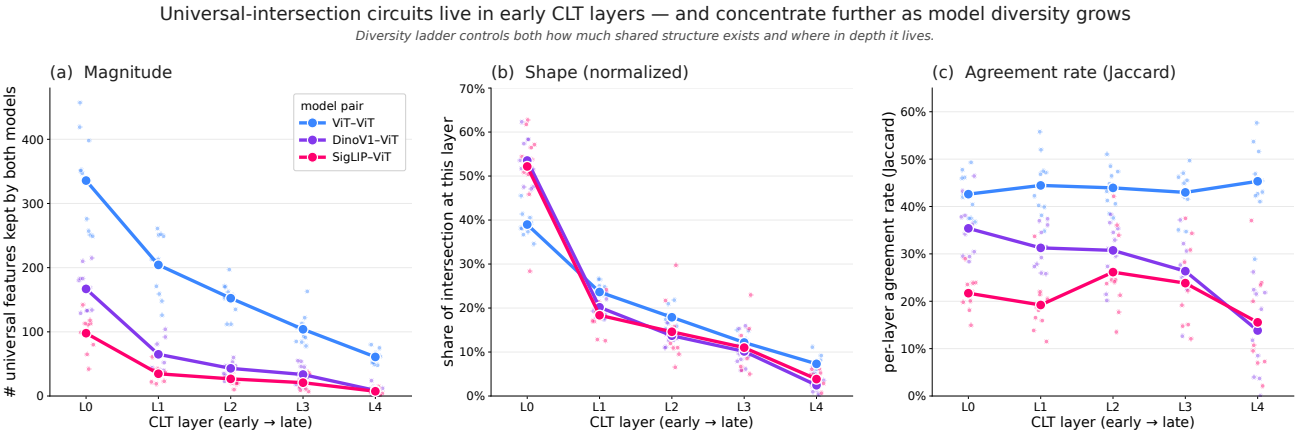

Figure 11. Universal-intersection circuits live in early CLT layers, and concentrate further as model diversity grows. Each colored line shows one model pair $M_{\in A,B}$ — ViT–ViT in blue, DinoV1–ViT in purple, SigLIP–ViT in pink — traced from early (left) to late (right) CLT layers. Panel (a) measures how many features both models keep in their per-class circuits at each layer — the size of the shared core. Panel (b) shows the same trajectory normalized per class to sum to one, so only the shape of the depth distribution is visible (factoring out total size). Panel (c) shows the agreement rate at each layer — the fraction of features kept by either model that both also keep — which controls for differences in circuit size. Each faint dot is one STL10 class (ten per pair); the bold line is the across-class mean.

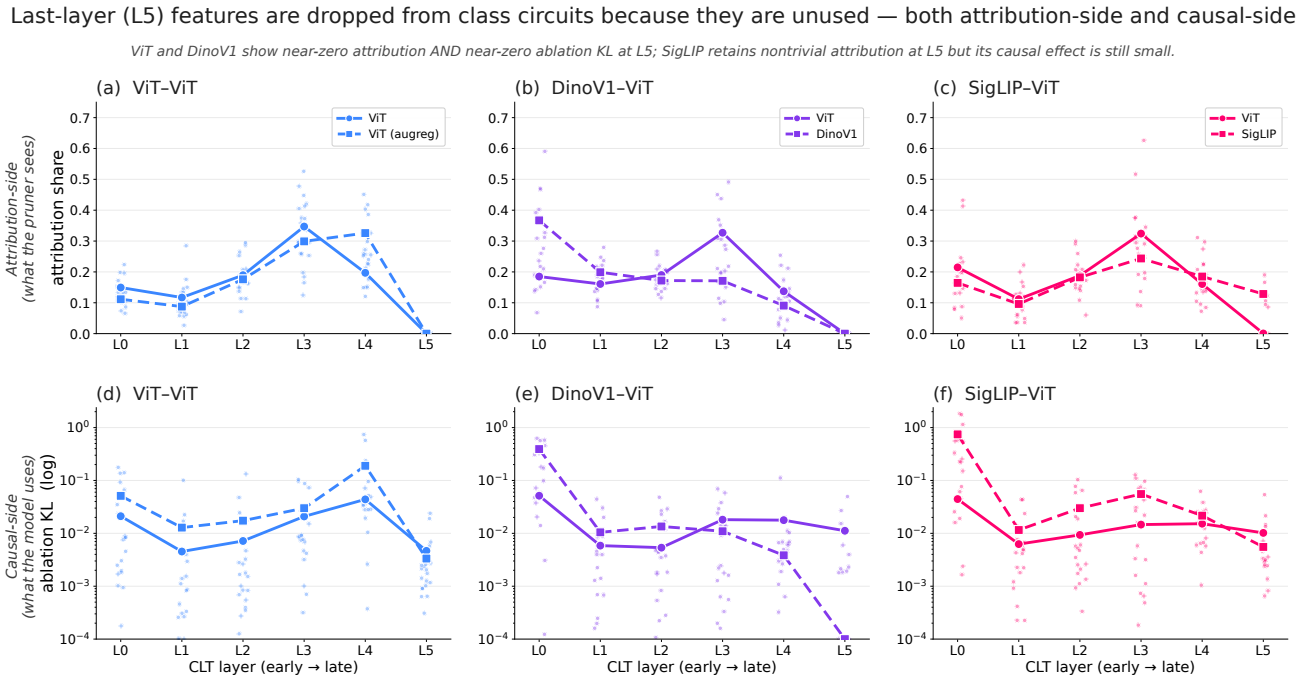

Figure 12. Why per-class circuits drop the last CLT layer. For each pair $M_{\in A,B}$ — ViT–ViT (blue), DinoV1–ViT (purple), SigLIP–ViT (pink) — the figure measures per-layer feature importance two ways. The top row (panels a–c) shows what the per-image circuit pruner sees: the share of feature-to-logit attribution at each layer. The bottom row (panels d–f, log scale) shows what the model actually loses if a layer's CLT features are zeroed end-to-end. Within each panel the solid + circle line is one model in the pair, the dashed + square line is the other (labeled in each top-row legend). Bold lines are means across ten STL10 classes; faint dots show per-class spread. At the last layer (L_5) both metrics are near zero for ViT and DinoV1 — the pruner correctly drops these features because they are unused. SigLIP retains some attribution at L_5 but its causal effect is still negligible. DinoV1 and SigLIP additionally concentrate causal weight extremely early (L_0 ablation alone costs much more KL than any later layer), while ViT models spread it across mid-layers.