ATCG: STRUCTURAL TRAIT MOLDING FOR COMBINATORIAL AUDITING OF DEEP CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep classifiers map learned embeddings to class labels through a flat prediction head and achieve high accuracy. However, when classes contain heterogeneous latent subpopulations, the opaque, unstructured embedding space makes the internal basis of predictions difficult to audit. Inspired by the combinatorial richness of modern biological taxonomy, we investigate the feasibility of representing intra-class heterogeneity through the combinatorial composition of "root elements." We introduce *Artificial-Trait Configurative-Generation* (ATCG), a structured framework that retrofits pretrained backbones with a *Pinball head*: a per-class binary hierarchy that learns a coordinate system of basis traits and induces a combinatorial barcode space that encodes intra-class heterogeneity. To structure this combinatorial space for auditing, ATCG aggregates class evidence from basis traits bottom-up via *Max-Dominance* through a hierarchy of intermediate-layer traits, while a geometric *Molding Bonus* discourages trait collapse by encouraging sibling specialization into complementary patterns, together yielding sparse basis-trait barcodes. The induced barcodes reveal whether each prediction's evidence is *consistent*, *conflicting*, or *insufficient*, and detect intra-class shifts that may be missed by class-label frequencies at the cohort level. We validate this on controlled Stacked-MNIST experiments where barcode-based monitoring identifies intra-class shifts, demonstrate it on Camelyon17-WILDS where barcode auditing relates prediction accuracy to patient-share profiles of basis traits, and show it generalizes to non-vision settings via TCGA RNA-seq classification. ATCG further integrates with Concept Bottleneck Models as a structural backend, exposing multiple latent realizations under a single concept.

1 INTRODUCTION

Modern deep classifiers achieve unprecedented accuracy by mapping high-dimensional feature embeddings through flat prediction heads. However, as data complexity grows, deep embeddings become increasingly opaque. When a class contains diverse latent subpopulations, a flat head collapses these nuances into an unstructured space, masking the internal basis of predictions and rendering them difficult to audit.

Addressing this lack of transparency is central to Explainable AI (XAI). For instance, Concept Bottleneck Models (CBMs) Koh et al. (2020) introduce interpretable concept layers that filter signals before they reach the prediction head, enabling human intervention in safety-critical tasks. In this work, we investigate a different path: representing intra-class heterogeneity through the combinatorial composition of "root elements." Inspired by biological taxonomy, where complex phenotypes are organized through combinations of genetic traits, we propose a framework that hierarchically organizes latent diversity into a traceable combinatorial code.

We introduce *Artificial-Trait Configurative-Generation* (ATCG), a structured retrofit that replaces the flat head with a per-class binary hierarchy of learned decision factors. Instead of routing each sample to a single leaf, ATCG allows multiple basis traits to activate simultaneously and aggregates evidence through Max-Dominance, while a geometric *Molding Bonus* encourages complementary specialization among sibling traits. The resulting trait barcodes make within-class evidence organization explicit.

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

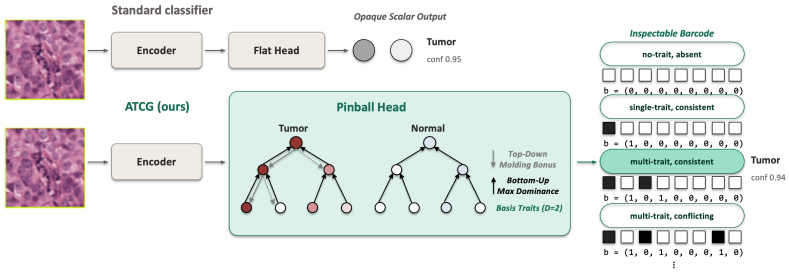


Figure 1: ATCG framework overview. Top: Standard classifier, Bottom: ATCG framework.

This structure supports auditing at two levels. At the sample level, the resulting barcodes distinguish predictions supported by consistent evidence from those with conflicting or insufficient evidence. At the cohort level, changes in barcode prevalence expose intra-class distribution shifts that may be missed by class-label frequencies. To leverage combinatorial richness and prevent node collapse, ATCG introduces a geometric *Molding Bonus* objective that encourages sibling traits to specialize into complementary, sparse patterns.

We evaluate ATCG on synthetic and natural imaging datasets, and on genomic data from The Cancer Genome Atlas (TCGA), to confirm portability beyond vision. On Camelyon17-WILDS, an unexpected accuracy gain triggers barcode-level inspection, which relates differences in prediction accuracy to the patient-share profiles of basis traits (Sec. 4.4). We also demonstrate ATCG's versatility as a tool for benchmarking generative model fidelity and as a structured backend for CBMs that admits multiple latent realizations under a single concept.

Contributions. This paper makes three primary contributions. **Architectural transparency:** we propose the ATCG framework and show that replacing the flat classification head with a per-class *Pinball Head*, shaped by Max-Dominance aggregation and a geometric *Molding Bonus*, yields sparse and inspectable trait activations and combinatorial barcodes. **Diagnostic utility:** these barcodes act as auditing signals at both the sample and cohort levels, flagging predictions without consistent evidence and localizing intra-class distribution shifts through changes in barcode prevalence. **Theoretical foundation:** we prove that the induced *Hierarchical Barcode Distance* is depth-monotonic and bi-Lipschitz equivalent under sparsity, providing a principled geometry for downstream barcode analysis and visualization. Beyond these core contributions, ATCG also plugs into existing concept-based pipelines as a structured backend, where each concept can admit multiple latent realizations.

2 Related Work

A vast body of XAI research focuses on *post-hoc* explanations, such as gradient-based attribution Selvaraju et al. (2017); Shrikumar et al. (2017) and perturbation methods Lundberg & Lee (2017). In contrast, *ante-hoc* approaches impose interpretable structure on the predictive mechanism itself. Representative examples include prototype-based methods such as ProtoPNet Chen et al. (2019) and ProtoTree Nauta et al. (2021). Concept-based methods include Testing with Concept Activation Vectors (TCAV) Kim et al. (2018) for post-hoc analysis, as well as CBMs Koh et al. (2020) and variants such as Post-hoc Concept Bottleneck Models (PCBM) Yuksekgonul et al. (2023), Concept Embedding Models (CEM) Espinosa Zarlenga et al. (2022), and Vision-Language-Guided Concept Bottleneck Models (VLG-CBM) Srivastava et al. (2024). ATCG is complementary to this latter family and can serve as a structured backend for concept-based pipelines (Sec. 5.2).

Hierarchical classification often relies on external taxonomies Yan et al. (2015) or learned routing mechanisms such as Neural-Backed Decision Trees (NBDT) Wan et al. (2021). These methods primarily organize *inter-class* decisions through a global tree. ATCG adopts an orthogonal notion of hierarchy: instead of routing samples across classes, it learns a *per-class* binary hierarchy of Artificial Traits that decomposes each class into structured evidence variants and synthesizes class evidence via Max-Dominance aggregation. Thus, whereas prior work focuses on routing between classes, ATCG organizes how each class is internally realized through explicit basis-trait composition.

3 METHOD

Artificial-Trait Configurative-Generation (ATCG) retrofits a pretrained backbone with a *per-class hierarchical* decision head, termed the *Pinball Head*. For each class, Pinball parameterizes class evidence with a hierarchy of learned *Artificial Traits*, aggregated bottom-up via Max-Dominance, yielding inspectable *trait activations* and discrete *trait barcodes* for auditing. To prevent trait collapse, the *Molding Bonus* provides top-down guidance that encourages complementary sibling specialization on positive samples and joint suppression on negative samples. Figure 1 summarizes the system.

3.1 PINBALL HEAD: PER-CLASS HIERARCHY VIA MAX-DOMINANCE

Given data $\{(x_i, y_i)\}_{i=1}^N$ with C classes, a backbone f_θ maps each input to an embedding $z = f_\theta(x) \in \mathbb{R}^K$. For each class c, Pinball installs a balanced binary hierarchy of depth D. We project the embedding to the basis trait layer via a mapping g_ϕ with scaled sigmoid activation: $\mathbf{t}^{(D)} = g_\phi(z) \in [0, \alpha]^{C \times 2^D}$, where $\alpha > 0$ is a fixed scaling hyperparameter (Appendix B.3).

In this structure, we refer to the bottom-layer nodes ($t_{c,j}^{(D)}$) as *basis traits*, the evidence units specialized for distinct intra-class variants. Nodes at intermediate layers serve as *Composite Traits*, the evidence units that aggregate basis traits to form progressively broader evidence scopes.

Prediction relies on a bottom-up aggregation rule inspired by the principle of genetic dominance:

$$t_{c,j}^{(d)}(x) = \max\Big\{t_{c,2j-1}^{(d+1)}(x),\, t_{c,2j}^{(d+1)}(x)\Big\}, \quad d = D-1, \ldots, 0. \tag{1}$$

Functioning as a logical disjunction, this operation enforces hierarchical consistency, guaranteeing that evidence detected by any basis trait propagates monotonically to the top-level node ($t_{c,1}^{(0)}$). The top-level value serves as the final class logit, and the prediction is $\hat{y} = \arg\max_c t_{c,1}^{(0)}(x)$ optimized by standard cross-entropy. By associativity of the maximum, $t_{c,1}^{(0)}(x) = \max_{j=1,\ldots,2^D} t_{c,j}^{(D)}(x)$: The class logit is the maximum activation over that class's basis traits, an operation related to maxout Goodfellow et al. (2013) and max-based channel-out Wang & JaJa (2014).

3.2 MOLDING BONUS: TOP-DOWN COMPLEMENTARITY AND SUPPRESSION

Max-Dominance alone guarantees signal propagation but does not prevent trait collapse: without explicit guidance, multiple traits may redundantly respond to the same evidence. ATCG therefore introduces the *Molding Bonus (MB)* to provide top-down specialization guidance. For each sibling pair $\mathbf{s} = (t_1, t_2)$, MB encourages two behaviors: (i) complementarity on *positive* samples, where siblings activate on complementary axes to capture distinct intra-class modes; and (ii) joint suppression on *negative* samples, where both remain inactive.

Path-conditional context assignment. Path-conditional context assignment is a training-only sample-selection rule that determines which samples contribute to the Molding Bonus for each sibling pair. The context is defined recursively. For the top-layer ($d = 0$), the context is strictly determined by the ground truth: the top-layer traits for class c are molded to be active for samples $y = c$ and inactive otherwise.

$$\mathcal{X}_{c,1}^{(0)+} = \{x \mid y = c\}, \qquad \mathcal{X}_{c,1}^{(0)-} = \{x \mid y \neq c\}. \tag{2}$$

For deeper layers ($d > 0$), valid context requires an *unambiguous branch assignment* from the parent. Consider a parent node j at depth d. We define its positive context $\mathcal{X}_{c,j}^{(d)+}$ as samples that (i) belong to class c, (ii) activate node j, and (iii) do *not* activate its sibling:

$$\mathcal{X}_{c,j}^{(d)+} = \Big\{x \;\Big|\; y = c,\; t_{c,j}^{(d)}(x) \geq \tau,\; t_{c,\mathrm{sib}(j)}^{(d)}(x) < \tau\Big\}, \tag{3}$$

here the threshold τ is specified in Appendix B. The negative set is the complement $\mathcal{X}_{c,j}^{(d)-} = \mathcal{X} \setminus \mathcal{X}_{c,j}^{(d)+}$. These sets are then used as the positive/negative contexts to mold the *child* sibling pair at depth $d + 1$. Intuitively, the term $t_{c,\mathrm{sib}(j)}^{(d)}(x) < \tau$ filters for path exclusivity: child-level specialization is applied only when the parent-level branch assignment is unambiguous, preventing noisy supervision from multi-path activations.

Geometric molding objective. Given $\mathcal{X}_{c,j}^{(d)+}$ and $\mathcal{X}_{c,j}^{(d)-}$, MB encourages the alignment of the empirical sibling distribution with a geometric template: positive samples are pushed toward the complementary axes $(\alpha, 0)$ and $(0, \alpha)$ (promoting bimodal complementarity), while negative samples are compressed toward the origin $(0, 0)$ (driving joint suppression). We implement this alignment as a differentiable density-matching objective that minimizes a KL divergence to a geometric template, using Kernel Density Estimation (KDE) for empirical density evaluation. We denote the total depth-aggregated bonus as $\mathrm{MB}(\theta, \phi)$. Full derivation of $\mathrm{MB}(\theta, \phi)$ and hyper-parameter specification are provided in Appendix C.

3.3 Training Objective

ATCG jointly optimizes predictive accuracy, structural specialization, and augmentation consistency:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{pin}} \mathcal{L}_{\text{pin}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}. \tag{4}$$

Here, $\mathcal{L}_{\text{CE}}$ is the standard cross-entropy loss. The second term, $\mathcal{L}_{\text{pin}} = -\mathrm{MB}(\theta, \phi)$, converts the MB into a loss term for minimization. When data augmentation is employed, $\mathcal{L}_{\text{con}}$ enforces representation consistency between an input x and an augmented view x' sampled from the same augmentation procedure:

$$\mathcal{L}_{\text{con}} = \mathbb{E}_{x,x'}\left[\|\mathbf{t}^{(D)}(x) - \mathbf{t}^{(D)}(x')\|_2^2\right]. \tag{5}$$

ATCG is trained via a two-stage protocol: Stage 1 trains the Pinball head with the backbone frozen, while Stage 2 additionally refines the encoder at a smaller learning rate.

3.4 Trait Barcodes and Hierarchical Distance

ATCG enables discrete analysis by thresholding basis-trait activations into binary *trait barcodes*[1] $\mathbf{b} \in \{0,1\}^{2^D}$, where $b_j = \mathbb{1}\{t_j^{(D)} \geq \tau\}$. Let $\mathbf{b}^{(d)}$ denote the induced binary pattern at depth d, obtained by propagating the thresholded terminal activations through Max-Dominance. To quantify structural divergence between two samples[2], we define a Hierarchical Trait Distance:

$$\mathcal{D}(i,j) = \sum_{d=0}^{D} w_d \cdot \left\|\mathbf{b}_i^{(d)} - \mathbf{b}_j^{(d)}\right\|_1. \tag{6}$$

We choose exponentially decaying weights w_d to guarantee *strict depth monotonicity*; under barcode sparsity, $\mathcal{D}$ is further bi-Lipschitz equivalent to a level-based ultrametric (Appendix A). High-level structural differences dominate low-level variations, so that distances respect the hierarchical structure for analysis and auditing.

4 Experiments

We validate ATCG across synthetic, real-world, and non-vision domains, addressing three questions: (1) does ATCG learn coherent trait hierarchies under controlled generative factors? (2) does the framework retain accuracy and consistent barcode patterns across diverse settings? (3) do the induced barcodes help identify trait patterns associated with differences in prediction accuracy?

4.1 Experimental Setup

Datasets. We use Stacked-MNIST to validate whether the learned hierarchy reflects controlled generative factor structure. We report in-domain accuracy on MNIST, Fashion-MNIST, CIFAR-10/100, and STL-10 LeCun et al. (1998); Xiao et al. (2017); Krizhevsky (2009); Coates et al. (2011). For realistic domain shift, we evaluate on Camelyon17-WILDS Koh et al. (2021), a multi-hospital histopathology benchmark. To verify portability beyond vision, we further evaluate on TCGA-10, a 10-way cancer-type classification task on RNA-seq tabular data The Cancer Genome Atlas Research

[1] Unrelated to persistence barcodes from topological data analysis; ATCG barcodes are discrete binary patterns over the per-class hierarchy, not multiscale topological features.

[2] Throughout this section and Appendix A, i and j denote sample indices, distinct from the node index j used in Eq. (1).

Network et al. (2013) obtained from UCSC Xena Goldman et al. (2020) (Appendix B). In our CBM extension (Sec. 5.2), we evaluate on CUB, CIFAR-10/100, and ImageNet Wah et al. (2011); Deng et al. (2009) to test scalability in concept-based pipelines.

Implementation. Following the two-stage protocol in Sec. 3.3, we use ResNet-18 for MNIST variants, ResNet-50 for natural-image and Camelyon17 benchmarks, and an MLP encoder for TCGA-10. Image inputs are resized to the backbone's default resolution: 32×32 for MNIST variants and 224×224 otherwise. Unless specified, we use depth $D = 4$, with $D = 5$ for Camelyon17-WILDS (binary task) and $D = 2$ for TCGA-10 (smaller per-class sample sizes). We use $\lambda_{\text{pin}} = 1.0$ and $\lambda_{\text{con}} = 1.0$. Full optimization and architecture details are provided in Appendix B.

4.2 STACKED-MNIST: FACTOR-ALIGNED HIERARCHIES

To demonstrate how ATCG learns an inspectable hierarchy of class-specific evidence, we construct a synthetic Stacked-MNIST dataset with known generative factors. Let (r, g, b) denote the MNIST digits independently sampled into the RGB channels, and define the class label as $y = (r \bmod 2)+2\times(g \bmod 5)$, yielding ten composite classes formed by five Green-digit groups and two Red parities. For example, $\mathcal{C}_0 = \{0, 2, 4, 6, 8\} \times \{0, 5\}$, $\mathcal{C}_1 = \{1, 3, 5, 7, 9\} \times \{0, 5\}$, and $\mathcal{C}_2 = \{0, 2, 4, 6, 8\} \times \{1, 6\}$. This design induces a controlled dependency structure: the Green channel determines the dominant grouping factor, the Red channel introduces secondary within-group variability, and the Blue channel is label-irrelevant noise. Because the Pinball hierarchy is per-class and Red parity is fixed within each class, an ideal hierarchy should therefore exhibit: (i) higher-level splits align with the Green-digit group, (ii) deeper branches align with the specific Red digit identity within each Green group, and (iii) limited sensitivity to Blue. Stacked-MNIST thus provides a diagnostic scaffold to test whether ATCG decomposes latent generative structure into auditable decision logic.

Quantitative analysis. We evaluate predictive performance and structural organization with three metrics. (i) Test Accuracy verifies that structural constraints preserve predictive performance. (ii) Average Silhouette Score measures within-class Red–Green factor separability. We compare ATCG's class-specific basis-trait activations with dimension-matched PCA embeddings (Appendix G.1). (iii) Factor purity quantifies the factor-alignment of binary trait barcodes at each depth, by averaging the majority fraction of activated samples that share the same factor value (Green/Red/Blue) across nodes. We report test accuracy on the test set, while silhouette and factor purity are computed on the training set to diagnose structural alignment with generative factors.

Table 1: Accuracy is evaluated on the test set; silhouette and purity on the training set. Values are three-run means, with SD reported for accuracy and silhouette.

		Joint-RG silhouette		**Factor purity (%, d=1 → d=4)**		
Model	**Test acc. (%)**	**Embedding (PCA)**	**ATCG basis traits**	**Green**	**Red**	**Blue**
Baseline	97.94 ± 0.13	0.026 ± 0.010	–	–	–	–
ATCG Stage 1	97.83 ± 0.04	0.026 ± 0.011	0.159 ± 0.030	$94.97 \rightarrow 98.09$	$23.97 \rightarrow 64.55$	$11.30 \rightarrow 13.11$
ATCG Stage 2	97.84 ± 0.08	0.172 ± 0.025	0.293 ± 0.039	$95.16 \rightarrow 97.85$	$23.91 \rightarrow 73.61$	$11.30 \rightarrow 13.97$

Table 1 shows comparable test accuracy across stages. Stage 1 basis traits show stronger Red–Green grouping than dimension-matched PCA embeddings. Stage 2 improves grouping in both representations, with basis traits retaining a higher silhouette (0.293 vs. 0.172). Factor purity also reveals refinement across depth: in Stage 2, Green purity increases from 95.16% to 97.85% and Red purity from 23.91% to 73.61%. Blue purity remains much lower ($11.30\% \rightarrow 13.97\%$), and Blue silhouettes remain slightly negative in all representations.

Qualitative interpretation. The learned hierarchy exhibits ATCG's bidirectional dynamics: bottom-up Max-Dominance aggregates Basis-Trait evidence into class scores, while top-down molding drives sibling complementarity and suppression. Figure 2 visualizes the branch of $\mathcal{C}_1$. The top three single-activation cases per basis trait are displayed. The full hierarchy is in Appendix Fig. 11. Stage 2 further reorganizes the embedding space rather than merely exposing pretrained backbone geometry (Appendix Fig. 10).

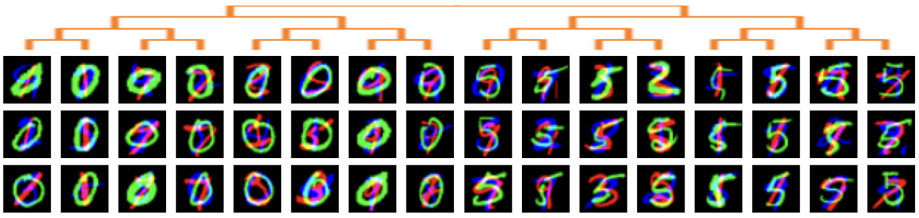


Figure 2: Learned trait hierarchy on Stacked-MNIST (single-activation subset). Higher-level splits align with the Green-digit group; deeper branches refine by specific Red digit identity within each group; the label-irrelevant Blue digit remains mixed.

4.3 Accuracy and Auditing on Real Datasets

Table 2 reports test accuracy across diverse benchmarks for the pretrained backbone (*Base*) and ATCG after Stage 1 and Stage 2, together with barcode-type breakdowns for Stage 2. We focus on two questions: (i) whether the structured retrofit preserves accuracy, and (ii) whether predictions supported by consistent evidence have higher observed accuracy than those with conflicting or insufficient evidence across settings. Stage 1 barcode-type breakdowns and seed-level standard deviations for both stages are reported in Appendix G.

Table 2: Test accuracy (%) across vision and non-vision benchmarks. Stage 2 columns report per-barcode-type breakdown as Acc (Prop.%); Camelyon17 denotes the WILDS-curated split.

		ATCG		Stage 2: Acc (Prop.%)			
Dataset	Base	Stage 1	Stage 2	Single	Multi-Intra	Multi-Inter	None
MNIST	**99.6**	99.5	99.5	99.5 (48)	99.9 (50)	64.1 (0.2)	83.7 (1.1)
F-MNIST	**94.8**	94.3	94.5	94.4 (54)	97.8 (42)	55.5 (1.6)	64.6 (2.3)
STL-10	**98.6**	98.3	98.4	99.0 (62)	99.9 (31)	72.4 (0.5)	88.7 (6.7)
CIFAR-10	**97.5**	97.2	96.9	96.3 (42)	99.2 (54)	60.3 (0.7)	75.1 (3.4)
CIFAR-100	**85.7**	84.4	84.5	87.2 (41)	95.2 (41)	52.6 (2.2)	53.5 (16)
CIFAR-100 (Super)	**91.8**	91.6	91.5	93.7 (67)	97.1 (25)	56.8 (1.9)	54.1 (5.9)
Histopathology (domain shift)							
Camelyon17 Val	94.2	94.2	**94.4**	94.0 (61)	97.7 (35)	55.8 (0.6)	63.0 (2.8)
Camelyon17 Test	87.1	89.6	**92.5**	94.8 (55)	93.1 (34)	49.6 (0.2)	76.8 (11)
Non-vision (RNA-seq)							
TCGA-10	**99.1**	98.6	98.9	99.2 (75)	99.5 (18)	83.3 (0.3)	94.4 (6.7)

Predictive compatibility. Across vision and non-vision benchmarks, ATCG remains competitive with the pretrained backbone, with only minor changes in accuracy (e.g., CIFAR-100: $85.74 \rightarrow 84.51$; TCGA-10: $99.09 \rightarrow 98.88$). On Camelyon17-WILDS, Stage 2 yields a gain of 5.36 percentage points on the OOD test split ($87.12 \rightarrow 92.48$); we trace this gap to barcode-level patterns in Sec. 4.4.

Barcode types and prediction accuracy. Beyond aggregate accuracy, ATCG provides a structure-derived auditing signal via terminal-level *barcode types*. We categorize predictions into four barcode types reflecting distinct configurations of decision evidence: *Single* (one active basis trait), *Multi-Intra* (multiple active basis traits within one class, providing consistent evidence), *Multi-Inter* (active basis traits across classes, providing conflicting evidence), and *None* (no active basis traits, indicating insufficient evidence). As shown in Table 2, *Single* and *Multi-Intra* have higher test accuracy than *Multi-Inter* and *None* in the evaluated datasets. To assess whether barcode types provide information beyond scalar confidence, we further analyze five in-distribution vision benchmarks, adjusting for maximum softmax probability (MSP) and classifier seed (Appendix G.2). *Multi-Intra* has lower estimated error odds than *Single* in all five datasets. The contrast between *Single*/*Multi-Intra* and *Multi-Inter*/*None* is significant in three of the five datasets.

4.4 CAMELYON17-WILDS: FROM ACCURACY GAP TO BARCODE-LEVEL DIAGNOSIS

Camelyon17-WILDS comprises H&E histopathology patches from multiple hospitals with distinct staining protocols and scanners, forming a realistic domain-shift benchmark. We observed an accuracy gain on the OOD test split, from 87.12 to 92.48 between Base and Stage 2 ($+5.36\%$; Table 2). We treat this unexpected gain as a trigger for barcode-level inspection. Figure 3 provides two complementary views of trait usage across Train, Val, and Test, suggesting a barcode-level shift.

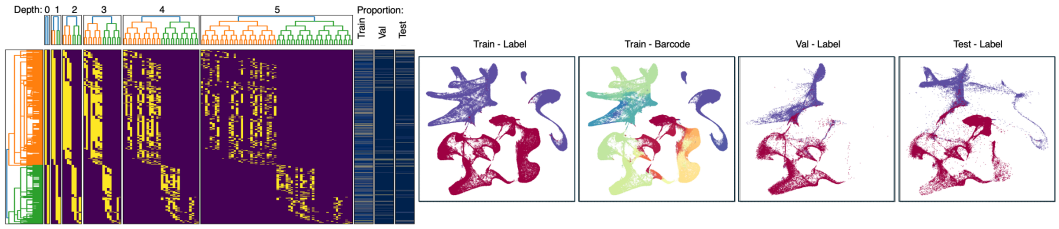


Figure 3: Trait usage on Camelyon17-WILDS. Top: integrated heatmap; rows are barcodes hierarchically clustered by barcode distance, columns follow the binary tree, side bars report Train/Val/Test prevalence. Bottom: UMAP of trait activations fitted on Train and projected to Val/Test.

Patient-share analysis links the gap to trait usage. In an illustrative run, we stratify basis traits by patient dominance into *Cross-Patient* (recurring across multiple patients) and *Single-Patient* (dominated by one patient); the asymmetry is visible in Figure 3 (formal definitions in Appendix E). Normal predictions are supported exclusively by Cross-Patient traits, while Tumor predictions combine shared foundations with nearly all Single-Patient traits. On the OOD test split, ground-truth tumor images have a mean Single-Patient usage of $\sim$21%, compared with $\sim$8% on Validation. Mean test accuracy across traits shows a larger ATCG–baseline gap for Single-Patient traits (98.2% vs. 94.1%) than for Cross-Patient traits (92.1% vs. 91.0%). The reported overall gain of 5.36 percentage points is specific to this setting. The central value of barcode auditing is the diagnostic capability itself: it relates accuracy differences, whether gains or losses, to specific trait activation patterns.

5 FURTHER APPLICATIONS

5.1 TRAIT BARCODE DIAGNOSTICS

ATCG's predicted-class trait barcodes enable monitoring for intra-class distribution shifts that may be missed by class-label frequencies. We formalize the monitoring procedure, validate it under controlled mode shifts on Stacked-MNIST, and apply it to benchmark generative models on CIFAR-10.

We monitor distribution shift through changes in the frequency of *predicted-class trait barcodes*: a loss in barcode frequency indicates *mode collapse*, while a gain indicates *mode invention*. Given a trained ATCG model, each sample yields a predicted-class barcode $\mathbf{b}_{\hat{y}} \in \{0,1\}^{2^D}$, and any dataset induces an empirical barcode distribution. We compare two such distributions, a reference and a target, with a *global* KL divergence and a *local* Z-test that flags individual barcodes as *Loss* or *Gain* anomalies under a Bonferroni-corrected cutoff over $C \cdot 2^{2^D}$ hypotheses (full protocol in Appendix F).

Controlled validation on Stacked-MNIST. We validate this claim on Stacked-MNIST (Sec. 4.2), where ground-truth generative factors are known. We construct six controlled scenarios from Missing and Novel manipulations at three magnitudes, $|\Delta| = 1, 2, 10$. In Missing scenarios, selected Red–Green factor combinations are removed from the target cohort; in Novel scenarios, selected combinations are removed from the reference cohort. Before examining global and local changes in discrete barcode distributions, we assess whether the manipulated mode shifts are detectable in three continuous representations: baseline embeddings, ATCG trait activations, and ATCG class logits using Gaussian-kernel maximum mean discrepancy (MMD), a standard kernel two-sample statistic Gretton et al. (2012). Table 3 reports permutation p-values for these continuous representations. MMD detects all six shifts in baseline embeddings and ATCG trait activations, while the ATCG class-logit test misses Novel-1 ($p = 0.126$). For discrete barcodes, directional KL summarizes global differences,

while Loss/Gain anomalies identify changes in individual barcode frequencies. Recall@3 is 1.0 in five scenarios and 0.6 for Novel-10.

Table 3: **Controlled shift diagnostics on Stacked-MNIST.** Gaussian-kernel MMD permutation p-values assess shifts in continuous representations. Predicted-class barcode diagnostics summarize global differences through directional KL and local frequency changes through Loss/Gain anomaly counts. Recall@3 measures the fraction of manipulated Red–Green combinations recovered.

	Continuous MMD p-values			**Barcode diagnostics**		
Scenario	**Baseline embedding**	**ATCG basis traits**	**ATCG logits**	**KL**	**Loss/Gain**	**Recall@3**
Missing-1	0.002	0.001	0.007	0.069078	2/0	1.0
Missing-2	0.001	0.001	0.001	0.211695	4/0	1.0
Missing-10	0.001	0.001	0.001	0.839331	22/0	1.0
Novel-1	0.012	0.001	0.126	0.035839	0/2	1.0
Novel-2	0.001	0.001	0.001	0.038031	0/5	1.0
Novel-10	0.001	0.001	0.001	0.045536	0/6	0.6

Benchmarking generative models. We apply the same diagnostics to benchmark generative models on CIFAR-10, using generated samples from EDM Karras et al. (2022) and the synthetic CIFAR-5M dataset Nakkiran et al. (2021). We evaluate class-conditional (EDM-Cond) and unconditional (EDM-Uncond) EDM samples alongside two independent CIFAR-5M draws (A and B), each comprising 50,000 samples. Table 4 shows lower FID and barcode KL for both EDM sample sets than for either CIFAR-5M draw. Building on this agreement, ATCG additionally identifies specific Loss and Gain barcodes (particular modes under- or over-represented relative to the original CIFAR-10). Barcode auditing thus diagnoses structural mismatches in generative models beyond what aggregate feature statistics (FID) capture, complementing standard generative benchmarks. Visualizations of barcode-frequency anomalies are in Appendix Fig. 12.

Table 4: **Benchmarking generative structural fidelity on CIFAR-10.** Comparison of FID and ATCG's barcode-level diagnostics. Both metrics favor EDM samples over CIFAR-5M draws; Loss/Gain counts (Bonferroni-corrected) further localize which barcodes drive each divergence.

Sample source	**FID ↓**	**KL ↓**	**Loss**	**Gain**
EDM (Cond / Uncond)	1.85 / 1.96	0.188 / 0.225	11 / 16	17 / 19
CIFAR-5M (A / B)	3.22 / 3.22	0.286 / 0.263	29 / 29	21 / 22

5.2 ATCG as a Backend for Concept Bottleneck Models

Beyond standard class prediction, ATCG can be used as a drop-in structural backend for Concept Bottleneck Models (CBMs) when human-interpretable intermediate concepts are desired. A standard CBM assigns one scalar predictor per concept (e.g., *wing*), collapsing heterogeneous visual cues into a single activation. In contrast, ATCG parameterizes each concept with multiple basis traits and aggregates them via Max-Dominance into a single concept score.

Predictive compatibility. We follow the VLG-CBM sequential setup Srivastava et al. (2024), directly reusing their released concept set and grounding-derived concept labels (VLM-proposed concepts localized by GroundingDINO). We replace the original Concept Bottleneck Layer (CBL) with an ATCG-installed CBL ($D{=}2$) that predicts the same concept labels from frozen image features, and fit the concept-to-class classifier on the resulting concept scores under VLG-CBM's recommended NEC$=5$ sparsity setting. Table 5 shows that VLG-CBM+ATCG matches the VLG-CBM baseline within small margins across datasets including ImageNet, indicating that the structured backend is predictive-compatible.

Qualitative evidence: multiple realizations per concept. Figure 4 illustrates how a single named concept can admit multiple realizations under ATCG. As an example, we focus on *red eyes*, a concept that the learned classifier identifies as shared across multiple bird classes. The left panel shows a

Table 5: **VLG-CBM accuracy with an ATCG backend.** Test accuracy (%) at NEC= 5. VLG-CBM+ATCG matches the baseline within small margins.

Method	CUB	CIFAR-10	CIFAR-100	ImageNet
VLG-CBM	75.54	90.00	67.91	73.58
VLG-CBM+ATCG	75.53	89.10	66.87	72.41

D=2 Pinball hierarchy in which basis traits are dominated by examples from different classes and progressively merge into higher-level Composite Traits via Max-Dominance. Because the ATCG-equipped CBL is trained only against concept supervision (class information enters only through the downstream sparse classifier), this hierarchy organizes *within-concept visual variants* that different classes tend to rely on, rather than a supervised partition by class. The right panel shows cross-class examples under the same concept, with grounding boxes confirming the shared visual cue.

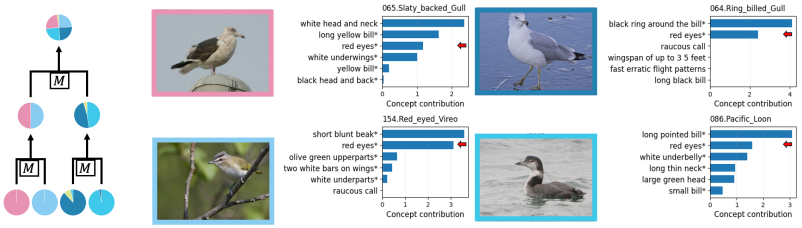


Figure 4: Multiple realizations of a shared concept (CUB). Left: a D=2 Pinball hierarchy for *red eyes* inside the CBL; basis traits merge via Max-Dominance into one concept score, with pie charts showing class composition per basis trait. Right: cross-class examples sharing the same concept, with top concept contributions per prediction.

6 Conclusion and Limitations

We presented ATCG, a framework that addresses intra-class heterogeneity by replacing flat classification heads with a per-class hierarchical Pinball head. By leveraging combinatorial trait barcodes and the Molding Bonus, ATCG exposes auditable decision evidence while maintaining competitive predictive performance. We further showed that ATCG can serve as a structured backend for concept bottleneck pipelines, supporting multiple realizations under a single concept.

Limitations and Future Work. ATCG has terminal output width $C2^D$ and $C(2^D - 1)$ sibling pairs subject to molding. Scalability is also constrained by positive support. Under balanced classes and an idealized balanced hierarchy, a mini-batch of size B provides approximately $B/(C2^d)$ positive samples to a parent context at depth d, while a training set of size N provides approximately $N/(C2^d)$ training examples for that context before path-exclusivity filtering; the realized support can be smaller in both cases. Empty-context masking prevents undefined molding terms but does not remove this information constraint. These considerations motivate choosing hierarchy depth according to class count, available support, and the within-class heterogeneity to be audited, rather than using a fixed depth across tasks. Practical directions for support-limited settings include strong-positive caching through memory banks and sampling strategies that increase the number of positive examples for selected classes within each update. Scaling deep hierarchies across very large class sets remains an open problem. Other limitations include the structural rather than inherently semantic nature of the learned traits, the Stage 2 tradeoff between stronger molding and preservation of the pretrained embedding geometry, and the still-limited empirical scope despite evaluations spanning vision, non-vision, and concept-backend settings. Trait barcodes summarize a model's evidence patterns rather than certify ground-truth correctness; findings on benchmarks such as Camelyon17-WILDS should therefore be interpreted as demonstrations of diagnostic capability rather than clinical claims.

AI USE STATEMENT

Generative AI tools were used to assist with literature retrieval and discovery, experimental planning and methodological discussion, development and checking of mathematical arguments, drafting and polishing manuscript text, and selected downstream data-processing, statistical-analysis, result-interpretation, and table-generation tasks. The core ATCG implementation and model-training code used to produce the reported models were written by the authors; AI assistance was used in selected downstream analysis and presentation workflows. All experiments were executed by the authors using their own training pipelines, and all reported numerical results, mathematical claims, statistical analyses, experimental conclusions, and bibliographic references were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

Implementation and experimental details, including training configurations, hyperparameters, dataset-specific settings, and statistical procedures, are provided in Appendices B–G. Assumptions and proofs for the theoretical results are provided in Appendix A.

REFERENCES

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223. PMLR, 2011.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

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

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Liò, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Mary J. Goldman, Brian Craft, Mim Hastie, Kristupas Repečka, Fran McDade, Akhil Kamath, Ayan Banerjee, Yunhai Luo, Dave Rogers, Angela N. Brooks, Jingchun Zhu, and David Haussler. Visualizing and interpreting cancer genomics data via the xena platform. *Nature Biotechnology*, 38 (6):675–678, 2020. doi: 10.1038/s41587-020-0546-8.

Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1319–1327. PMLR, 2013.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677. PMLR, 2018.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 2020.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664. PMLR, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. In *International Conference on Learning Representations*, 2021.

Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14933–14943, 2021.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153. PMLR, 2017.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations Workshop Track*, 2015.

Divyansh Srivastava, Ge Yan, and Tsui-Wei Weng. VLG-CBM: Training concept bottleneck models with vision-language guidance. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

The Cancer Genome Atlas Research Network, John N. Weinstein, Eric A. Collisson, Gordon B. Mills, Kenna R. Mills Shaw, Brad A. Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M. Stuart. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10): 1113–1120, 2013. doi: 10.1038/ng.2764.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. URL `https://www.vision.caltech.edu/datasets/cub_200_2011/`.

Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Suzanne Petryk, Sarah Adel Bargal, and Joseph E. Gonzalez. NBDT: Neural-backed decision tree. In *International Conference on Learning Representations*, 2021.

Qi Wang and Joseph JaJa. From Maxout to Channel-Out: Encoding information on sparse pathways. In *Artificial Neural Networks and Machine Learning – ICANN 2014*, volume 8681 of *Lecture Notes in Computer Science*, pp. 273–280. Springer, 2014. doi: 10.1007/978-3-319-11179-7_35.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2740–2748, 2015.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023.

A Theoretical Analysis of Metric Geometry

This section analyzes geometric properties of the proposed *Hierarchical Barcode Distance*. We formalize two complementary statements: (1) **Strict Depth Monotonicity**, ensuring that *coarser hierarchy-level disagreements* dominate finer-level differences under a worst-case dense-activation scenario; and (2) **Conditional Bi-Lipschitz Equivalence**, showing that when induced barcodes are sufficiently sparse, the distance is bi-Lipschitz equivalent (up to bounded distortion) to a level-based ultrametric defined by the configured hierarchy.

A.1 Definitions and Preliminaries

Let D be the fixed hierarchy depth, and let $\mathbf{b}_i^{(d)} \in \{0,1\}^{2^d}$ denote the induced binary activation pattern at depth d for sample i (obtained by thresholding basis-trait activations and propagating them through Max-Dominance). We define the hierarchical structural distance

$$\mathcal{D}(i,j) = \sum_{d=0}^{D} w_d \cdot \|\mathbf{b}_i^{(d)} - \mathbf{b}_j^{(d)}\|_1, \quad \text{where } w_d = 2^{-(Dd+1)}. \tag{7}$$

The weight schedule is chosen so that earlier (coarser) levels dominate later (finer) levels in a provable, worst-case sense. For distinct terminal barcodes, we define the *first divergence depth*

$$\delta(i,j) = \min\{d \mid \mathbf{b}_i^{(d)} \neq \mathbf{b}_j^{(d)}\}. \tag{8}$$

We define the reference level-based ultrametric by

$$\mathcal{D}^*(i,j) := \begin{cases} 0, & \mathbf{b}_i^{(D)} = \mathbf{b}_j^{(D)}, \\ w_{\delta(i,j)} = 2^{-(D\delta(i,j)+1)}, & \text{otherwise.} \end{cases} \tag{9}$$

Identical terminal barcodes also satisfy $\mathcal{D}(i,j) = 0$. Intuitively, $\mathcal{D}^*$ measures how high in the hierarchy two samples first disagree, ignoring lower-level variation.

A.2 Strict Depth Monotonicity

We first show that the distance ordering is strictly preserved even in the theoretical worst case where all deeper layers exhibit maximal disagreement (dense activation).

Theorem A.1 (Strict No-Avalanche Condition)**.** *Let $(\mathbf{b}_i, \mathbf{b}_j)$ be a pair of barcodes diverging at depth δ, and let $(\mathbf{b}_x, \mathbf{b}_y)$ be a pair diverging at depth $\delta+1$ with maximum possible Hamming distance in all subsequent layers. For all $\delta \in [0, D-1]$, strict monotonicity holds:*

$$\min \mathcal{D}(i,j) > \max \mathcal{D}(x,y). \tag{10}$$

Proof. We compare the minimum contribution induced by a first divergence at depth δ against the maximum possible accumulated contribution from all deeper layers starting from $\delta+1$.

1. Lower Bound (Contribution from the first divergence at δ): A divergence at depth δ implies distinct decision paths. Consequently, differences must exist at depth δ and persist or propagate to at least one active branch in subsequent layers.

$$V_{\text{early}}(\delta) \geq w_\delta + \sum_{k=\delta+1}^{D} w_k \cdot 1 > w_\delta. \tag{11}$$

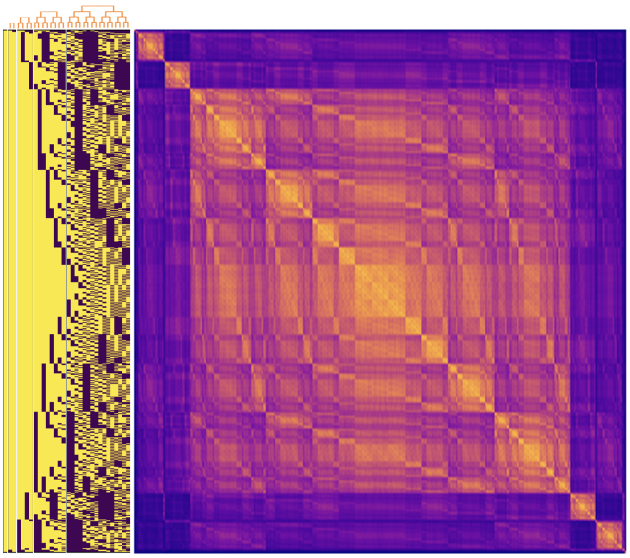


Figure 5: **Visualizing depth-dominated geometry and barcodes.** (Right) Pairwise Hierarchical Trait Distance $\mathcal{D}$ computed over *all* binary barcodes at depth $D{=}4$ (the space of possible patterns has size $2^{2^D} = 2^{16}$). The nested block structure illustrates a depth-dominated, ultrametric-like geometry: distances are governed primarily by the first divergence depth $\delta(i, j)$. (Left) The corresponding binary barcodes, ordered by hierarchical clustering under $\mathcal{D}$. The hierarchy shown at the top indicates the Pinball-defined tree topology used to organize the barcodes. Shallower traits (left columns) define coarse groupings, while deeper traits (right columns) refine them into finer substructures, yielding an inspectable decision topology.

(Even conservatively assuming only 1 bit difference per subsequent layer).

2. Upper Bound (Worst-case accumulated contribution from deeper layers): At depth k, at most 2^k coordinates may differ. Define the maximal contribution of layer k as:

$$T_k = w_k \cdot 2^k = 2^{-((D-1)k+1)}. \tag{12}$$

The total late contribution is the sum of these maximal terms:

$$V_{\text{late}}(\delta) \leq \sum_{k=\delta+1}^{D} T_k = T_{\delta+1} \sum_{j=0}^{D-\delta-1} r^j. \tag{13}$$

The sum is a geometric series with ratio $r = 2^{-(D-1)}$. Since the finite sum is strictly less than the infinite series sum (and $r \leq 1/2$ for $D \geq 2$):

$$V_{\text{late}}(\delta) < T_{\delta+1} \cdot \frac{1}{1-r} \leq 2T_{\delta+1} \tag{14}$$

3. Strict Dominance Analysis: We first compare the leading term w_δ to the leading deeper-layer term $T_{\delta+1}$:

$$\frac{w_\delta}{T_{\delta+1}} = 2^{D-\delta-1}. \tag{15}$$

- **Case** $\delta \leq D-2$**:** The ratio is ≥ 2, so $w_\delta \geq 2T_{\delta+1}$. Combining this with the bound above:
$$V_{\text{early}}(\delta) > w_\delta \geq 2T_{\delta+1} > V_{\text{late}}(\delta). \tag{16}$$
- **Case** $\delta = D-1$**:** The ratio is 1, so $w_{D-1} = T_D$. The sum contains exactly one term, so $V_{\text{late}} = T_D$. However, the early contribution must include the difference at depth D (since distinct paths imply distinct leaves):
$$V_{\text{early}}(\delta) \geq w_{D-1} + w_D > w_{D-1} = T_D = V_{\text{late}}(\delta). \tag{17}$$

Thus, strict monotonicity holds universally. □

A.3 Conditional Bi-Lipschitz Equivalence via Terminal Sparsity

Strict depth monotonicity ensures a well-defined hierarchy-level ordering, but it does not by itself bound metric distortion: if many bits are active at deep layers, late-layer terms can still accumulate nontrivially (even though they cannot overturn the depth ordering). We therefore state a conditional stability result under a sparsity assumption on the induced barcodes. The Max-Dominance aggregation rule allows us to formulate this assumption purely at the terminal depth.

Lemma A.2 (Sparsity monotonicity under Max-Dominance)**.** *Under Max-Dominance, a node at depth d is active if and only if at least one of its children at depth $d+1$ is active. Each active node at depth d therefore corresponds to at least one active child at depth $d+1$, which gives*

$$\left\|\mathbf{b}_i^{(d)}\right\|_1 \;\le\; \left\|\mathbf{b}_i^{(d+1)}\right\|_1 \quad \textit{for all } d = 0, \dots, D-1. \tag{18}$$

Consequently, a terminal-depth bound $\|\mathbf{b}_i^{(D)}\|_1 \le K$ implies $\|\mathbf{b}_i^{(k)}\|_1 \le K$ for all $k \le D$.

Theorem A.3 (Bi-Lipschitz Equivalence Under Terminal Sparsity)**.** *Suppose terminal-depth sparsity: there exists a constant K such that $\|\mathbf{b}_i^{(D)}\|_1 \le K$ for all samples i. Then, for any pair (i, j) with first divergence depth $\delta = \delta(i, j)$, the metric $\mathcal{D}$ is bi-Lipschitz equivalent to the ultrametric $\mathcal{D}^*$:*

$$\mathcal{D}^*(i,j) \;\le\; \mathcal{D}(i,j) \;\le\; \frac{2K}{1-2^{-D}} \cdot \mathcal{D}^*(i,j). \tag{19}$$

Proof. Let $\delta = \delta(i, j)$.

Lower Bound: Since the paths for i and j diverge at layer δ, they differ in at least one coordinate. Thus:

$$\mathcal{D}(i,j) \ge w_\delta \left\|\mathbf{b}_i^{(\delta)} - \mathbf{b}_j^{(\delta)}\right\|_1 \ge w_\delta \cdot 1 = \mathcal{D}^*(i,j). \tag{20}$$

Upper Bound: By Lemma A.2, terminal-depth sparsity propagates to all layers: $\|\mathbf{b}_i^{(k)}\|_1 \le K$ for all $k \le D$. Using the triangle inequality, the distance at any layer is bounded by:

$$\left\|\mathbf{b}_i^{(k)} - \mathbf{b}_j^{(k)}\right\|_1 \le \left\|\mathbf{b}_i^{(k)}\right\|_1 + \left\|\mathbf{b}_j^{(k)}\right\|_1 \le 2K. \tag{21}$$

The total distance is a weighted sum. We bound the finite sum by the infinite geometric series with first term w_δ and ratio 2^{-D}:

$$\mathcal{D}(i,j) \le 2K \sum_{k=\delta}^{D} w_k < 2K \sum_{m=0}^{\infty} w_\delta (2^{-D})^m = 2K \cdot \frac{w_\delta}{1-2^{-D}}. \tag{22}$$

Substituting $\mathcal{D}^*(i, j) = w_\delta$, we obtain the final bound:

$$\mathcal{D}(i,j) \le \frac{2K}{1-2^{-D}} \cdot \mathcal{D}^*(i,j). \tag{23}$$

□

Empirical indicator of sparsity. The distortion constant in Theorem A.3 depends only on the terminal-depth sparsity bound K. Appendix D reports the mean number of active traits $\bar{k}$ on CIFAR-100 (Superclass), serving as a direct empirical indicator that $\|\mathbf{b}^{(D)}\|_1$ remains small for typical samples.

Interpretation. Strict depth monotonicity (Theorem A.1) guarantees a worst-case ordering principle: pairs that diverge earlier in the hierarchy have strictly larger distance under $\mathcal{D}$ than any pair whose first divergence occurs deeper, even if deeper-layer differences are maximized. Under the additional sparsity regime of Theorem A.3, $\mathcal{D}$ further admits bounded distortion relative to the level-based ultrametric $\mathcal{D}^*$. Together, these results provide a metric-geometric interpretation of retrieval and clustering: hierarchy-level divergence controls the ranking, and sparsity controls the within-level geometric stability.

B Implementation Details

We provide detailed specifications for reproducing ATCG, including the Pinball head architecture, geometric molding grid definitions, and optimization hyperparameters. Table 6 summarizes the Pinball configurations used in all reported experiments. Code will be released in the camera-ready version.

Table 6: **Pinball head configurations across reported experiments.** The structure balances depth D against class count C, ensuring optimization stability under a batch size of 256.

Benchmark	Backbone	d_{in}	d_{mlp}	C	D	Basis Traits
Stacked-MNIST (Table 1)						
Stacked-MNIST	ResNet-18	512	2048	10	4	160
Standard Benchmarks (Table 2)						
MNIST	ResNet-18	512	2048	10	4	160
Fashion-MNIST	ResNet-18	512	2048	10	4	160
CIFAR-10	ResNet-50	2048	2048	10	4	160
STL-10	ResNet-50	2048	2048	10	3	80
CIFAR-100 (Fine)	ResNet-50	2048	2048	100	2	400
TCGA-10	MLP	128	128	10	2	40
Hierarchical Analysis (Table 11)						
CIFAR-100 (Super)	ResNet-50	2048	2048	20	3	160
Histopathology (domain shift) (Sec 4.4)						
Camelyon17-WILDS	ResNet-50	2048	2048	2	5	64
Architecture Validation						
CIFAR-10 (ViT)	ViT-Base	768	2048	10	4	160
CIFAR-100 (Super; ViT)	ViT-Base	768	2048	20	3	160

TCGA-10 dataset and backbone. We obtained TCGA RNA-seq expression data from the UCSC Xena platform Goldman et al. (2020), using TCGA-specific (per-cohort) normalization rather than Pan-Cancer normalization. We selected the 10 cancer types with the largest sample counts to ensure stable per-class statistics: breast invasive carcinoma (BRCA), kidney renal clear cell carcinoma (KIRC), head and neck squamous cell carcinoma (HNSC), lung adenocarcinoma (LUAD), thyroid carcinoma (THCA), lung squamous cell carcinoma (LUSC), prostate adenocarcinoma (PRAD), stomach adenocarcinoma (STAD), bladder urothelial carcinoma (BLCA), and liver hepatocellular carcinoma (LIHC). We randomly partitioned the dataset into 80% training and 20% testing sets. The input is a 20,531-dimensional gene expression vector; we use an MLP encoder ($20531 \to 512 \to 128$) as the backbone, with the Pinball head depth $D = 2$ given the smaller per-class sample sizes.

B.1 Pinball Head Architecture

The Pinball head replaces the final linear classifier of the backbone. As defined in Section 3.1, it implements a mapping $g_\phi : \mathbb{R}^{d_{\text{in}}} \to [0, \alpha]^{C \cdot 2^D}$ that projects the backbone embedding $z = f_\theta(x)$ directly to the terminal-level (depth-D) trait activations across the C class-conditional hierarchies, i.e., $\mathbf{t}^{(D)} = g_\phi(z)$.

Implementation of g_ϕ. We parameterize g_ϕ as a two-layer MLP with a scaled sigmoid output:

$$g_\phi(z) = \alpha \cdot \sigma(\mathbf{W}_2 \, \mathrm{ReLU}(\mathbf{W}_1 z + \mathbf{b}_1) + \mathbf{b}_2) \,, \tag{24}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{mlp}} \times d_{\text{in}}}$, $\mathbf{W}_2 \in \mathbb{R}^{(C \cdot 2^D) \times d_{\text{mlp}}}$, and $\mathbf{b}_1, \mathbf{b}_2$ are biases.

- **Hidden layer:** maps $z \in \mathbb{R}^{d_{\text{in}}}$ to a hidden representation in $\mathbb{R}^{d_{\text{mlp}}}$.
- **Output layer:** produces $C \cdot 2^D$ basis trait activations (the depth-D bottom-layer nodes of all C hierarchies).
- **Scaled activation:** we use $\alpha \cdot \sigma(\cdot)$ with $\alpha = 10$, yielding $\mathbf{t}^{(D)} \in [0, \alpha]^{C \cdot 2^D}$.

Given $\mathbf{t}^{(D)}$, all internal activations $\mathbf{t}^{(d)}$ ($d < D$) are computed deterministically via the Max-Dominance rule (Section 3.1).

Configuration strategy and constraints. Choosing (C, D) trades off representational capacity against optimization stability under a fixed mini-batch size (256). In particular, Molding Bonus estimation requires sufficient positive density per molded sibling pair. Since the expected number of positives per class per mini-batch is $\approx 256/C$, deeper hierarchies can quickly lead to sparse or unstable branch-level contexts for large C. We therefore adopt the following configurations (Table 6):

- **MNIST, Fashion-MNIST, and CIFAR-10 ($C = 10$):** $D = 4$ (total basis traits $= C \cdot 2^D = 160$) provides granularity without overly diluting positive density.
- **Large class count ($C = 100$):** for CIFAR-100 (Fine), we use $D = 2$ (400 basis traits) to keep each class hierarchy shallow and learnable given limited positives per class per mini-batch.
- **Small class count ($C = 2$):** for Camelyon17-WILDS, we use $D = 5$ (64 basis traits) to increase expressivity while maintaining ample positive density.

B.2 Geometric Molding Template Definition

The Molding Bonus minimizes the divergence between sibling-trait activations and fixed geometric templates. Before specifying the grid coordinates, we clarify an indexing convention that avoids ambiguity between (i) the depth used to define contexts and (ii) the depth of the sibling pair being molded.

Indexing convention. We use d to index the depth of a *parent* node that defines the context sets, and we use $\ell = d+1$ to index the depth of the corresponding *sibling pair* being molded. Equivalently, contexts $\mathcal{X}_{c,j}^{(d)\pm}$ are defined at depth d, while molding grids $\mathcal{G}^{(\ell)\pm}$ and the layer-wise bonus act on sibling pairs at depth ℓ.

Structural constants. The template geometry is anchored by two hyperparameters:

- **Scaling factor ($\alpha = 10$)[3]:** trait activations lie in $[0, \alpha]$. We use grid coordinates within $[1, \alpha - 1]$.
- **Decision threshold ($\tau = 5.5$):** defines the logical boundary: $t \geq \tau$ is "active," and $t < \tau$ is "inactive."

All grid values below are chosen to position sibling basis-traits relative to this boundary.

Discrete value sets. We construct the grids using three sets of coordinates:

- **High range (active):** target values for an explicitly activated sibling basis-trait,
$$V_{\text{high}} = \{8, 9\} \quad (\gg \tau).$$
- **Low range (suppressed):** target values for strong joint suppression,
$$V_{\text{low}} = \{1, 2\} \quad (\ll \tau).$$
- **Mid range (depth-adaptive, but sub-threshold):** when one sibling is high, the other is encouraged to remain *sub-threshold* while allowing depth-dependent geometric relaxation. For sibling depth $\ell \in \{1, \ldots, D\}$, define
$$u^{(\ell)} = 2 + 2 \cdot \frac{\ell - 1}{D - 1}, \qquad V_{\text{mid}}^{(\ell)} = \{u^{(\ell)} - 1,\ u^{(\ell)}\}.$$

[3] A unit-scale parameterization ($\alpha = 1$) is equivalent under coordinated rescaling of the activation cutoff, geometric-template locations, KDE bandwidth, and logit scale; α is a numerical parameterization and does not increase representational capacity.

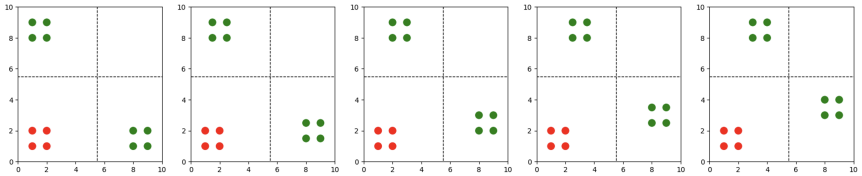


Figure 6: **Geometric molding template grids $\mathcal{G}^{(\ell)\pm}$ at sibling-pair depths $\ell = 1, \ldots, D$.** Each panel shows the joint activation space of a sibling pair $(t_1, t_2) \in [0, \alpha]^2$. Dashed lines at $\tau = 5.5$ mark the active/inactive boundary used for context assignment. **Inactive template** (p^-, red dots) concentrates near the origin $(0, 0)$ to encourage joint suppression on negative-context samples. **Active template** (p^+, green dots) concentrates near the complementary corners $(\alpha, 0)$ and $(0, \alpha)$ to encourage exactly one sibling to activate on positive-context samples. Across depths, the active-template mid-range relaxes toward τ at deeper levels, allowing sibling traits to remain geometrically close yet logically distinct (Appendix B.2).

Template construction. Using these sets, the molding grids are defined as follows.

- **Inactive grid ($\mathcal{G}^{(\ell)-}$):** universal joint suppression. For negative context, both sibling basis-traits are encouraged to be low:

$$\mathcal{G}^{(\ell)-} = \{(t_1, t_2) \mid t_1 \in V_{\text{low}},\ t_2 \in V_{\text{low}}\}. \tag{25}$$

- **Active grid ($\mathcal{G}^{(\ell)+}$):** depth-adaptive complementarity. For positive context, exactly one sibling basis-trait is encouraged to be high while the other remains sub-threshold:

$$\mathcal{G}^{(\ell)+} = \underbrace{\{(t_1, t_2) \mid t_1 \in V_{\text{high}},\ t_2 \in V_{\text{mid}}^{(\ell)}\}}_{\text{Sibling 1 high}} \cup \underbrace{\{(t_1, t_2) \mid t_1 \in V_{\text{mid}}^{(\ell)},\ t_2 \in V_{\text{high}}\}}_{\text{Sibling 2 high}}. \tag{26}$$

Design note. Molding is defined on sibling pairs, hence the first applicable sibling depth is $\ell = 1$ (with parent depth $d = 0$). At this first sibling level ($\ell = 1$), $V_{\text{mid}}^{(1)} = \{1, 2\} = V_{\text{low}}$, enforcing strict orthogonality between sibling basis-traits. At deeper levels ($\ell \to D$), $V_{\text{mid}}^{(\ell)}$ relaxes upward (e.g., $\{3, 4\}$) while still satisfying $\max(V_{\text{mid}}^{(\ell)}) < \tau$. This enables sub-cutoff similarity: deeper-level sibling basis-traits may be geometrically closer while remaining logically distinct under the thresholding rule.

B.3 Optimization and Hyperparameters

We adopt a retrofit training protocol that equips a standard feature extractor with transparent, trait-based decision logic. We first train the backbone (initialized with ImageNet-1k V2 weights) on the target dataset to establish a strong baseline. We then apply the two-stage fine-tuning strategy to retrofit the Pinball head and incorporate the structural and consistency regularizers.

Training schedule. To account for dataset scale, we adjust the number of epochs to reach stable convergence without unnecessary computational overhead:

- **Standard benchmarks (CIFAR, STL):** baseline training for 100 epochs; retrofit Stage 1 for 20 epochs; retrofit Stage 2 for 20 epochs.
- **Large-scale benchmark (Camelyon17-WILDS):** baseline training for 30 epochs; retrofit Stage 1 for 5 epochs; retrofit Stage 2 for 5 epochs, reflecting the substantially larger sample size ($>$ 300k images).

Data augmentation for consistency. The trait consistency loss $\mathcal{L}_{con}$ encourages agreement between the *basis-trait activations* induced by two views (x, x') of the same image.

- **Reference view (x):** standard preprocessing (Resize + Normalize).

- **Augmented view (x'):** AutoAugment (CIFAR/STL/Camelyon). For fair comparison, the baseline backbones are trained using the same augmentation policy.

Table 7: **Training configurations.** Hyperparameters for baseline pre-training and the ATCG retrofit phases. Camelyon17-WILDS uses fewer epochs due to its large scale.

Configuration	Value
Phase 0: Baseline pre-training	
Initialization	ImageNet-1k (V2)
Epochs (Std / Camelyon)	**100 / 30**
Batch size	256
Learning rate	0.01
Phase 1: ATCG retrofit (Stage 1)	
Backbone status	**Frozen**
Epochs (Std / Camelyon)	**20 / 5**
LR (Head η_ϕ)	0.01
LR (Backbone η_θ)	0.0
Phase 2: ATCG retrofit (Stage 2)	
Backbone status	**Unfrozen**
Epochs (Std / Camelyon)	**20 / 5**
LR (Head η_ϕ)	0.01
LR (Backbone η_θ)	0.001
Optimization strategy	
Optimizer	SGD (Momentum=0.9, Weight Decay=10^{-4})
LR Scheduler	Cosine annealing (per stage)
Loss weights ($\lambda_{CE}, \lambda_{pin}, \lambda_{con}$)	1.0, 1.0, 1.0
Pinball parameters	
Scaling factor (α)	10.0
Cutoff threshold (τ)	5.5
Decay factor (γ)	0.4
Gaussian kernel bandwidth (h)	1.5

Hyperparameter choices.

- **Loss balancing.** We optimize $\mathcal{L}_{total} = \lambda_{CE}\mathcal{L}_{CE} + \lambda_{pin}\mathcal{L}_{pin} + \lambda_{con}\mathcal{L}_{con}$ with $\lambda_{CE} = \lambda_{pin} = \lambda_{con} = 1.0$ in all experiments.
- **Activation scale and cutoff.** We set $\alpha = 10$ and use $\tau = 5.5$ to define an active/inactive split for binarizing basis-trait activations.
- **Hierarchical decay ($\gamma = 0.4$).** The aggregate weight per depth scales as $(2\gamma)^d$. The choice $\gamma = 1/2$ corresponds to equal per-depth weight; we use $\gamma = 0.4 < 1/2$ so that shallower layers receive proportionally more weight, emphasizing coarse structure before deeper-level specialization.
- **Optimization.** We use SGD with momentum 0.9 and weight decay 10^{-4}, and apply cosine annealing within each training stage.

Computational efficiency. ATCG is a lightweight retrofit module that does not require training from scratch. On CIFAR-10 with a ResNet-50 backbone, the Pinball head adds $\sim$16% trainable parameters (4.5M / 28M). The full two-stage training completes in $\sim$2.5 hours on a single NVIDIA Quadro RTX 8000 (48 GB; 48 mins Stage 1, 99 mins Stage 2). We also confirm architectural generality by applying the same Pinball head to a Vision Transformer backbone (ViT-B/16; `vit_base_patch16_224`) Dosovitskiy et al. (2021) without modifying the backbone.

C Molding Mechanism

Templates and differentiable density matching. Consider a sibling pair of trait activations $\mathbf{s}(x) = (t_1(x), t_2(x)) \in [0, \alpha]^2$. For each class c and depth d, we define the context sets used to mold sibling

pairs. At the top level ($d = 0$), the context is determined directly by labels: positives are samples with $y = c$ and negatives are samples with $y \neq c$ (Section 3.2). For deeper levels ($d > 0$), the context for a sibling pair is defined *path-conditionally* using the parent/sibling state as in Eq. (3). We encode the desired sibling behaviors on positives versus negatives using two fixed target templates on $[0, \alpha]^2$:

- **Active template p^+:** a bimodal distribution concentrating mass near the complementary corners $(\alpha, 0)$ and $(0, \alpha)$.
- **Inactive template p^-:** a unimodal distribution concentrating mass near the origin $(0, 0)$.

Given a context set $\mathcal{X}$, we estimate the empirical density of sibling activations by Gaussian KDE:

$$\hat{p}_\Theta(\mathbf{u}; \mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \kappa_h\big(\mathbf{u} - \mathbf{s}(x)\big), \qquad \kappa_h(\mathbf{v}) = \frac{1}{2\pi h^2} \exp\Big(- \frac{\|\mathbf{v}\|_2^2}{2h^2} \Big), \tag{27}$$

where h is the bandwidth and $\Theta := (\theta, \phi)$ collects all trainable parameters that affect $\mathbf{s}(x)$. In a mini-batch, the context set $\mathcal{X}_{c,j}^{(d)\pm}$ may be empty, e.g., at deeper layers where the path-conditional criteria of Eq. (3) filter out all samples in the batch. In such cases $\hat{p}_\Theta(\cdot; \mathcal{X})$ is undefined, so we mask the corresponding local bonus term. Positive and negative bonuses are normalized separately by the sum of depth weights over their respective nonempty node contexts. Eqs. (29)–(30) describe the case in which all contexts are nonempty.

Bonus as template-to-empirical KL alignment. We define the local molding objective by encouraging $\hat{p}_\Theta(\cdot; \mathcal{X}^+)$ and $\hat{p}_\Theta(\cdot; \mathcal{X}^-)$ to match p^+ and p^-, respectively. Concretely, we minimize the *template-to-empirical* KL divergence $\mathrm{KL}(p^\pm \,\|\, \hat{p}_\Theta(\cdot; \mathcal{X}^\pm))$. Since $p^\pm$ are fixed, minimizing this KL is equivalent (up to an additive constant) to maximizing $\mathbb{E}_{\mathbf{g} \sim p^\pm}[\log \hat{p}_\Theta(\mathbf{g}; \mathcal{X}^\pm)]$.

We approximate this expectation by evaluating $\log \hat{p}_\Theta$ on a fixed grid $\mathcal{G}^\pm$ that is chosen to concentrate on the high-mass regions of the corresponding template:

$$\widehat{\mathrm{MB}}(\mathcal{X}^\pm; \mathcal{G}^\pm) = \frac{1}{|\mathcal{G}^\pm|} \sum_{\mathbf{g} \in \mathcal{G}^\pm} \log \hat{p}_\Theta(\mathbf{g}; \mathcal{X}^\pm). \tag{28}$$

For nonempty contexts, log densities are evaluated using `logsumexp` for numerical stability. The implementation uses $\epsilon = 10^{-15}$ to stabilize context-count and template-weight normalization.

Thus, $\widehat{\mathrm{MB}}(\cdot)$ is a *bonus* that increases when the empirical activation density assigns higher probability mass to the template regions.

In practice (Figure 6), we instantiate sibling-depth-specific grids $\mathcal{G}^{(d+1)+}$ and $\mathcal{G}^{(d+1)-}$ to stabilize optimization, while retaining the same objective across depths.

Hierarchical aggregation. We aggregate local bonuses across classes and hierarchy nodes. For each depth $d \in \{0, \ldots, D-1\}$, we average over all classes $c \in \{1, \ldots, C\}$ and all parent nodes $j \in \{1, \ldots, 2^d\}$:

$$\widehat{\mathrm{MB}}^{(d)} = \frac{1}{C \cdot 2^d} \sum_{c=1}^{C} \sum_{j=1}^{2^d} \sum_{s \in \{+,-\}} \widehat{\mathrm{MB}}\Big(\mathcal{X}_{c,j}^{(d)s}; \mathcal{G}^{(d+1)s} \Big). \tag{29}$$

Finally, the total molding bonus is the depth-weighted average:

$$\widehat{\mathrm{MB}}_{\mathrm{total}} = \frac{1}{\sum_{k=0}^{D-1} (2\gamma)^k} \sum_{d=0}^{D-1} (2\gamma)^d \, \widehat{\mathrm{MB}}^{(d)}, \tag{30}$$

where $\gamma \in (0, 1/2)$ ensures deeper layers are downweighted (i.e., $2\gamma < 1$).

D Ablation Studies

This appendix isolates the contribution of each component in ATCG and examines sensitivity to architectural and optimization hyperparameters. We use CIFAR-100 and its superclass taxonomy as a controlled testbed for quantifying structural quality and the generalization of interpretability under a fixed hierarchy.

Table 8: **Component-wise ablation on Camelyon17-WILDS.** The Pinball head provides a structured decision pathway. The structural term $\mathcal{L}_{\text{pin}}$ (Molding Bonus regularizer) reduces the proportion of unused terminal-depth basis traits. The consistency term $\mathcal{L}_{\text{con}}$ encourages consistent basis-trait activations across augmented views, improving utilization and OOD accuracy.

Config	Pinball Head	$\mathcal{L}_{\text{pin}}$	$\mathcal{L}_{\text{con}}$	Dead Basis Traits	Val (OOD)	Test (OOD)
Baseline	✗	✗	✗	N/A	94.2	87.1
Pinball Only	✓	✗	✗	90.1%	94.3	87.7
Pinball $+\mathcal{L}_{\text{pin}}$	✓	✓	✗	25.5%	94.4	89.2
Pinball $+\mathcal{L}_{\text{con}}$	✓	✗	✓	44.3%	94.5	88.5
Pinball $+\mathcal{L}_{\text{pin}} + \mathcal{L}_{\text{con}}$	✓	✓	✓	5.7%	94.4	92.5

Benchmarks. We use two complementary benchmarks:

- **Camelyon17-WILDS (Appendix D.1):** component-wise analysis under realistic domain shift. We report OOD validation and test accuracy, and we report **Dead Basis Traits** on the clean training split to diagnose basis-trait utilization.
- **CIFAR-100 (Super) (Appendix D.2 and Appendix D.3):** structural sensitivity and structural-property analysis. All metrics are reported on the test set to confirm that the learned interpretability generalizes to unseen data.

Metrics. Unless stated otherwise, Jaccard metrics are computed on trait barcodes at depth D.

- **Accuracy:** top-1 classification accuracy.
- **Sibling Jaccard (Sib-J):** overlap between *sibling* basis-traits, computed as the mean Intersection-over-Union (IoU) between their binary activation vectors over images. Moderate overlap can reflect shared predictive factors, while near-unity overlap indicates redundancy.
- **Cousin Jaccard (Cous-J):** mean IoU between subclasses within the *same* superclass. This measures local separation inside a superclass.
- **Inter-Superclass Jaccard (Inter-J):** mean IoU between classes from *different* superclasses. This should be near zero.
- **Mean active traits ($\bar{k}$):** the average number of active basis traits per image. This is a sparsity indicator for the induced barcodes.
- **Dead Basis Traits:** the fraction of terminal-depth basis traits that are never activated on the specified split. This measures utilization of the available trait capacity.

D.1 Component-wise Analysis (Camelyon17-WILDS)

We evaluate the necessity of the proposed components on Camelyon17-WILDS. Table 8 shows complementary roles across accuracy and trait utilization.

- **Structural regularization ($\mathcal{L}_{\text{pin}}$) promotes non-degenerate trait utilization.** With the Pinball head alone, the Dead Basis Traits is 90.1%. Adding $\mathcal{L}_{\text{pin}}$ reduces it to 25.5%, while OOD Test accuracy increases from 87.7% to 89.2%.
- **Consistency and structural molding provide complementary effects.** Adding $\mathcal{L}_{\text{con}}$ alone reduces the Dead Basis Traits to 44.3% and gives 88.5% OOD Test accuracy. Combining $\mathcal{L}_{\text{pin}}$ and $\mathcal{L}_{\text{con}}$ further reduces the Dead Basis Traits to 5.7% and reaches 92.5% OOD Test accuracy. Thus, neither regularizer alone reproduces the full combination.

Hierarchy-depth robustness. We used $D = 5$ for Camelyon17 because the task is binary and the class-specific hierarchy can allocate additional capacity to within-class variation while retaining substantial per-class support. To assess sensitivity to this choice, we repeated the same three-seed procedure at $D = 3$ and $D = 4$ using the same backbone checkpoints, data splits, seed schedule, and Stage 1/Stage 2 training protocol. OOD Test accuracy was $93.20 \pm 0.16\%$, $92.54 \pm 0.37\%$, and $92.48 \pm 1.28\%$ for $D = 3, 4, 5$, respectively, compared with $87.12 \pm 1.66\%$ for the backbone.

The barcode-type accuracy pattern was qualitatively similar across depths: Single and Multi-Intra predictions remained substantially more accurate than Multi-Inter and None predictions, although the exact barcode proportions changed with capacity.

Table 9: **Camelyon17 depth robustness.** Values are three-seed means ± population SDs. Overall accuracy is evaluated on the OOD Validation and Test splits; barcode-type accuracy is evaluated on the OOD Test split.

		Overall accuracy (%)		Barcode-type accuracy (%)			
Depth	**Traits/class**	**Val**	**Test**	**Single**	**Multi-Intra**	**Multi-Inter**	**None**
Backbone	–	94.17 ± 0.28	87.12 ± 1.66	–	–	–	–
$D = 3$	8	94.16 ± 0.47	93.20 ± 0.16	95.34 ± 0.36	96.71 ± 1.48	18.27 ± 18.27	83.58 ± 1.26
$D = 4$	16	94.34 ± 0.37	92.54 ± 0.37	94.60 ± 0.83	94.12 ± 2.52	34.42 ± 11.85	84.07 ± 2.02
$D = 5$	32	94.38 ± 0.28	92.48 ± 1.28	94.76 ± 0.85	93.08 ± 5.75	49.62 ± 3.99	76.83 ± 7.18

D.2 Structural Sensitivity Analysis: CIFAR-100 (Super)

CIFAR-100 provides 100 fine classes grouped into 20 superclasses: CIFAR-100 (Super). We use the superclass partition to quantify whether ATCG learns a meaningful coarse-to-fine structure. All results in Table 10 are reported on the CIFAR-100 test set.

Impact of depth (D). Accuracy is broadly similar across depths (Acc ≈ 91.4–91.5), so accuracy alone is not a sufficient selection criterion. Structural metrics reveal a clear trade-off:

- $D = 2$ **(capacity bottleneck):** Sib-J is very low (0.046), indicating near-orthogonality across sibling basis-traits. This can over-separate evidence and limit representation of shared sub-concepts.
- $D = 4$ **(redundancy):** Sib-J increases substantially (0.447) without an accuracy gain, indicating redundant sibling basis-traits that activate on similar subsets.
- $D = 3$ **(balanced):** $D = 3$ avoids the rigid orthogonality of $D = 2$ and the redundancy of $D = 4$. It yields comparable accuracy with more balanced overlap statistics.

Regularization weights. Varying λ_{pin} shows a consistent trade-off between structure and classification. Small λ_{pin} increases overlap and activity ($\bar{k}$), while large λ_{pin} can over-constrain the model and reduce accuracy. In contrast, results are less sensitive to λ_{con} in this regime, with relatively stable accuracy and overlap metrics.

Structural constants. Changing γ and τ produces small differences in both accuracy and structural metrics, indicating that the method is not finely tuned to a narrow constant choice.

Connection to bounded distortion. The conditional bi-Lipschitz result in Appendix A depends on sparsity of the induced barcodes. Here, the mean number of active terminal-depth basis traits $\bar{k}$ serves as an empirical proxy for barcode sparsity, and remains low and stable under the default setting (Table 10), consistent with the sparsity regime assumed by the theory.

D.3 Structural Properties of Learned Traits: CIFAR-100 (Super)

We further validate that ATCG recovers a meaningful coarse-to-fine organization using the CIFAR-100 superclass grouping.

Depth-wise overlap. For images x_i and x_j at depth d, let $\mathbf{b}_i^{(d)}, \mathbf{b}_j^{(d)} \in \{0,1\}^{2^d}$ be their binarized barcodes. We compute

$$\mathrm{Jaccard}(i, j; d) = \frac{\|\mathbf{b}_i^{(d)} \wedge \mathbf{b}_j^{(d)}\|_1}{\|\mathbf{b}_i^{(d)} \vee \mathbf{b}_j^{(d)}\|_1},$$

Table 10: **Comprehensive structural sensitivity analysis on CIFAR-100 test set.** We report accuracy and structural metrics across tree depth (D), regularization weights (λ_{pin}, λ_{con}), and structural constants (γ, τ). **Sib-J:** sibling Jaccard overlap. **Cous-J:** within-superclass overlap. **Inter-J:** cross-superclass overlap. **Mean $\bar{k}$:** mean number of active basis traits per image. Default settings are highlighted in **bold**.

Configuration	Acc (%)	Sib-J	Cous-J	Inter-J	Mean $\bar{k}$
1. Depth Analysis *(Capacity matching)*					
$D = 2$ (4 traits per class)	91.5	0.046	0.002	0.000	1.02
$\mathbf{D = 3}$ **(8 traits per class)**	**91.4**	**0.227**	**0.007**	**0.000**	**1.23**
$D = 4$ (16 traits per class)	91.4	0.447	0.012	0.000	1.62
2. Structure Weight *(λ_{pin})*					
$\lambda_{pin} = 0.1$	91.6	0.376	0.019	0.000	1.43
$\lambda_{pin} = 0.5$	91.4	0.250	0.010	0.000	1.27
$\lambda_{\mathbf{pin}} = \mathbf{1.0}$	**91.4**	**0.227**	**0.007**	**0.000**	**1.23**
$\lambda_{pin} = 2.0$	91.1	0.191	0.005	0.000	1.18
$\lambda_{pin} = 5.0$	90.9	0.156	0.003	0.000	1.20
3. Consistency Weight *(λ_{con})*					
$\lambda_{con} = 0.1$	91.7	0.216	0.007	0.000	1.23
$\lambda_{con} = 0.5$	91.5	0.228	0.007	0.000	1.24
$\lambda_{\mathbf{con}} = \mathbf{1.0}$	**91.4**	**0.227**	**0.007**	**0.000**	**1.23**
$\lambda_{con} = 2.0$	91.2	0.227	0.007	0.000	1.21
$\lambda_{con} = 5.0$	91.0	0.232	0.006	0.000	1.19
4. Structural Constants *(γ)*					
$\gamma = 0.3$	91.1	0.257	0.006	0.000	1.25
$\gamma = \mathbf{0.4}$	**91.4**	**0.227**	**0.007**	**0.000**	**1.23**
$\gamma = 0.5$	91.5	0.202	0.008	0.000	1.21
5. Cutoff Threshold *(τ)*					
$\tau = 4.5$	91.7	0.214	0.016	0.000	1.27
$\tau = \mathbf{5.5}$	**91.4**	**0.227**	**0.007**	**0.000**	**1.23**
$\tau = 6.5$	91.3	0.220	0.007	0.000	1.20

and average it over two groups: (1) *within-superclass* pairs from different fine classes under the same superclass, and (2) *across-superclass* pairs from different superclasses. A well-formed hierarchy should show higher overlap at shallow depths for within-superclass pairs, followed by decreasing overlap at deeper depths due to specialization. Across-superclass overlap should remain near zero at all depths.

Table 11: **Intra-class disentanglement on CIFAR-100 (Super).** Image-level Jaccard overlap between trait activations at each hierarchy depth d. **With MB** uses the full ATCG regularization; **No MB** ablates the Molding Bonus. **Intra** averages over pairs of images from different fine classes within the same superclass; **Inter** averages over pairs from different superclasses. With MB, Intra overlap drops sharply with depth ($0.994 \rightarrow 0.056$), reflecting hierarchical decomposition. Without MB, overlap remains ≈ 1 at all depths, validating the necessity of $\mathcal{L}_{pin}$.

	With MB		No MB	
d	**Intra**	**Inter**	**Intra**	**Inter**
0	0.9940	0.0001	0.9981	0.0001
1	0.4325	0.0000	0.9838	0.0001
2	0.1554	0.0000	0.9759	0.0001
3	0.0558	0.0000	0.9759	0.0001

Quantitative and qualitative results. Table 11 reports overlap statistics across depths. Figure 7 provides qualitative examples of fine classes within a superclass. These results show that shallow basis-traits capture shared superclass cues, while deeper basis-traits progressively specialize.

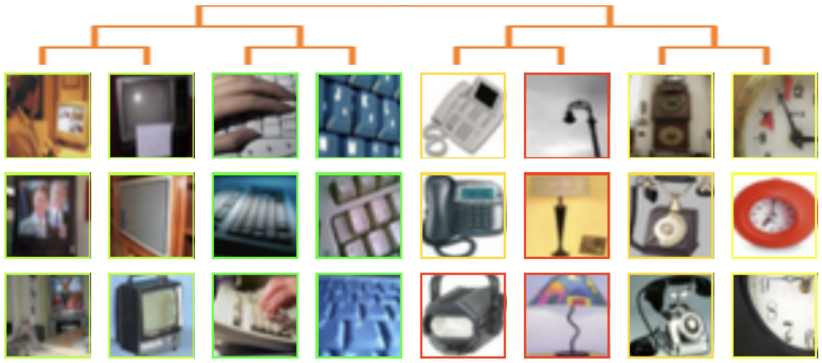
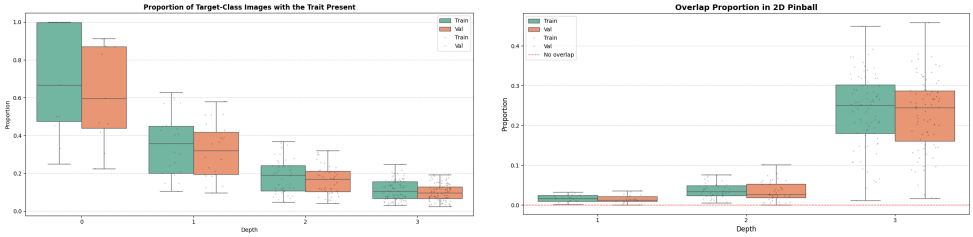


Figure 7: **Hierarchical organization on CIFAR-100.** Representative samples from different fine classes (e.g., Keyboard, Lamp) within a single superclass, grouped by their trait barcodes. Colored borders mark activation of specific basis traits along the hierarchy: shallow traits capture shared superclass cues while deeper traits progressively specialize toward fine classes.

Generalization and sibling overlap diagnostics. We further examine whether trait behavior generalizes beyond the training set and whether sibling basis-traits are redundant. Figure 8(a) compares activation-rate distributions between train and validation splits across depths. Figure 8(b) reports sibling overlap statistics (IoU) across depths.



(a) Trait activation rate **(b) Sibling overlap (IoU)**

Figure 8: **Structural diagnostics on CIFAR-100. (a)** Train and validation activation distributions are closely aligned across depths, indicating that basis traits generalize beyond the training set. **(b)** Sibling overlap is low at shallow depths and increases moderately at deeper levels, reflecting controlled basis-trait sharing rather than full redundancy.

D.4 Max-Dominance vs. mean aggregation

Complementing Sec. D.1, we isolate the contribution of the Max-Dominance aggregation rule by keeping the per-class hierarchy and the full training objective fixed (MB and consistency both enabled) and replacing Max-Dominance with mean aggregation across siblings.

Under the non-degenerate regime established by the Molding Bonus, mean aggregation produces substantially denser activations ($\bar{k}: 2.32 \rightarrow 5.83$) at comparable OOD accuracy, whereas Max-Dominance yields a sparser, more inspectable trait representation. Combined with Sec. D.1, this summarizes how the three components specialize: the per-class hierarchy provides the substrate for within-class evidence variants, the Molding Bonus prevents collapse and ensures non-degenerate utilization, and Max-Dominance selects sparse, inspectable representations among non-degenerate alternatives.

Table 12: **Max-Dominance vs. mean aggregation on Camelyon17-WILDS.** Dead-trait ratio and mean active traits ($\bar{k}$) on the clean training split; OOD accuracy on the WILDS Val/Test splits. Mean$_{\pm \text{SD}}$ over 3 seeds.

Model	Dead traits (train) ↓	$\bar{k}$ (train) ↓	Val (OOD) ↑	Test (OOD) ↑
Max+MB	$0.057_{\pm 0.052}$	$2.321_{\pm 0.132}$	$94.38_{\pm 0.28}$	$92.48_{\pm 1.28}$
Mean+MB	$0.000_{\pm 0.000}$	$5.828_{\pm 0.428}$	$91.04_{\pm 1.62}$	$91.72_{\pm 1.50}$

E CAMELYON17-WILDS: TRAIT PROFILING AND RELIANCE ANALYSIS

This appendix provides the full method and quantitative results for the patient-share analysis summarized in Section 4.4.

Phase 1: trait profiling via patient share. The patient-share results below describe one illustrative $D = 5$ Stage-2 run. Traits with fewer than 10 training activations are excluded as Rare; the remaining traits are categorized using training data, with these categories held fixed on Validation and Test.

We quantify whether a trait is broadly supported across individuals or dominated by a specific training patient. For each trait t and patient p, we compute a patient-normalized activation rate

$$r_{t,p} \;=\; \frac{\#\{\text{activations of trait } t \text{ in patient } p\}}{\#\{\text{images of patient } p\}}. \tag{31}$$

We normalize across patients to obtain the contribution share

$$\text{Share}_{t,p} \;=\; \frac{r_{t,p}}{\sum_j r_{t,j}}. \tag{32}$$

Using a 50% patient-share threshold, we categorize traits into two types: *Cross-Patient Traits* ($\max_p \text{Share}_{t,p} < 0.5$), supported by evidence patterns that recur across multiple individuals; and *Single-Patient Traits* ($\max_p \text{Share}_{t,p} \geq 0.5$), dominated by one patient source.

Mapping traits to their target classes reveals a pronounced asymmetry. Normal predictions are supported exclusively by Cross-Patient Traits (32/32). The Tumor class additionally relies on nearly all Single-Patient Traits, combining cross-patient and patient-specific evidence.

Phase 2: adaptive reliance. We stratify trait usage by ground-truth class and split. Usage is the per-image fraction of active non-Rare traits belonging to a given type, counting traits across both class-specific hierarchies. Among NORMAL images with at least one active non-Rare trait, mean Cross-Patient usage exceeds 99.9% across splits. For TUMOR images, mean Single-Patient usage rises from $\sim$8% on Validation to $\sim$21% on Test; all tumor images are included, with zero assigned to images having no active non-Rare traits. This reliance trend coincides with the barcode-template prevalence shift visible in Figure 3.

On the OOD Test set, we compute accuracy on the images activating each trait, then average equally across test-active traits within each category. These mean conditional accuracies are 92.1% versus 91.0% for ATCG versus baseline among Cross-Patient traits, with a larger difference among Single-Patient traits (98.2% versus 94.1%). The activation-conditioned sample sets can overlap.

Trait-localized evidence. We additionally localize trait-specific evidence using Grad-CAM (Selvaraju et al., 2017) and Guided Backpropagation (Springenberg et al., 2015), computed with respect to each trait's activation score. Figure 9 shows examples for two sibling traits (A/B), contrasting patches that activate only A, only B, or both. In the co-activation case, the two siblings consistently highlight distinct spatial regions within the same patch, often corresponding to distinct morphological cues. As a complementary expert check, we consulted a practicing pathologist, who interpreted Trait A as tumor regions rich in tumor-infiltrating lymphocytes and Trait B as infiltrating tumor cells (some with cytoplasmic vacuolation); the co-activated siblings localized to complementary, largely non-overlapping regions. We note that these are provided as qualitative evidence; quantitative factor-level validation is left to future work.

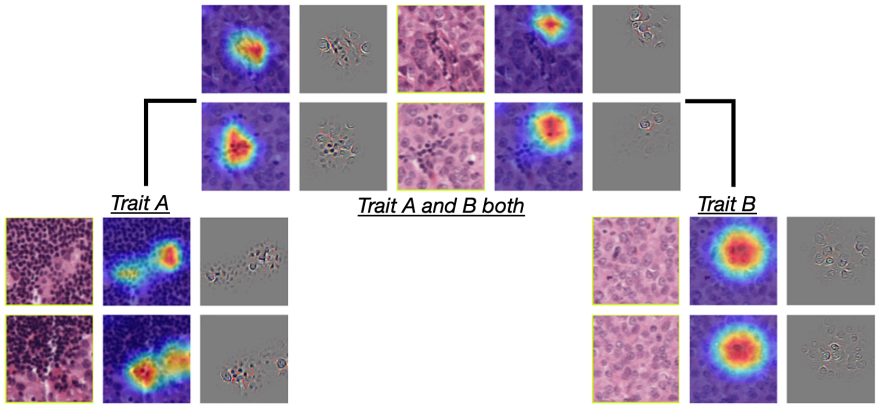


Figure 9: **Trait-wise localization on Camelyon17-WILDS.** For two representative tumor patches: the input image (left), trait-specific Grad-CAM heatmaps (middle), and Guided Backpropagation (GBP, right). When multiple traits are active, each trait localizes to a distinct evidence region; co-activation highlights complementary regions within the same patch, decomposing the decision into multiple localized evidence pathways rather than relying on a single feature.

F BARCODE DISTRIBUTION DIAGNOSTICS: DIVERGENCE AND ANOMALY TESTS

This appendix specifies the barcode-based monitoring procedure used in our diagnostic experiments. The same protocol applies to both the Stacked-MNIST shift simulation and other generative settings.

Empirical predicted-class barcode distributions. Given a trained ATCG model and a dataset split $\mathcal{S}$, each sample $x \in \mathcal{S}$ yields a predicted label $\hat{y}(x)$ and a predicted-class trait barcode $\mathbf{b}_{\hat{y}(x)}(x) \in \{0,1\}^{2^D}$ by thresholding terminal-depth basis-trait activations. We map each predicted-class barcode to an integer ID $\kappa \in \{0, \ldots, K-1\}$, where $K = C \cdot 2^{2^D}$ counts all (class, barcode) pairs (C class trees, each admitting at most 2^{2^D} distinct terminal barcodes), and compute the empirical frequency vector $P \in [0,1]^K$ by counting barcode IDs:

$$P(\kappa) = \frac{n(\kappa)}{N}, \qquad N = \sum_{\kappa=0}^{K-1} n(\kappa), \tag{33}$$

where $n(\kappa)$ is the number of occurrences of barcode ID κ in the split.

Global drift via KL divergence. To quantify global differences between reference and target barcode distributions, we compute a directional KL divergence. For each cohort, we distribute a total prior mass $\lambda = 1$ uniformly over the K barcode identities:

$$\widetilde{P}(\kappa) = \frac{n(\kappa) + \lambda/K}{N + \lambda}. \tag{34}$$

Here, N is the cohort size, and all logarithms are natural. The KL score is then

$$D_{\mathrm{KL}}\big(\widetilde{P}_{\mathrm{ref}} \,\|\, \widetilde{P}_{\mathrm{tgt}}\big) = \sum_{\kappa=0}^{K-1} \widetilde{P}_{\mathrm{ref}}(\kappa) \, \log \frac{\widetilde{P}_{\mathrm{ref}}(\kappa)}{\widetilde{P}_{\mathrm{tgt}}(\kappa)}. \tag{35}$$

Local anomaly detection via Z-scores (mode loss and gain). To localize which barcodes change significantly, we compute a per-mode Z-score using an unpooled large-sample normal approximation. Let $P_{\mathrm{ref}}(\kappa) = n_{\mathrm{ref}}(\kappa)/N_{\mathrm{ref}}$ and $P_{\mathrm{tgt}}(\kappa) = n_{\mathrm{tgt}}(\kappa)/N_{\mathrm{tgt}}$. We define

$$Z_\kappa = \frac{P_{\mathrm{tgt}}(\kappa) - P_{\mathrm{ref}}(\kappa)}{\sqrt{\frac{P_{\mathrm{ref}}(\kappa)\big(1-P_{\mathrm{ref}}(\kappa)\big)}{N_{\mathrm{ref}}} + \frac{P_{\mathrm{tgt}}(\kappa)\big(1-P_{\mathrm{tgt}}(\kappa)\big)}{N_{\mathrm{tgt}}} + \delta}}, \tag{36}$$

where δ is a small constant for numerical stability. We flag significant anomalies via a two-sided cutoff z_{bonf}:

$$\textbf{Loss}: \ Z_\kappa < -z_{\text{bonf}}, \qquad \textbf{Gain}: \ Z_\kappa > z_{\text{bonf}}. \tag{37}$$

Multiple-testing correction and cutoff. We use a Bonferroni-corrected two-sided cutoff for family-wise error rate α:

$$z_{\text{bonf}} = \Phi^{-1}\Big(1 - \frac{\alpha}{2K}\Big), \tag{38}$$

where $\Phi^{-1}(\cdot)$ is the standard normal inverse CDF and K is the number of monitored hypotheses, one per barcode ID. For both the Stacked-MNIST and CIFAR-10 diagnostics, $D = 4$ and $C = 10$, giving $K = 655{,}360$ and, with $\alpha = 0.05$,

$$z_{\text{bonf}} \approx 5.376. \tag{39}$$

Numerical constants and continuous-representation tests. The barcode analyses in Tables 3 and 4 both use Eq. (34) with $\lambda = 1$. The local anomaly test uses the unsmoothed empirical proportions in Eq. (36) and $\delta = 10^{-8}$ as a numerical stability offset.

For the continuous representations in Table 3, we use the exact biased quadratic Gaussian-RBF MMD^2 statistic with a pooled label-invariant median squared-distance bandwidth and 999 fixed-size permutations. The reported raw Monte Carlo value is

$$p = \frac{1 + E}{1 + 999},$$

where E is the number of permuted statistics at least as large as the observed statistic. These are raw per-scenario p-values; no across-scenario Holm correction is applied.

CIFAR-10 generative diagnostics. Table 4 uses cached outputs from a single Stage 2 ATCG model with $C = 10$, $D = 4$, and activation threshold 5.5. The reference comprises the 50,000 CIFAR-10 training images; each generated cohort contains 50,000 samples. All four comparisons use the same model and reference cohort.

Factor-level localization via Recall@L (Stacked-MNIST only). When ground-truth latent factors are available, we measure whether detected anomalies localize the manipulated factors. Let $\mathcal{A} = \{\kappa : |Z_\kappa| > z_{\text{bonf}}\}$ be the set of anomalous barcode IDs. For each anomalous barcode $\kappa \in \mathcal{A}$, we tabulate the Red–Green factor combinations among its associated samples from the reference cohort in Missing scenarios and from the target cohort in Novel scenarios. Let $\text{Top-}L(\kappa)$ denote the L most frequent factor combinations among those samples. Given a ground-truth shifted factor set $\mathcal{M}_\Delta$, we define

$$\text{Recall@}L = \frac{\big|\mathcal{M}_\Delta \cap \big(\cup_{\kappa \in \mathcal{A}} \text{Top-}L(\kappa)\big)\big|}{|\mathcal{M}_\Delta|}. \tag{40}$$

In our experiments, we report Recall@3.

F.1 Stacked-MNIST diagnostics: shift construction

This appendix specifies the controlled shift simulation used in Sec. 5.1.

For each scenario in Table 3, we independently generate reference and target cohorts of 60,000 observations each from MNIST test images, subject to the exclusions specified below. The representation models are fixed, and all diagnostics for a scenario use the same cohorts. The continuous representations are the baseline's 512-dimensional embeddings, ATCG's 160 terminal basis-trait activations across all ten classes, and its ten class logits.

Generative factors and composite labels. Each Stacked-MNIST image is formed by independently sampling three MNIST digits $(r, g, b) \in \{0, \ldots, 9\}^3$ for the Red/Green/Blue channels. We define the class label as

$$y = (r \bmod 2) + 2 \cdot (g \bmod 5) \in \{0, \ldots, 9\}. \tag{41}$$

Thus each class corresponds to a Red-parity choice and a Green-digit group. We encode each raw *Red–Green factor combination* (r, g) by the mode ID

$$m = 10r + g \in \{0, \ldots, 99\}, \tag{42}$$

with r in the tens place and g in the ones place (e.g., $m = 45$ denotes $r = 4$, $g = 5$). This agrees with the stored raw RGB target $100r + 10g + b$. We denote by $\mathcal{M}$ the set of all 100 such combinations; each class label y corresponds to 10 distinct combinations in $\mathcal{M}$.

Controlled factor shifts. For Missing scenarios, we exclude selected Red–Green mode IDs from the target cohort; for Novel scenarios, we exclude them from the reference cohort. We consider three regimes: (i) one combination, $\Delta = \{45\}$, corresponding to $(r, g) = (4, 5)$; (ii) two combinations, $\Delta = \{40, 45\}$, sharing Red digit $r = 4$ and differing between Green digits $g = 0$ and $g = 5$ within the same composite class; and (iii) ten combinations, $\Delta = \{40, \ldots, 49\}$, covering all Green digits at fixed Red digit $r = 4$ and spanning five composite classes. These exclusions need not preserve class prevalence.

G TWO-STAGE RESULTS WITH SEED-LEVEL VARIANCE

This appendix complements Table 2 with seed-level variance estimates and a separate Stage 1 breakdown. We report mean$_{\pm\text{SD}}$ over 3 seeds for both stages across all benchmarks. Stage 1 (Table 13) attaches the Pinball head to a frozen backbone, suitable when the encoder must remain unchanged; Stage 2 (Table 14) additionally adapts the encoder. The two tables share an identical structure to Table 2, with each barcode-type cell reporting per-type test accuracy together with the fraction of test samples in that type. For rare barcode types (*Multi-Inter* and, on certain splits, *None*), the supporting sample count per seed can be very small (Prop $< 1\%$); the resulting Acc variance can be amplified by small supporting sample counts.

Table 13: **Stage 1 results with seed-level variance.** Test accuracy and barcode-type breakdown across nine benchmarks. Values are mean$_{\pm\text{SD}}$ over 3 seeds. Each barcode-type cell reports per-type accuracy with the fraction of test samples in parentheses. Variance for rare types (notably *Multi-Inter*) can be amplified by small supporting sample counts.

Dataset	Stage 1 Acc	Stage 1 barcode-type: Acc$_{\pm\text{SD}}$ (Prop%$_{\pm\text{SD}}$)			
		Single	Multi-Intra	Multi-Inter	None
MNIST	$99.47_{\pm 0.05}$	$99.28_{\pm 0.10}$ ($31.22_{\pm 0.40}$)	$99.95_{\pm 0.01}$ ($64.15_{\pm 0.37}$)	$62.17_{\pm 22.73}$ ($0.13_{\pm 0.03}$)	$95.13_{\pm 0.90}$ ($4.51_{\pm 0.08}$)
F-MNIST	$94.34_{\pm 0.08}$	$92.73_{\pm 0.36}$ ($37.89_{\pm 0.44}$)	$98.20_{\pm 0.05}$ ($55.26_{\pm 0.48}$)	$55.40_{\pm 2.89}$ ($1.42_{\pm 0.25}$)	$76.35_{\pm 0.58}$ ($5.42_{\pm 0.18}$)
STL-10	$98.28_{\pm 0.04}$	$98.74_{\pm 0.17}$ ($52.06_{\pm 0.82}$)	$99.85_{\pm 0.01}$ ($39.42_{\pm 1.09}$)	$72.60_{\pm 3.47}$ ($0.73_{\pm 0.01}$)	$89.70_{\pm 0.56}$ ($7.80_{\pm 0.29}$)
CIFAR-10	$97.17_{\pm 0.10}$	$96.52_{\pm 0.45}$ ($33.60_{\pm 0.55}$)	$99.46_{\pm 0.12}$ ($59.01_{\pm 0.65}$)	$60.31_{\pm 10.35}$ ($0.81_{\pm 0.08}$)	$84.46_{\pm 1.32}$ ($6.58_{\pm 0.16}$)
CIFAR-100	$84.44_{\pm 0.06}$	$88.01_{\pm 0.73}$ ($37.29_{\pm 0.31}$)	$96.01_{\pm 0.14}$ ($40.76_{\pm 0.51}$)	$57.79_{\pm 2.76}$ ($1.90_{\pm 0.17}$)	$56.83_{\pm 0.37}$ ($20.05_{\pm 0.26}$)
CIFAR-100 (Super)	$91.56_{\pm 0.15}$	$93.94_{\pm 0.23}$ ($59.89_{\pm 0.64}$)	$97.97_{\pm 0.17}$ ($30.06_{\pm 0.74}$)	$58.97_{\pm 1.16}$ ($1.79_{\pm 0.07}$)	$58.07_{\pm 1.20}$ ($8.26_{\pm 0.19}$)
Histopathology (domain shift)					
Camelyon17 Val	$94.17_{\pm 0.18}$	$94.81_{\pm 0.59}$ ($59.12_{\pm 9.35}$)	$97.88_{\pm 1.10}$ ($32.01_{\pm 8.99}$)	$54.74_{\pm 4.19}$ ($0.26_{\pm 0.15}$)	$75.74_{\pm 0.99}$ ($8.61_{\pm 1.87}$)
Camelyon17 Test	$89.59_{\pm 1.38}$	$93.52_{\pm 1.40}$ ($42.95_{\pm 7.77}$)	$90.61_{\pm 5.64}$ ($25.27_{\pm 6.01}$)	$23.84_{\pm 17.37}$ ($0.05_{\pm 0.05}$)	$83.31_{\pm 0.68}$ ($31.74_{\pm 5.93}$)
Non-vision (RNA-seq)					
TCGA-10	$98.60_{\pm 0.15}$	$99.37_{\pm 0.20}$ ($53.59_{\pm 2.45}$)	$99.67_{\pm 0.24}$ ($36.44_{\pm 1.45}$)	$71.71_{\pm 6.44}$ ($1.80_{\pm 0.38}$)	$94.58_{\pm 1.75}$ ($8.17_{\pm 1.04}$)

Table 14: **Stage 2 (full ATCG) results with seed-level variance.** Same structure as Table 13, with the backbone additionally fine-tuned. Mean±SD over 3 seeds.

Dataset	Stage 2 Acc	Stage 2 barcode-type: Acc$_{\pm\text{SD}}$ (Prop%$_{\pm\text{SD}}$)			
		Single	Multi-Intra	Multi-Inter	None
MNIST	$99.47_{\pm 0.07}$	$99.47_{\pm 0.11}$ ($48.29_{\pm 0.27}$)	$99.93_{\pm 0.03}$ ($50.44_{\pm 0.32}$)	$64.12_{\pm 12.93}$ ($0.15_{\pm 0.02}$)	$83.71_{\pm 3.57}$ ($1.11_{\pm 0.02}$)
F-MNIST	$94.53_{\pm 0.16}$	$94.44_{\pm 0.23}$ ($53.73_{\pm 0.80}$)	$97.80_{\pm 0.04}$ ($42.31_{\pm 0.75}$)	$55.53_{\pm 4.90}$ ($1.62_{\pm 0.15}$)	$64.59_{\pm 2.00}$ ($2.34_{\pm 0.17}$)
STL-10	$98.43_{\pm 0.01}$	$98.99_{\pm 0.11}$ ($61.98_{\pm 0.58}$)	$99.88_{\pm 0.03}$ ($30.83_{\pm 0.71}$)	$72.37_{\pm 3.48}$ ($0.53_{\pm 0.05}$)	$88.68_{\pm 1.30}$ ($6.67_{\pm 0.19}$)
CIFAR-10	$96.86_{\pm 0.13}$	$96.28_{\pm 0.21}$ ($42.36_{\pm 0.10}$)	$99.17_{\pm 0.12}$ ($53.56_{\pm 0.22}$)	$60.25_{\pm 2.41}$ ($0.69_{\pm 0.03}$)	$75.11_{\pm 0.25}$ ($3.39_{\pm 0.21}$)
CIFAR-100	$84.51_{\pm 0.16}$	$87.22_{\pm 0.25}$ ($41.05_{\pm 0.49}$)	$95.24_{\pm 0.26}$ ($41.19_{\pm 0.32}$)	$52.64_{\pm 0.63}$ ($2.18_{\pm 0.25}$)	$53.50_{\pm 1.01}$ ($15.59_{\pm 0.39}$)
CIFAR-100 (Super)	$91.46_{\pm 0.06}$	$93.66_{\pm 0.23}$ ($67.34_{\pm 0.82}$)	$97.12_{\pm 0.12}$ ($24.84_{\pm 0.71}$)	$56.78_{\pm 1.90}$ ($1.88_{\pm 0.22}$)	$54.08_{\pm 0.92}$ ($5.94_{\pm 0.29}$)
Histopathology (domain shift)					
Camelyon17 Val	$94.38_{\pm 0.28}$	$94.00_{\pm 0.79}$ ($61.36_{\pm 10.40}$)	$97.71_{\pm 1.15}$ ($35.30_{\pm 10.33}$)	$55.81_{\pm 3.05}$ ($0.56_{\pm 0.36}$)	$62.99_{\pm 7.99}$ ($2.78_{\pm 0.98}$)
Camelyon17 Test	$92.48_{\pm 1.28}$	$94.76_{\pm 0.85}$ ($54.75_{\pm 15.95}$)	$93.08_{\pm 5.75}$ ($33.98_{\pm 13.49}$)	$49.62_{\pm 3.99}$ ($0.18_{\pm 0.12}$)	$76.83_{\pm 7.18}$ ($11.09_{\pm 3.88}$)
Non-vision (RNA-seq)					
TCGA-10	$98.88_{\pm 0.08}$	$99.21_{\pm 0.06}$ ($74.71_{\pm 0.62}$)	$99.50_{\pm 0.42}$ ($18.24_{\pm 0.88}$)	$83.33_{\pm 23.57}$ ($0.31_{\pm 0.04}$)	$94.43_{\pm 0.45}$ ($6.74_{\pm 0.58}$)

Stage 1 provides usable structure; Stage 2 strengthens it. Across all nine benchmarks, Stage 1 already delivers stable barcode-type categorization with low seed-level variance on the dominant types

(*Single* and *Multi-Intra*, jointly covering $\geq 75\%$ of test samples on most splits). Differences between Stage 1 and Stage 2 in raw test accuracy are mostly small ($< 1\%$ in either direction), with the notable exception of the Camelyon17 OOD test split ($89.59 \pm 1.38 \rightarrow 92.48 \pm 1.28$). On Stacked-MNIST, the three-run mean Trait-16 Joint-RG silhouette increases from 0.159 (Stage 1) to 0.293 (Stage 2), and terminal-depth Red-factor purity increases from 64.55% to 73.61% (Table 1). Stage 1 is a viable deployment mode when the backbone must remain frozen; Stage 2 is the appropriate choice when backbone fine-tuning is permitted and stronger structural organization is the goal.

G.1 STACKED-MNIST: THREE-RUN STRUCTURAL DIAGNOSTICS

Table 15 supplements Table 1. At $D = 4$, each class-specific hierarchy has $2^D = 16$ basis traits; ATCG uses their activations for the ground-truth class. For each model and run, PCA is fitted to the full 60,000-example training set of 512-dimensional penultimate embeddings, retaining the first 16 components without whitening. Both representations are evaluated on these same training observations using Euclidean silhouette scores, with the ten (r, g) digit identities as groups within each composite class. Scores are averaged equally across the ten classes, and three-run summaries report mean $\pm$ population SD. The paired trait-minus-PCA-16-embedding Joint-RG differences are 0.13346 ± 0.01922 for Stage 1 and 0.12134 ± 0.01443 for Stage 2; these are descriptive run-level summaries.

Table 15: **Supporting Stacked-MNIST structural diagnostics.** Panel (a) reports individual-run Joint-RG silhouettes. Panels (b) and (c) report three-run means and population SDs. All values use the full training set.

(a) Joint-RG silhouette by run

Representation	Run 0	Run 1	Run 2
Baseline embedding PCA-16	0.03420	0.03093	0.01219
Stage 1 embedding PCA-16	0.03417	0.03231	0.01036
Stage 1 Trait-16	0.17889	0.18156	0.11676
Stage 2 embedding PCA-16	0.20650	0.16075	0.14759
Stage 2 Trait-16	0.34629	0.28041	0.25216

(b) Blue silhouette

Model	Embedding PCA-16	Trait-16
Baseline	-0.01916 ± 0.00073	–
Stage 1	-0.01914 ± 0.00074	-0.01984 ± 0.00079
Stage 2	-0.01878 ± 0.00009	-0.01836 ± 0.00110

(c) Factor purity (%)

Model	Depth	Green	Red	Blue
Stage 1	1	94.9701 ± 1.6849	23.9686 ± 0.7156	11.3008 ± 0.0286
Stage 1	4	98.0868 ± 0.2759	64.5489 ± 3.4035	13.1120 ± 0.9098
Stage 2	1	95.1648 ± 1.8357	23.9133 ± 0.6943	11.2954 ± 0.0040
Stage 2	4	97.8516 ± 0.8327	73.6089 ± 2.6747	13.9735 ± 1.2532

G.2 MSP-ADJUSTED BARCODE-TYPE ANALYSIS

We analyze Stage 2 predictions on the five in-distribution image test sets in Table 16, using three classifier seeds. For image i and seed s, the outcome is prediction error, $Y_{is} = \mathbf{1}\{\hat{y}_{is} \neq y_i\}$. Maximum softmax probability (MSP) is clipped to $[10^{-6}, 1 - 10^{-6}]$, transformed to logit-MSP, and standardized within each seed using its mean and population SD, yielding z_{is}. We fit a separate logistic regression for each dataset:

$$\operatorname{logit} \Pr(Y_{is} = 1) = \alpha_s + \beta z_{is} + \sum_{k \in \{I, X, N\}} \gamma_k \, \mathbf{1}\{T_{is} = k\},$$

where α_s is a seed-specific intercept, T_{is} is the barcode type, and I, X, and N denote *Multi-Intra*, *Multi-Inter*, and *None*; *Single* is the reference category. Confidence intervals and two-sided tests use sample-ID cluster-robust covariance because each image is evaluated under all three seeds.

We report two contrasts. First, we compare the equal-category averages of adjusted log error odds for *Single*/*Multi-Intra* versus *Multi-Inter*/*None*, with odds ratio $\mathrm{OR}_A = \exp[(\gamma_I - \gamma_X - \gamma_N)/2]$. This is a contrast in log odds, not a regression on two pooled groups. The *Multi-Intra* versus *Single* contrast has $\mathrm{OR}_B = \exp(\gamma_I)$; values below one indicate lower adjusted error odds for *Multi-Intra*.

Table 16: **MSP-adjusted barcode-type contrasts.** Odds ratios (OR), 95% confidence intervals, and two-sided p-values on five in-distribution test sets.

	Single/Multi-Intra vs. Multi-Inter/None		**Multi-Intra vs. Single**	
Dataset	**OR [95% CI]**	$\boldsymbol{p}$	**OR [95% CI]**	$\boldsymbol{p}$
MNIST	$0.539\ [0.302, 0.963]$	0.036805	$0.170\ [0.090, 0.324]$	6.50×10^{-8}
F-MNIST	$1.052\ [0.871, 1.269]$	0.600563	$0.422\ [0.365, 0.488]$	2.15×10^{-31}
STL-10	$0.484\ [0.265, 0.885]$	0.018383	$0.177\ [0.071, 0.436]$	1.71×10^{-4}
CIFAR-10	$0.959\ [0.750, 1.227]$	0.741310	$0.398\ [0.323, 0.491]$	6.56×10^{-18}
CIFAR-100	$0.502\ [0.443, 0.569]$	4.20×10^{-27}	$0.621\ [0.557, 0.693]$	1.33×10^{-17}

The *Multi-Intra* versus *Single* contrast shows lower adjusted error odds in all five datasets ($p < 0.05$). The *Single*/*Multi-Intra* versus *Multi-Inter*/*None* contrast is significant on MNIST, STL-10, and CIFAR-100.

H SUPPLEMENTARY FIGURES

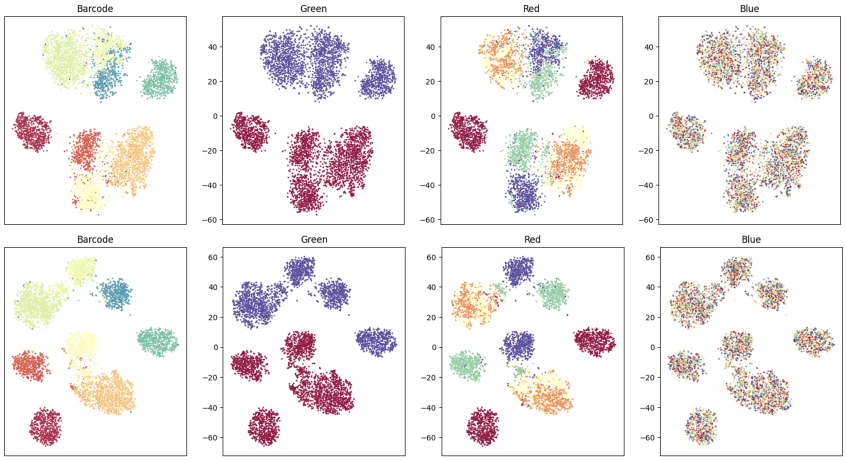


Figure 10: **Evolution of Stacked-MNIST embedding geometry (t-SNE).** Feature embeddings after Stage 1 (top) and Stage 2 (bottom), with points colored by predicted-class trait barcode (leftmost panel) and by each ground-truth generative factor (Green, Red, Blue). The Stage 1 backbone remains frozen. After Stage 2, the projections show more distinct grouping by the Red and Green factors, while the label-irrelevant Blue digit remains visually mixed within clusters.

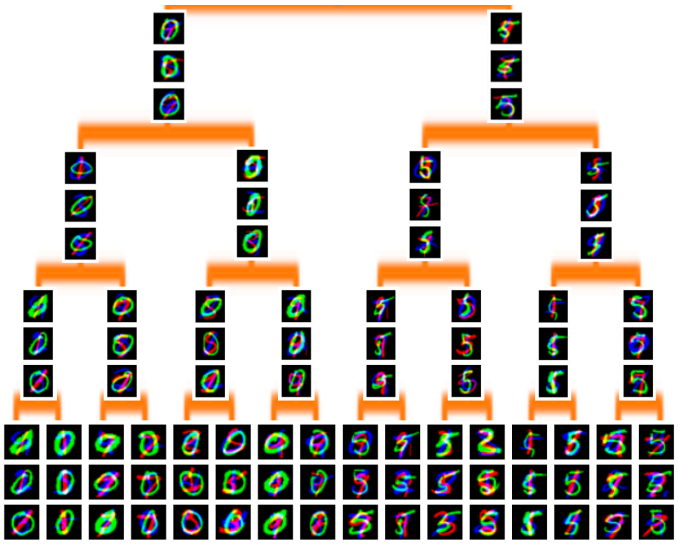


Figure 11: **Learned trait hierarchy on Stacked-MNIST (path-exclusive vs. mixed cases).** Each thumbnail shows a representative sample routed to a node of the learned hierarchy. The *bottom row* visualizes path-exclusive samples whose predicted-class barcode activates a *single* terminal-depth basis-trait, yielding the most specific decision patterns. Moving *upward*, each parent node aggregates samples in which *both* sibling basis-traits at that level are active, illustrating how ATCG composes coarse-to-fine structure by merging mixed-activation evidence. Higher-level splits align primarily with **Green**-group identity, while deeper branches refine by the specific **Red** digit identity within each group; the label-irrelevant **Blue** digit remains mixed within nodes.

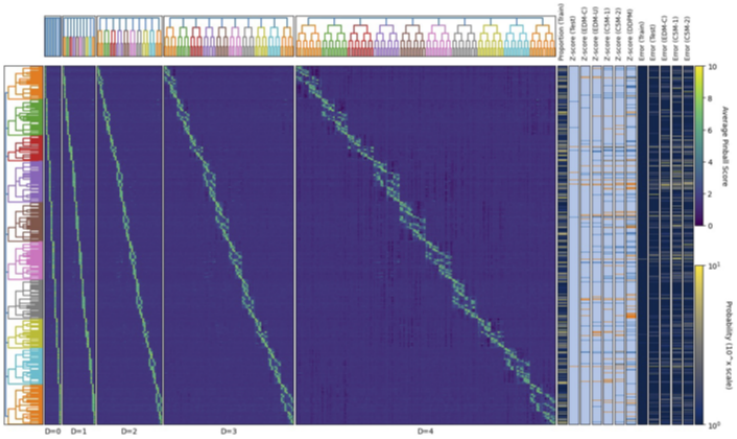


Figure 12: **Visual diagnosis of generative structure fidelity on CIFAR-10.** Decision-structure alignment between training data and generative models, integrated into a single heatmap. *Main panel*: trait activation profiles (columns) for distinct barcodes (rows); rows are ordered by single-linkage clustering on hierarchical trait distance, grouping similar barcodes into coherent blocks, and columns follow the Pinball binary-tree structure. *Right panel*: per-barcode reference frequency P_{ref} (first column) and per-generator Z-scores relative to the reference (remaining columns). Blue bands ($Z < -z_{\mathrm{bonf}}$) localize collapsed decision branches (Loss); red bands ($Z > z_{\mathrm{bonf}}$) localize invented modes (Gain).