Supplementary Material

A. Semantic Interpretation of Geometric Regularity: Texture-vs-Shape Hypothesis

In the main text, we established that our proposed method mitigates Concept Leakage by smoothing the concept manifold geometry. In this supplementary analysis, we refer to the Concept-Centric Transformer trained with our bi-level perturbation strategy as "Robust CCT" [64], distinguishing it from the standard baseline, "Vanilla CCT". We provide a semantic interpretation of the geometric regularity achieved by Robust CCT by linking it to the *Texture-vs-Shape Hypothesis* [24]. We posit that the "high curvature" observed in Vanilla CCT is not random noise, but a geometric manifestation of the model's reliance on high-frequency local textures [38, 40–42].

The link between geometry and semantics [39] is rooted in the spectral properties of visual features. Local texture cues (e.g., fur, scales, or bark) are inherently high-frequency and spatially volatile; separating classes based on these subtle variations requires the decision boundary to form complex, highly curved surfaces. In contrast, global object shapes represent low-frequency structures that vary smoothly across the input space, naturally mapping to flatter, more linear manifolds. Consequently, standard models that minimize training error without geometric constraints tend to latch onto texture shortcuts, resulting in highly curved manifolds prone to Concept Leakage.

Our core hypothesis is that by explicitly penalizing high curvature via bi-level perturbations, Robust CCT effectively acts as a "geometric low-pass filter" on the representation learning process. This forces the model to abandon brittle, high-frequency texture cues in favor of robust, low-frequency structural features.

Figure A-1 empirically validates this hypothesis using the *Stylized-ImageNet* dataset [24], which creates a conflict between shape (e.g., a cat) and texture (e.g., elephant skin). While standard CNNs (blue dots) and Vanilla CCT (green stars) fall into the texture trap, heavily predicting based on surface patterns, Robust CCT (red stars) demonstrates a decisive shift toward the shape-biased region. This alignment with human perception (red diamonds) confirms that the induced geometric regularity does not merely stabilize numerical outputs; it fundamentally alters the semantic nature of the learned representations. By flattening the manifold, Robust CCT successfully steers the model to perceive objects through their robust global forms rather than their incidental surface textures.

B. Further Related Works

This section provides an extended review of explainable AI (XAI) methodologies, contextualizing our focus on intrinsic concept-based interpretability.

Overview. Recent advancements in machine learning have increasingly prioritized the development of models that offer transparent and interpretable justifications for their predictions. Explainability approaches are broadly categorized into *post hoc* techniques and *intrinsically interpretable* models.

Post-hoc Explanations. Post-hoc methods aim to elucidate the decision-making process of a trained black-box model without altering its internal structure. These methods typically employ feature attribution techniques [2, 54, 67] to quantify the contribution of individual input features. Prominent examples include activation maximization [62, 81, 91] and saliency-based visualizations [73, 75, 78], which have been widely adopted to analyze the feature sensitivity of Convolutional Neural Networks (CNNs).

Complementary to feature attribution, attention-based mechanisms have been utilized to highlight discriminative input regions [22, 28, 29, 45, 95–98]. While these techniques offer valuable insights into a model's focus, they are limited by their retrospective nature; they describe *where* the model looked, but do not necessarily explain *what* semantic features drove the decision or guarantee that the highlighted regions are causally linked to the prediction.

Intrinsic Concept-based Models. To address the limitations of post-hoc approximations, intrinsic concept-based approaches have emerged as a robust alternative. These models are designed to align their internal representations directly with human-understandable semantic concepts [4, 12, 13, 27, 31, 47–50, 69, 89, 90, 92]. By constraining the reasoning process to operate on high-level concepts rather than pixel-level features, these methods enhance interpretability while maintaining competitive predictive performance.

A notable architecture in this domain is the Concept Transformer (CT) [69], which decomposes predictions into structured, multidimensional concept streams. While CTs effectively improve transparency, they often rely on predefined regions or attention maps to extract concepts, potentially limiting their flexibility in capturing holistic or semantic structures compared to fully end-to-end architectures like the Concept-Centric Transformer (CCT) used in this work.

C. Dataset Statistics

Table A-1 presents the statistical breakdown of the benchmark datasets used in our experiments. Since these datasets do not provide official validation splits, we manually re-

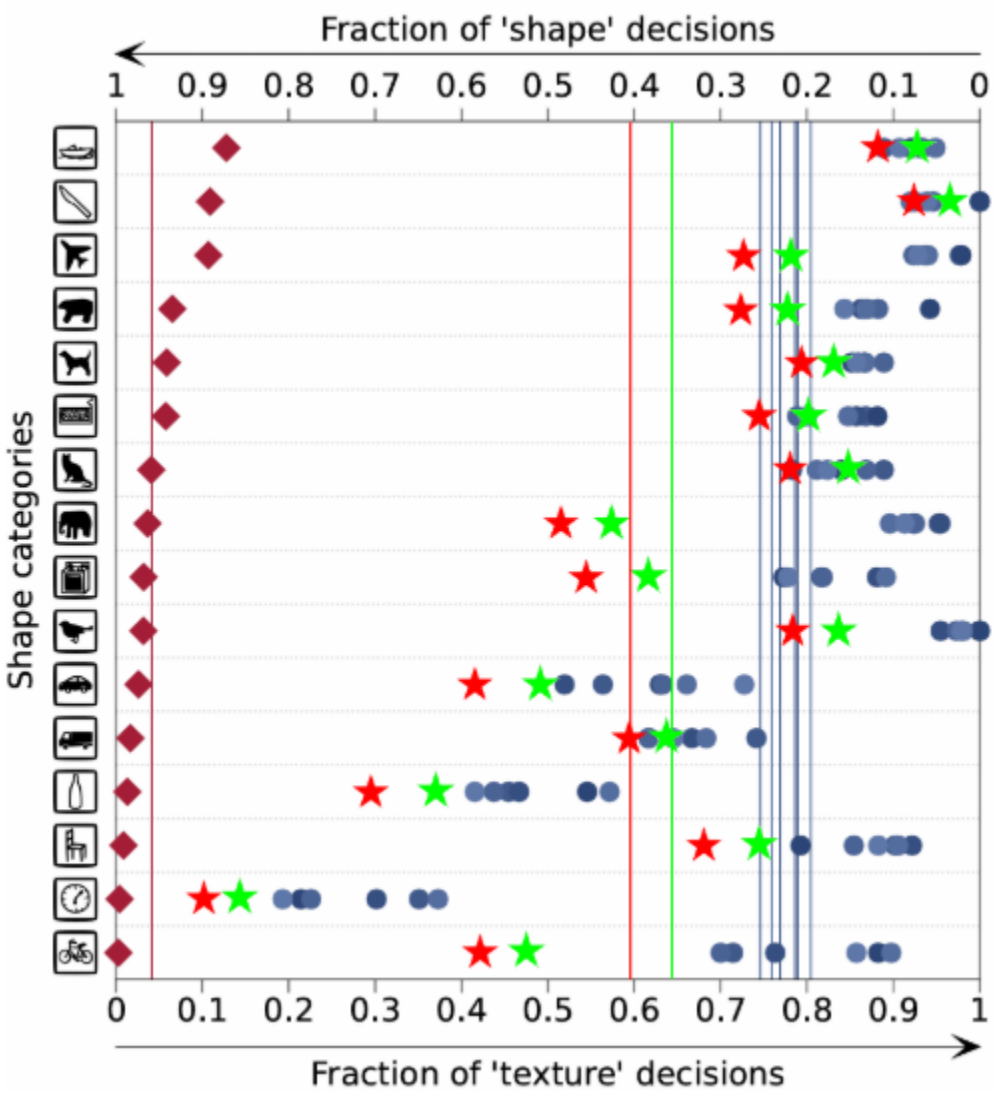


Figure A-1. **Geometric Regularization Induces Human-like Shape Bias.** We evaluate the semantic alignment of learned representations using the *Conflicting Cues* dataset [24], where images contain conflicting shape and texture information (e.g., a cat silhouette with elephant skin). The x-axis quantifies the texture bias: values to the right indicate reliance on local textures (machine-like), while values to the left indicate reliance on global shapes (human-like). Standard CNNs (blue dots) and Vanilla CCT (green stars) cluster heavily in the texture-biased region. In contrast, **Robust CCT** (red stars) exhibits a decisive distributional shift toward the left, aligning significantly closer to Human perception (red diamonds). This empirical evidence confirms that our bi-level perturbation strategy effectively suppresses high-frequency texture noise, compelling the model to ground predictions in robust, global structural concepts. (Figure format adapted from [25].)

served a portion of the training set for hyperparameter tuning.

CUB-200-2011. For the CUB-200-2011 dataset, we followed the preprocessing protocols detailed in [69]. We filtered the original 312 binary concepts by retaining only those present in at least 45% of samples within each class and appearing across a minimum of 8 classes. This filtering process yielded a final set of 108 concepts, which were further categorized into 13 global concepts and 95 spatial concepts based on individual attribute analysis.

ImageNet. Unlike CUB-200-2011, the ImageNet dataset [16] does not contain ground-truth concept annotations. To accommodate this, we treated the concept slots as fully latent variables and set the number of concepts to 50.

Table A-1. **Dataset statistics.** The symbol † indicates that the input resolutions were rescaled to match the ViT backbone requirements. For both datasets, we adhere to the experimental protocols described in Wang et al. [87] and Hong et al. [43].

	Dataset	
Metric	**CUB-200-2011**	**ImageNet**
Input size	$3 \times 224 \times 224^{\dagger}$	$3 \times 224 \times 224^{\dagger}$
# Classes	200	1,000
# Concepts	108 (13 Global / 95 Spatial)	50
# Training samples	5,994	255,224
# Validation samples	1,000	10,000
# Test samples	4,794	20,000

Table A-2. **Training hyperparameter configurations.** Values separated by commas indicate distinct settings for different stages or module components (e.g., specific learning rates for backbone vs. concept heads).

	Dataset	
Hyperparameter	**CUB-200-2011**	**ImageNet**
Batch size	16	256
Epochs	50	10
Warmup iters.	$10, 20$	$10, 20$
Weight decay	$1 \times 10^{-3}, 1 \times 10^{-4}$	$1 \times 10^{-3}, 1 \times 10^{-5}$
Attention sparsity	$0.2, 0.5$	$0.0, 0.2$

D. Experimental Environment

D.1. Hardware Specification

All experiments were conducted on a high-performance computing server equipped with the following specifications:

- **CPU:** Intel® Core™ i7-6950X CPU @ 3.00GHz (Turbo up to 3.50 GHz)
- **RAM:** 128 GB DDR4 2400MHz
- **GPU:** NVIDIA GeForce Titan Xp (Pascal Architecture, 12 GB GDDR5X)

D.2. Model Architectures

We employed Vision Transformers (ViT) [18] as the backbone feature extractors for all models. The implementation was based on the `timm` (PyTorch Image Models) library, integrated within the Hugging Face™ ecosystem.

E. Implementation Details

Table A-2 summarizes the training hyperparameters used for the CUB-200-2011 and ImageNet experiments. Our adversarial noise injection strategy draws primarily from established methods in [58, 84, 94]. For the CCT architecture, we adopted the configuration from [43] with minor adjustments to the warm-up iterations, number of concept slots, and slot dimensions to ensure stability and optimal convergence. These hyperparameters were tuned to balance the diversity of learned concepts with the model's overall representational capacity. For all other baseline models, we strictly adhered to the configurations reported in their respective original publications.

F. Additional Experimental Results

F.1. Geometric Complexity and Data Efficiency

In the main text, we argued that our bi-level perturbation strategy simplifies the representation geometry by suppressing high curvature. Here, we investigate the practical implication of this geometric simplification on *Data Efficiency*. Theoretically, a smoother, lower-curvature manifold possesses lower geometric complexity, implying that fewer training samples should be required to accurately estimate the decision boundaries. If our method truly flattens the manifold, it should exhibit superior generalization in data-scarce regimes compared to baselines that learn complex, volatile boundaries.

To validate this, we report the clean classification accuracy of models trained with varying fractions of the available training data, ranging from 10% to 100%. Experiments were conducted on both CUB-200-2011 and ImageNet-200 to assess whether the observed trends hold across fine-grained and larger-scale recognition settings.

Table A-3 presents a comprehensive comparison. Consistent with our geometric hypothesis, Robust CCT significantly outperforms the Vanilla baseline across all splits. Crucially, the performance gap is most pronounced in extremely low-data regimes (e.g., 10% and 30% splits), where the Vanilla model struggles to define its complex decision boundaries given sparse supervision. As the volume of training data increases, the gap naturally narrows, as sufficient data allows even the complex manifold of the baseline to be adequately sampled.

These results confirm that our method acts as a powerful *geometric regularizer*. By enforcing linearity, it reduces the sample complexity of the learning task, enabling the model to construct robust concept representations even when supervision is severely limited. Full performance results are provided in Table A-4 and Table A-5.

F.2. Adversarial Robustness as a Geometric Consequence

In the main text, we hypothesized that the high curvature of the concept manifold acts as a primary vulnerability, amplifying input noise into significant semantic distortions. Here, we validate this hypothesis through the lens of *Adversarial Robustness*. Mathematically, adversarial attacks exploit regions of high local curvature (large gradients) to flip model predictions with minimal perturbations. Therefore, if our method successfully flattens the manifold as claimed, it should intrinsically reduce the local Lipschitz constant of

Table A-3. **Data Efficiency Performance on CUB-200-2011 and ImageNet-200.** Clean accuracy is reported for models trained with varying subsets of the training data (10% to 100%). Robust CCT outperforms other baselines across most settings, particularly in low-data regimes. Best results are highlighted in **bold**.

	CUB-200-2011 Accuracy (%)							ImageNet-200 Accuracy (%)					
Model	**10%**	**30%**	**50%**	**70%**	**90%**	**100%**	**Model**	**10%**	**30%**	**50%**	**70%**	**90%**	**100%**
ProtoConcept [32]	34.4	37.8	41.7	66.1	77.5	85.2	BotCL [87]	43.4	58.5	69.1	75.1	80.3	83.0
ProtoPFormer [89]	21.1	26.0	53.1	66.4	72.9	84.9	ProtoPFormer [89]	41.2	54.6	63.1	69.3	74.8	83.4
ProtoPool [71]	20.0	34.0	58.8	65.9	71.6	87.6	ProtoPool [71]	42.5	55.9	64.4	70.6	75.6	76.5
ProtoPNet [12]	33.7	45.3	59.0	65.1	78.4	84.8	ProtoPNet [12]	43.1	56.5	65.0	71.2	76.6	77.7
Vanilla CCT	51.0	72.4	76.5	90.0	90.1	90.3	Vanilla CCT	43.4	58.5	66.1	70.1	78.3	83.7
Robust CCT (Ours)	**65.3**	**76.6**	**80.5**	**90.9**	**91.0**	**91.3**	**Robust CCT (Ours)**	**52.6**	**65.8**	**74.0**	**79.8**	**85.7**	**85.9**

Table A-4. **Data efficiency evaluation on CUB-200-2011 (Full performance table).** Clean accuracy reported for models trained with limited data fractions (10%–90%). Robust CCT demonstrates superior performance, particularly in low-data regimes. Best results are in **bold**.

Model	**Training Data Fraction**				
	10%	**30%**	**50%**	**70%**	**90%**
Part R-CNN [95]	44.3	57.9	61.7	65.8	70.2
B-CNN [51]	42.8	56.2	65.1	71.4	77.9
ST-CNN [46]	45.1	58.6	67.4	73.5	78.9
ProtoPNet [12]	33.7	45.3	59.0	65.1	78.4
ProtoPool [71]	20.0	34.0	58.8	65.9	71.6
ProtoPFormer [89]	21.1	26.0	53.1	66.4	72.9
ProtoConcept [32]	34.4	37.8	41.7	66.1	77.5
Deformable ProtoPnet [17]	23.1	25.3	49.9	53.6	70.1
SPDA-CNN [93]	47.2	60.8	69.5	75.6	80.9
CEMM [92]	11.1	19.4	46.8	54.9	62.1
CT [69]	50.6	64.1	72.5	78.3	83.4
Vanilla CCT [43]	51.0	72.4	76.5	90.0	90.1
Robust CCT (Ours)	**65.3**	**76.6**	**80.5**	**90.9**	**91.0**

the mapping, naturally hardening the model against attacks.

To verify this, we subject both the Robust and Vanilla models to rigorous adversarial evaluations. We include full result tables to offer a transparent view of model behavior across various attack vectors, perturbation strengths, and metrics. See Table A-6 and Table A-7 for details.

Methodology. We evaluate robustness against attacks constrained by the ℓ_∞ norm. The objective of these attacks is to generate a minimally perturbed input $\mathbf{x}^{adv}$ that maximizes the loss $\mathcal{L}$ within an ϵ-ball around the original input $\mathbf{x}_0$:

$$\max_{\mathbf{x}^{adv}} \mathcal{L}(f(\mathbf{x}^{adv}), y) \quad \text{s.t.} \quad \|\mathbf{x}^{adv} - \mathbf{x}_0\|_\infty \leq \epsilon, \tag{1}$$

where y denotes the correct label and ϵ represents the perturbation budget. We employ three standard attack strategies:

- *Projected Gradient Descent (PGD)* [55]: PGD iteratively refines the perturbation by following the gradient of the loss surface. The update rule at step $t+1$ is:

$$\mathbf{x}_{t+1}^{adv} = \Pi_{\mathbf{x}_0,\epsilon}\left(\mathbf{x}_t^{adv} + \alpha \cdot \text{sgn}(\nabla_{\mathbf{x}} \mathcal{L}(f(\mathbf{x}_t^{adv}), y))\right), \tag{2}$$

  where $\Pi_{\mathbf{x}_0,\epsilon}$ is the projection onto the ϵ-ball. This method directly probes the local curvature; a model with a flattened geometry (smaller gradients) will inherently limit the effectiveness of these updates.
- *Fast Gradient Sign Method (FGSM)* [30]: A single-step variant of PGD.
- *AutoAttack* [15]: An ensemble of parameter-free attacks (APGD-CE, APGD-DLR, FAB, Square Attack) used to prevent gradient masking and ensure a rigorous assessment.

Results on ImageNet-1K. Table A-6 reports Clean Accuracy (CA) and Robust Accuracy (RA). While standard Vision Transformers and Vanilla CCT succumb easily to attacks due to their volatile boundaries, **Robust CCT** maintains significantly higher robust accuracy. This confirms that flattening the manifold effectively dampens the ampli-

Table A-5. **Data efficiency evaluation on ImageNet-200 (Full performance table).** Clean accuracy reported for models trained with limited data fractions (10%–90%). Robust CCT demonstrates superior performance, particularly in low-data regimes. Best results are in **bold**.

Model	Training Data Fraction				
	10%	**30%**	**50%**	**70%**	**90%**
ProtoPFormer [89]	41.2	54.6	63.1	69.3	74.8
ProtoPool [71]	42.5	55.9	64.4	70.6	76.0
ProtoPNet [12]	43.1	56.5	65.0	71.2	76.6
Deform-ProtoPNet [17]	42.1	54.2	63.0	70.2	75.6
CT [69]	4.1	5.6	12.6	17.2	21.3
BotCL [87]	43.4	58.5	66.1	70.1	78.3
Vanilla CCT [43]	47.4	60.7	69.1	75.1	80.3
Robust CCT (Ours)	**52.6**	**65.8**	**74.0**	**79.8**	**85.7**

fication of input noise.

Results on CUB-200-2011. Table A-7 details performance on the CUB dataset. Robust CCT consistently achieves higher robustness and lower performance degradation compared to Vanilla CCT and other baselines like ProtoPNet. These results provide strong empirical evidence that our geometric regularization strategy successfully constructs a "safety buffer" around decision boundaries, preventing small perturbations from causing concept drift or prediction failure.

F.3. From Local Flatness to Global Structure

In the main text, we focused on "local stability"—ensuring that small perturbations do not disrupt concept activations. Here, we investigate whether this *local* geometric regularization induces coherent structure at the *global* scale of the latent space. Theoretically, if the manifold is locally flattened everywhere, the global topology should essentially become "untangled," leading to better class separation and more reliable probabilistic behavior.

Figure A-2 validates this by visualizing t-SNE [82] embeddings on CUB-200-2011. While Vanilla ViT and CCT show some degree of overlap or fragmentation, **Robust CCT** exhibits significantly tighter and more distinct class clusters. This confirms that penalizing local curvature effectively propagates to the global level, resolving topological irregularities that often plague standard models.

This improved geometry is further quantified by the calibration metrics in Table A-8. A highly curved, irregular manifold often leads to overconfident yet incorrect predictions in regions of uncertainty. By flattening these regions, Robust CCT achieves consistently lower calibration errors (ECE [65], Brier score [11], and NLL [23]) and a higher clustering V-measure [70]. These results provide tangible evidence that our method successfully disentangles the high-dimensional concept space, aligning the model's predicted confidence with its true geometric reliability.

G. Semantic Disentanglement via Geometric Stability

In the main text, we proposed that geometric regularity is a prerequisite for reliable interpretability. Here, we extend this by demonstrating that this regularity also drives clear semantic disentanglement in unsupervised settings. We hypothesize that the separation of foreground and background is not accidental, but a geometric necessity imposed by our training. Background textures often contain high-frequency variations (high curvature), whereas foreground objects possess stable structural forms (low curvature). By enforcing local stability, the model is compelled to segregate these geometrically distinct modalities into separate concept slots to minimize the regularization penalty.

Figure A-3 validates this on ImageNet, a dataset without ground-truth concept labels. Remarkably, Robust CCT autonomously disentangles the scene into distinct foreground and background components. For instance, in the diverse animal classes shown, the model consistently assigns specific slots (Concept 1) to the stable object of interest—such as spiders, dogs, or sharks—while assigning separate slots (Concept 2) to background environments. This decomposition highlights that our geometric regularization acts as a *structural prior*, encouraging the model to organize its latent space around robust, semantic features even in the absence of explicit supervision.

H. Ablation Study

We further validate our design choice of restricting perturbations to the early training phase. Figure A-4 illustrates the clean accuracy as a function of the perturbation duration (percentage of total epochs).

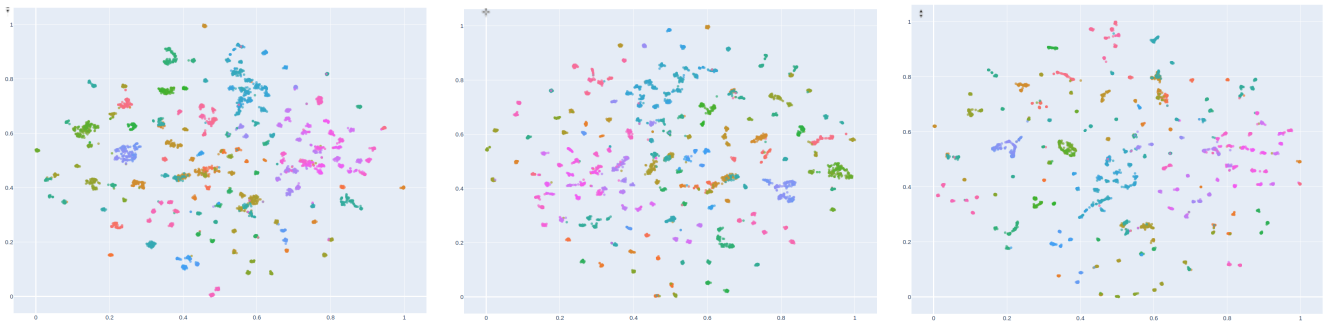
Performance peaks when perturbations are applied for the first 10% of epochs. Extending the duration beyond

Table A-6. **Adversarial robustness on ImageNet-1K.** Comparison of Clean Accuracy (CA) and Robust Accuracy (RA) under PGD and AutoAttack. Robust CCT demonstrates superior stability compared to the vanilla backbone and other standard architectures. (Baselines adapted from [74]).

Model	CA	RA against PGD				RA against AutoAttack			
		1e-3	3e-3	5e-3	1e-2	1e-3	3e-3	5e-3	1e-2
Robust CCT	77.6	61.1	30.1	13.3	1.5	55.5	8.0	0.8	0.0
Vanilla CCT	77.4	57.7	28.2	9.2	1.3	51.0	7.7	0.0	0.0
ViT-S/16	77.6	55.4	24.6	10.2	1.0	48.1	6.0	0.5	0.0
ViT-B/16	75.7	48.9	14.6	6.0	0.9	39.8	5.4	0.6	0.0
ViT-L/16	79.2	55.1	23.4	9.9	1.8	46.6	8.5	1.0	0.0
ViT-SAM-B/16	76.7	63.4	37.0	20.1	3.8	59.8	26.0	8.4	0.1
ViT-B/16-Res	84.0	45.5	8.4	2.3	0.1	27.7	0.9	0.0	0.0
T2T-ViT-14	80.1	37.1	7.0	1.8	0.3	12.9	0.1	0.0	0.0
T2T-ViT-24	82.2	27.7	12.3	3.4	0.2	20.8	0.3	0.0	0.0
Dist-Deit-B/16	81.8	50.0	12.4	3.2	0.2	42.7	3.4	0.2	0.0
Swin-S/4	81.8	40.0	12.4	3.2	0.2	7.9	0.1	0.0	0.0
MLP-Mixer-B/16	73.8	41.9	10.7	1.8	0.0	34.5	3.8	0.3	0.0
ConvNeXt-S	82.7	42.4	8.1	2.6	0.0	15.9	0.0	0.0	0.0
SEResNeXt50	75.7	35.4	4.9	0.8	0.0	21.6	0.6	0.0	0.0
ResNeXt-32x4d-ssl	80.3	23.0	2.9	1.2	0.0	6.5	0.0	0.0	0.0
ResNet50-swsl	81.2	24.7	2.9	1.2	0.0	8.1	0.0	0.0	0.0
ResNet18	70.0	24.9	2.0	0.6	0.1	14.3	0.4	0.0	0.0
ResNet50-32x4d	77.6	13.2	0.2	0.0	0.0	13.2	0.2	0.0	0.0
ShuffleNet	69.4	15.0	0.4	0.0	0.0	6.1	0.0	0.0	0.0
MobileNet	71.9	16.7	0.4	0.0	0.0	7.8	0.0	0.0	0.0
VGG16	71.6	26.3	3.2	1.3	0.0	16.7	0.5	0.0	0.0

Table A-7. **Adversarial robustness on CUB-200-2011.** Robust Accuracy under PGD, FGSM, and AutoAttack. Robust CCT shows the smallest accuracy drop under attack.

Attack	ϵ	α	**Metric**	**Vanilla CCT**	**Robust CCT**	**ProtoPNet**	**ProtoPool**
PGD [55]	1e-3	2e-4	Robust Accuracy (↑)	71.1	**74.7**	13.1	16.5
			Accuracy drop (↓)	19.2	**16.6**	71.7	71.1
	3e-3	8e-4	Robust Accuracy (↑)	27.5	**35.0**	5.8	5.2
			Accuracy drop (↓)	62.8	**56.3**	79.0	82.4
	5e-3	1.2e-3	Robust Accuracy (↑)	9.3	**13.5**	0.0	1.1
			Accuracy drop (↓)	81.0	**77.8**	84.8	86.5
	1e-2	2.5e-3	Robust Accuracy (↑)	1.9	**2.4**	0.0	0.8
			Accuracy drop (↓)	88.4	**88.9**	84.8	86.8
AutoAttack [15]	1e-3	N/A	Robust Accuracy (↑)	60.0	**62.2**	3.2	5.3
			Accuracy drop (↓)	30.3	**29.1**	81.6	82.3
	3e-3	N/A	Robust Accuracy (↑)	10.0	**15.0**	0.1	2.0
			Accuracy drop (↓)	80.3	**76.3**	84.7	85.6
	5e-3	N/A	Robust Accuracy (↑)	3.3	**5.7**	0.0	1.1
			Accuracy drop (↓)	87.0	**85.6**	84.8	86.5
	1e-2	N/A	Robust Accuracy (↑)	0.0	0.0	0.0	0.0
			Accuracy drop (↓)	90.3	91.3	84.8	87.6
FGSM [30]	1e-3	N/A	Robust Accuracy (↑)	77.5	**82.4**	17.6	25.3
			Accuracy drop (↓)	12.8	**8.9**	67.2	62.3
	5e-3	N/A	Robust Accuracy (↑)	47.1	**60.0**	7.2	15.0
			Accuracy drop (↓)	43.2	**31.3**	77.6	72.6



(a) Vanilla ViT (b) Vanilla CCT (c) Robust CCT (Ours)

Figure A-2. **Impact of Geometric Regularization on Latent Topology.** t-SNE visualizations on CUB-200-2011 reveal that Robust CCT forms more separable and compact class clusters compared to baselines. This suggests that our local flattening strategy effectively untangles the global manifold structure.

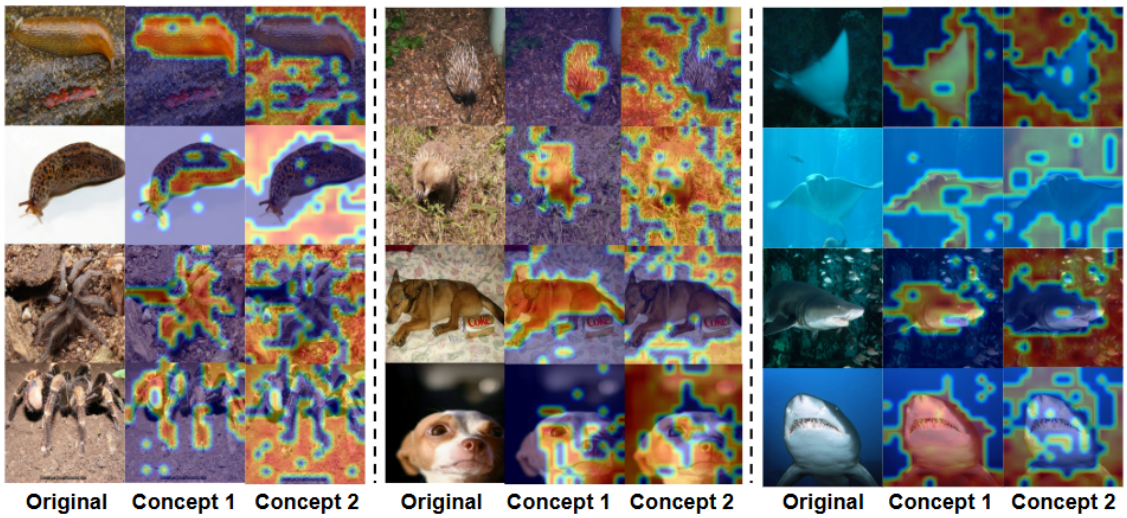


Figure A-3. **Semantic Disentanglement (ImageNet).** By enforcing geometric stability, Robust CCT autonomously separates stable foreground objects (Concept 1) from volatile background elements (Concept 2). This demonstrates that geometric regularization naturally induces semantic decomposition, preserving interpretability even without explicit concept labels.

Table A-8. **Quantitative Assessment of Global Structure.** Robust CCT demonstrates superior probabilistic calibration and clustering quality. Lower ECE and Brier scores indicate that the flattened geometry leads to confidence scores that are better aligned with actual accuracy.

Model	ECE (↓)	Brier (↓)	NLL (↓)	V-measure (↑)
Vanilla ViT	5e-4	0.18	0.50	0.92
Vanilla CCT	2e-4	0.15	0.40	0.94
Robust CCT	**1e-4**	**0.14**	**0.38**	**0.95**

this point causes a sharp decline in accuracy, dropping to 72.1% at the 25% mark. This temporal sensitivity offers an important insight into the learning dynamics of concept manifolds: geometric regularization is most effective during the *formative phase* of training, where it guides the initialization of the manifold topology toward a flat, robust configuration. However, once the global structure is established, prolonged noise injection acts as excessive regularization, preventing the model from fine-tuning its decision boundaries for precision. Thus, a "short burst" of geometric constraint is sufficient to permanently inoculate the model against instability.

I. Full Performance Comparison

For completeness, we provide the full performance breakdown for CUB-200-2011 and ImageNet-200 in Tables A-9, and A-10.

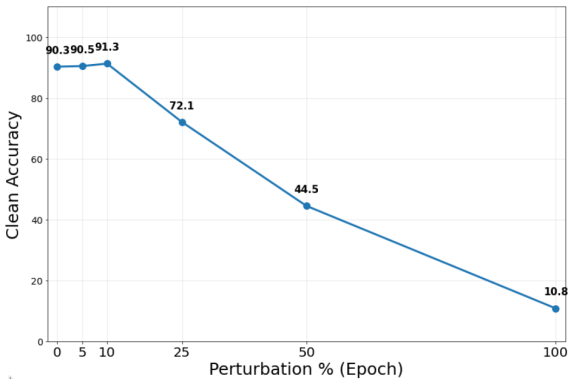


Figure A-4. **Effect of perturbation schedule on accuracy.** Applying perturbations for only the initial 10% of training yields optimal performance. This suggests that geometric regularization is crucial for the initial shaping of the manifold, but should be relaxed to allow for fine-grained convergence.

Table A-9. **Performance comparison on CUB-200-2011 (Full Data).** All baseline results are sourced from Hong et al. [43], except for Robust CCT. Our method achieves the highest accuracy.

Method	**Test Accuracy (%)**
Part R-CNN [95]	76.4
SPDA-CNN [93]	85.1
2-level attn. [88]	77.9
ProtoPNet [12]	84.8
ProtoPool [71]	87.6
ProtoPFormer [89]	84.9
ProtoConcept [32]	85.2
Deformable ProtoPnet [17]	86.5
B-CNN [51]	85.1
ST-CNN [46]	84.1
CEMM [92]	77.1
Vanilla CCT [43]	90.3 ± 0.1
Robust CCT (Ours)	$\mathbf{91.3 \pm 0.2}$

Table A-10. **Performance comparison on ImageNet-200 (Full Data).** Robust CCT outperforms all interpretable baselines. Note that CT's low performance is attributed to the lack of ground-truth concept supervision.

Model	**Test Accuracy (%)**
PIP-Net [61]	N/A
ProtoPFormer [89]	83.4 ± 2.2
ProtoPool [71]	76.5 ± 0.8
ProtoPNet [12]	77.7 ± 0.3
Deform-ProtoPNet [17]	76.1 ± 0.3
CT [69]	27.0 ± 0.2
BotCL [87]	83.0
Vanilla CCT [43]	83.7 ± 0.2
Robust CCT (Ours)	$\mathbf{85.9 \pm 0.1}$