# Multi-Granularity Concept Whitening for Neural Network Interpretability

Russell Barton
Duke University
russellba@zillowgroup.com

Viet Hung Le
Duke University
hung.le@duke.edu

Yunhong Shan
Duke University
islinashan@google.com

Alexander Katopodis
Duke University
alexkato29@gmail.com

Jon Donnelly
Duke University
jon.donnelly@duke.edu

Eric Chen
Duke University
eric.y.chen@duke.edu

Chaofan Chen
University of Maine
chaofan.chen@maine.edu

Cynthia Rudin
Duke University
cynthia@cs.duke.edu

## Abstract

*Interpretability remains a central challenge in deep learning, especially for fine-grained classification tasks. Concept Whitening (CW) represents a promising step towards interpretable deep learning: by aligning axes within a neural network's latent space with human-specified concepts, CW helps users understand how neural networks reason. However, CW leaves many axes unlabeled, undermining the interpretability of the latent space. We address this problem by introducing Multi-Granularity Concept Whitening (MGCW), a novel framework that reformulates CW to partition the latent space into hierarchical, high-level concept subspaces, each containing both labeled and free (unlabeled) axes that correspond to sub-concepts. This design ensures that emergent concepts stay localized within relevant semantic regions (high-level concepts), enhancing both structure and interpretability. Experiments on CUB-200-2011 and COCO/Places365 show that MGCW boosts concept purity and interpretability while preserving classification accuracy. Our findings establish MGCW as a powerful tool for learning structured, interpretable representations in deep networks with minimal performance trade-offs.*

## 1. Introduction

Deep neural networks have come to dominate predictive tasks across many domains. These models are accurate but tend to follow a largely opaque reasoning process. Recent work has sought to address this problem by building inherently interpretable extensions to deep neural networks that maintain predictive performance while adding interpretability. Concept Whitening (CW) [4] is a promising approach in this direction: it aligns individual axes in the latent space of a neural network with known, human-understandable concepts.

However, CW requires a set of finely labeled images for each concept of interest and as such cannot typically assign a concept to every axis in the latent space. The remaining axes are left unlabeled, and the network can entangle them arbitrarily. This significantly undermines interpretability, especially for fine-grained classification tasks such as those in the CUB-200-2011 dataset [27]. Moreover, CW leaves users without any way to integrate *partial* and *overlapping* concept annotations—for example, a user may have a set of "beak" images and a separate set of "hooked beak" images.

In this work, we propose Multi-Granularity Concept Whitening (MGCW), which addresses these issues by partitioning the latent space around high-level concepts and allowing each subspace to contain both labeled and unlabeled (free) sub-concepts. This partition ensures that each axis in the latent space is interpreted relative to a specific sub-concept or a known high-level concept region. Within each subspace, MGCW supports a winner-takes-all mechanism that allows unlabeled sub-concepts to emerge while remaining constrained to a meaningful high-level concept, improving interpretability without requiring dense annotations. Figure 1 provides an illustrative overview of this structure.

Across two benchmark datasets, we show that MGCW provides improved semantic consistency of concepts relative to prior work without substantially harming the accuracy of the original network. Moreover, the hierarchical nature of MGCW enables users to embed partial knowledge about the concepts in an image (*e.g.*, this contains *some* kind of beak). Finally, we demon-

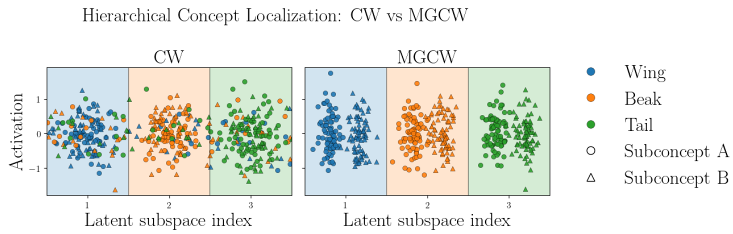



Figure 1. A conceptual overview of MGCW. CW aligns axes with high-level concepts but does not restrict where emergent sub-structures form, allowing activations to spill into unrelated semantic regions. MGCW partitions the latent space into concept-specific subspaces (colored bands), ensuring that fine-grained variations (marker shapes) remain confined within their designated high-level concept.

strate qualitatively that the concept axes learned by MGCW are semantically consistent. Code is available at https://github.com/rbarton124/MultiGranularityCW.

## 2. Related Work

Research on enhancing neural network interpretability has followed two main approaches: providing post-hoc explanations, and developing inherently interpretable models.

Post-hoc explanation methods include activation maximization [6, 19], perturbation-based approaches like Local Interpretable Model-agnostic Explanations (LIME) [23] and SHapley Additive exPlanations (SHAP) [17], saliency mapping [2, 3, 10, 24], and concept extraction approaches like Testing with Concept Activation Vectors (TCAV) [11]. However, each of these approaches can suffer from instability, produce unfaithful explanations, or suffer from correlations in the extracted concepts, leading to misleading interpretations of model behavior [1–4, 10, 12, 20, 24].

Inherently interpretable models overcome the faithfulness issues of post-hoc explanations by making the reasoning process of a model directly understandable. In Concept Whitening (CW), a module that de-correlates and normalizes the latent space is added to a neural network (NN), aligning individual axes with predefined concepts [4]. While not originally emphasized, this structure implicitly concentrates concept-specific information along single axes. In contrast to TCAV, which represents concepts as unconstrained directions, CW's axis-aligned formulation provides a more structured and localized representation of concept-specific information. CW has been applied to interpret models in high-stakes settings, particularly in healthcare [8, 21]. Dai et al. [5] proposed a hierarchical concept whitening (CW) module to clarify the latent hierarchical relationship between label concepts. Specifically, they added a second constraint to preserve parent-child relationships between concepts on top of the original CW constraint to align latent space axes with the corresponding concept dimensions. However, unlike our work, they do not consider "free sub-concepts" – the idea that not all images within a high-level concept are labeled with the maximum possible granularity – and therefore require an explicit fine-grained concept dataset for each axis, unlike MGCW. Concept bottleneck models (CBMs) are a related but distinct approach to improving model interpretability by computing human-specified concepts from images and using these concepts to form predictions, instead of aligning the latent space [13, 16, 25, 28, 30–32]. However, there is research showing that CBM's concepts may neither correspond to anything meaningful in the input space nor reliably base concept predictions on relevant input features [18, 22].

Our work builds on the original Concept Whitening [4] and Dai's hierarchical extension [5]. We generalize concept whitening to a wider range of concept relationships beyond parent-child hierarchies and extend it to discover unlabeled concepts alongside labeled ones, without the faithfulness issues associated with concept bottleneck models.

## 3. Multi-Granularity Concept Whitening

**Notation and Preliminaries** We consider an image classification dataset with $n$ samples $\mathcal{D} := \{(\mathbf{X}_i, y_i)\}_{i=1}^n$, where $\mathbf{X}_i \in \mathbb{R}^{3\times H\times W}$ is a 3-channel image and $y_i \in \{1, 2, \dots, C\}$ is the class label. Let $f : \mathbb{R}^{3\times H\times W} \to \mathbb{R}^C$ denote a deep neural network classifier. We add MGCW at a chosen intermediate layer of $f$: let $f_1 : \mathbb{R}^{3\times H\times W} \to \mathbb{R}^{D\times H'\times W'}$ denote the portion of $f$ before the target layer, and let $f_2 : \mathbb{R}^{D\times H'\times W'} \to \mathbb{R}^C$ denote the remainder, so that $f = f_2 \circ f_1$. For the auxiliary concept-alignment objective, as in [4], we additionally define a pooling mechanism $g : \mathbb{R}^{D\times H'\times W'} \to \mathbb{R}^D$ and pooled latent vectors $\mathbf{z}_i := g(f_1(\mathbf{X}_i))$. These pooled vectors are used only for concept alignment; the main classification objective continues to operate on the full spatial feature map $f_1(\mathbf{X}_i)$. Our goal is to structure the pooled latent space so that its axes are orthogonal, concept-aligned, and organized hierarchically.

Let $\mathcal{Q}_{\text{sub}} := \{1, 2, \dots, K_{\text{sub}}\}$ denote the set of $K_{\text{sub}}$ labeled sub-concepts of interest, and let $\mathcal{Q}_{\text{high}} := \{S_{k_{\text{high}}}\}_{k_{\text{high}}=1}^{K_{\text{high}}}$ be a collection of $K_{\text{high}}$ high-level concept subspaces such that each $k_{\text{sub}} \in \mathcal{Q}_{\text{sub}}$ belongs to exactly one $S_{k_{\text{high}}} \in \mathcal{Q}_{\text{high}}$. We

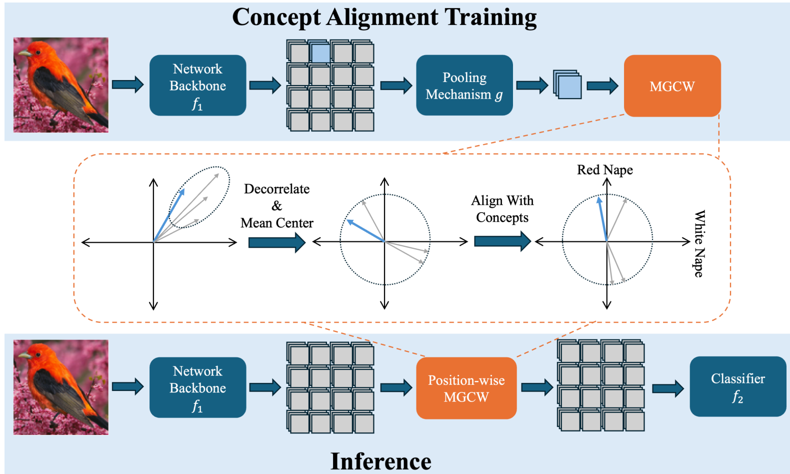



Figure 2. **MGCW training and inference.** MGCW replaces a batch norm block at an intermediate layer of a neural network. During training, the main classification loss updates both $f_1$ and $f_2$ using the full spatial feature map, while a pooling mechanism $g$ is used only in the concept-alignment step to produce a single $D$-dimensional vector per image for optimizing the rotation matrix $\mathbf{Q}$. During inference, the learned transformation is applied position-wise to every $D$-dimensional feature vector in the representation, yielding a $D \times H' \times W'$ transformed feature map, before the classifier head $f_2$.

do not require that $\mathcal{Q}_{\text{high}}$ is only a partition of the labeled sub-concepts: each set $S_{k_{\text{high}}}$ may also contain one or more *free sub-concepts*—axes associated with the high-level concept for which no labeled sub-concept exists. Thus, for a free sub-concept $j$, we have $j \notin \mathcal{Q}_{\text{sub}}$ but $j \in S_{k_{\text{high}}}$.

Intuitively, each set inside $\mathcal{Q}_{\text{high}}$ groups some number of sub-concepts into an associated high-level concept; for example, given sub-concepts indicating "long beak," "short beak," and "forked tail," the high-level concept "beak" may consist of "long beak," "short beak," and one or more free sub-concepts, while the "tail" concept consists of "forked tail" and a free sub-concept "unspecified tail type." We assume we are given an auxiliary dataset $\mathcal{D}^{\text{sub}}_{k_{\text{sub}}} := \{\mathbf{X}_i^{(k_{\text{sub}})}\}_{i=1}^{n_{k_{\text{sub}}}}$ of $n_{k_{\text{sub}}}$ images for each labeled sub-concept $k_{\text{sub}} \in \mathcal{Q}_{\text{sub}}$, and similarly a set $\mathcal{D}^{\text{high}}_{k_{\text{high}}} := \{\mathbf{X}_i^{(k_{\text{high}})}\}_{i=1}^{n_{k_{\text{high}}}}$ of $n_{k_{\text{high}}}$ images for each high-level concept. The notion of free sub-concepts allows us to leverage images with partial concept labels; for instance, an image that is known to contain some kind of beak without a specific beak type may appear in the high-level "beak" concept dataset, but not in any labeled sub-concept dataset.

We optimize our model $g \circ f_1$ to align one concept with each axis through two transformations. First, as in [4], we perform a whitening operation $\psi(\mathbf{z}) := \mathbf{W}(\mathbf{z} - \mu)$, where $\mathbf{W} \in \mathbb{R}^{D \times D}$ is a whitening matrix defined by $\mathbf{W}^\top \mathbf{W} = \mathbf{\Sigma}^{-1}$ for covariance matrix $\mathbf{\Sigma}$ and $\mu := \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i$ is the mean embedding across our training set. We then apply a learned rotation matrix $\mathbf{Q} \in \mathbb{R}^{D \times D}$ to $\psi(\mathbf{z})$, with the ultimate goal of producing axis-aligned concepts. We denote the complete Multi-Granularity Concept Whitening layer by $\Psi(\mathbf{z}) := \mathbf{Q}\psi(\mathbf{z})$.

During inference, $\Psi$ is applied to each of the $H' \times W'$ feature vectors in the representation space of $f_1$. Letting $\Psi_{\text{inference}} : \mathbb{R}^{D \times H' \times W'} \to \mathbb{R}^{D \times H' \times W'}$ denote this position-wise application of $\Psi$, the inference process of a network with MGCW is $f_2 \circ \Psi_{\text{inference}} \circ f_1$. Figure 2 illustrates this.

**Learning Hierarchical Concept Alignment** We aim to solve the following optimization problem:

$$\max_{\mathbf{q}_1,\dots,\mathbf{q}_k} \left[\mathcal{L}_{CW_{\text{sub}}} + \lambda \mathcal{L}_{CW_{\text{high}}}\right] \text{ subject to } \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}, \quad (1)$$

where

$$\mathcal{L}_{CW_{\text{sub}}} = \sum_{k_{\text{sub}} \in \mathcal{Q}_{\text{sub}}} \frac{1}{n_{k_{\text{sub}}}} \sum_{\mathbf{X}_i^{(k_{\text{sub}})} \in \mathcal{D}^{\text{sub}}_{k_{\text{sub}}}} \mathbf{q}_{k_{\text{sub}}}^\top \psi\left(g \circ f_1(\mathbf{X}_i^{(k_{\text{sub}})})\right)$$

$$\mathcal{L}_{CW_{\text{high}}} = \sum_{\substack{S \in \mathcal{Q}_{\text{high}} \\ k_{\text{high}} \in S : k_{\text{high}} \notin \mathcal{Q}_{\text{sub}}}} \frac{1}{n_{k_{\text{high}}}} \sum_{\mathbf{X}_i^{(k_{\text{high}})} \in \mathcal{D}^{\text{high}}_{k_{\text{high}}}} \max_{j \in S} \mathbf{q}_j^\top \psi\left(g \circ f_1(\mathbf{X}_i^{(k_{\text{high}})})\right)$$

for some hyperparameter value $\lambda \in \mathbb{R}^+$. The first term, $\mathcal{L}_{CW_{\text{sub}}}$, sums over the concept dataset for each labeled sub-concept and encourages each sample to align with its designated axis. The second term, $\mathcal{L}_{CW_{\text{high}}}$, iterates over samples associated with a high-level concept but **not** a specific labeled sub-concept and encourages each sample to align with the most responsive axis in the corresponding subspace. This enables free sub-concepts to emerge naturally within a specified high-level concept region, regulated by $\lambda$.

In practice, we aim to solve (1) while maintaining predictive performance via an alternating optimization procedure. In each training epoch, we optimize $f_1$ and $f_2$ to minimize cross entropy over $\mathcal{D}$ using batched stochastic gradient descent.

After every $N_{\text{batches}}$ steps of gradient descent, we perform a *concept alignment step* where we directly minimize the loss term $\left(-\mathcal{L}_{CW_{\text{sub}}} - \lambda\mathcal{L}_{CW_{\text{high}}}\right)$ with respect to $\mathbf{Q}$, subject to $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$. As in [4], we solve this constrained optimization problem using the Cayley transform. For each optimization step $t$, we set

$$\mathbf{Q}^{(t+1)} := \left(\mathbf{I} + \tfrac{\eta}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} - \tfrac{\eta}{2}\mathbf{A}\right)\mathbf{Q}^{(t)},$$

where $\eta$ is a learning rate and $\mathbf{A} := \mathbf{G}(\mathbf{Q}^{(t)})^\top - \mathbf{Q}^{(t)}\mathbf{G}^\top$, where $\mathbf{G}$ denotes the gradient of $\mathcal{L}_{CW_{\text{sub}}} + \lambda\mathcal{L}_{CW_{\text{high}}}$ with respect to $\mathbf{Q}$.

**Constructing Concept Datasets** In practice, the images from each concept dataset are unlikely to contain a singular concept. For example, a natural image of a bird will often contain the beak, tail, and feet of a bird, which may all be concepts of interest. This complicates the learning process.

To mitigate this issue, we apply bounding-box annotations (or relevant masks) to crop down to only the image region corresponding to the target concept, and resize this crop to the size of a standard input image. In doing so, the network sees images that predominantly contain the part-of-interest (*e.g.*, a beak region) for each concept. All reported experiments use this crop-and-resize preprocessing. We also tested redaction-style variants in preliminary experiments, but they produced weaker purity on a representative model, so we focus on cropped concept images in this work.

## 4. Experiments

We evaluate MGCW on two benchmark datasets: CUB-200-2011 [27] and Places365 [33]. For CUB-200, we use the provided part annotations as concepts: for example, the "beak" high-level concept contains "hooked beak" and "short beak" as sub-concepts. For Places365, we use COCO [15] object classes as concepts. Both concept datasets provide spatially localized annotations, allowing us to crop each concept image to the relevant region (Section 3). Because CW is sensitive to the number of concepts, we experiment with two concept-set sizes per dataset. For CUB-200, the *Small* set contains 2 high-level concepts (*eye* and *nape*) subdivided into a total of 9 sub-concepts; the *Large* set contains 9 high-level concepts (*back*, *beak*, *belly*, *eye*, *general*, *leg*, *nape*, *tail*, and *throat*) and 36 sub-concepts. For COCO, the *Small* set contains 2 high-level concepts with 20 sub-concepts; the *Large* set contains 4 high-level concepts and

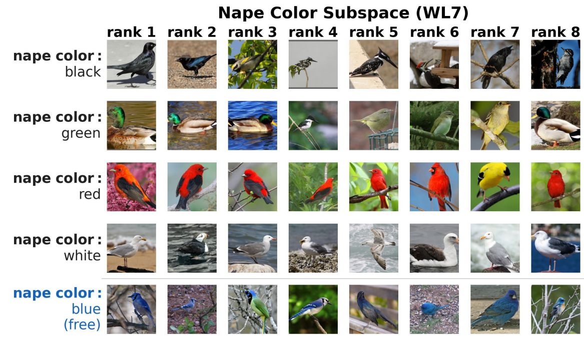



Figure 3. **Nape color subspace at WL7: labeled and free axes side by side.** Each row shows the top-8 most-activated images for one axis within the *nape color* high-level subspace. Rows above the divider correspond to labeled sub-concepts (e.g., *nape color: white*, *nape color: red*); the row below is the free (unlabeled) axis. The free axis consistently retrieves birds with coherent nape coloring not captured by any labeled sub-concept, demonstrating that MGCW's winner-takes-all mechanism localizes emergent structure within the correct semantic region rather than letting it drift to unrelated axes.

38 sub-concepts, organized by COCO supercategory groupings such as *animals*, *food*, and *sports equipment*.

We employed ImageNet pretrained VGG-16 [26], DenseNet-161 [9], ResNet-18 and ResNet-50 [7] models, replacing various batch norm layers with MGCW (termed "Whitened Layers" or WL). Each model variant was trained for 200 epochs, with checkpoints selected based on validation accuracy. Each whitened layer result corresponds to a separately trained model with MGCW inserted at that single layer.

**Free Sub-Concepts Find Consistent Patterns** We first evaluate qualitatively whether "free sub-concepts" – axes that have a high-level concept label like "beak," but not a fine-grained one – learn consistent semantic concepts. We consider the high-level *nape color* concept, in which we provide sub-concept labels for black, green, red, and white nape colors. We include all of these images in the high-level *nape color* concept dataset, but also include images of birds with blue napes. We purposely do not provide a sub-concept label for blue napes. Thus, the question is: can MGCW isolate *nape color: blue* images to a single free axis in the *nape color* subspace *without* specific labels?

Figure 3 presents the top-8 most highly activated images for all axes within the "nape color" high-level subspace at WL7. As expected, each labeled axis retrieves images whose birds visually exhibit the corresponding attribute: the *nape color: white* row is populated entirely by birds with pale or white napes, and the *nape color: red* row primarily surfaces birds with vivid red nape patches. The free axis,

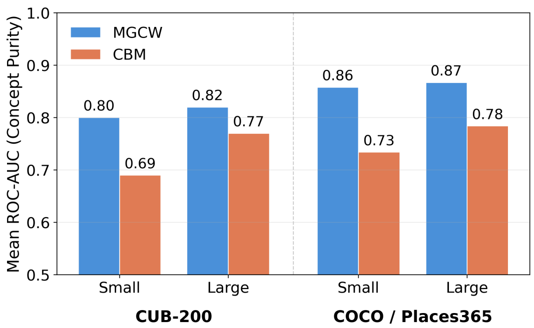



Figure 4. **Concept purity: MGCW vs. CBM.** Mean ROC-AUC purity for MGCW and a Concept Bottleneck Model (CBM) on Small and Large concept sets for CUB-200 (a) and COCO/Places365 (b). For both models, the seventh (CUB) or eighth (Places365) layer of a ResNet-18 is modified with the respective module. MGCW consistently achieves higher purity than CBM across both datasets and concept-set sizes.

separated by a horizontal divider, also produces a visually coherent retrieval — the top images share a consistent coloring pattern not represented by any explicitly labeled sub-concept. This demonstrates that MGCW's hierarchical partitioning successfully isolated the *nape color: blue* images to the free axis without an explicit labeled dataset for this concept. This analysis is also useful for identifying model failures: in the *nape color: green* row, several images depict birds with green foliage in the background rather than a green nape, revealing annotation noise that limits purity.

The framework also permits multiple free sub-concepts within the same high-level concept subspace. In additional CUB-200 Large runs with multiple free sub-concepts per subspace, training remained stable across WL1–WL8 (test top-1 accuracy 73.79–75.65), and the strongest hierarchical metrics again appeared at later layers (mean hierarchical AUC 0.838 and masked Hit@3 0.613 at WL7). Since these multi-free runs were exploratory rather than the main focus of the paper, we use them only as supporting evidence that MGCW can accommodate more than one free sub-concept within a high-level concept.

We also examine how concept axis quality varies with the layer at which MGCW is inserted. Figure 8 compares independently trained models across insertion positions: models with MGCW at early layers produce concept axes that respond to varied, loosely related images, while models with MGCW at later layers yield tighter, semantically consistent clusters. Critically, the free axis in the later-layer models converges to a coherent nape-color pattern visually distinct from all labeled sub-concepts, demonstrating that winner-takes-all alignment not only confines free axes to the correct part region but also drives genuine concept specialization when placed in representationally rich layers. We examine this layer-placement effect in greater depth in Section 4.

**MGCW Improves Concept Purity** Next, we study how well each concept axis represents the desired concept, which we refer to as "purity." Intuitively, we would like images $\mathbf{X}_i^{k_\ell} \in \mathcal{D}_{k_\ell}$ containing a concept $k_\ell$ to have a higher activation along the axis assigned to $k_\ell$ than other images. We use AUROC to measure how well this property holds; we consider samples containing $k_\ell$ "positive" samples, and consider all other samples negative. We treat the activation along axis $k_\ell$ for a given image as the "prediction" for purity AUROC. We compare the purity of MGCW to that of a concept bottleneck model using the same backbone and concepts; for this model, we treat the logit associated with the given concept as the "prediction" for purity AUROC.

Figure 4 compares the mean purity across all concepts for MGCW against a Concept Bottleneck Model [13] on both CUB-200 and COCO/Places365. MGCW substantially outperforms CBM on both datasets and both concept-set sizes. On CUB-200, MGCW improves purity from 0.69 to 0.80 on the small concept set and from 0.77 to 0.82 on the large set. On COCO/Places365, MGCW achieves 0.86 purity on the small set versus 0.73 for CBM, and 0.87 versus 0.78 on the large set. These consistent gains suggest that structured subspace alignment provides more reliable concept representations than dense bottleneck supervision.

We also study how purity varies with the layer at which MGCW is inserted. Each point in Figure 5 represents an independently trained model with MGCW (or CW as the baseline) placed at a single layer. For base CW, we provided the sub-concepts of QCW as the concept set. MGCW produces similar or better purity than CW at every insertion point. The largest gains appear when the module is placed at the first layer, especially on Places365: shallow feature spaces encode shared sub-concept patterns where CW's global alignment is most prone to cross-concept leakage, and MGCW's explicit subspace partitioning prevents this entanglement. Both methods achieve their strongest purity when inserted at deeper layers, reflecting the greater semantic specificity of deeper CNN feature spaces.

The two datasets reveal an informative contrast. On CUB-200, MGCW maintains a substantial purity advantage across all insertion points (e.g., 0.822 vs. 0.670 for CW at WL7 on the Large set), while on COCO/Places365 the gap narrows and largely disappears at deeper layers. This reflects each dataset's concept structure: CUB's high-level concepts group anatomically adjacent body regions (nape, back, belly, throat) whose sub-concepts easily bleed across boundaries, whereas COCO's high-level concepts group categorically distinct object types (animals, food, sports equipment) that flat CW already separates well. The

| Dataset | Model | Opt WL | M-AUC | B-AUC |
|---|---|---|---|---|
| CUB | RN-18 (S) | 7 | **0.834** | 0.691 |
| | RN-18 (L) | 7 | **0.865** | 0.616 |
| | RN-50 (L) | 11 | **0.871** | 0.645 |
| Places365 | RN-18 (S) | 8 | **0.861** | 0.759 |
| | RN-18 (L) | 8 | **0.887** | 0.758 |
| | RN-50 (L) | 16 | **0.901** | 0.787 |

Table 1. **Mean concept purity within a high-level concept subspace ("Masked AUC") versus across the entire representation space ("Baseline AUC").** Each M-AUC and B-AUC value is averaged over sub-concepts for the model selected at the optimal whitened layer (Opt WL) for that dataset/model setting. Across both datasets and concept sets, purity is substantially higher when restricting evaluation to the corresponding high-level concept subspace. RN = ResNet, (S) = small concept set, (L) = large concept set. M-AUC = masked AUC, B-AUC = baseline AUC.

masked (concept-conditional) AUC results in Table 1 corroborate this, with a $\sim$40% within-subspace improvement on CUB-200 Large versus only $\sim$17% on Places365 Large. Taken together, these patterns confirm that MGCW provides the greatest benefit precisely where cross-concept leakage is highest—and that its hierarchical partitioning is achieving its intended effect.

**MGCW Supports Partial Knowledge** In practice, users may know partial information about an image – for example, they may know that the image contains a bird's wings but not its beak, while being unable to describe the wing type precisely. The hierarchical nature of MGCW allows us to evaluate how "pure" wing concepts are when compared only to other wing concepts. For each sub-concept, we compute a concept-level AUC and report the mean across sub-concepts. Baseline AUC uses the full representation and all other concepts as negatives, while "masked AUC" restricts evaluation to the corresponding high-level concept subspace, isolating fine-grained distinctions among sibling sub-concepts. This evaluates the model's ability to reason within a known semantic category (e.g., wings) rather than across all possible concepts.

Table 1 presents the results of this analysis. Providing partial high-level information to MGCW substantially improves concept purity in all cases. On CUB-200 Large with ResNet-18, we see a roughly 40% increase in AUC when reasoning within one high-level concept rather than across all of them. Similarly, on Places365 Large with ResNet-18, we observe a roughly 20% improvement with masking.

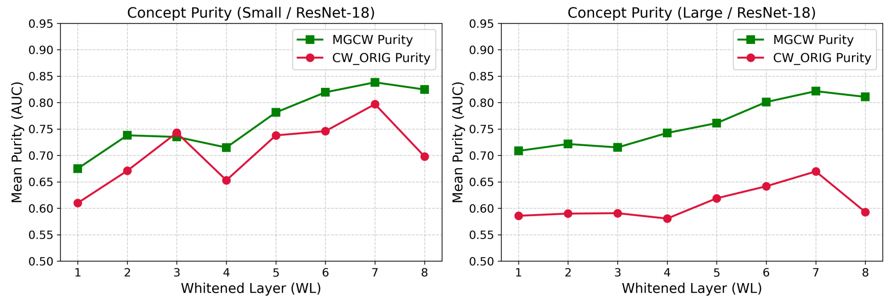
We note that this analysis is not well-defined under the original CW formulation. While CW aligns individual axes with labeled concepts, it does not partition the latent space into semantically scoped subspaces. As a result, any within-concept masking would be applied only at evaluation time and would not reflect structural constraints in the learned representation. In contrast, MGCW explicitly allocates axes to high-level concept subspaces, making masked evaluation a direct probe of the model's geometric organization rather than a post-hoc filtering procedure.

**MGCW Maintains Accuracy** We next ask whether these interpretability gains come at a cost to accuracy. We train a separate ResNet-18 model for each insertion point (WL1–WL8) on both CUB-200 and Places365, using both concept sets, and compare each to the pretrained baseline.

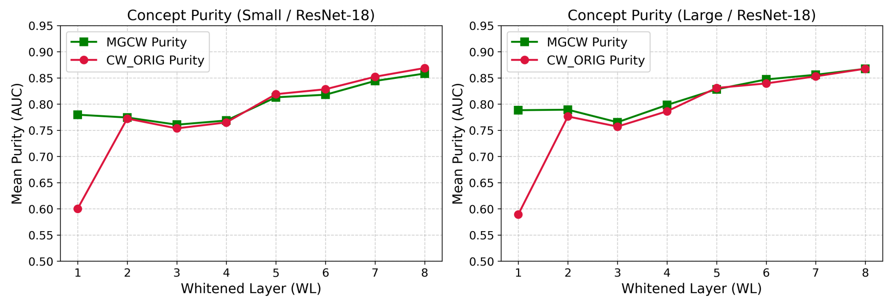
Figure 6 shows the accuracy of each independently trained model. MGCW typically preserves accuracy reasonably well, with most reductions below 2% and all reductions below 5% across the settings studied, although some insertion points do incur noticeable degradation. Taken together, these results suggest that MGCW's interpretability benefits can often be obtained at a manageable predictive cost, but only when the insertion layer is chosen carefully.

Accuracy varies with insertion point: models with MGCW at mid-depth layers (WL3–WL4) maintained the highest accuracy, while certain deeper insertions (WL5 for CUB-200, WL8 for Places365) led to larger degradation. The optimal insertion point differs between datasets, suggesting that MGCW placement should be tuned per domain. The Small concept set occasionally matches or exceeds the accuracy of the Large set, suggesting that annotation quality and concept-set scale influence how well the backbone adapts to the MGCW constraint. The deepest layer is therefore not always optimal: although late layers are often the most semantically specific, they are also the most specialized for the final prediction task, so imposing concept alignment too close to the classifier can reduce flexibility needed for downstream classification. To contextualize MGCW's accuracy impact, we also evaluated the impact of adding MGCW to four standard architectures (ResNet-18, ResNet-50, DenseNet-161, VGG16) fine tuned for each of CUB-200 and Places365. In each case, we add MGCW at the layer that yielded the best validation accuracy for that architecture. Figure 7 presents the result of this analysis. We found that, in ResNet-50 and DenseNet-161, adding MGCW slightly *increased* the overall accuracy of the model for CUB-200. In ResNet-18 and VGG-16, and in all models on Places365, we observed a slight ($< 2\%$) decrease in accuracy after adding MGCW. Across architectures, MGCW does not substantially harm predictive accuracy when inserted at an appropriate layer.

**MGCW Semantics Are Better at Deeper Insertion Points** Finally, we examine how the semantic quality of




(a) CUB-200 concepts




(b) COCO/Places365 concepts

Figure 5. **Concept purity by MGCW insertion layer (ResNet-18).** Each point is an independently trained model with MGCW (green squares) or CW (crimson circles) inserted at that single layer. Top row: CUB-200. Bottom row: COCO/Places365. MGCW consistently matches or exceeds CW purity across all insertion points, with the largest gains when the module is placed at shallow layers where global alignment is most prone to cross-concept leakage.

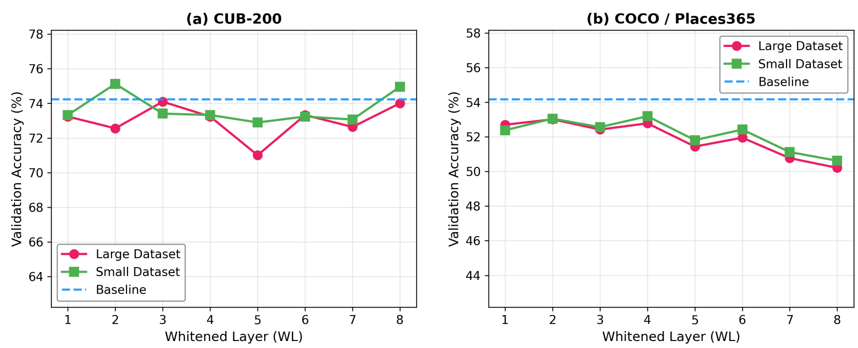



Figure 6. **Classification accuracy by MGCW insertion layer (ResNet-18).** Each point is an independently trained model with MGCW inserted at that single layer. Validation accuracy for Small and Large concept sets versus the pretrained baseline, on CUB-200 (left) and Places365 (right). Accuracy is broadly maintained across insertion points, with mid-depth placements achieving the best trade-off between interpretability and accuracy.

MGCW concept axes depends on the layer at which the module is inserted. Figure 8 presents grids for CUB-200 and COCO sub-concepts, where each column is a separately trained model with MGCW at that single layer. In models where MGCW is placed at WL1–WL2, concept axes lack specificity and are often confounded by background context, reflecting the limited semantic structure in shallow feature maps. In models with MGCW at WL5–WL6, color-based concept axes consistently retrieve matching images. By WL7–WL8, even structurally demanding concept axes such as *tail shape: forked tail* (CUB) and *sports: tennis racket* (COCO) produce specific, semantically tight exemplars. This pattern mirrors the natural hierarchy of CNN feature spaces: shallower layers encode texture and global color, middle layers develop part-selective structure, and later layers carry fine-grained attribute information suitable for concept alignment. At the same time, the final layer is not always the best insertion point, because the top of the network must still preserve flexibility for the classification objective; the best trade-off therefore often occurs slightly

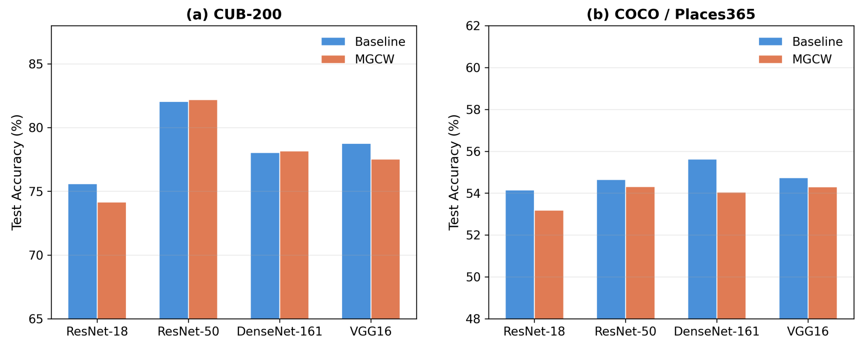
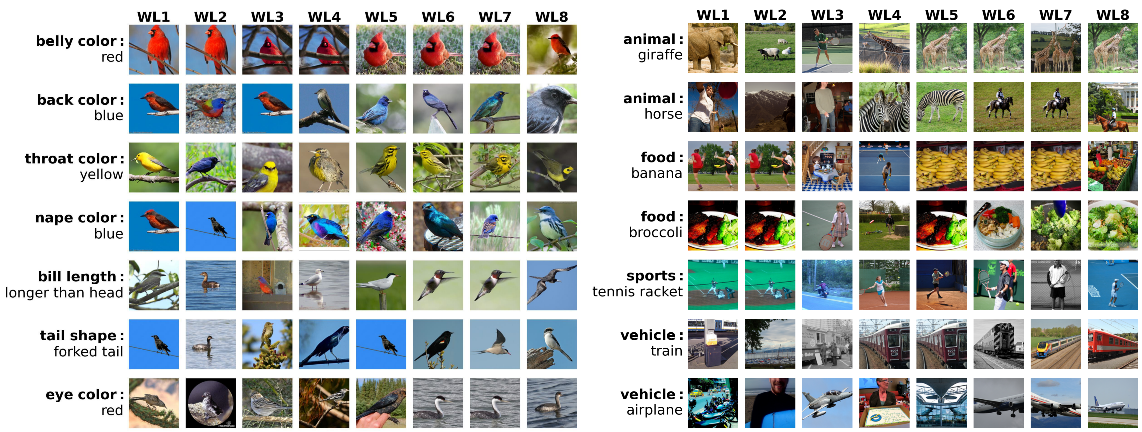



Figure 7. **MGCW vs. baseline accuracy across architectures.** Test accuracy of MGCW (at optimal WL) versus pretrained baselines for ResNet-18, ResNet-50, DenseNet-161, and VGG16 on CUB-200 (a) and COCO/Places365 (b).




(a) CUB-200

(b) COCO/Places365

Figure 8. **Concept axis quality by MGCW insertion layer (Large dataset, ResNet-18).** Each column corresponds to an independently trained model with MGCW inserted at that single layer (WL1–WL8). Each row is a sub-concept; each cell shows the image that maximally activates the corresponding axis in that model. Models with MGCW at shallower layers respond to whole-image color and background cues, while models with MGCW at deeper layers isolate the specific part-attribute named by the sub-concept. Concepts requiring fine-grained spatial discrimination (e.g., tail shape) require deeper insertion to achieve specific retrievals than coarser color attributes.

before the final block.

## 5. Conclusion

We introduced Multi-Granularity Concept Whitening (MGCW), an extension of Concept Whitening [4] that partitions the latent space into hierarchical subspaces containing both labeled and free axes. By explicitly structuring the geometry of the representation space, MGCW supports concepts at different levels of granularity and can identify fine-grained concepts even without exhaustive fine-grained labels. Across the CUB-200 [27] and COCO/Places365 [15, 33] datasets, MGCW improves concept purity relative to standard CW and concept bottlenecks while enabling masked AUC evaluation under partial knowledge. MGCW suggests effective within-subspace separation among fine-grained sibling concepts. These interpretability gains come at minimal cost: with appropriate layer placement, MGCW maintains classification accuracy close to standard pretrained baselines. Our results show that interpretability benefits from not only axis alignment but also from structured constraints on latent geometry itself.

Like CW, MGCW modifies only the rotation matrix during concept alignment while leaving the backbone frozen. To achieve stronger concept separation, future work could jointly optimize the backbone parameters with respect to the concept alignment loss. Additionally, MGCW currently relies on predefined high-level partitions; learning subspace structure jointly with axis alignment would move MGCW toward automatic high-level concept discovery. Future work could also adapt MGCW beyond convolutional backbones, such as Vision Transformers [29] and multimodal foundation models [14].

## Acknowledgment

We gratefully acknowledge financial support from the National Institutes of Health/NIDA under grant number 5R01-DA054994, and from the National Science Foundation under grant numbers OIA-2218063 and IIS-2442039. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under grant number DGE 213975.

## References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Proceedings of Conference on Advances in Neural Information Processing Systems*, pages 9505–9515, 2018. 2

[2] Julius Adebayo, Michael Muelly, Harold Abelson, and Been Kim. Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *International conference on learning representations*, 2022. 2

[3] Revoti Prasad Bora, Philipp Terhörst, Raymond Veldhuis, Raghavendra Ramachandra, and Kiran Raja. Slice: Stabilized lime for consistent explanations for image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10988–10996, 2024. 2

[4] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. 1, 2, 3, 4, 8

[5] Haixing Dai, Lu Zhang, Lin Zhao, Zihao Wu, Zhengliang Liu, David Liu, Xiaowei Yu, Yanjun Lyu, Changying Li, Ninghao Liu, Tianming Liu, and Dajiang Zhu. Hierarchical semantic tree concept whitening for interpretable image classification, 2023. 2

[6] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459/. 4

[8] Junlin Hou, Jilan Xu, and Hao Chen. Concept-attention whitening for interpretable skin lesion diagnosis. arXiv preprint arXiv:2404.05997, 2024. Submitted April 9, 2024; revised November 6, 2024. 2

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, July 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.243. URL https://ieeexplore.ieee.org/document/8099726/. 4

[10] Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, 150:228–234, 2021. 2

[11] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 2

[12] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 267–280, 2019. 2

[13] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 2, 5

[14] Meir Yossef Levi and Guy Gilboa. The double-ellipsoid geometry of clip. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. 8

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4, 8

[16] Nan Liu, Yilun Du, Shuang Li, Joshua B. Tenenbaum, and Antonio Torralba. Unsupervised compositional concepts discovery with text-to-image generative models, 2023. URL https://arxiv.org/abs/2306.05357. 2

[17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 2

[18] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? *arXiv preprint arXiv:2105.04289*, 2021. 2

[19] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep

generator networks. *Advances in neural information processing systems*, 29, 2016. 2

[20] Angus Nicolson, Lisa Schut, Alison Noble, and Yarin Gal. Explaining explainability: Recommendations for effective use of concept activation vectors. *Transactions on Machine Learning Research*, 2025. 2

[21] Michela Proietti, Alessio Ragno, Biagio La Rosa, Rino Ragno, and Roberto Capobianco. Explainable AI in drug discovery: self-interpretable graph neural network for molecular property prediction using concept whitening. *Machine Learning*, 113(4):2013–2044, October 2024. doi: 10.1007/s10994-023-06369-y. 2

[22] Naveen Raman, Mateo Espinosa Zarlenga, Juyeon Heo, and Mateja Jamnik. Do concept bottleneck models respect localities? *arXiv preprint arXiv:2401.01259*, 2024. 2

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 2

[24] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019. 2

[25] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. 2

[26] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. 4

[27] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1, 4, 8

[28] Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification, 2023. URL https://arxiv.org/abs/2304.10131. 2

[29] Yuzhu Wang, Manni Duan, and Shu Kong. Attention to the burstiness in visual prompt tuning! In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 8

[30] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification, 2023. URL https://arxiv.org/abs/2211.11158. 2

[31] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models, 2023. URL https://arxiv.org/abs/2205.15480.

[32] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off, 2022. URL https://arxiv.org/abs/2209.09056. 2

[33] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6): 1452–1464, 2017. 4, 8