PACE: PREFERENCE-AWARE EARLY EXITING FOR CONCEPT BOTTLENECK MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

While concept bottleneck models (CBMs) offer concept-level explanations of model predictions, modern CBMs typically rely on computationally intensive vision-language backbones to construct and align their concepts. This makes efficient deployment more difficult, especially when inference must preserve informative concept-level evidence rather than produce only class labels. To address this problem, we present *PACE*, a preference-aware multi-exit framework that supports early exiting in CBMs. Specifically, PACE keeps the visual backbone fixed and trains additional concept exits at intermediate depths to reduce inference cost while preserving the intermediate concept bottlenecks. Building on our core training mechanism, *Concept-Preference Guided Distillation* (CPGD), PACE addresses the difficulty of learning shallow concepts by exploiting preferences regarding concept depth. Instead of forcing every shallow concept to imitate the ultimate bottleneck representation, CPGD constructs concept-specific teachers at multiple depths and uses fusion-confidence-weighted supervision to better train shallow exits. At deployment, PACE introduces *Readiness-Calibrated MaxProb* (RCMax), a lightweight decision rule that retains MaxProb as the primary early-exit criterion while calibrating exit decisions based on key-concept support extracted from the current bottleneck. Comprehensive experiments on well-known benchmarks show that PACE achieves a superior balance between accuracy and efficiency compared to state-of-the-art (SOTA) baselines, with speedups ranging from $1.73\times$ to $2.48\times$.

1 INTRODUCTION

Although deep visual recognition models have achieved state-of-the-art (SOTA) performance in various tasks, their inherent black-box nature constrains their deployment in safety-critical scenarios, such as medical image diagnosis [29, 8], environmental perception for autonomous driving [18, 5], and high-precision industrial defect inspection [16]. As a promising interpretable visual recognition model, the Concept Bottleneck Model (CBM) [13] links visual representations to category predictions via explicit concepts, providing human-readable intermediate evidence for recognition. This concept-driven interpretable architecture makes CBM well-suited for the aforementioned safety-critical scenarios, where transparency, auditability, and intervention capabilities [23] are crucial. Due to rich open-vocabulary priors encoded in pre-trained Vision-Language Models (VLMs), recent CBM approaches [17, 36, 26, 22] increasingly rely on VLMs such as CLIP [21] and OpenCLIP [3] to construct, align, or filter concepts, reducing reliance on manual concept annotation and improving the scalability of concept spaces. Despite these critical advances, this VLM-driven paradigm also makes CBMs highly dependent on computationally expensive ViT-based vision–language backbones. Considering that the vast majority of safety-critical scenarios operate on resource-constrained devices with strict latency requirements, the practical deployment of CBMs requires a significant reduction in computational cost, while still fully preserving informative and actionable concept-level evidence.

To meet the strict efficiency requirements of VLM-based CBM deployment, early exit methods [28, 12, 41, 35] offer a promising paradigm for accelerating inference in resource-constrained environments. These methods equip networks with intermediate exits, enabling computationally easy samples to be processed at shallower layers while directing more complex instances to deeper ones. This adaptive inference mechanism has been shown to achieve a favorable accuracy-efficiency trade-off on standard vision benchmarks. However, a significant limitation emerges when these standard early-exit methods are applied to CBMs. Existing methods primarily treat intermediate exits as early classifiers and make routing decisions based exclusively on class-level signals, such as confidence, margin, entropy, or

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.

prediction stability [41, 7, 1, 10]. Although this design paradigm has proven effective for traditional black-box models, it fails to account for the unique architectural properties of CBMs. Within a multi-exit CBM architecture, an intermediate exit operates not merely as an early-stage classifier, but more fundamentally as an intermediate concept bottleneck. This bottleneck layer is critical to preserving both the model's interpretability and intervenability. Consequently, introducing early exits to CBMs induces a critical requirement typically unaddressed by standard methods: intermediate exits must not only accelerate predictions but also preserve high-quality, usable concept-level evidence to support those predictions. Failure to incorporate concept-level criteria compromises the intrinsic interpretability that defines the CBM framework.

We identify two key challenges in achieving this objective. **Challenge 1: Shallow exits often struggle to form usable concept bottlenecks.** In standard CBMs, the link from visual features to concepts is typically realized using either a linear probe or a small MLP. Near the final layer, visual embeddings already capture substantial semantic information, so such simple readout heads tend to perform well. By contrast, features in early ViT layers are more spatially localized and less semantically mature, making it difficult for basic readouts to reliably recover semantic concepts from these intermediate activations. In addition, end-to-end training of multi-exit architectures can create optimization interference between shallow and deep exits. Losses from early exits can reshape the shared backbone in ways that degrade deeper representations, ultimately harming late-exit accuracy or lowering the quality of the final concept bottleneck. To eliminate this confounding factor, we use a decoupled frozen-backbone design: the visual backbone remains fixed, the shared concept-to-class readout is trained separately and then fixed, and intermediate concept exits are trained to recover bottlenecks compatible with this readout. This ensures that the final CBM endpoint is not influenced by shallow-exit losses and centers the challenge on constructing effective shallow concept bottlenecks. **Challenge 2: Shallow concept bottlenecks require appropriate training supervision.** A typical approach in multi-exit architectures is to distill shallow-exit predictions from the final exit [11, 39, 4]. In CBMs, this amounts to assuming that the deepest concept bottleneck serves as the best universal teacher for all shallower concepts. However, this is not always true, as evidence for different concepts emerges gradually with increasing depth. As shown in Figure 1, group-level readiness varies across semantic concept groups, and image-level examples further show that different concepts in the same image can peak at different exits. For example, low-level visual properties such as *texture* or *tail* may already be reliably inferred at early exits, whereas more abstract or fine-grained concepts such as *pattern markings* often stabilize only at deeper exits. Uniformly forcing all shallow concepts to match the final bottleneck ignores depth-specific preferences and can reduce the usefulness of shallow representations. *Therefore, a key challenge in CBM-based designs is how to fully exploit the inherent advantages of early exit to accelerate inference processes while preserving both prediction accuracy and meaningful concept-level evidence.*

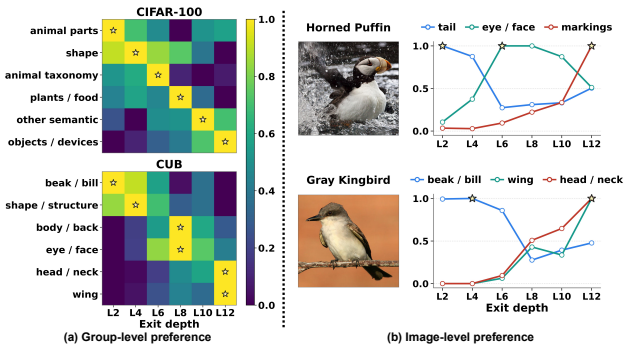


Figure 1: A motivating example. **(a)** Group-level readiness varies across exit depths. **(b)** Image-level examples show different concepts peaking at different exits. Stars mark preferred exits. These observations motivate preference-aware supervision for shallow concept bottlenecks.

To tackle these challenges, we present *PACE*, a preference-aware early-exit framework for CBMs. PACE is motivated by the observation that concept evidence emerges progressively across depths and that different concepts can exhibit different depth preferences. Rather than using intermediate exits solely as class predictors, PACE leverages concept-depth preferences to enhance shallow bottleneck recovery and relies on concept-level support to determine whether an intermediate bottleneck is sufficiently mature for early prediction. This paper makes the following four major contributions:

- We formulate CBM early exiting as concept-supported adaptive inference, evaluating intermediate exits not only by class confidence but also by the usability of their concept bottlenecks.

- We propose CPGD, a concept-preference-guided training method that builds concept-wise, multi-depth teachers and uses fusion-confidence-weighted supervision to improve shallow bottlenecks.
- We introduce RCMax, a lightweight readiness-calibrated exit rule that uses key-concept support from the current bottleneck to calibrate MaxProb-based early exits without extra backbone compute.
- We conduct comprehensive experiments on five standard recognition benchmarks under a shared multi-exit CBM setup, showing that PACE achieves a stronger practical accuracy-efficiency trade-off than adapted early-exit baselines, while delivering cumulative-MAC speedups of $1.73\times$-$2.48\times$.

2 Related Work

CBMs and Multi-layer Concepts. CBMs [13] make predictions via an explicit intermediate concept layer, improving interpretability and enabling concept-level intervention at test time. Classical CBMs use manually specified, annotated concepts, whereas recent post-hoc [38], label-free [17], or vision-language-guided CBMs [36, 26, 20, 22, 21] reduce this cost by constructing or discovering concept spaces with language, vision-language, or foundation models. These approaches enhance the scalability and usability of concept-based prediction but focus mainly on concept construction, faithfulness, intervention, or final-layer explanation. Beyond flat concept sets, recent work explores richer concept embeddings [6], hierarchical spaces [34], spatially grounded concepts [27], and multi-granularity structures [19, 2]. MVP-CBM [31] further shows that different concepts align better with features from different visual layers, and that tying all concepts to the final layer can weaken concept–feature correspondence, suggesting depth-dependent concept evidence. However, existing hierarchical and multi-layer CBMs mainly use internal concept activations to organize, aggregate, or refine final explanations and predictions, without treating progressive concept formation as an opportunity for adaptive computation, in which intermediate bottlenecks support early prediction.

Early Exit and Shallow-Exit Training. Early exit is an adaptive inference strategy that inserts intermediate exits into a network, allowing samples to stop before the final layer [28]. Existing work improves the accuracy–efficiency trade-off via exit-head design, multi-exit training, and inference-time exit policies. For model design, BranchyNet [28] adds intermediate classifiers, MSDNet [12] builds multi-scale dense features for anytime prediction, and LGViT [35] strengthens ViT early exits with local perception and global aggregation heads. For training, knowledge distillation [11, 37] and self-distillation [39] enhance shallow exits by aligning them with deeper outputs or features, while recent multi-exit methods [37, 25, 9, 4, 10] address optimization conflicts and late-exit degradation. For inference, exit policies often use confidence, margin, entropy, prediction stability, or expert aggregation, as in PABEE [41], F-PABEE [7], and BEEM [1]. However, directly applying standard early-exit methods to CBMs is non-trivial. An intermediate CBM exit must expose a usable concept bottleneck, not just class logits; simple linear or MLP readouts can underfit shallow ViT features. To eliminate head capacity as a confound, we use the same stronger, heterogeneous exit-head family for all methods. Shallow-exit training in CBMs also differs from class-level distillation: concepts (attributes, parts, textures, shapes, scene cues) may emerge at different depths, so uniformly mimicking the final bottleneck can ignore preferred concept depths. Finally, standard exit gates rely solely on class-level signals, whereas CBMs provide concept-level evidence at each exit.

To the best of our knowledge, PACE is the first attempt to study dynamic early-exit inference for CBMs with explicit multi-depth concept bottlenecks. By coupling preference-aware training of shallow bottlenecks with concept-guided exit calibration, PACE supports efficient CBM deployment while fully exploiting, rather than discarding, intermediate concept evidence.

3 Method

3.1 Problem Formulation and Overview of PACE

For a given input image x, a frozen vision-language backbone produces intermediate feature representations at multiple depths. Each exit yields a concept representation, and a shared concept-to-class readout transforms the resulting concept logits into class logits. The deepest exit serves as the full-depth endpoint, whereas the shallower exits enable dynamic inference. Let $\mathcal{E}$ be the set of all exits, and let $\mathcal{E}_s \subset \mathcal{E}$ denote the subset of shallow exits. For an exit $l \in \mathcal{E}$, the backbone representation is $h_l(x)$, the concept probe is f_l, and the shared classifier is g:

$$z_l(x) = f_l(h_l(x)) \in \mathbb{R}^K, \qquad u_l(x) = g(z_l(x)) \in \mathbb{R}^C, \qquad p_l(y \mid x) = \mathrm{softmax}(u_l(x)),$$

where K is the number of concepts and C is the number of classes. Here, concept extraction is handled by the exit-specific probes f_l, while concept-to-class prediction is handled by a shared readout g. Accordingly, we adopt a decoupled, two-stage CBM training scheme. The concept-to-class classifier g is trained in its own stage and then kept fixed. During multi-exit training, all exits share the same frozen readout, so differences between exits reflect how well they reconstruct concept bottlenecks compatible with g, not exit-specific heads. To disentangle PACE from probe-capacity effects, we employ the same heterogeneous family of exit heads for PACE and all adapted baselines. This design reduces the chance that performance gaps are due to exit-head capacity and instead links them more directly to the training objective and deployment policy. Please refer to Appendix D for additional ablations with simpler linear and MLP heads.

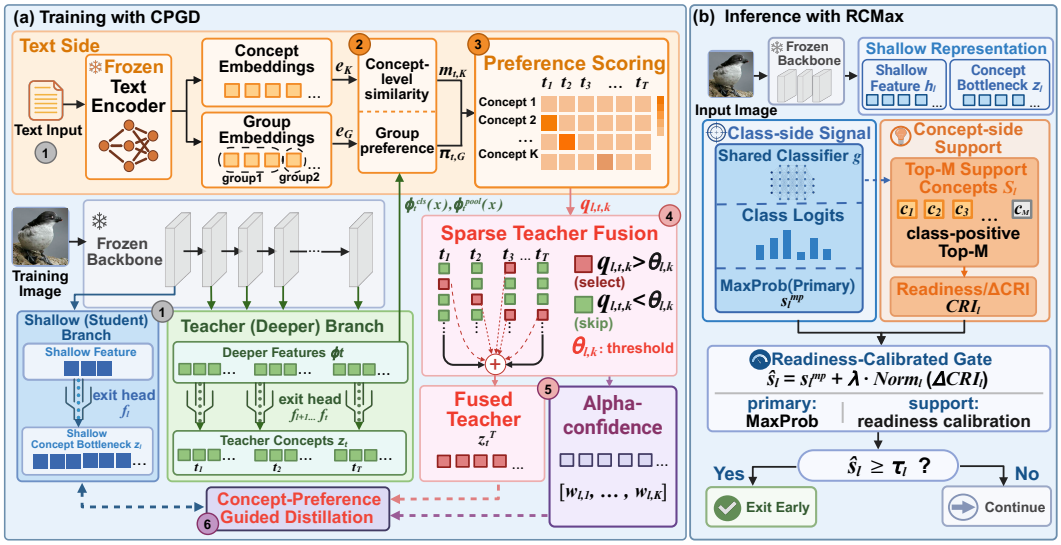


Figure 2: Overview of PACE. **(a)** CPGD constructs concept-wise fused teachers for shallow concept bottlenecks using sample-wise concept-depth preferences. **(b)** RCMax calibrates MaxProb-based early-exit decisions with key-concept support from the current bottleneck.

Since each exit corresponds to a concept bottleneck, an early prediction should be made only when the current bottleneck provides concept-level justification for the predicted class. Figure 2 depicts PACE as a framework consisting of two main components. In Figure 2(a), CPGD exploits text-side concept and group embeddings, together with deeper visual features, to infer preferred depths for each concept. These depth preferences specify which deeper exits provide the most suitable evidence for each shallow concept. CPGD then selectively fuses the logits from the chosen deeper concepts into concept-specific teachers and scales the distillation loss based on the concentration of the teacher-depth distribution. In Figure 2(b), RCMax retains the shared classifier and MaxProb score as the class-level indicator, and verifies that class-relevant key concepts are adequately instantiated in the current bottleneck before allowing an early exit. The following subsections describe the main training and inference components, while additional implementation details are provided in the Appendix A.

3.2 Concept-Preference Guided Distillation (Training with CPGD)

CPGD Overview and Procedure. A common way to train multi-exit models is to distill earlier exits from the final one. In the case of CBMs, this effectively assumes that the deepest concept bottleneck serves as a single, uniform teacher for all shallower concepts. This assumption can be suboptimal because concept evidence becomes accessible at different depths: low-level visual features may appear early, while fine-grained or abstract concepts often depend on deeper layers. Thus, for a given shallow exit, different concepts may be better guided by teachers at different depths. CPGD addresses this training limitation by assembling concept-specific teachers from multiple deeper exits.

As shown in Figure 2(a), CPGD can be decomposed into six steps: text–image input preparation, evidence scoring, concept-depth preference scoring, sparse teacher fusion, alpha-confidence weighting, and the final training objective.

Step 1: Text-image inputs and teacher candidates. CPGD first prepares the text-side and image-side inputs used for preference estimation. On the text side, the frozen text encoder maps concept phrases and group-level text descriptions into concept embeddings e_k and group embeddings e_G. On the image side, the frozen visual backbone provides the shallow feature $h_l(x)$ for the student exit and deeper features for candidate teacher exits. For a shallow exit $l \in \mathcal{E}_s$, all teacher candidates are exits at greater depth, $\mathcal{T}(l) = \{t \in \mathcal{E} : t > l\}$. These text and visual representations are then used to compute group-level and concept-level preferences.

Step 2: Group and concept evidence scoring. Let $G(k)$ represent the semantic group associated with concept k. For each candidate teacher at depth t, CPGD fuses a group-level preference with a concept-level similarity. Concretely, the group evidence score is computed as $a_{t,G}(x) = \phi_t^{\text{cls}}(x)^\top e_G$, where $\phi_t^{\text{cls}}(x)$ is a projected visual representation and e_G is the text embedding corresponding to semantic group G. We then transform this score into a group preference via

$$\pi_{t,G}(x) = \text{softmax}_G\left(\frac{\sigma(a_{t,G}(x))}{\tau_1}\right). \tag{1}$$

For a given concept k, the similarity at the concept level is defined as $m_{t,k}(x) = \phi_t^{\text{pool}}(x)^\top e_k$, where $\phi_t^{\text{pool}}(x)$ denotes the pooled visual representation and e_k denotes the text embedding corresponding to concept k. The preference score for concept k at layer t is then given by $s_{t,k}(x) = \pi_{t,G(k)}(x) m_{t,k}(x)$. The group-level component provides a coarse semantic signal at each depth, whereas the concept-level component further sharpens the preference among concepts within the same group. Details regarding projection and normalization are deferred to the Appendix B. The resulting score $s_{t,k}(x)$ provides the unnormalized preference of teacher depth t for concept k.

Step 3: Concept-depth preference scoring. We normalize these scores across teacher depths, yielding a sample-wise and concept-wise distribution that reflects depth preferences:

$$q_{l,t,k}(x) = \text{softmax}_{t\in\mathcal{T}(l)}\left(\frac{s_{t,k}(x)}{\tau_2}\right). \tag{2}$$

Step 4: Sparse teacher fusion. A sparse teacher mask chooses a limited subset of teacher depths, and $\alpha_{l,t,k}(x)$ represents the normalized sparse fusion weight computed from $q_{l,t,k}(x)$. The fused teacher for concept k at exit l is

$$z_{l,k}^T(x) = \sum_{t\in\mathcal{T}(l)} \alpha_{l,t,k}(x) z_{t,k}(x). \tag{3}$$

This teacher accounts for samples, concepts, and the depth of representation. More information on the sparse teacher depth selection mask, the fallback rule used when no depths are selected, and the top-κ constraint on chosen teacher depths can be found in the Appendix B.

Step 5: Alpha-Confidence Weighting. The same fusion weights also serve as a concentration-driven proxy for confidence during distillation:

$$\rho_{l,k}(x) = \max_{t\in\mathcal{T}(l)} \alpha_{l,t,k}(x), \qquad w_{l,k}(x) = \frac{\rho_{l,k}(x)}{\frac{1}{BK}\sum_{i=1}^{B}\sum_{k'=1}^{K}\rho_{l,k'}(x_i) + \epsilon}, \tag{4}$$

where B denotes the mini-batch size. A more concentrated teacher distribution places higher emphasis, whereas a more diffuse distribution reduces the influence of the associated concept supervision.

Step 6: CPGD Training Objective. Each shallow exit preserves the concept-level supervision term $\mathcal{L}_{\text{concept}}^{(l)}$ as well as the class-level self-distillation term $\mathcal{L}_{\text{KD}}^{(l)}$ derived from the final exit. When training exit l, all teacher logits $z_{t,k}(x)$ for $t \in \mathcal{T}(l)$ are treated as stop-gradient targets, ensuring that the CPGD loss only updates the shallow exit and does not back-propagate into the backbone or any deeper teacher exits. In addition, CPGD penalizes the mismatch between the shallow exit's concept logits and those of the fused concept teacher:

$$\mathcal{L}_{\text{CPGD}}^{(l)} = \frac{1}{BK}\sum_{i=1}^{B}\sum_{k=1}^{K} w_{l,k}(x_i)\left\|z_{l,k}(x_i) - z_{l,k}^T(x_i)\right\|_2^2, \quad \mathcal{L}^{(l)} = \mathcal{L}_{\text{concept}}^{(l)} + \lambda_{\text{KD}}\mathcal{L}_{\text{KD}}^{(l)} + \lambda_{\text{CPGD}}\mathcal{L}_{\text{CPGD}}^{(l)}. \tag{5}$$

Thus, CPGD enhances the recovery of shallow concept bottlenecks by guiding each shallow concept logit to align with a preference-weighted combination of deeper teacher signals, instead of requiring all concepts to uniformly mimic the final bottleneck representation.

3.3 READINESS-CALIBRATED MAXPROB (INFERENCE WITH RCMAX)

During inference, PACE checks whether the current concept bottleneck provides sufficient evidence to justify an early exit.

Class-side signal. As shown in Figure 2(b), RCMax retains the standard MaxProb signal as the primary class-side criterion. PACE uses the shared concept-to-class classifier g to produce class logits at exit l and defines

$$s_l^{\mathrm{mp}}(x) = \max_y p_l(y \mid x), \tag{6}$$

which reflects the classifier's confidence but does not indicate whether the current concept bottleneck offers sufficient class-supporting evidence. RCMax therefore augments this class-side signal with concept-side support.

Concept-side support. Readiness-Calibrated MaxProb refines this class-level confidence with a lightweight, CBM-specific calibration step that checks whether the predicted class is actually supported by its most important concepts.

For each class c, let $\mathcal{K}(c)$ represent the set of the top-M concepts that have positive classifier weights for that class, and denote by $\omega_{c,k}$ their corresponding L1-normalized weights. For exit l and concept k, we compute validation statistics $\mu_{l,k}$ and $\beta_{l,k}$ from the concept activations, where $\mu_{l,k}$ is a quantile and $\beta_{l,k}$ is a truncated standard deviation. The concept readiness score is

$$r_{l,k}(x) = \sigma\left(\frac{z_{l,k}(x) - \mu_{l,k}}{\beta_{l,k} + \epsilon}\right). \tag{7}$$

Let $\hat{y}_l$ denote the predicted class and $\tilde{y}_l$ denote the runner-up class. We define the class-conditioned readiness index and readiness gap as

$$\mathrm{CRI}_l(c; x) = \sum_{k \in \mathcal{K}(c)} \omega_{c,k} r_{l,k}(x), \qquad \Delta_l^{\mathrm{cri}}(x) = \max\left\{\mathrm{CRI}_l(\hat{y}_l; x) - \mathrm{CRI}_l(\tilde{y}_l; x), 0\right\} \tag{8}$$

Readiness-calibrated gate. The function Norm_l denotes a per-exit robust min-max normalization, estimated on the validation set using the 10^{th} and 90^{th} percentiles; the full expression is provided in the appendix. RCMax combines the class-side MaxProb score and the concept-side readiness gap into a calibrated exit score:

$$\tilde{s}_l(x) = s_l^{\mathrm{mp}}(x) + \lambda\, \mathrm{Norm}_l\big(\Delta_l^{\mathrm{cri}}(x)\big), \qquad \ell(x) = \min\{l \in \mathcal{E}_s : \tilde{s}_l(x) \geq \tau\}.$$

During inference, exits are evaluated sequentially: if the calibrated score at exit l exceeds the threshold τ, PACE terminates the forward pass and outputs the prediction at that exit; otherwise, the sample continues to the next exit. If no shallow exit meets the threshold, the sample defaults to the final exit. Accordingly, RCMax factorizes the exit decision into class-side confidence and concept-side evidence sufficiency, instead of relying on confidence alone as a proxy for readiness. Since RCMax reuses the current exit's concept logits and only performs vector operations on precomputed key concepts, it introduces no additional backbone forward pass.

4 EXPERIMENTS

To evaluate the performance of PACE, we conducted experiments on five recognition benchmarks under an identical multi-exit CBM setup. All approaches are implemented in PyTorch and are evaluated with frozen CLIP/OpenCLIP vision backbones. Training follows a decoupled, two-stage procedure: the shared concept-to-class classifier is first trained independently; then, during multi-exit training, both the backbone and the classifier are kept frozen, and only the concept probes, exit heads, and CPGD-specific teacher-selection parameters are updated. This section aims to answer the following three Research Questions (RQs).

RQ1 (Superiority of PACE): What advantages does PACE offer over adapted early-exit baselines?

RQ2 (Benefits of Components): Where do the gains of CPGD come from, and what additional benefit does RCMax provide?

RQ3 (Preservation of Concept Evidence): Does PACE preserve usable intermediate concept evidence while remaining lightweight for deployment?

4.1 EXPERIMENTAL SETTINGS

Datasets. We conducted experiments on CUB [30], CIFAR-100, CIFAR-10 [14], TinyImageNet [15, 24], and Places365 [40]. Note that these datasets span a range of recognition settings, including fine-grained categorization, coarse object recognition, ImageNet-based small-scale classification, and large-scale scene recognition. We treat TinyImageNet as an ImageNet-derived benchmark; further details about its source and data splits are provided in Appendix A.

CBM Instantiation. For each dataset, we build a six-exit label-free CBM [17] on a frozen CLIP/OpenCLIP backbone [21, 3], with exits at $\{\mathrm{L}_l \mid l = 2k,\ k \in \mathbb{N},\ 1 \le k \le 6\}$. The final exit L_{12} serves as the full-depth endpoint. We use a shared sparse linear concept-to-class classifier as a fixed readout across all exits. Following the decoupled two-stage protocol described in Section 3, this classifier is trained separately and then kept fixed during multi-exit training. Thus, early exits differ in how well they recover concept bottlenecks compatible with the same readout, rather than in the use of different class heads. All methods share the same backbone, concept set, exit locations, heterogeneous exit heads, shared classifier, and validation protocol; details are in Appendix A.

Baselines. We compare PACE with the full-depth CBM and early-exit baselines adapted to the same multi-exit CBM setting. *Full-depth CBM* evaluates the L_{12} endpoint without early exit. *BYOT / UD* is our uniform-distillation baseline, where shallow exits are trained with a shared final-depth distillation objective [39]. *EnViT* adapts representative ViT early-exit training to the same CBM exits [4]. *BEEM-A* is adapted to our CBM exits following the same validation-selection protocol [1]. *PABEE* uses prediction stability as the exit criterion [41]. *F-PABEE*$_{\mathrm{cls}}$ applies patience to class predictions, while *F-PABEE*$_{\mathrm{con}}$ applies patience to concept-side predictions [7]. All adapted baselines share the same backbone, exit locations, concept vocabulary, shared classifier, exit-head capacity, and validation-selection protocol. Detailed adaptations are provided in Appendix A.2.

Evaluation Metrics. We report Top-1 accuracy and speedup relative to the full-depth L_{12} CBM: Following standard early-exit evaluation, speedup is computed as the ratio between the total full-depth inference cost and the realized cost under dynamic exiting: $\mathrm{Speedup} = \frac{NC_{\mathrm{L}_{12}}}{\sum_{j=1}^{N} C_{\ell(x_j)}}$, where N is the number of samples, $\ell(x_j)$ is the selected exit, and C_ℓ is the cumulative MAC cost up to exit ℓ. C_ℓ includes the frozen backbone, exit concept head, and shared concept-to-class classifier. RCMax reuses current-exit concept logits and adds only vector-level readiness computation over precomputed key concepts, with no additional backbone forward pass. Details are in Appendix C. Thresholds and readiness-calibration weights are selected on the validation split. Table 1 reports validation-selected operating points, Figure 3 reports accuracy–speedup frontiers, and later ablations use forced early exits or matched practical target speedups as specified below.

Table 1: Performance comparison in terms of accuracy and efficiency. We present test accuracy and the achieved cumulative-MAC speedup for operating points chosen via validation. Bold values denote the best or jointly best performance among early-exit approaches, excluding the full-depth baseline.

Method	CUB		CIFAR-100		CIFAR-10		TinyImageNet		Places365	
	Acc. ↑	Speedup ↑	Acc. ↑	Speedup ↑	Acc. ↑	Speedup ↑	Acc. ↑	Speedup ↑	Acc. ↑	Speedup ↑
Full-depth CBM	68.31	1.00×	75.85	1.00×	94.93	1.00×	75.03	1.00×	50.36	1.00×
BYOT / UD	61.14	1.50×	71.42	1.80×	93.54	2.45×	70.08	1.81×	44.94	1.81×
EnViT	60.66	1.53×	72.27	1.70×	93.47	2.45×	70.10	1.83×	45.09	1.81×
BEEM-A	55.65	1.52×	67.81	1.73×	87.68	2.46×	61.71	1.80×	41.52	1.79×
PABEE	60.98	1.67×	68.66	2.06×	93.26	1.73×	65.88	1.99×	45.73	1.87×
F-PABEE$_{\mathrm{cls}}$	60.95	1.50×	70.46	1.72×	90.17	2.45×	65.59	1.79×	44.93	1.81×
F-PABEE$_{\mathrm{con}}$	59.83	1.51×	65.75	1.69×	87.14	2.48×	64.54	1.81×	43.42	1.80×
PACE (ours)	**61.33**	**1.73**×	**72.64**	**2.07**×	**93.74**	**2.48**×	**70.10**	**2.00**×	**46.22**	**1.89**×

4.2 PERFORMANCE COMPARISON WITH ADAPTED EARLY-EXIT BASELINES (RQ1)

To evaluate PACE, we compare it with adapted early-exit baselines in the same multi-exit CBM setting. Table 1 shows validation-selected operating points within dataset-specific practical speedup ranges, along with corresponding test accuracy and realized speedup. These ranges exclude both trivial

near-full-depth points and extremely low-accuracy, high-speed points, and Figure 3 supplements this with full accuracy–speedup frontiers. Table 1 assesses complete early-exit policies, while Tables 2 and 3 separately analyze PACE training- and deployment-side contributions. Overall, PACE improves the practical trade-off by matching or exceeding early-exit accuracy while achieving higher realized speedups in these practical regions.

Figure 3 illustrates the accuracy–speedup trade-off frontiers over different operating points. This frontier-based view complements Table 1 by demonstrating that the comparison does not depend on any single threshold. We concentrate on the practically relevant region characterized by a small loss in accuracy, where early exits are most suitable for deployment, and the quality of shallow concept bottlenecks has the greatest impact.

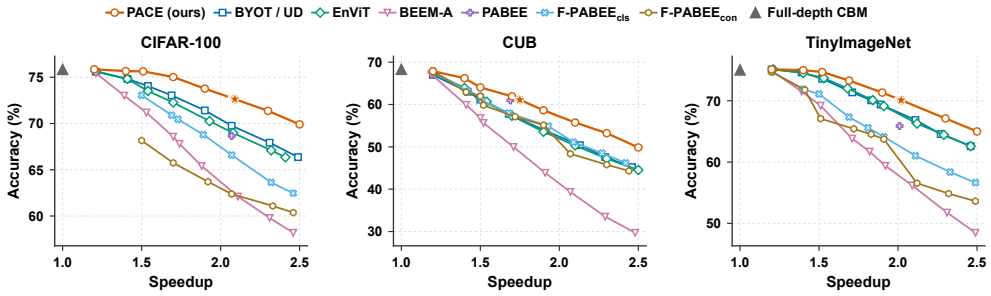


Figure 3: Accuracy-speedup trade-offs. Each subfigure compares PACE with adapted baselines under the same multi-exit CBM setup. The full-depth CBM reference is marked at $1.0\times$ speedup.

4.3 Ablation Studies on CPGD and RCMax (RQ2)

Impact of CPGD on Shallow Concept Bottlenecks. We first isolate CPGD's training-time effect before applying readiness calibration. We define Avg. Early Acc. as the unweighted mean accuracy of forced predictions at L_2, L_4, and L_6, which evaluates shallow-bottleneck recovery independently of the deployment-time gate.

Table 2: Training impact on early-exit predictions. Component-wise CPGD ablation on CIFAR-100; Δ is measured relative to uniform multi-depth supervision.

Variant	Avg. Early Acc. ↑	Δ
Final-only / UD	57.53	-3.03
Uniform multi-depth	60.56	0.00
Top-1 preferred	61.23	+0.67
Concept-only dense	60.57	+0.01
Full-preference dense	61.21	+0.65
Sparse w/o confidence	61.28	+0.72
Full CPGD	**61.68**	**+1.12**

Table 2 shows that uniform multi-depth supervision already improves Avg. Early Acc. from 57.53 to 60.56. Preference-aware teacher selection provides further gains: Top-1 preferred and full-preference dense improve over uniform supervision by $+0.67$ and $+0.65$ points, respectively, while group-level preference contributes $+0.64$ points. Sparse selection provides a smaller additional gain, and confidence weighting further improves performance to 61.68, yielding a total $+1.12$-point gain over uniform multi-depth supervision.

We further evaluate the complete CPGD configuration over seeds 1, 42, and 1024 on CUB, CIFAR-100, and TinyImageNet. CPGD outperforms UD in all $9/9$ dataset-seed pairs, with mean improvements of $+4.68 \pm 0.64$, $+4.38 \pm 0.20$, and $+2.66 \pm 0.48$ points, respectively. Complete cross-dataset results and three-seed summaries are provided in Appendix D.1.

Impact of RCMax on Deployment Decisions. We next examine the deployment-time contribution of readiness calibration. Table 3 compares UD and CPGD under MaxProb and RCMax at comparable practical target speedups. These operating points are chosen to compare the four variants under similar compute regimes rather than at their individually optimized points. Thus, UD+MaxProb versus CPGD+MaxProb isolates the training-side effect of CPGD, while CPGD+MaxProb versus PACE isolates the additional contribution of RCMax.

Table 3: Module-wise PACE ablation at comparable practical speedups. Each entry reports test accuracy and cumulative-MAC speedup as "Accuracy / Speedup."

Dataset	Target	UD + MaxProb	UD + RCMax	CPGD + MaxProb	PACE
CUB	1.5×	60.93 / 1.51×	60.98 / 1.51×	63.61 / 1.55×	**64.34 / 1.54**×
CIFAR-100	1.8×	72.23 / 1.77×	72.26 / 1.77×	74.41 / 1.79×	**74.57 / 1.78**×
TinyImageNet	2.2×	65.29 / 2.22×	65.42 / 2.22×	67.77 / 2.22×	**68.16 / 2.20**×
Places365	1.9×	44.47 / 1.90×	44.73 / 1.90×	45.72 / 1.90×	**46.03 / 1.93**×

Across all datasets, replacing UD with CPGD under the same MaxProb gate produces the larger improvement, indicating that the primary gains arise from training-side shallow-bottleneck recovery. RCMax provides smaller but consistent complementary gains without additional backbone forward passes: across three seeds on CUB, CIFAR-100, and TinyImageNet, it improves over MaxProb in all 9/9 comparisons, with mean paired gains of $+0.88 \pm 0.27$, $+0.10 \pm 0.06$, and $+0.21 \pm 0.06$ points, respectively. Full results are provided in Appendix D.3.

CPGD also transfers to two newer CBM families, DN-CBM [22] and CFM [32], improving Avg. Early Acc. by $+4.55$ and $+2.58$ points, respectively; full settings are provided in Appendix D.5.

4.4 Concept-Level Semantic Preservation (RQ3)

We next examine whether the improvement in shallow predictive accuracy comes at the expense of concept-level semantic alignment. We keep the original 370-dimensional label-free CUB concept vocabulary, trained checkpoints, and evaluation protocol unchanged, and expand the mapping to human-annotated CUB attributes from 44 to 132 original concepts (35.7% coverage), corresponding to 107 unique semantic targets. We construct the mapping independently of model outputs. Since this evaluation uses forced predictions at $\mathrm{L}_2, \mathrm{L}_4, \mathrm{L}_6$ without RCMax, we report the training-side method as CPGD.

As shown in Table 4, CPGD improves Avg. Early Acc. by $+4.68$ points while the mean semantic-alignment metrics remain stable or improve slightly across all four measures. The same conclusion holds after collapsing synonymous mappings to the 107 unique semantic targets. These results indicate that the gain in shallow predictive utility is not obtained at the expense of semantic alignment over the substantially broader mapped portion of the original label-free vocabulary. Additional controlled attribute evaluations, intervention analyses, and concept-space diagnostics are provided in Appendix D.4.

Table 4: Expanded semantic evaluation on CUB. Results are averaged over seeds 1, 42, and 1024 using 132 mapped concepts from the original 370-dimensional vocabulary. Δ denotes the mean paired difference computed before rounding.

Metric	UD	CPGD	Δ
Avg. Early Acc.	37.92	**42.59**	**+4.68**
AUROC	69.17	**69.66**	**+0.50**
Weighted AUROC	68.69	**69.18**	**+0.49**
mAP	30.01	**30.16**	**+0.14**
BAcc	64.01	**64.26**	**+0.25**

5 Conclusion

We introduced PACE, a preference-aware framework that moves CBM efficiency beyond final-depth explanation toward inference-time concept formation. PACE exploits the observation that concept evidence emerges progressively across exits: CPGD uses concept-wise multi-depth teachers to strengthen shallow bottlenecks, while RCMax calibrates early-exit decisions with key-concept support from the current bottleneck. Across five datasets under a common multi-exit CBM protocol, PACE improves the practical accuracy–efficiency trade-off, with module analyses showing that CPGD provides the main gains and RCMax yields consistent deployment-side improvements. Future work includes broader backbones and exit budgets, hardware latency evaluation, and stronger quantitative measures of progressive evidence for concepts.

AI USE STATEMENT

Generative AI tools were used to assist with drafting and polishing portions of the manuscript and with retrieving relevant literature. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final manuscript.

ETHICS STATEMENT

This work uses publicly available benchmark datasets and pretrained models and does not involve the collection of new human-subject data. We adhere to the ICLR Code of Ethics. Potential risks associated with incomplete or biased concept vocabularies and their use in high-stakes settings are discussed in Appendix G.

REPRODUCIBILITY STATEMENT

We provide dataset and split details, model and exit configurations, training hyperparameters, baseline adaptations, and gate-selection procedures in Appendix A; detailed CPGD implementation in Appendix B; cumulative-MAC accounting in Appendix C; and additional ablations, multi-seed evaluations, and robustness analyses in Appendix D.

REFERENCES

[1] Divya Jyoti Bajpai and Manjesh Kumar Hanawal. BEEM: Boosting performance of early exit DNNs using multi-exit classifiers as experts. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[2] Itay Benou and Tammy Riklin-Raviv. Show and tell: Visually explainable deep neural nets via spatially-aware concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30063–30072, 2025.

[3] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829, 2023.

[4] Yonghao Dong, Qiang He, Penghong Rui, Zhenzhe Zheng, Zhao Li, Feifei Chen, Hai Jin, and Yun Yang. EnViT: Enhancing the performance of early-exit vision transformers via exit-aware structured dropout-enabled self-distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 20852–20860, 2026.

[5] Jessica Echterhoff, An Yan, Kyungtae Han, Amr Abdelraouf, Rohit Gupta, and Julian McAuley. Driving through the concept gridlock: Unraveling explainability bottlenecks in automated driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7346–7355, 2024. doi: 10.1109/WACV57701.2024.00718.

[6] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lió, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pp. 21400–21413, 2022.

[7] Xiangxiang Gao, Wei Zhu, Jiasheng Gao, and Congrui Yin. F-PABEE: Flexible-patience-based early exiting for single-label and multi-label text classification tasks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[8] Yunhe Gao, Difei Gu, Mu Zhou, and Dimitris Metaxas. Aligning human knowledge with visual concepts towards explainable medical image classification. In *Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 46–56. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72117-5_5.

[9] Cheng Gong, Yao Chen, Qiuyang Luo, Ye Lu, Tao Li, Yuzhi Zhang, Yufei Sun, and Le Zhang. Deep feature surgery: Towards accurate and efficient multi-exit networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[10] Jianing He, Qi Zhang, Hongyun Zhang, Xuanjing Huang, Usman Naseem, and Duoqian Miao. COSEE: Consistency-oriented signal-based early exiting via calibrated sample weighting mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pp. 24023–24031, 2025.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[12] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[13] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 5338–5348, 2020.

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

[15] Ya Le and Xuan S. Yang. Tiny ImageNet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[16] Yi Liu, Changsheng Zhang, and Xingjun Dong. A survey of real-time surface defect inspection methods based on deep learning. *Artificial Intelligence Review*, 56:12131–12170, 2023. doi: 10.1007/s10462-023-10475-7.

[17] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[18] Daniel Omeiza, Helena Webb, Marina Jirotka, and Lars Kunze. Explanations in autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10142–10162, 2022. doi: 10.1109/TITS.2021.3122865.

[19] Konstantinos P. Panousis, Dino Ienco, and Diego Marcos. Coarse-to-fine concept bottleneck models. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pp. 105171–105199, 2024.

[20] Katharina Prasse, Patrick Knab, Sascha Marton, Christian Bartelt, and Margret Keuper. DCBM: Data-efficient visual concept bottleneck models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 49752–49782, 2025.

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

[22] Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 444–461, 2024.

[23] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019. doi: 10.1038/s42256-019-0048-x.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[25] Wonchul Son, Jaemin Na, Junyong Choi, and Wonjun Hwang. Densely guided knowledge distillation using multiple teacher assistants. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9395–9404, 2021.

[26] Divyansh Srivastava, Ge Yan, and Tsui-Wei Weng. VLG-CBM: Training concept bottleneck models with vision-language guidance. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pp. 79057–79094, 2024.

[27] Ao Sun, Yuanyuan Yuan, Pingchuan Ma, and Shuai Wang. Boosting concept bottleneck models with supervised, hierarchical concept learning. OpenReview, 2025.

[28] Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2016. doi: 10.1109/ICPR.2016.7900006.

[29] Bas H. M. van der Velden, Hugo J. Kuijf, Kenneth G. A. Gilhuijs, and Max A. Viergever. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470, 2022. doi: 10.1016/j.media.2022.102470.

[30] 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.

[31] Chunjiang Wang, Kun Zhang, Yandong Liu, Zhiyang He, Xiaodong Tao, and S. Kevin Zhou. MVP-CBM: Multi-layer visual preference-enhanced concept bottleneck model for explainable medical image classification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 529–537, 2025.

[32] Kai Wittenmayer, Sukrut Rao, Amin Parchami-Araghi, Bernt Schiele, and Jonas Fischer. CFM: Language-aligned concept foundation model for vision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

[33] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 11205–11216, 2021.

[34] Haodong Xie, Yujun Cai, Rahul Singh Maharjan, Yiwei Wang, Federico Tavella, and Angelo Cangelosi. Hierarchical, interpretable, label-free concept bottleneck model. *arXiv preprint arXiv:2604.02468*, 2026.

[35] Guanyu Xu, Jiawei Hao, Li Shen, Han Hu, Yong Luo, Hui Lin, and Jialie Shen. LGViT: Dynamic early exiting for accelerating vision transformer. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pp. 9103–9114, 2023.

[36] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19187–19197, 2023.

[37] Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. ViTKD: Practical guidelines for ViT feature knowledge distillation. *arXiv preprint arXiv:2209.02432*, 2022.

[38] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[39] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3713–3722, 2019.

[40] 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, 2018.

[41] Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. BERT loses patience: Fast and robust inference with early exit. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pp. 18330–18341, 2020.

A Supplementary Implementation Details

Datasets. Table 5 provides an overview of the datasets employed in our experiments. For each dataset, we partition the original training set into training and validation subsets using a fixed random seed, and we rely on the official test or validation split for reporting final results. We use the validation split to select classifiers, determine gate thresholds, and choose operating points; we compute all primary reported metrics on a separate held-out evaluation split. For Places365, we adopt a subset protocol because training on the full training set is computationally expensive: we use 120k samples for training, 12k samples from the training set for validation, and an additional 12k images for held-out evaluation.

Existing Assets. We rely exclusively on public benchmark datasets [30, 14, 15, 24, 40] and publicly released CLIP/OpenCLIP [21, 3] backbones for research evaluation. We reference the original publications for all datasets and pretrained models employed in this study and do not distribute any datasets or pretrained weights ourselves. Users are expected to adhere to the licenses and usage terms specified by the original dataset and model providers.

Table 5: Dataset settings. "Val" is the validation subset to select thresholds and hyperparameters.

Dataset	Classes	Train	Val	Test / eval split
CUB	200	5,391	599	5,790 official test images
CIFAR-100	100	45,000	5,000	10,000 official test images
CIFAR-10	10	45,000	5,000	10,000 official test images
TinyImageNet	200	90,000	10,000	10,000 official validation images
Places365	365	120,000	12,000	12,000 held-out validation images

Backbones and Exits. We keep the CLIP/OpenCLIP vision–language backbones frozen and train only the concept probes, the exit heads, and the sparse concept-to-class readout layer. For CIFAR-10 and CIFAR-100, we use OpenCLIP ViT-L/14, whereas for CUB, TinyImageNet, and Places365, we use CLIP ViT-B/16. Across all datasets, we place six exits at $\{\mathrm{L}_l \mid l = 2k,\ k \in \mathbb{N},\ 1 \le k \le 6\}$, with L_{12} serving as the full-depth CBM endpoint used for evaluation. For the 24-block ViT-L/14 backbone, these exit labels correspond to physical transformer blocks $\{4, 8, 12, 16, 20, 24\}$, respectively, so L_{12} denotes the actual final backbone block. All early-exit methods share the same exit positions across all datasets.

Concept Vocabulary and Grouping. For each dataset, we employ the label-free concept vocabulary specified in Table 6. These concepts are organized into broad semantic categories for preference-guided concept distillation and readiness calibration. We construct the grouping files via lexical rules applied to the filtered concept vocabulary; they are not derived from the validation or test splits.

Table 6: Concept vocabulary and grouping statistics.

Dataset	Concepts	Groups	Concept/group metadata
CUB	370	12	`cub_filtered.txt`, lexical part groups
CIFAR-100	892	12	`cifar100_filtered.txt`, lexical attribute groups
CIFAR-10	143	12	`cifar10_filtered.txt`, lexical attribute groups
TinyImageNet	2,050	19	`tinyimagenet_filtered.txt`, lexical attribute groups
Places365	2,544	27	`places365_filtered.txt`, lexical scene groups

Shared Concept-to-Class Classifier. We train the concept-to-class classifier once using the concept representation from the final exit (L12) and then share it across all exits. This setup guarantees that differences between early exits stem from the quality of their concept bottlenecks and their gating behavior, rather than from variations in the class-head capacity. We employ a sparse linear readout trained with GLM-SAGA [33] over a small grid of regularization strengths. In the main experiments, we choose a regularization value of $\lambda = 3 \times 10^{-4}$, with the maximum learning rate fixed at 0.1.

A.1 Exit Heads and Training Recipe

Exit-Head Architecture. For shallow exits, all methods rely on the same heterogeneous exit-head family across all datasets. Unless specified otherwise, this heterogeneous exit head consists of local perception heads at $\mathrm{L}_2, \mathrm{L}_4, \mathrm{L}_6$ and global aggregation heads at $\mathrm{L}_8, \mathrm{L}_10, \mathrm{L}_{12}$. This setup prevents shallow concept readouts from being underpowered and ensures that comparisons primarily reflect differences in training objectives and exit policies, rather than in exit-head capacity.

Uniform Distillation Baseline. UD trains the shallow exits so that, given the same frozen backbone, concept vocabulary, exit heads, and shared classifier, they reproduce the concept distribution of the final exit. It serves as the baseline multi-exit CBM for evaluating confidence- and patience-gating mechanisms.

EnViT Adaptation. The EnViT baseline employs the same six CBM exits and the same heterogeneous head family, but incorporates structured shallow-exit training and dropout according to the adapted EnViT procedure. The dropout rates are set to $0.0, 0.1, 0.2, 0.3, 0.4$ for $\mathrm{L}_2, \mathrm{L}_4, \mathrm{L}_6, \mathrm{L}_8, \mathrm{L}_{10}$, respectively. Since EnViT was initially developed for class-level early exiting in ViT, we reuse its training-side shaping strategy at the corresponding concept-bottleneck exits and leave the shared concept-to-class classifier unchanged.

PACE Training. PACE employs CPGD as its training-side component and RCMax as the deployment-time gating mechanism. CPGD performs preference-guided concept distillation, using the `alpha_conf` weighting scheme and a fused concept teacher. For each trained multi-exit model, we choose the gate or patience decision rule based on the validation split and then evaluate it once on the separate evaluation split. To ensure fairness, all methods within a given dataset use the same frozen backbone, concept vocabulary, exit locations, heterogeneous family of exit heads, shared concept-to-class classifier, validation split, and test/evaluation split.

Table 7: Main training hyperparameters.

Hyperparameter	Value
Random seed / split seed	42 / 42
Multi-seed analysis	{1, 42, 1024}
Train-validation split	90% / 10% of training split
Probe optimizer	Adam
Probe learning rate	1×10^{-3}
Probe epochs	5 for CIFAR; 20 for CUB; 10 for TinyImageNet/Places365
Probe batch size	256; 512 for Places365 subset runs
Inference batch size	256 for CIFAR; 512 for TinyImageNet/Places365
Shared classifier	GLM-SAGA sparse linear head
Classifier epochs	20
Classifier batch size	256; 512 for Places365 subset runs
Classifier regularization grid	$\{3 \times 10^{-4}\}$ in the main runs
Shallow self-distillation temperature	2.0
Shallow self-distillation weight	0.5
Operating-point selection	validation-selected gate in the target speedup region
CPGD teacher mode	fused
CPGD preference weight mode	`alpha_conf`
τ_1, τ_2 for preference teacher	0.2, 0.2
Teacher candidates per concept	2
Preference distillation weight	1.0

A.2 Baselines and Gate Selection

Confidence Gates. MaxProb triggers an early exit when the highest class probability at that exit exceeds a threshold determined on the validation set. Margin instead relies on the difference between the Top-1 and Top-2 class probabilities rather than the maximum probability alone. All confidence thresholds are tuned offline using stored exit logits, so gate comparisons do not require retraining.

Readiness-Calibrated MaxProb. RCMax extends MaxProb by incorporating a concept-readiness calibration factor. During validation, we choose both the calibration weight and the confidence threshold. The approach is deliberately lightweight: it relies solely on quantities already computed by the concept bottleneck and introduces no additional learned gating network.

BEEM-A Adaptation. BEEM-A interprets intermediate exits as experts whose contributions are weighted by their accuracy. We adapt BEEM-A to the same six-exit CBM setting by using the stored class logits from each exit. During inference, neighboring exits that predict the same class pool their confidence scores into a cumulative ensemble score, with each exit weighted by its validation accuracy. A sample is considered finished once this cumulative score surpasses a threshold chosen on the validation set. The backbone, concept vocabulary, exit heads, and shared concept-to-class classifier all remain fixed, and the operating point is determined using the same validation procedure as for the other adapted baselines.

Patience Baselines. PABEE exits when class predictions remain stable across consecutive exits. F-PABEE relaxes exact class agreement by using distributional similarity, evaluated in both class and concept spaces. For fairness, patience baselines are reproduced offline using the same saved exit outputs and the same validation-selection procedure.

Operating-Point Selection. For Table 1, each method is selected on the validation split within a practical speedup region and then evaluated once on the test split. The main practical region is $1.5\times$–$2.5\times$, except Places365 where we use $1.0\times$–$2.0\times$ because the full-depth accuracy is lower and the dataset is harder. CIFAR-10 uses a more aggressive $\sim 2.5\times$ point because the full-depth model is near saturated.

B Concept-Preference Computation and CPGD Details

Text Embeddings and Group Descriptions. We use the frozen CLIP/OpenCLIP text encoder to encode each concept phrase into a concept embedding e_k. For each semantic group G, we construct a group-level text description by concatenating the concept phrases assigned to that group and encode it with the same frozen text encoder to obtain e_G. The concept groups are fixed before training and are not learned from the validation or test split. We reuse these text embeddings for preference scoring and do not update them during CPGD training.

Group Preference. For each teacher depth t, we compute a group-level evidence score between a projected visual feature $\phi_t^{\mathrm{cls}}(x)$ and the text embedding e_G of semantic group G:

$$a_{t,G}(x) = \phi_t^{\mathrm{cls}}(x)^\top e_G.$$

The group preference is obtained by applying a sigmoid followed by a temperature-scaled softmax over semantic groups:

$$\pi_{t,G}(x) = \mathrm{softmax}_G\left(\frac{\sigma(a_{t,G}(x))}{\tau_1}\right).$$

This term estimates which semantic groups are more supported at each teacher depth.

Concept-Level Similarity. For concept k, we compute a concept-level similarity between the pooled visual feature $\phi_t^{\mathrm{pool}}(x)$ and the concept text embedding e_k:

$$m_{t,k}(x) = \phi_t^{\mathrm{pool}}(x)^\top e_k.$$

Let $G(k)$ denote the semantic group of concept k. The concept-layer preference score is

$$s_{t,k}(x) = \pi_{t,G(k)}(x) m_{t,k}(x).$$

The group-level component provides broad semantic support at each layer, while the concept-level component further refines preference within each group.

For each teacher depth, the intermediate visual representation is passed through the frozen backbone's post-normalization and visual projection, followed by ℓ_2 normalization. The pooled representation is obtained by token averaging before applying the same frozen projection and normalization. These operations introduce no additional trainable parameters.

Teacher-Depth Preference. For a shallow exit l, the candidate teacher depths are all deeper exits,

$$\mathcal{T}(l) = \{t \in \mathcal{E} : t > l\}.$$

We normalize the concept-layer preference scores over teacher depths:

$$q_{l,t,k}(x) = \mathrm{softmax}_{t \in \mathcal{T}(l)} \left(\frac{s_{t,k}(x)}{\tau_2} \right).$$

This gives a sample-wise and concept-wise teacher-depth preference.

Sparse Teacher Selection and Fusion. A sparse mask $M_{l,t,k}(x) \in \{0, 1\}$ is used to choose a limited subset of teacher depths for each sample and concept. Inspired by the adaptive thresholding approach in MVP-CBM-style sparse multi-layer concept fusion [31], we begin by determining the maximum and minimum teacher-depth preferences for concept k:

$$q_{l,k}^{\max}(x) = \max_{t \in \mathcal{T}(l)} q_{l,t,k}(x), \qquad q_{l,k}^{\min}(x) = \min_{t \in \mathcal{T}(l)} q_{l,t,k}(x).$$

The selection threshold is

$$\theta_{l,k}(x) = \sigma(\eta_l) \left(q_{l,k}^{\max}(x) - q_{l,k}^{\min}(x) \right) + q_{l,k}^{\min}(x),$$

where η_l is a learnable scalar for shallow exit l. We then define the binary sparse mask

$$M_{l,t,k}(x) = \mathbf{1}[q_{l,t,k}(x) \geq \theta_{l,k}(x)].$$

To allow gradients to pass through the thresholding operation, we use a straight-through estimator (STE). Specifically, we form a soft mask

$$\widetilde{M}_{l,t,k}(x) = \sigma \left(\frac{q_{l,t,k}(x) - \theta_{l,k}(x)}{\tau_{\mathrm{STE}}} \right),$$

and use

$$M_{l,t,k}^{\mathrm{STE}}(x) = M_{l,t,k}(x) + \widetilde{M}_{l,t,k}(x) - \mathrm{sg}\Big(\widetilde{M}_{l,t,k}(x) \Big)$$

during training, where $\mathrm{sg}(\cdot)$ denotes stop-gradient. Thus, the forward pass uses the binary sparse mask, while the backward pass follows the differentiable soft mask. The scalar η_l is initialized to zero and optimized jointly with the shallow exit head using Adam; we use $\tau_{\mathrm{STE}} = 0.02$ in all experiments. For stability, the sigmoid parameterization constrains $\theta_{l,k}(x)$ to lie between $q_{l,k}^{\min}(x)$ and $q_{l,k}^{\max}(x)$. In practice, we also cap the number of selected teacher depths with a small top-κ budget and fall back to the largest-preference teacher if the threshold mask is empty. During training, we use $M_{l,t,k}^{\mathrm{STE}}(x)$ in the fusion weights; during evaluation, we use the hard mask $M_{l,t,k}(x)$. For notation, let

$$\widehat{M}_{l,t,k}(x) = \begin{cases} M_{l,t,k}^{\mathrm{STE}}(x), & \text{during training,} \\ M_{l,t,k}(x), & \text{during evaluation.} \end{cases}$$

The normalized sparse fusion weight is

$$\alpha_{l,t,k}(x) = \frac{q_{l,t,k}(x) \widehat{M}_{l,t,k}(x)}{\sum_{t' \in \mathcal{T}(l)} q_{l,t',k}(x) \widehat{M}_{l,t',k}(x) + \epsilon}.$$

The fused teacher for concept k at shallow exit l is

$$z_{l,k}^{T}(x) = \sum_{t \in \mathcal{T}(l)} \alpha_{l,t,k}(x) z_{t,k}(x).$$

The number of selected teacher depths is capped by the teacher-candidate budget listed in Table 7.

Alpha-Confidence Weighting. The same fusion weights provide a reliability estimate for concept distillation:

$$\rho_{l,k}(x) = \max_{t \in \mathcal{T}(l)} \alpha_{l,t,k}(x).$$

The mini-batch normalized weight is

$$w_{l,k}(x) = \frac{\rho_{l,k}(x)}{\frac{1}{BK} \sum_{i=1}^{B} \sum_{k'=1}^{K} \rho_{l,k'}(x_i) + \epsilon}.$$

A concentrated teacher-depth preference yields a larger distillation weight, while diffuse teacher preferences downweight the corresponding concept supervision.

Label-Free Concept Target and Base Concept Loss. Since no binary concept annotations are used in the label-free setting, concept supervision is obtained from the frozen CLIP/OpenCLIP image-text alignment space. Let $\bar{v}(x)$ be the normalized final image embedding of the frozen backbone, $\bar{e}_k$ be the normalized text embedding of concept k, and γ be the CLIP logit scale. The pseudo concept logit for concept k is

$$r_k(x) = \gamma \bar{v}(x)^\top \bar{e}_k.$$

At exit l, the concept probe maps the exit feature to a normalized embedding $\bar{u}_l(x)$, and its concept logits are

$$z_{l,k}(x) = \gamma \bar{u}_l(x)^\top \bar{e}_k.$$

The base label-free concept loss is a temperature-scaled KL distillation loss over the concept vocabulary:

$$\mathcal{L}^{(l)}_{\text{concept}} = \frac{1}{B} \sum_{i=1}^{B} \tau_c^2 \mathrm{KL}\left(\mathrm{softmax}\left(\frac{r(x_i)}{\tau_c} \right) \middle\| \mathrm{softmax}\left(\frac{z_l(x_i)}{\tau_c} \right) \right).$$

Thus, the label-free runs do not use manually annotated binary concept labels or explicit negative-concept sampling; all concepts in the vocabulary participate through the soft pseudo-label distribution.

CPGD Objective. Every shallow exit preserves both concept-level supervision and class-level self-distillation derived from the final exit. CPGD further incorporates an alpha-confidence–weighted concept distillation loss:

$$\mathcal{L}^{(l)}_{\text{CPGD}} = \frac{1}{BK} \sum_{i=1}^{B} \sum_{k=1}^{K} w_{l,k}(x_i) \left\| z_{l,k}(x_i) - z^T_{l,k}(x_i) \right\|_2^2.$$

The final shallow-exit loss is

$$\mathcal{L}^{(l)} = \mathcal{L}^{(l)}_{\text{concept}} + \lambda_{\text{KD}} \mathcal{L}^{(l)}_{\text{KD}} + \lambda_{\text{CPGD}} \mathcal{L}^{(l)}_{\text{CPGD}}.$$

During shallow-exit optimization, the deeper teacher logits $z_{t,k}(x)$ that form $z^T_{l,k}(x)$ are handled as stop-gradient targets. Consequently, the shallow-exit loss does not propagate gradients into the frozen backbone or the deeper exit heads. Training is restricted to the shallow exit head and the teacher-selection parameter η_l, with gradients for η_l flowing through the STE defined above.

C Cumulative-MAC Accounting

In the main tables, we compute speedup from cumulative MACs. For an input x_j, let $\ell(x_j) \in \{\mathrm{L}_2, \mathrm{L}_4, \mathrm{L}_6, \mathrm{L}_8, \mathrm{L}_{10}, \mathrm{L}_{12}\}$ be the selected exit and C_ℓ be the cumulative cost of exiting at ℓ. We report

$$\text{Speedup} = \frac{N C_{L12}}{\sum_{j=1}^{N} C_{\ell(x_j)}}.$$

All reported speedups are derived from stored per-sample exit choices made under the specified gating or patience strategy. Each C_ℓ accounts for the frozen ViT computations up to the chosen exit layer, the heterogeneous exit head, and the shared concept-to-class classifier. MaxProb and RCMax reuse the already computed class and concept logits. Their additional computations are limited to comparisons, top-k selection, scalar normalization, and weighted summations; these operate at the vector level and require no extra backbone or exit-head forward passes.

Table 8: Per-exit cumulative cost employed for MACs-based speedup estimation. Values are expressed in GMACs per image.

Dataset	C_{L_2}	C_{L_4}	C_{L_6}	C_{L_8}	$C_{\mathrm{L}_{10}}$	$C_{\mathrm{L}_{12}}$
CUB	3.076	5.984	8.892	11.784	14.692	17.600
CIFAR-100	6.977	13.716	20.454	27.160	33.898	40.636
CIFAR-10	6.977	13.715	20.453	27.159	33.898	40.636
TinyImageNet	3.076	5.984	8.892	11.785	14.693	17.601
Places365	3.077	5.985	8.893	11.786	14.693	17.601

D Additional Results

D.1 Additional CPGD Results

We provide additional results for the training-side effect of CPGD. Table 9 reports the complete cross-dataset comparison between uniform distillation (UD) and CPGD under the original experimental setting. Table 10 further reports the stability of the complete CPGD configuration over seeds 1, 42, and 1024. All results use forced predictions at L_2, L_4, L_6, without applying an early-exit gate.

Table 9: Cross-dataset training impact of CPGD. "Avg. Early Acc." is the unweighted mean accuracy of forced predictions at L_2, L_4, L_6, and Δ denotes the improvement over UD.

Method	CUB	CIFAR-100	TinyImageNet	Places365
UD	38.08	57.53	52.11	36.04
CPGD	**42.98**	**61.68**	**54.66**	**37.51**
Δ	**+4.90**	**+4.15**	**+2.55**	**+1.47**

Table 10: Three-seed stability of CPGD. Values are the mean $\pm$ sample standard deviation over seeds 1, 42, and 1024. "Avg. Early Acc." is averaged over forced L_2, L_4, L_6 predictions.

Dataset	UD	CPGD	CPGD $-$ UD	Wins
CUB	37.92 ± 0.17	$\mathbf{42.59 \pm 0.78}$	$\mathbf{+4.68 \pm 0.64}$	3/3
CIFAR-100	57.92 ± 0.39	$\mathbf{62.30 \pm 0.57}$	$\mathbf{+4.38 \pm 0.20}$	3/3
TinyImageNet	52.35 ± 0.33	$\mathbf{55.01 \pm 0.37}$	$\mathbf{+2.66 \pm 0.48}$	3/3

Across the four datasets in Table 9, CPGD improves Avg. Early Acc. over UD by 1.47–4.90 points. The three-seed results in Table 10 further show that the improvement is not specific to a single initialization: CPGD outperforms UD in all $9/9$ evaluated dataset-seed pairs.

Group-Level Depth Preferences. We further examine whether semantic concept groups exhibit heterogeneous depth preferences. On CIFAR-100, $9/12$ groups (75.0%) prefer a non-final teacher depth, while on CUB $8/12$ groups (66.7%) prefer a non-final depth. The preferred layers are distributed across multiple intermediate exits rather than collapsing to L_{12}, providing additional evidence that different semantic groups benefit from different teacher depths.

D.2 Exit-Head Capacity Ablation

Table 11 presents a comparison between basic readout heads and the heterogeneous exit head on a representative dataset. All configurations share the same frozen backbone, concept vocabulary, exit positions, common concept-to-class classifier, and validation setup. The objective is to demonstrate that straightforward linear or MLP readouts can underfit shallow concept bottlenecks, whereas the heterogeneous head prevents exit-head capacity from becoming the primary limiting factor.

Table 11: Exit-head capacity ablation on CUB under the same uniform-distillation early-exit protocol. "Avg. Early Acc." is the unweighted mean forced prediction accuracy at L_2, L_4, L_6.

Exit head	Avg. Early Acc. ↑	L_{12} Acc. ↑	Notes
Linear probe	10.39	58.38	simple linear readout
MLP probe	5.80	42.71	shallow nonlinear readout
Heterogeneous head	**38.08**	**68.31**	LPH/GAH head family

D.3 Additional RCMax Analyses

We provide additional analyses of RCMax covering three-seed stability, sensitivity to calibration hyperparameters, robustness to validation-set composition, and comparison with simpler concept-aware gates.

Table 12: Three-seed stability of RCMax over MaxProb. Each entry reports the paired test-accuracy improvement of RCMax using the same CPGD checkpoint. Values are in percentage points.

Dataset	Val. Target	Seed 1	Seed 42	Seed 1024	Mean $\pm$ SD
CUB	1.73×	+1.05	+0.57	+1.00	+0.88 $\pm$ 0.27
CIFAR-100	2.15×	+0.11	+0.15	+0.03	+0.10 $\pm$ 0.06
TinyImageNet	2.15×	+0.16	+0.28	+0.19	+0.21 $\pm$ 0.06

Three-Seed Stability of RCMax. We further evaluate RCMax over seeds 1, 42, and 1024 using the same CPGD checkpoint as the corresponding MaxProb baseline. For each seed, RCMax calibration and gate thresholds are selected on the validation split at the specified target speedup and then evaluated on the held-out test split.

RCMax improves over MaxProb in all 9/9 dataset–seed comparisons, showing that its complementary deployment-side gain is consistent across independent training runs.

Sensitivity to Readiness Calibration. At closely matched target MAC speedups, validation-selected RCMax improves test accuracy over MaxProb by $+0.48$, $+0.10$, $+0.13$, and $+0.38$ points on CUB, CIFAR-100, TinyImageNet, and Places365, respectively. To further examine sensitivity, we fix $q = 0.5$ and top-$k = 16$, and vary the calibration weight λ within $[0.025, 0.15]$. The resulting gains over MaxProb range from $+0.24$ to $+1.05$ on CUB, from $+0.26$ to $+0.52$ on Places365, from 0.00 to $+0.16$ on TinyImageNet. These results indicate that RCMax is most useful when concept readiness provides information beyond class confidence, while remaining approximately neutral when the additional readiness signal is less informative.

Robustness to Validation-Set Composition. We repeat the complete RCMax calibration and threshold-selection procedure over 100 stratified resamples containing 80% of the validation set. This analysis uses the same trained model and does not involve retraining. For each resample, the normalization statistics, calibration weight, and exit thresholds are estimated from that resample alone. RCMax outperforms MaxProb in 96% of the resamples, with an average paired accuracy improvement of $+0.78 \pm 0.39$ points, indicating that the observed benefit is robust to moderate changes in validation-set composition.

Comparison with Simpler Concept-Aware Gates. We further compare RCMax with two simpler concept-aware alternatives on CUB. Global Readiness augments MaxProb with the normalized mean readiness over all concepts, while Predicted-class Readiness uses only concepts associated with the current top-1 prediction. RCMax instead uses the positive readiness gap between the predicted and runner-up classes.

Table 13: Comparison of RCMax with simpler concept-aware gates on CUB.

Gate	Test Acc. ↑	MAC Speedup ↑
MaxProb	63.54	1.554×
Global Readiness	63.64	1.553×
Predicted-class Readiness	63.58	1.538×
RCMax	**64.02**	**1.569**×

The simpler concept-aware scores provide only small changes relative to MaxProb, whereas RCMax achieves the highest accuracy and MAC-based speedup in this controlled comparison. This supports the use of the readiness gap, which measures whether the current concept bottleneck provides stronger evidence for the predicted class than for its strongest competing class.

Sample-Wise Exit Behavior. For the selected RCMax policy on CUB, 7.75%, 24.54%, 22.02%, 6.18%, 4.32%, and 35.18% of samples exit at L_2, L_4, L_6, L_8, L_{10}, and L_{12}, respectively. Thus, 54.31% of samples exit by L_6, while harder samples continue to deeper layers. This distribution shows that RCMax produces genuinely sample-adaptive exit decisions rather than collapsing to a single preferred depth.

D.4 Additional Concept-Level Evaluation

We provide additional concept-level analyses complementing the main semantic-preservation results in Section 4.4. We first provide a more detailed three-seed analysis of the expanded original-vocabulary evaluation, followed by a controlled evaluation using the official CUB attribute vocabulary, simulated concept interventions, and an auxiliary concept-space compatibility diagnostic.

Expanded Original-Vocabulary Evaluation. We retain the original 370-dimensional label-free CUB concept vocabulary and the existing trained checkpoints, and expand the semantic mapping to the official human-annotated CUB attributes. The expanded mapping covers 132 of the 370 original concepts (35.7%), compared with 44 concepts (11.9%) under the original strict matching protocol. After merging synonymous mappings, these 132 concepts correspond to 107 unique semantic targets. The mapping is constructed independently of model outputs, and no model is retrained for this semantic evaluation.

We evaluate UD and CPGD using forced predictions at L_2, L_4, and L_6 over seeds 1, 42, and 1024. Table 14 reports the corresponding three-seed semantic-alignment results.

Table 14: Three-seed semantic evaluation using 132 mapped concepts from the original 370-dimensional CUB vocabulary. Values are mean $\pm$ sample standard deviation over seeds 1, 42, and 1024. Δ is the paired CPGD–UD difference computed within each seed and then averaged.

Metric	UD	CPGD	Δ
AUROC	69.17 ± 0.86	$\mathbf{69.66 \pm 0.71}$	$\mathbf{+0.50 \pm 0.23}$
Weighted AUROC	68.69 ± 0.89	$\mathbf{69.18 \pm 0.68}$	$\mathbf{+0.49 \pm 0.23}$
mAP	30.01 ± 0.24	$\mathbf{30.16 \pm 0.41}$	$\mathbf{+0.14 \pm 0.36}$
BAcc	64.01 ± 0.44	$\mathbf{64.26 \pm 0.37}$	$\mathbf{+0.25 \pm 0.07}$

Together with the $+4.68$-point improvement in Avg. Early Acc. reported in Table 4, the results show that CPGD improves shallow predictive utility while the mean semantic-alignment metrics remain stable or improve slightly across all four measures.

The conclusion is unchanged after collapsing synonymous mappings to the 107 unique semantic targets. The paired improvements are $+0.47 \pm 0.22$ in AUROC, $+0.47 \pm 0.22$ in weighted AUROC, $+0.12 \pm 0.43$ in mAP, and $+0.21 \pm 0.03$ in balanced accuracy. This indicates that the observed trend is not driven by duplicated synonymous mappings.

Controlled Evaluation with Official CUB Attributes. As a complementary controlled diagnostic, we evaluate CPGD using the official CUB attribute vocabulary rather than the original label-free concept vocabulary. This setting provides substantially broader human-attribute coverage, although it should be viewed as a controlled semantic diagnostic rather than the original experimental setting. Among the 302 evaluable attributes, we compare UD and CPGD at forced L_2, L_4, and L_6 exits while keeping the full-depth endpoint fixed.

Averaged over the three early exits, CPGD improves AUROC from 68.57 to 68.86, weighted AUROC from 65.90 to 66.09, mAP from 20.32 to 20.81, and balanced accuracy from 62.90 to 63.04. The detailed per-exit results are reported in Table 15.

Table 15: Controlled concept-level evaluation using the official CUB attribute vocabulary. Results compare UD and CPGD at individual exits. The shared full-depth L_{12} endpoint is unchanged.

Exit	AUROC (UD / CPGD)	mAP (UD / CPGD)	BAcc. (UD / CPGD)
L_2	63.97 / **64.50**	16.93 / **17.44**	59.18 / **59.61**
L_4	69.47 / **69.88**	20.60 / **21.14**	63.68 / **63.80**
L_6	**72.29** / 72.20	23.42 / **23.85**	**65.82** / 65.70
L_{12}	81.22 / 81.22	37.69 / 37.69	73.64 / 73.64

The semantic metrics remain stable or improve slightly overall, while the full-depth representation is unchanged. This controlled experiment therefore provides additional evidence that the improvement

in shallow predictive accuracy is not accompanied by a systematic degradation in human-attribute alignment.

Simulated Concept Interventions. We additionally evaluate whether the resulting concept bottlenecks remain usable for concept-level intervention. We consider samples that are initially misclassified by both UD and CPGD and apply the same simulated intervention protocol with a maximum budget of five concept queries. Under this protocol, Guided Success@5 increases from 2.57% with UD to 3.57% with CPGD, while the Oracle Success@5 upper bound increases from 7.75% to 9.84%.

Table 16: Simulated concept-intervention results on initially misclassified CUB samples under the same five-query budget.

Method	Guided Success@5 ↑	Oracle Success@5 ↑
UD	2.57	7.75
CPGD	**3.57**	**9.84**

These results provide complementary evidence that CPGD preserves the usefulness of the intermediate concept bottleneck for intervention under this protocol. We emphasize that this is a simulated intervention diagnostic rather than a human-user study.

Concept-Space Compatibility Diagnostic. Finally, as an auxiliary representation-level diagnostic, we measure the z-score-normalized cosine similarity between shallow concept logits and the final-depth concept logits, averaged over forced exits at L_2, L_4, and L_6. Because the same concept-to-class classifier is shared across exits, compatibility with the mature final-depth concept representation provides an additional measure of shallow bottleneck recovery. The final-depth bottleneck is used only as a common reference, and this diagnostic is complementary to, rather than a substitute for, the human-attribute evaluations above.

Table 17: Concept-space compatibility diagnostic. We report z-score-normalized cosine similarity between shallow concept logits and final-depth concept logits, averaged over forced exits at L_2, L_4, and L_6. Values are multiplied by 100 for readability.

Dataset	UD	CPGD	Δ
CUB	91.11	**92.52**	**+1.40**
CIFAR-100	84.17	**86.42**	**+2.25**
TinyImageNet	76.86	**80.17**	**+3.31**
Places365	75.77	**80.93**	**+5.15**

Across all four datasets, CPGD increases shallow-to-final concept-space agreement relative to UD. Together with the direct human-attribute evaluations above, this auxiliary diagnostic supports the view that CPGD improves shallow concept bottlenecks while retaining compatibility with the mature concept representation used by the shared classifier.

D.5 Transfer to Newer CBM Families

We further examine whether the training-side benefit of CPGD transfers beyond the label-free CBM configuration used in the main experiments. We consider two newer CBM families, DN-CBM [22] and CFM [32], and evaluate the average forced prediction accuracy at L_2, L_4, and L_6. All results in this subsection use seed 42. These experiments isolate the training-side effect of CPGD and do not apply the dynamic RCMax gate.

DN-CBM on CIFAR-100. For DN-CBM, UD and CPGD use the same frozen OpenAI CLIP ViT-B/16 carrier, 4096-dimensional SAE concept representation, concept-to-class classifier, exit locations, data split, and full-depth initialization. The fixed L_{12} accuracy is 81.30%. Under this controlled setting, CPGD increases Avg. Early Acc. from 42.09% to 46.64%, corresponding to a $+4.55$-point improvement over UD.

CFM on Places365. For CFM, both variants use the same frozen CLIP-DINOiser ViT-B/16 LAION2B carrier, 8192-dimensional Matryoshka SAE concept representation, official linear probe, spatially max-pooled patch concepts, exit locations, data split, and full-depth initialization. The fixed L_{12} accuracy is 55.09%. CPGD increases Avg. Early Acc. from 23.54% to 26.11%, yielding a $+2.58$-point improvement.

Table 18: Training-side transfer of CPGD to newer CBM families. "Avg. Early Acc." is the unweighted mean Top-1 accuracy of forced predictions at L_2, L_4, and L_6.

Model / Dataset	UD	CPGD	Δ
DN-CBM / CIFAR-100	42.09	**46.64**	**+4.55**
CFM / Places365	23.54	**26.11**	**+2.58**

These results indicate that the training-side improvement from CPGD is not specific to the original label-free CBM instantiation used in the main experiments. Because these experiments use forced shallow exits rather than the complete dynamic PACE pipeline, we interpret them specifically as evidence of training-side transfer.

E Qualitative Examples of Progressive Concept Evidence

The qualitative case studies supplement the quantitative findings and should not be viewed as an independent interpretability benchmark. Their role is to show how multi-exit CBMs reveal concept evidence at various depths, and how intermediate bottlenecks can already provide strong support for a class before the final exit. For each image, we display the predicted class and the most highly activated concepts from L_2 through L_{12}. Highlighted concepts indicate the principal supporting concepts for the predicted class under the shared concept-to-class classifier.

Figure 4 presents two illustrative CUB cases. At shallower exits, the predicted concepts tend to be broad, generic, or only loosely aligned with the true class label. As we move deeper into the network, the predicted class becomes accurate, and the activated concepts grow more class-specific, such as *a pointed bill* for Belted Kingfisher and *a large, orange bill* or *a red spot on the beak* for White Pelican. The orange boxes mark the exits PACE/RCMax chose. In both cases, PACE stops before the final layer, once an intermediate bottleneck already yields concepts that substantively support the correct class. These examples align with the concept–depth preferences exploited by CPGD and highlight that early exiting in CBMs should account for intermediate evidence of concepts, rather than relying solely on the final prediction.

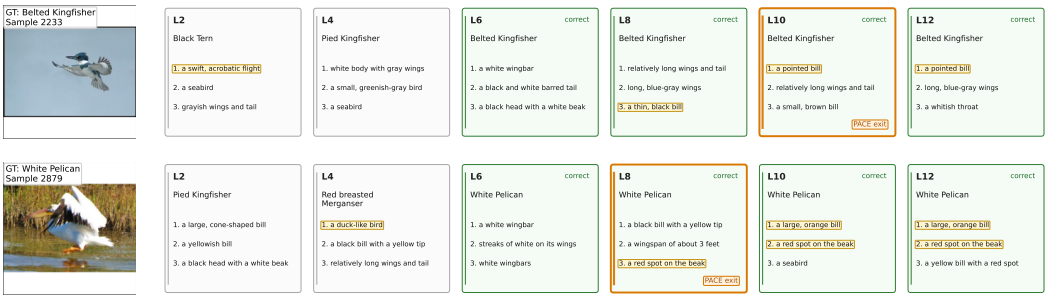


Figure 4: Qualitative examples of progressive concept evidence across exits. For each CUB image, we show the predicted class and top activated concepts from L_2 to L_{12}. Green borders mark exits whose prediction matches the ground truth; orange borders mark the exit chosen by PACE/RCMax. Highlighted concepts are key support concepts for the prediction. Early exits reveal coarse or generic attributes, while deeper exits show more class-specific concepts. PACE exits before the final layer when the intermediate concept bottleneck already provides sufficient support.

F Compute Resources

All experiments employ frozen CLIP/OpenCLIP vision backbones; only the concept probes, exit heads, concept-to-class classifiers, and the sparse teacher-selection parameters used in CPGD are trainable. We conducted the reported experiments on a server with Ubuntu, Intel Xeon Platinum 8352V CPUs, 256 GiB of system memory, and NVIDIA GeForce RTX 4090 GPUs with 24GB memory. For a single dataset with six exits, training typically took from several GPU-hours up to roughly one GPU-day, depending on dataset size and whether cached features were reused, not counting initial pilot and debugging runs. MAC measurements and gate-threshold sweeps were carried out offline using stored exit logits and were thus far less computationally demanding than training.

G Limitations and Broader Impacts

Limitations. PACE relies on accurate deeper exits because CPGD forms concept-specific teachers using the deeper bottlenecks. If these deeper exits yield poor concept estimates, the combined teacher signals may carry less useful information. As with standard CBMs, our multi-exit CBM exhibits an inherent trade-off between accuracy and interpretability due to its prediction through an explicit concept bottleneck. The concept layer provides human-interpretable intermediate evidence but also restricts the information available to the final classifier, potentially lowering accuracy relative to unconstrained black-box models. Accordingly, our comparisons emphasize methods that are adapted to the same CBM framework, rather than asserting direct speedups over their original black-box versions. RCMax relies on validation-set statistics for readiness normalization and calibration-weight selection, and thus may need to be recalibrated whenever the deployment distribution, concept set, or label space changes. We report efficiency in terms of cumulative MACs, a hardware-agnostic proxy for computational cost; actual wall-clock latency may vary with batching, memory access patterns, kernel implementations, and hardware details.

Broader Impacts. PACE aims to provide efficient and interpretable inference for concept bottleneck models. On the positive side, it can reduce inference cost and surface intermediate concept-level evidence, enabling users to better examine model behavior. However, risks arise if automatically induced concept vocabularies are incomplete, biased, or otherwise misleading, leading users to place undue confidence in concept-based explanations. For high-stakes use, practitioners should carefully vet the concept vocabulary, track calibration performance, and analyze failure modes under the target data distribution.