ReCoG: Relational Concept Graph for Training-Free Personalization

Changseon Yu
GSAIM
Chung-Ang University
ckdtjs358@cau.ac.kr

Hyeokjun Kweon
GSAIM
Chung-Ang University
hyeokjunkweon@cau.ac.kr

Abstract

Training-free personalization adapts a frozen vision-language model (VLM) to recognize user-defined concepts from only a few exemplars by retrieving concept entries at inference time. While recent retriever-based methods enrich each concept with descriptive evidence, they largely treat concepts in isolation, so the retrieved information can be plausible yet non-discriminative, especially when the personalized concept set contains highly confusable, fine-grained categories. We argue that personalization is inherently **relational**: correct recognition depends on the specific distinctions between a concept and its nearest alternatives. To this end, we propose **ReCoG**, a **Re**lational **Co**ncept **G**raph for retrieval-augmented personalization. ReCoG represents concepts as nodes and augments them with directed edges that explicitly capture discriminative cues between concept pairs, elicited by a frozen VLM during database construction. At inference, ReCoG retrieves a shortlist and resolves ambiguity via explicit pairwise comparisons conditioned on the corresponding relational edges, selecting the concept that consistently wins against competing candidates. We also introduce **FINGER** benchmark to test personalized recognition on confusable concept sets. Across FINGER, ReCoG substantially outperforms prior methods, with the largest gains in the most confusable regimes. ReCoG also achieves consistently best performance on established personalization benchmarks including captioning and VQA, confirming that relational evidence is broadly beneficial beyond fine-grained settings. Project Page: https://csy-98.github.io/ReCoG/

1 Introduction

Personalizing vision-language model (VLM) to user-specific concepts is a central capability for interactive AI [9, 17, 1, 19, 8, 11, 36, 7, 24, 22]. In many applications, users want a model to recognize or generate content consistent with a set of personal concepts, such as similar-looking products, given only a few examples. While fine-tuning can achieve high fidelity, it is often costly, slow, and difficult to repeat whenever new concepts are introduced.

This has motivated a line of training-free personalization methods [11, 8, 29, 15, 26, 16, 14]. This approach first builds a lightweight concept database, which comprises the sample images and features of the personal concepts. Here, the features are typically visual embeddings or text descriptions obtained by VLM, by using the sample images as inputs. At the time of inference, when a query is given, the relevant concepts are retrieved from the established concept database. The retrieved concepts are used as input to the frozen backbone VLM. Recent retriever-based approaches such as RAP [11] and R2P [8] have shown that this paradigm can be effective and scalable.

However, we find that such approaches focus on representations that primarily describe each concept in isolation, *i.e.*, without considering the other concepts. Even if plausible descriptions exist for

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

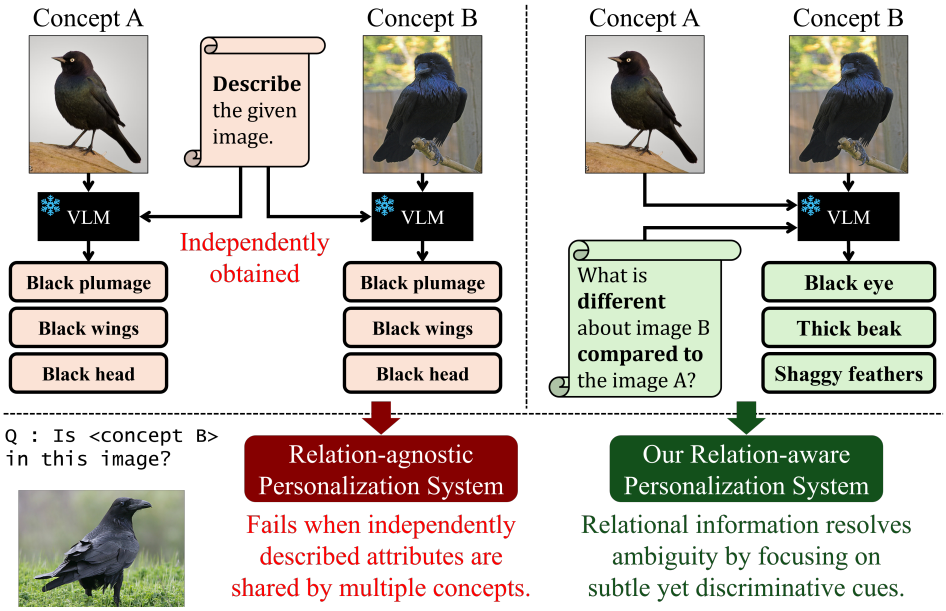

Figure 1: Illustration of why relations matter. Left: When VLM describes each concept independently, it produces salient yet generic attributes shared across confusable concepts, leading to ambiguity. Right: When VLM is conditioned on the inter-concept relation (*e.g.*, asking what differs between candidates), it can elicit discriminative cues that ease recognition even under fine-grained setting.

each concept, they are not guaranteed to be helpful for identifying that concept with respect to other concepts in the personalized set. As depicted in Fig. 1, a description may capture the most salient properties of a concept, but still misses the less salient yet important cues for recognition. These challenges become particularly severe in fine-grained regimes where concepts are highly similar. This suggests that concept identity in personalization should be formulated relationally rather than purely unary. What matters is not only a description, but also the **distinction** that makes discrimination.

In this light, we propose to consider **relations among concepts** for personalization more explicitly. Training-free personalization can become more discriminative and more robust, by eliciting contrastive evidence during database construction and by performing relative comparisons among competing candidates during retrieval. To this end, we propose **Re**lational **Co**ncept **G**raph (**ReCoG**), a novel training-free personalization method based on the relations among the concepts, by explicitly encoding and exploiting them. ReCoG represents concepts as nodes and introduces directed edges that store discriminative textual information specific to a concept pair. These edges capture the attributes and cues that are most useful for distinguishing two nearby concepts, instead of repeating generic descriptions that may be shared by multiple nodes.

At inference time, we shift the focus from retrieving the most similar node to retrieving the most helpful distinction for disambiguation. ReCoG performs pairwise comparisons between candidates using the relational edges associated with each pair, identifying which concept better matches the query under the relevant pair-specific distinctions. This relational structure allows the system to select evidence that is aligned with the actual ambiguities present in the query.

Furthermore, we introduce a FINe-Grained pERsonalization (**FINGER**) benchmark to evaluate fine-grained personalization. First, we employ CUB-200-2011 [31], a classic fine-grained bird dataset, and construct two regimes: CUB-Easy, formed by randomly sampling species, and CUB-Hard, formed by selecting highly similar species using CUB's attribute annotations to intentionally maximize inter-concept confusability. Second, we include a CLEVR-based task [13] where each concept corresponds to a scene configuration and images rendered from different viewpoints belong to the same concept, providing a controlled environment with systematically adjustable difficulty.

On FINGER benchmark, ReCoG remarkably outperforms prior approaches, demonstrating that explicitly encoding distinctions is crucial for overcoming fine-grained confusion where relation-agnostic descriptions often fail. In addition, on conventional benchmarks [1, 19, 8], ReCoG also

consistently achieves best performance on all tasks (recognition, image captioning, and VQA). This strongly confirms that ReCoG is a promising framework even under coarse-grained settings.

2 Related Works

2.1 Personalization on VLMs and MLLMs

Recent vision-language models (VLMs) and multimodal large language models (MLLMs) have shown strong capabilities in various multimodal tasks [35, 12, 2, 6, 10]. However, their knowledge is mostly generic. They can recognize common categories such as "dog", but they do not naturally recognize user-specific instances such as "user's own dog". Personalization of VLMs/MLLMs aims to incorporate user-provided concepts into downstream responses. Existing work has studied how frozen VLM representations can be extended to such concepts through learned embeddings, concept-specific modules, or lightweight alignment mechanisms [7, 1, 19, 22, 21]. Recent work further expands this direction from single-concept image personalization to multi-concept or online settings [3, 4, 34, 20, 5] and video applications [30, 37]. Despite their effectiveness, many of these approaches require additional optimization or auxiliary modules, which can limit scalability.

2.2 Training-Free Personalization

Training-free personalization avoids updating model parameters for each new user or concept. RAP [11] proposes a retrieval-augmented framework for MLLM personalization based on retrieval-augmented generation (RAG) [17, 9]. It stores user-specific concepts as a form of database, retrieves relevant concepts with a multimodal retriever, and injects the retrieved concept information into the MLLM input to generate personalized responses. R2P [8] further extracts descriptive texts for personalized concepts and compares retrieved candidate concepts via Chain-of-Thought (CoT) reasoning [32]. Recent training-free systems similarly build external or lightweight personal memories by storing descriptive textual metadata, visual prompts, or concept-representative visual tokens [27, 33, 28]. However, these methods primarily represent concepts independently using descriptive evidence. In contrast, ReCoG focuses on relations among personalized concepts, enabling VLM to perform more effective reasoning and discrimination of user-specific concepts.

2.3 Personalization Benchmarks

Early benchmarks focus on whether models can learn and use user-specific concepts for personalized tasks [7, 1, 19, 22]. Subsequent works extend it to compositional settings, including chimeric concept-context composition and multi-concept interactions [25, 8, 3]. More recent benchmarks further improve realism and scale by covering diverse modalities and concept types [27, 5, 4]. Despite these contributions, benchmarks with carefully controlled fine-grained distinctions among personalized concepts remain limited. For example, Yo'LLaVA++ introduced in Jarvis [33] targets fine-grained recognition, but it is limited to text modality. Since distinguishing subtle differences among user-specific visual concepts is crucial for VLM personalization, we introduce FINGER, a novel benchmark designed to evaluate fine-grained personalization across varying levels of difficulty.

3 Preliminaries

3.1 Problem Formulation

In this paper, we study a training-free personalization setting where a pre-trained VLM is adapted to recognize a user-defined set of concepts without updating its parameters. To enable a frozen VLM to handle novel concepts, we follow a two-stage paradigm commonly adopted in prior works:

(i) Build a lightweight database from a small support set per concept.

(ii) Retrieve relevant entries from the database to guide the VLM.

Let $\mathcal{C} = \{1, \ldots, N\}$ denotes the indices of N novel concepts $\{c_i\}_{i=1}^{N}$ for personalization. We assume that each concept is associated with a small support set of exemplars:

$$\mathcal{S}_i = \{(x_{i,m}^s, z_{i,m}^s)\}_{m=1}^{M_i}, \tag{1}$$

where $x^s_{i,m}$ is the m-th support image for the i-th concept. Notably, $z^s_{i,m}$ is optional information associated with $x^s_{i,m}$, such as textual description or embedding vectors. For simplicity, we focus on the single-exemplar case, *i.e.*, the number of exemplars $M_i = 1$ and omit the exemplar index m. The full personalization database is constructed by combining the entire support sets: $\mathcal{S} = \bigcup_{i=1}^{N} \mathcal{S}_i$.

This database serves as a source of personal knowledge at inference time, where the retriever selects the concept relevant to an input query. Specifically, given a query image x^q with an unknown concept label $y \in \mathcal{C}$, the goal is to predict the correct concept. To this end, `Retriever` returns the concept that is most relevant to the query from the database $\mathcal{S}$:

$$\hat{y} = \texttt{Retriever}(x^q, \mathcal{S}), \tag{2}$$

where $\hat{y} \in \mathcal{C}$ denotes the top-1 retrieved concept.

3.2 Conventional Relation-Agnostic Approaches

To the best of our knowledge, existing retrieval-based personalization methods construct the database $\mathcal{S}$ without considering relations among concepts. We refer to these approaches as "Relation-Agnostic". In such methods, each concept is represented independently by a standalone entry. We review representative relation-agnostic instantiations under the formulation in Sec. 3.1.

A widely used strategy is to employ a pre-trained CLIP model. Let $\texttt{CLIP}_v$ denote the CLIP visual encoder. For each concept c_i, we compute and store the CLIP embedding of its support image:

$$f^s_i = \texttt{CLIP}_v(x^s_i) \quad \text{and} \quad z^s_i \leftarrow \{f^s_i\}. \tag{3}$$

At inference time, the stored embeddings are used for retrieval. Specifically, given a query image x^q, we compute its embedding $f^q = \texttt{CLIP}_v(x^q)$ and perform nearest-neighbor search over the database:

$$\hat{y} = \texttt{Retriever}(x^q, \mathcal{S}) = \underset{i \in \mathcal{C}}{\operatorname{argmax}} \ \operatorname{sim}(f^q, f^s_i)\,, \tag{4}$$

where $\operatorname{sim}(\cdot, \cdot)$ can be a cosine similarity or a negative Euclidean distance.

This approach pre-computes an embedding for each concept during database construction, so inference reduces to a simple nearest-neighbor retrieval, making it computationally efficient. However, the representational capacity of a single CLIP feature can be limited for fine-grained personalization. Consequently, many works have sought to enrich each concept with more informative descriptions.

For example, R2P [8] further augments each concept with textual information extracted by a VLM, in addition to the CLIP embedding. Given the support image x^s_i for concept c_i, R2P employs a frozen VLM to generate a concise text that well describes the concept, and the database is expanded as:

$$t^s_i = \texttt{VLM}(x^s_i | P_{\text{desc}}) \quad \text{and} \quad z^s_i \leftarrow \{f^s_i, t^s_i\}. \tag{5}$$

where P_{desc} denotes the prompt for obtaining the descriptive text.

At inference time, R2P follows a two-stage pipeline. First, it retrieves a candidate set $\mathcal{K} \subset \mathcal{C}$ by selecting the top-K nearest concepts under CLIP similarity. Second, R2P re-ranks the candidates by asking a frozen VLM to verify whether the query image x^q belongs to each candidate, conditioned on both the support image x^s_i and the textual descriptions t^s_i:

$$(\ell^{\text{Yes}}_i, \ell^{\text{No}}_i) = \texttt{VLM}(x^q, x^s_i, t^s_i | P_{\text{belong}}), \tag{6}$$

where P_{belong} denotes the verification prompt that instructs the model to answer in a `Yes/No` format. R2P then selects the concept with the highest confidence for `Yes` among the candidates:

$$\hat{y} = \texttt{Retriever}(x^q, \mathcal{S}) = \underset{i \in \mathcal{K}}{\operatorname{argmax}} \ \frac{\ell^{\text{Yes}}_i}{\ell^{\text{Yes}}_i + \ell^{\text{No}}_i}. \tag{7}$$

4 Methods

4.1 Importance of Relation Among Concepts in Personalization

As discussed above, prior works have focused on improving the *descriptiveness* of per-concept representations (by using CLIP embeddings or augmenting them with VLM-generated textual attributes).

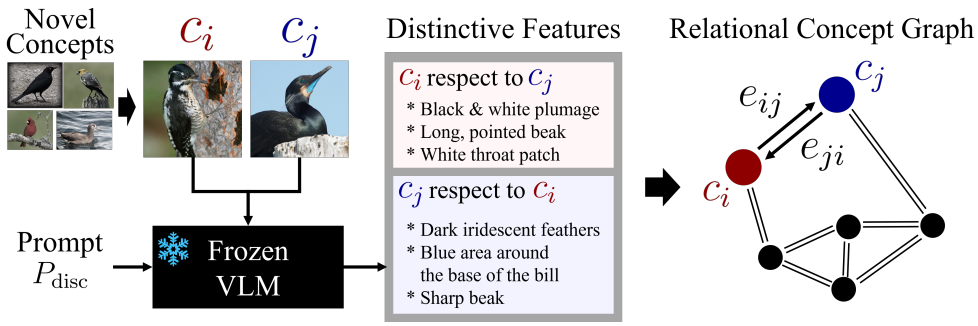

Figure 2: Given a set of novel concepts, we use a frozen VLM with a discriminative prompt P_{disc} to elicit directed contrastive evidence between concept pairs. For each ordered pair (c_i, c_j), the VLM summarizes what makes c_i different from c_j, producing an edge e_{ij} (and vice versa). Collecting these directed edges yields the **Relational Concept Graph**, which provides explicit inter-concept relations.

However, being descriptive and being discriminative are not always aligned. Even if plausible descriptions exist for each concept, they are not guaranteed to be *discriminative* with respect to other concepts in the personalized set. A description may capture most salient properties of a concept while still missing the few attributes that distinguish it from its closest alternatives.

In fine-grained regimes, *i.e.*, when the concepts to be personalized are highly similar to one another, this issue becomes particularly severe. For example, in **CUB**, many bird species share highly similar global appearances (*e.g.*, "a dark bird with a streaked breast"). Descriptive approaches can easily focus on the *most salient* attributes that characterize each species in isolation, while overlooking subtle yet crucial cues that separate visually confusable species (*e.g.*, slight differences in head markings, wing-bar patterns, or bill shape). In short, within the context of personalization, information extracted independently per concept can be descriptive yet insufficient to ensure strong separability.

A similar limitation also appears in the retrieval phase. Most existing methods assess, for each concept independently, how likely the query belongs to that concept, and then select the concept with the highest score. However, this strategy implicitly assumes that the resulting scores are well-calibrated and comparable across concepts. In practice, both CLIP similarities and VLM-based verification scores can be sensitive to prompt phrasing, visual context, and concept-specific ambiguities, making their absolute scale unreliable for global ranking. Consequently, choosing the maximum among independently computed confidences can be brittle, especially under fine-grained situation.

We argue that both limitations stem from ignoring *relations among concepts*. If database construction and retrieval are explicitly conditioned on inter-concept relations—*e.g.*, by eliciting contrastive evidence during description generation and by performing relative comparisons among competing candidates during retrieval—personalization can become more discriminative and more robust. Motivated by this insight, we propose Relational Concept Graph (**ReCoG**), which incorporates concept relations as an important signal throughout both database construction and retrieval.

4.2 Constructing Relation-Aware Database

For each concept c_i, we compute the CLIP embedding f_i^s of its support image x_i^s and store it, as in Eq. (3). The key difference of ReCoG is that, in addition to per-concept information, we explicitly store **relations among concepts**. To this end, we define a directed concept graph: $\mathcal{G} = (\mathcal{C}, \mathcal{E})$, where each node corresponds to a concept and edges encode their discriminative relation.

For a pair of concepts (c_i, c_j), we ask a VLM to describe what makes c_i different from c_j, as shown in Fig. 2. We denote the resulting textual evidence as a directed edge e_{ij}:

$$e_{ij} = \texttt{VLM}(x_i^s, x_j^s \mid P_{\text{disc}}), \tag{8}$$

where P_{disc} is a prompt for eliciting discriminative evidence (Sec. D). Note that e_{ij} and e_{ji} are generally different, as they are produced from opposite perspectives: e_{ij} describes the difference from the viewpoint of c_i, whereas e_{ji} describes it from the viewpoint of c_j. In practice, we obtain both edges at once, from a single VLM comparison for each unordered concept pair.

We then augment each concept entry with its outgoing discriminative edges:

$$\mathcal{S}_i = \{(x_i^s, z_i^s)\} \quad \text{where} \quad z_i^s \leftarrow \Big\{ f_i^s,\ \{e_{ij}\}_{j \in \mathcal{A}(i)} \Big\}, \tag{9}$$

where $\mathcal{A}(i) = \mathcal{C} \setminus \{i\}$ is the set of concepts for which we store outgoing edges. By explicitly storing how concepts differ from one another, the database captures evidence that is directly useful for discrimination, rather than describing each concept in isolation.

4.3 Discriminative Retrieval via Explicit Comparison

At inference time, we begin by pruning the search space from $\mathcal{C}$ to a candidate set $\mathcal{K} \subset \mathcal{C}$ using CLIP similarity, similar to R2P. Within $\mathcal{K}$, the candidates are close in the CLIP space, so distinguishing them requires more discriminative evidence. Here, we perform more fine-grained retrieval using the constructed graph-based database. Specifically, ReCoG directly performs **relative comparisons** between candidates, instead of independently scoring each candidate.

For each unordered pair (u, v) in $\mathcal{K}$, we ask a frozen VLM to decide whether the query is closer to c_u or c_v, conditioned on the relational edges e_{uv} and e_{vu}:

$$(\ell_{u \succ v}, \ell_{v \succ u}) = \texttt{VLM}(x^q, x_u^s, x_v^s, e_{uv}, e_{vu} \mid P_{\text{comp}}), \tag{10}$$

where P_{comp} is a prompt asking the model to perform a pairwise comparison (Sec. D). Here, $\ell_{u \succ v}$ denotes the logit indicating that the query is closer to c_u than c_v, and vice versa. We use both e_{uv} and e_{vu} since both perspectives provide complementary evidence. The winner of this comparison is

$$w(u, v) = \begin{cases} u, & \ell_{u \succ v} > \ell_{v \succ u}, \\ v, & \text{otherwise.} \end{cases} \tag{11}$$

We then aggregate match outcomes in a round-robin manner:

$$W_u = \sum_{v \in \mathcal{K} \setminus \{u\}} \mathbf{1}\big[w(u, v) = u\big], \tag{12}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. Ties are possible in principle; in such cases, we select the one with the highest CLIP similarity. Otherwise, the concept with the largest number of wins is returned:

$$\hat{y} = \texttt{Retriever}(x^q, \mathcal{S}) = \operatorname*{argmax}_{u \in \mathcal{K}} W_u. \tag{13}$$

4.4 Sparse ReCoG for Computational Efficiency

Constructing relational edges for all concept pairs requires $\mathcal{O}(N^2)$ VLM calls during database construction. Although this cost is incurred only once (offline), we additionally consider a simple sparse variant to reduce this overhead. Instead of using the full adjacency $\mathcal{A}(i) = \mathcal{C} \setminus \{i\}$, we construct $\mathcal{A}(i)$ using CLIP similarity among support images. Specifically, for each concept c_i, we select its top-k nearest concepts as relational neighbors:

$$\mathcal{A}(i) = \text{TopK}_{j \in \mathcal{C} \setminus \{i\}} \, \text{sim}(f_i^s, f_j^s). \tag{14}$$

We then construct relational edges only for concept pairs that appear in at least one selected adjacency set. This yields at most Nk pair proposals and often fewer actual VLM calls (due to duplication), while focusing on visually similar concepts where discriminative evidence is most likely to be useful.

At inference time, sparse construction may leave some candidate pairs without precomputed relational edges. There are two possible ways to handle such missing edges. The first option is to generate the missing edge on demand using the same discriminative prompt as in database construction. This preserves the full pairwise comparison behavior for the encountered pair, at the cost of an additional inference-time VLM call; once generated, the edge can also be cached into the database for future use. The second option is to avoid any additional VLM call and treat the missing comparison as a tie, assigning no win to either candidate. This may introduce a small performance drop because some pairwise comparisons are skipped during round-robin, but it keeps inference strictly within the precomputed sparse graph. In our experiments (Sec. 6.6), we find that this simple no-call strategy already provides a favorable trade-off between computational cost and recognition accuracy.

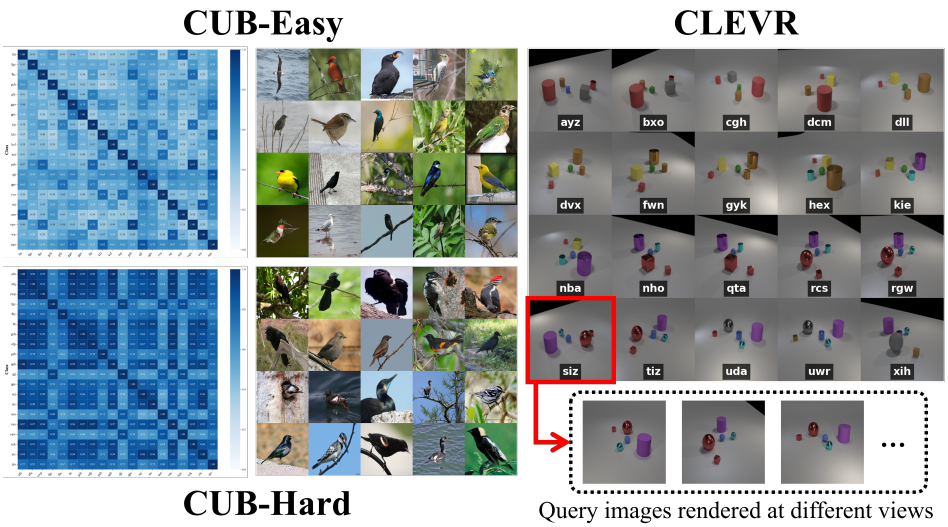

Figure 3: The proposed FINGER Benchmark. **Left**: 20 concepts in CUB-Easy (top) and CUB-Hard (bottom). We also provide the pairwise concept similarity matrices computed from CUB's visual attribute vectors (darker means higher similarity, *i.e.*, more difficult). **Right**: 20 concepts in CLEVR, where each concept corresponds to a specific scene configuration.

5 FINGER Benchmark

In practical personalization scenarios, the challenge is often fine-grained: users may want an AI system to distinguish between highly similar concepts that differ only in subtle visual details (*e.g.*, closely related species or visually similar parts in manufacturing). Motivated by this, we introduce a **FINe-Grained pERsonalization (FINGER)** Benchmark that explicitly targets concept sets with high visual confusability, as in Fig. 3.

The first task is based on CUB-200-2011 dataset [31], which contains 200 bird species with subtle inter-class variations. A useful property of CUB is that each image has a vector of semantic visual attributes (*e.g.*, body color), enabling controlled construction of challenging concept sets. Using CUB, we build two tasks with different levels of difficulty. Each task contains $N = 20$ concepts; for each concept, we provide one support image and nineteen query images, resulting in 400 images per task.

We construct **CUB-Easy** task by sampling 20 species at random from the 200 classes. We therefore refer to this task as "Easy," although it remains challenging. In contrast, **CUB-Hard** task is intentionally designed to be more confusable. Specifically, we select visually similar concepts by leveraging CUB attributes, for example by focusing on species that score highly along attributes related to `black` appearance. CUB-Hard matches CUB-Easy in configuration, but is considerably more difficult: the system must retrieve the correct concept from a set of visually similar, predominantly black birds given a query that is itself a black bird.

We also introduce a **CLEVR**-based task to provide a more controlled environment [13]. CLEVR framework supports compositional scene generation with objects of different shapes, colors, sizes, and materials. We treat an entire scene configuration (*i.e.*, a set of objects and their spatial arrangement) as a concept, and regard images rendered from different viewpoints as belonging to that concept. For this task, we use 20 concepts with 1 support image and 9 query images per concept (200 images total). This benchmark is particularly useful because its difficulty can be systematically adjusted by controlling the degree of overlap in scene attributes. More details are provided in Appendix (Sec. A).

6 Experimental Results

6.1 Setup

Datasets/Metrics. In addition to our FINGER benchmark, we evaluate ReCoG on three established benchmarks: MyVLM (29 concepts) [1], Yo'LLaVA (40 concepts) [19], and PerVA (329 concepts) [8]. For recognition task, we report positive accuracy (P.Acc.), measured on positive queries that include

Table 1: Recognition and captioning performance on the proposed FINGER benchmark. Baseline denotes the setting using the relation-agnostic description (the same as R2P [8]) and performs retrieval in a pairwise manner in Sec. 4.3. ReCoG uses the relation-aware concept graph and pairwise retrieval.

Methods	CUB-Easy				CUB-Hard				CLEVR-Default			
	P.Acc	N.Acc	Wtd.	Recall	P.Acc	N.Acc	Wtd.	Recall	P.Acc	N.Acc	Wtd.	Recall
R2P [8]	87.9	99.4	93.6	76.1	63.8	98.1	81.0	55.0	61.1	98.0	79.6	24.4
Baseline	50.0	97.3	73.7	65.0	41.8	96.9	69.4	47.9	51.7	97.5	74.6	14.8
ReCoG (**Ours**)	**96.3**	**99.8**	**98.1**	**80.8**	**86.3**	**99.3**	**92.8**	**62.1**	**87.2**	**99.3**	**93.3**	**41.7**

the target concept, and negative accuracy (N.Acc.), measured on negative queries that exclude the target concept. As same as R2P [8], for each target concept, we exhaustively evaluates images from all other concepts as negative examples. However, we measured the negative accuracy of the Baseline and ReCoG from their top-1 retrieval results, as the goal was to evaluate the output of the personalization retriever. Wtd. denotes the uniformly weighted average of them: (P.Acc.+N.Acc.)/2. For captioning and VQA tasks, we report hard recall and accuracy, respectively, following R2P [8].

Implementation. We construct the database using CLIP [23] ViT-L. We use MiniCPM-o-2.6 [35] as VLM by default. Results with LLaVA [18] are in Appendix (Sec. E). Similar to R2P [8], we construct database using a single reference image per concept, without negative samples. We conduct our experiments on a single RTX Pro 6000 GPU.

6.2 Results on FINGER Benchmark

Table 1 reports the results on our FINGER Benchmark, and it cleanly illustrates why relational understanding is the key component in training-free personalization. **ReCoG dominates across all tasks and metrics**. On Wtd., ReCoG achieves 98.1% on CUB-Easy, 92.8% on CUB-Hard, and 93.3% on CLEVR, outperforming the strongest prior work R2P [8] by +4.5, +11.8, and +13.7 percent points, respectively. The gains are even larger on P.Acc. This is notable because our benchmark comprising highly confusable concepts, yet ReCoG still maintains substantial performance.

These results also highlight how the conventional relation-agnostic personalization approach struggles with fine-grained confusion. R2P [8] is a substantially strong method that performs well on conventional benchmarks (refer to Table 3). On FINGER benchmark, however, it faces difficulty due to the visually confusing concepts (*e.g.*, 63.8% P.Acc. on CUB-Hard and 61.1% P.Acc. on CLEVR). This behavior is consistent with our motivation: concept descriptions generated in isolation are often plausible but fail to identify the specific distinctions needed to separate a concept from its closest confounders.

Finally, the comparison between R2P [8] and Baseline isolates the role of retrieval strategy. Our Baseline uses the same descriptive features as R2P [8] but replaces separated verification with comparison-based retrieval. If pairwise retrieval alone were the main source of gain, we would expect a meaningful improvement over R2P [8] instead, Baseline is worse. This implies that pairwise comparison is not always beneficial when the evidence is not aligned with the real ambiguity.

Fig. 4 shows ReCoG's superiority intuitively. The left confusion matrix corresponds to Baseline, and the right matrix corresponds to ReCoG. The baseline exhibits substantial off-diagonal mass. This pattern is consistent with the nature of CUB-Hard, where concepts share very similar global appearance. In contrast, ReCoG produces a near-diagonal structure, indicating that ReCoG consistently reduces confusion across the entire fine-grained concept set. More qualitative results of ReCoG on FINGER benchmark can be found in Appendix (Sec. G).

Table 2 further provides a controlled analysis of CLEVR difficulty. Here, K denotes scene complexity and hop represets distance between the concepts (i.e., 1-hop is more fine-grained). For more details, please refer to Appendix (Sec. A). As K increases, all methods generally suffer a drop in positive accuracy, indicating that larger scenes introduce more distractors and make personalized matching harder. Nevertheless, ReCoG consistently outperforms R2P and the Baseline. The margin is especially pronounced in the more confusable 1-hop regime, where concepts differ only by a small local change. Even in the relatively easier 2-hop setting, ReCoG maintains a clear advantage, reaching 81.4% compared with 68.4% for R2P [8] and 40.0% for the Baseline.

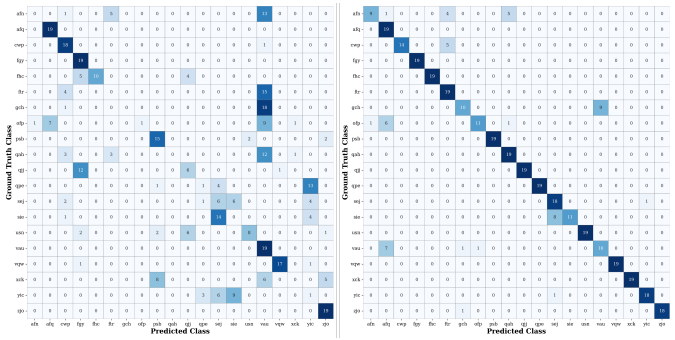

Figure 4: Conf. matrices of Baseline (left) and ReCoG (right).

K	Methods	1-hop	2-hop
4	R2P	61.7 / 98.0	75.0 / 98.7
	Baseline	52.8 / 97.5	68.3 / 98.3
	ReCoG	**91.4 / 99.5**	**91.7 / 99.6**
5	R2P	53.8 / 97.6	72.2 / 98.5
	Baseline	36.1 / 96.6	50.5 / 97.4
	ReCoG	**76.6 / 98.8**	**82.2 / 99.1**
6	R2P	46.6 / 97.2	68.4 / 98.3
	Baseline	30.0 / 96.3	40.0 / 96.8
	ReCoG	**74.4 / 98.7**	**81.4 / 99.0**

Table 2: P.Acc/N.Acc comparison across different K and hop distances on CLEVR task.

Table 3: Recognition/captioning performance on [7, 19, 8] and VQA performance on Yo'LLaVA

Methods	MyVLM[1]				Yo'LLaVA[19]				PerVA[8]			
	P.Acc	N.Acc	Wtd.	Recall	P.Acc	N.Acc	Wtd.	Recall	P.Acc	N.Acc	Wtd.	Recall
MyVLM	96.6	90.9	93.8	0.02	-	-	-	-	66.0	58.5	62.2	-
Yo'LLaVA	97.0	95.7	96.4	0.1	94.9	89.8	92.4	0.2	75.1	69.0	72.0	-
RAP	94.4	98.8	96.6	88.0	86.0	99.2	92.2	79.7	92.9	85.2	89.0	64.1
R2P	96.3	98.5	97.4	91.4	91.1	97.7	94.4	87.1	90.2	92.8	91.8	72.5
ReCoG	**98.6**	**99.9**	**99.3**	**92.9**	**95.0**	**99.9**	**97.4**	**90.4**	**97.7**	**99.9**	**98.8**	**73.7**

Methods	Accuracy
Yo'LLaVA [19]	92.9
RAP [11]	93.2
GPT-4V+VP [19]	86.6
LLaVA [19]	89.9
R2P [8]	96.5
ReCoG	**97.0**

6.3 Results on Conventional Personalization Benchmarks

Table 3 reports results on three established benchmarks (MyVLM [1], Yo'LLaVA [19], and PerVA [8]), comparing ReCoG against both training-based [1, 19] and training-free [11, 8] approaches. ReCoG achieves the best performance across all tasks (recognition, captioning, and VQA) and all benchmarks, consistently outperforming prior methods under the same evaluation protocol. The results indicate that the relational evidence encoded by ReCoG is not only beneficial in our challenging FINGER benchmark, but also transfers to conventional personalization scenarios where concepts are less confusable, yielding substantial performance even on the relatively saturated benchmarks.

6.4 Analysis on Sparse ReCoG

We evaluate sparse ReCoG on CUB-Hard to analyze the trade-off between database construction cost and recognition accuracy. In this experiment, we use the tie strategy in Sec. 4.4: if a candidate pair does not have precomputed relational edges, the comparison is treated as a tie during round-robin aggregation. Alternatively, if additional inference-time VLM calls are acceptable occasionally, missing edges can be constructed online. This yields the same behavior as full ReCoG.

As shown in Fig. 5, increasing k rapidly improves performance with only a small number of additional VLM calls. Even with $k = 1$, Sparse ReCoG already outperforms R2P [8] while requiring only 13 VLM calls for graph construction. More notably, with $k = 2$, it reaches 85.0% P.Acc. using only 26 VLM calls, which is close to the full ReCoG performance of 86.3% obtained with all 190 pairwise edges. After this point, the performance largely saturates, suggesting that most useful relational evidence comes from a small set of visually similar neighboring concepts. These results indicate that sparse ReCoG could provide a practical cost-accuracy trade-off.

6.5 Isolating Relational Evidence from Pairwise Retrieval

ReCoG combines two components: relation-aware evidence construction (Sec. 4.2) and explicit pairwise retrieval (Sec. 4.3). To examine their contributions, we conduct an ablation on CUB-Hard by varying how the textual evidence is obtained and how the retrieved candidates are scored. We compare independently generated concept descriptions with pair-specific relational evidence, and candidate-wise verification with pairwise comparison followed by round-robin aggregation.

As shown in Table 4, R2P [8] uses concept-wise descriptions and verifies each candidate independently, whereas the Baseline reported in Table 1 uses the same descriptions with ReCoG's pairwise retrieval procedure. The Baseline performs worse than R2P (41.8% vs. 63.8%), indicating that pairwise comparison does not necessarily improve recognition when conditioned on independently generated descriptions. To evaluate relational evidence separately from pairwise retrieval, we also consider a candidate-wise variant that aggregates the outgoing edges $\{e_{ij}\}_{j \in \mathcal{K} \setminus \{i\}}$ for each retrieved candidate c_i. The resulting evidence is used to verify each candidate independently, following the scoring procedure of R2P. This variant improves P.Acc. to 74.0%, suggesting that contrastively generated evidence is more discriminative even without pairwise retrieval.

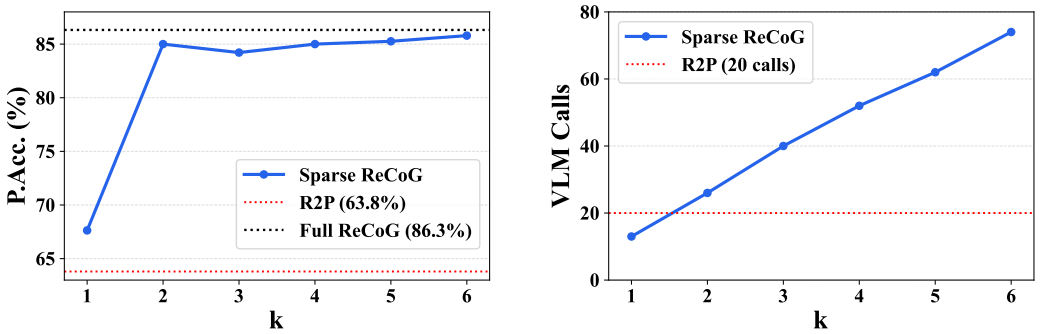

Figure 5: Effect of sparse graph construction on CUB-Hard. Left: P.Acc. of ReCoG when relational edges are sparsely constructed. Right: the number of VLM calls required for sparse database construction after removing duplicated concept pairs. Red dashed lines are those of R2P [8].

Table 4: Effect of textual evidence and retrieval strategy on CUB-Hard. All results are P.Acc.(%)

Textual evidence	**Candidate-wise verification**	**Pairwise comparison**
Concept-wise descriptions	R2P [8] (63.8)	Baseline (41.8)
Pair-specific relational evidence	Relational verification (74.0)	**ReCoG (86.3)**

6.6 Scalability to Large and Dynamic Concept Libraries

To further analyze the scalability, we conducted an additional large-scale experiment using all 200 species in CUB, treating each species as a distinct concept. For the missing edges in Sparse ReCoG, we consider two strategies. In the tie strategy, a candidate pair without a precomputed relation edge is treated as a tie. Consequently, Sparse ReCoG may perform fewer than pairwise comparisons. For example, the tie strategy requires an average of 2.46 VLM calls per query (2.79 s), compared with 3 calls for R2P [8] (3.83 s). In the on-demand strategy, a missing relation is generated when the corresponding candidate pair is encountered for the first time and is then cached for all subsequent queries. Under our default setting of $M = 3$, this requires at most three additional calls. However, the same pair never requires another construction call once its relation has been cached. In our 200-concept experiment, the on-demand strategy requires an average of 3.281 VLM calls per query (3.86 s) during the first evaluation pass, and this decreases to 3 calls per query after caching. Even with $K_{\text{graph}} = 1$ and the no-call tie strategy, Sparse ReCoG (56.2%) outperforms R2P [8] (48.6%) in P.Acc., while requiring fewer VLM calls in both offline database construction and online inference.

7 Conclusion

We presented ReCoG, a training-free personalization framework that explicitly models relations among user-defined concepts. ReCoG constructs a Relational Concept Graph whose directed edges encode pair-specific discriminative evidence. At inference time, ReCoG leverages these relational edges to perform explicit pairwise comparisons among retrieved candidates. We further introduced FINGER, a benchmark designed to evaluate personalized recognition under controlled fine-grained difficulty. Extensive experiments show that ReCoG substantially improves over prior training-free personalization methods on FINGER, especially in the most confusable regimes.

Limitation. Although sparse ReCoG shows that the construction cost can be substantially reduced while maintaining strong performance, scalability may still become challenging when the number of concepts grows significantly or when the database is updated online. Developing more adaptive and efficient graph sparsification strategies is therefore an interesting direction for future work.

Acknowledgements and Disclosures

This research was supported by the ANCHOR through the Seoul ANCHOR Center, funded by the Ministry of Education (MOE) and the Seoul Metropolitan Government (2026-ANCHOR-01-024-06, 30%), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2026-25510069, 20%), and Institute of Information & communications Technology Planning & Evaluation (IITP, AI Computing Support Project for R&D, 50%) grant funded by the Korea government(MSIT).

References

[1] Alaluf, Y., Richardson, E., Tulyakov, S., Aberman, K., Cohen-Or, D.: Myvlm: Personalizing vlms for user-specific queries. In: European Conference on Computer Vision. pp. 73–91. Springer (2024)

[2] Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022)

[3] An, R., Yang, S., Zhang, R., Lu, M., Jiang, T., Zeng, K., Luo, Y., Cao, J., Liang, H., Chen, Y., et al.: Mc-llava: Multi-concept personalized vision-language model. arXiv preprint arXiv:2411.11706 (2024)

[4] An, R., Yang, S., Zhang, R., Shen, Z., Lu, M., Dai, G., Liang, H., Guo, Z., Yan, S., Luo, Y., et al.: Unictokens: Boosting personalized understanding and generation via unified concept tokens. arXiv preprint arXiv:2505.14671 (2025)

[5] Bai, H., Wang, R., Du, Z., Zhao, Y., Zhang, F., Chen, H., Zhu, X., Zheng, B., Zhao, X.: Online-pvlm: Advancing personalized vlms with online concept learning. arXiv preprint arXiv:2511.20056 (2025)

[6] Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)

[7] Cohen, N., Gal, R., Meirom, E.A., Chechik, G., Atzmon, Y.: "this is my unicorn, fluffy": Personalizing frozen vision-language representations. In: European conference on computer vision. pp. 558–577. Springer (2022)

[8] Das, D., Talon, D., Wang, Y., Mancini, M., Ricci, E.: Training-free personalization via retrieval and reasoning on fingerprints. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9683–9692 (2025)

[9] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., Wang, H., et al.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 **2**(1), 32 (2023)

[10] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)

[11] Hao, H., Han, J., Li, C., Li, Y.F., Yue, X.: Rap: Retrieval-augmented personalization for multimodal large language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14538–14548 (2025)

[12] Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European conference on computer vision. pp. 709–727. Springer (2022)

[13] Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2901–2910 (2017)

[14] Kim, J., Kim, W., Park, W., Do, J.: Mmpb: It's time for multi-modal personalization (2025), https://arxiv.org/abs/2509.22820

[15] Kim, M., il Lee, K., Joo, S., Lee, H., Thonet, T., Jung, K.: Drift: Decoding-time personalized alignments with implicit user preferences (2025), https://arxiv.org/abs/2502.14289

[16] Lee, K., Park, J., Gim, J., Choi, W., Hwang, K., Kim, J., Im, S.: A training-free style-personalization via svd-based feature decomposition (2025), https://arxiv.org/abs/2507.04482

[17] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems **33**, 9459–9474 (2020)

[18] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)

[19] Nguyen, T., Liu, H., Li, Y., Cai, M., Ojha, U., Lee, Y.J.: Yo'llava: Your personalized language and vision assistant. Advances in Neural Information Processing Systems **37**, 40913–40951 (2024)

[20] Oh, Y., Yu, S., Park, J., Moon, H.C., Mok, J., Yoon, S.: Contextualized visual personalization in vision-language models. arXiv preprint arXiv:2602.03454 (2026)

[21] Pham, C., Phan, H., Doermann, D., Tian, Y.: Personalized large vision-language models. arXiv preprint arXiv:2412.17610 (2024)

[22] Pi, R., Zhang, J., Han, T., Zhang, J., Pan, R., Zhang, T.: Personalized visual instruction tuning. arXiv preprint arXiv:2410.07113 (2024)

[23] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)

[24] Richardson, C., Zhang, Y., Gillespie, K., Kar, S., Singh, A., Raeesy, Z., Khan, O.Z., Sethy, A.: Integrating summarization and retrieval for enhanced personalization via large language models. arXiv preprint arXiv:2310.20081 (2023)

[25] Rosasco, A., Berti, S., Pasquale, G., Malafronte, D., Sato, S., Segawa, H., Inada, T., Natale, L.: Concon-chi: Concept-context chimera benchmark for personalized vision-language tasks. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 22239–22248 (2024)

[26] Rout, L., Chen, Y., Ruiz, N., Kumar, A., Caramanis, C., Shakkottai, S., Chu, W.S.: Rb-modulation: Training-free personalization of diffusion models using stochastic optimal control (2024), https://arxiv.org/abs/2405.17401

[27] Seifi, S., Dorovatas, V., Reino, D.O., Aljundi, R.: Personalization toolkit: Training free personalization of large vision language models. arXiv preprint arXiv:2502.02452 (2025)

[28] Seifi, S., Gardier, S., Dorovatas, V., Reino, D.O., Aljundi, R.: Ego: Embedding-guided personalization of vision-language models. arXiv preprint arXiv:2603.09771 (2026)

[29] Shi, J., Xiong, W., Lin, Z., Jung, H.J.: Instantbooth: Personalized text-to-image generation without test-time finetuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8543–8552 (2024)

[30] Shi, Y., Yan, W., Xu, G., Li, Y., Chen, Y., Li, Z., Yu, F., Li, M., Yeo, S.Y.: Pvchat: Personalized video chat with one-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23321–23331 (2025)

[31] Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S., et al.: The caltech-ucsd birds-200-2011 dataset. Tech. rep., Technical Report CNS-TR-2011-001, California Institute of Technology (2011)

[32] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)

[33] Xu, B., Feng, J., Lu, S., Luo, Y., Yan, S., Liang, H., Lu, M., Zhang, W.: Jarvis: Towards personalized ai assistant via personal kv-cache retrieval. arXiv preprint arXiv:2510.22765 (2025)

[34] Yang, S., Ji, H., Lu, S., Chen, J., Xu, B., Lu, M., Zhang, Y., Dong, W., Zhang, W.: Small-large collaboration: Training-efficient concept personalization for large vlm using a meta personalized small vlm. arXiv preprint arXiv:2508.07260 (2025)

[35] Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Li, H., Zhao, W., He, Z., et al.: Minicpm-v: A gpt-4v level mllm on your phone. arXiv preprint arXiv:2408.01800 (2024)

[36] Yeh, C.H., Russell, B., Sivic, J., Heilbron, F.C., Jenni, S.: Meta-personalizing vision-language models to find named instances in video. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 19123–19132 (2023)

[37] Zheng, Y., An, R., Lin, X., Liu, Y., Yang, S., Zhang, H., Li, H., Zhang, Q., Zhang, R., Li, G., et al.: Pearl: Personalized streaming video understanding model. arXiv preprint arXiv:2603.20422 (2026)

Table A5: Scene-to-scene attribute change in CLEVR-Default. Each row indicates an actual edit applied when moving from scene t to scene $t+1$.

From Scene	To Scene	Object idx	Attribute	From value	To value
scene00	scene01	2	color	blue	green
scene01	scene02	3	material	metal	rubber
scene02	scene03	3	shape	cylinder	cube
scene02	scene03	4	color	gray	yellow
scene03	scene04	0	color	red	brown
scene04	scene05	2	shape	cube	sphere
scene05	scene06	0	material	rubber	metal
scene06	scene07	1	material	rubber	metal
scene07	scene08	1	color	brown	cyan
scene08	scene09	4	material	rubber	metal
scene08	scene09	0	color	brown	purple
scene09	scene10	2	color	green	blue
scene10	scene11	4	color	yellow	red
scene11	scene12	3	material	rubber	metal
scene12	scene13	4	shape	cube	sphere
scene13	scene14	1	shape	cylinder	sphere
scene14	scene15	0	material	metal	rubber
scene15	scene16	2	shape	sphere	cylinder
scene16	scene17	4	color	red	gray
scene16	scene17	2	material	rubber	metal
scene17	scene18	3	color	red	brown
scene18	scene19	4	material	metal	rubber
scene18	scene19	3	material	metal	rubber

A Additional Details and Experiments on CLEVR

We construct a FINGER benchmark using a CLEVR-based pipeline. Scene rendering is implemented with custom scripts built on top of the original CLEVR utilities, using a fixed base scene and predefined object/material dictionaries. Each concept is defined over object-level attributes from a controlled vocabulary, including shape (cube, sphere, cylinder), color (8 values), material (rubber, metal), and size (small, large).

A.1 CLEVR-Default in the Main Paper

For the CLEVR-Default experiments in the main paper, we construct the concepts by generating 20 concepts organized as a chain-like procedure. Here, each concept is derived from the previous one by changing the attributes of one or two objects, while keeping the overall scene layout highly similar across adjacent concepts. In the benchmark used in the main paper, these edits are restricted to shape, color, and material changes, while object size and object positions are kept unchanged across neighboring concepts. Each concept corresponds to a fixed object configuration and is rendered into 10 multi-view images, yielding 200 images in total. For evaluation, we use one image per concept as the support example and the remaining nine as query images.

Table A5 lists the actual scene-to-scene edits for one instantiation of this benchmark (denoted as CLEVR). The table makes the chain construction explicit: each transition from scene t to scene $t+1$ is realized by one or two local attribute edits on individual objects. Most transitions involve a single modification, such as changing the color of object 2 from blue to green (scene00→scene01) or the material of object 3 from metal to rubber (scene01→scene02). A smaller number of transitions involve two simultaneous edits, e.g., scene02→scene03, scene08→scene09, scene16→scene17, and scene18→scene19. Importantly, all edits remain local and limited to shape, color, or material, so consecutive scenes stay highly similar despite representing different personalized concepts.

A.2 Difficulty-controlled CLEVR

In addition to the chain-based benchmark used in the main paper, we further construct **difficulty-controlled** CLEVR variants to analyze the behavior of personalization methods under systematically varying scene complexity and inter-concept distance. These additional benchmarks are parameterized

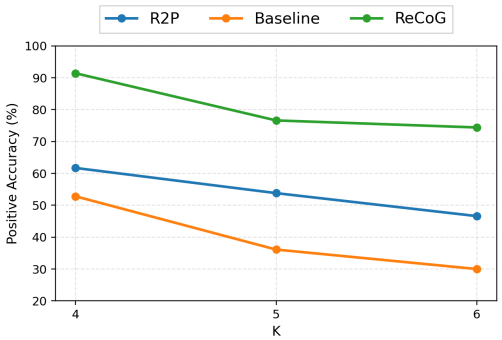

Figure A6: Performance comparison on difficulty-controlled CLEVR with different base scene complexities K (1-hop).

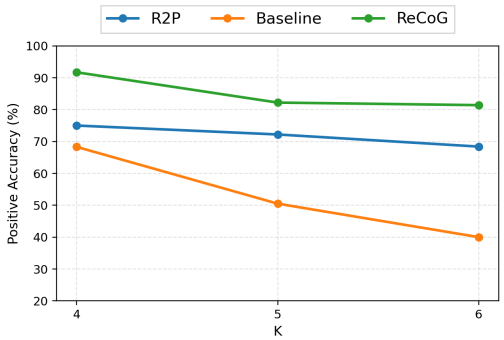

Figure A7: Performance comparison on difficulty-controlled CLEVR with different base scene complexities K (2-hop).

by the base number of objects $K \in \{4, 5, 6\}$ and by the hop size, which determines how far each derived scene is from its parent scene.

We define an "edit unit" as either changing one attribute (shape, color, size, or material) of a single object, or adding/removing one object. A *1-hop* modification applies one such edit unit, whereas a *2-hop* modification applies two edit units sequentially, possibly on the same object or on different objects. Using this definition, we generate two families of concept sets.

For the **1-hop** setting, we first sample a random root scene with K objects. We then generate four scenes by applying one 1-hop modification to the root. From each of these four scenes, we further generate four additional scenes using another 1-hop modification. After discarding the initial root scene, this yields a set of 20 scenes in total.

For the **2-hop** setting, we follow the same procedure, except that each expansion step uses a 2-hop modification instead of a 1-hop one. Again, after removing the root scene, we obtain 20 scenes. During generation, we explicitly remove duplicate scenes to avoid degenerate cycles or accidental returns to previously generated configurations.

Compared with the CLEVR benchmark in the main paper, these additional variants provide two explicit difficulty knobs. First, increasing K makes the scene more complex by increasing the number of objects. Second, the hop size controls the separation between concepts: 1-hop concepts are more locally confusable, while 2-hop concepts are typically more separated in concept space. This allows a more fine-grained analysis of how personalization methods behave as the task becomes either more cluttered or more subtle.

A.3 Experimental Results on Difficulty-controlled CLEVR

Table 2 in the main paper and Fig. A6-A7 summarize the results. Several trends are consistent across all settings. First, increasing K consistently makes the task more difficult. As the number of objects grows from 4 to 6, positive accuracy drops for all methods in both the 1-hop and 2-hop settings. This confirms that larger scenes introduce more distractors and make fine-grained matching substantially harder.

Second, the 1-hop setting is more challenging than the 2-hop setting, especially in terms of positive accuracy. Although 2-hop involves more edits, the resulting concepts are generally farther apart and therefore less confusable than 1-hop concepts, which differ by only a single local change. This effect is clearly visible for both R2P and the baseline, whose positive accuracy improves noticeably in the 2-hop setting.

Third, ReCoG consistently achieves the best performance across all values of K and under both hop settings. The advantage is particularly large in the more confusable 1-hop regime. For example, at $K = 6$, ReCoG achieves 74.4% positive accuracy in the 1-hop setting, compared with 46.6% for R2P and 30.0% for the baseline. Even in the 2-hop setting, where concepts are relatively easier to

Table A6: Performance comparison for experiments on changing positional arrangement. Values are reported in Positive Accuracy/Negative Accuracy (%).

K	Methods	P.Acc / N.Acc
4	R2P	27.8 / 96.2
	Baseline	18.0 / 95.7
	ReCoG	**33.2 / 96.5**

Table A7: Comparative results on CUB-Hard: MiniCPM-o-2.6 vs. LLaVA-1.5. Bold indicates the best performance within each model group.

Methods	MiniCPM-o-2.6			LLaVA-1.5		
	P.Acc	N.Acc	Wtd.	P.Acc	N.Acc	Wtd.
Baseline	41.8	96.9	69.4	36.1	96.6	66.4
ReCoG(Ours)	**86.3**	**99.3**	**92.8**	**89.4**	**99.5**	**94.5**

separate, ReCoG still maintains a clear margin, reaching 81.4% at $K = 6$, compared with 68.4% and 40.0% for R2P and the baseline, respectively.

Finally, ReCoG shows notably stronger robustness as scene complexity increases from $K = 5$ to $K = 6$. In both the 1-hop and 2-hop settings, its positive accuracy drops only marginally, whereas the competing methods suffer substantially larger degradations. For example, in the 1-hop setting ReCoG decreases from 76.6% to 74.4%, while R2P drops from 53.8% to 46.6% and the baseline from 36.1% to 30.0%. Likewise, in the 2-hop setting ReCoG remains nearly unchanged (82.2% to 81.4%), in contrast to the larger declines of R2P (72.2% to 68.4%) and the baseline (50.5% to 40.0%). This suggests that ReCoG scales more gracefully to more cluttered scenes and preserves its discriminative ability even in harder personalization regimes.

A.4 CLEVR with Changing Positional Arrangement

We additionally consider a harder CLEVR setting in which the main source of variation is positional arrangement. Concretely, we begin with the four scenes generated by a 1-hop modification and, for each of them, create four additional 1-hop variants. In this second stage, the modification is restricted to changes in the *xy positions* of objects, rather than their attributes or identities. This setup makes the task significantly more difficult, because candidate scenes remain almost identical in object composition and differ mainly in spatial layout.

As expected, Table A6 shows that the overall performance degrades under this setting, indicating that positional rearrangement is a particularly challenging factor for training-free personalization. Nevertheless, ReCoG still consistently and substantially outperforms both the baseline and R2P, showing that explicit comparison remains effective even in this harder regime. These results suggest that, while purely spatial changes without strong attribute differences are still difficult to resolve reliably, our relation-aware retrieval strategy provides a clear advantage over relation-agnostic alternatives.

B Construction and Variance of CUB-Hard

CUB contains 200 birds and 312 class-level visual attributes. Among these attributes, we identified 16 attributes corresponding to black coloration in different body regions, such as wing or upperparts. For each species, we computed the mean score across these 16 attributes and ranked all 200 species. CUB-Hard consists of the 20 species with the highest scores. The support/query images were randomly selected.

To analyze the variance of CUB-Hard, we constructed three alternative concept sets using the same procedure. Specifically, we selected white, brown, and grey coloration, respectively. ReCoG consistently outperforms R2P with low variation across concept sets (see Table A8).

Concept set	R2P P.Acc.	ReCoG P.Acc.
Black	63.8%	**86.3%**
White	64.5%	**90.8%**
Brown	69.0%	**95.3%**
Grey	63.2%	**92.1%**
Mean $\pm$ std.	65.1 $\pm$ 2.3%	**91.1 $\pm$ 3.2%**

Table A8: Performance comparison across different concept sets based on coloration.

Compare the two {category} concepts identified by the concept-identifiers and highlight what makes each concept unique.

- Image 1: <example1>
- Image 2: <example2>

Your response MUST follow EXACTLY the JSON format shown below (and nothing else):

```
{
  "example1 distinct features":  ["feat1", "feat2", "feat3"],
  "example2 distinct features":  ["feat1", "feat2", "feat3"]
}
```

IMPORTANT:
- Do not include any additional text or commentary.
- Any deviation from this format will be considered incorrect.

Figure A8: Prompt template for relation-aware database construction.

C Impact of Employed VLMs

To examine whether the effectiveness of ReCoG depends on a specific VLM backbone, we additionally replace the recognition model used in the main paper (MiniCPM-o-2.6) with LLaVA-1.5 and evaluate on CUB-Hard. Table A7 shows that ReCoG consistently outperforms the baseline under both backbones. With MiniCPM-o-2.6, ReCoG improves the weighted accuracy from 69.4 to 92.8. The same trend is preserved with LLaVA-1.5, where ReCoG improves the weighted accuracy from 66.4 to 94.5. A similar pattern is observed for positive accuracy: under LLaVA-1.5, ReCoG achieves 89.4, substantially higher than the baseline score of 36.1. These results indicate that the gain from ReCoG does not rely on a particular VLM choice. Even when the recognition backbone is replaced, incorporating relation-aware concept retrieval consistently provides large improvements over the relation-agnostic baseline. This suggests that ReCoG is a general and backbone-compatible personalization framework.

D Details on VLM Prompting

D.1 Prompt Template for Relation-aware Database Construction

In this section, we provide the prompt used for database construction (P_{disc} in Sec. 4.2). Figure A8 shows the template for eliciting discriminative concept features from a frozen VLM.

Given two support images corresponding to a pair of concepts, the prompt asks the VLM to explicitly compare them and identify what makes each concept distinct from the other. Unlike generic captioning or independent concept description, this prompt is designed to produce *relational* and *contrastive* evidence, i.e., features that are useful specifically for distinguishing one concept from a visually similar alternative. This pairwise comparison is the key ingredient for constructing the directed edges in our relational concept graph.

To ensure that the VLM output can be parsed and stored reliably, we enforce a strict JSON-only response format. For each ordered concept pair, the model returns two lists of distinct features, one

You are a helpful AI agent specializing in image analysis and object recognition.
You are given three images:
- **Image 1:** Query image (the image to classify)
- **Image 2:** Reference image for concept "{concept1_name}"
- **Image 3:** Reference image for concept "{concept2_name}"

Additionally, you are provided with information about these two concepts:
Distinctive features of {concept1_name}: ["feat1", "feat2", "feat3"]
Distinctive features of {concept2_name}: ["feat1", "feat2", "feat3"]

Task:
- Compare Image 1 (query) with Image 2 and Image 3.
- Determine which concept the query image belongs to.
- Focus on the distinctive features provided above to make your decision.

Question:
- Provide your reasoning based on the distinctive features.
- Generate your response in JSON format:
  `{"Reasoning": "...", "Answer": "..."}`
- Output only the JSON response. DO NOT output any additional text.

Figure A9: Prompt template for comparison-based retrieval.

for each concept from its own perspective. These textual outputs are then used as the discriminative edge evidence. By repeating this process over concept pairs, we obtain a database that captures not only per-concept visual representations, but also explicit inter-concept differences that are directly useful for fine-grained personalized recognition.

D.2 Prompt Template for Comparison-based Retrieval

In this section, we provide the prompt used for discriminative retrieval via explicit comparison (P_{comp} in Sec. 4.3). Figure A9 shows the template used at inference time for comparison-based retrieval.

Given a query image and two candidate support images, the prompt asks a frozen VLM to determine which candidate concept is more compatible with the query. Importantly, the model is not asked to make this decision based only on global visual similarity. Instead, it is additionally provided with the concept-specific distinctive features extracted during database construction, so that the comparison can focus on the most discriminative evidence between the two candidates. This design directly reflects the retrieval procedure of ReCoG. After CLIP-based pruning produces a small candidate set, fine-grained ambiguity remains among visually similar concepts. This prompt resolves such ambiguity by forcing the VLM to compare two candidates side by side while conditioning on their relation-aware textual evidence. In this way, retrieval is performed through relative judgment rather than independent scoring of each concept.

For each comparison, the model returns a short reasoning trace together with its final choice. These pairwise decisions are then aggregated across all candidate pairs, as described in Sec. 4.3, to determine the final prediction.

E Validation on Realistic Open-Domain Personalized Concepts

To further validate ReCoG, we additionally conducted an experiment on CelebA. We treated each identity ID as a personalized concept. For each concept, we used one support image and 19 query images. We constructed all 190 pairwise relational edges among the 20 concepts. Under support-image seed 42, ReCoG improves P.Acc. from 76.1% to 94.2% (see Table A9). This result provides additional evidence that ReCoG is effective not only for birds or synthetic scenes, but also for fine-grained recognition of real human identities.

Dataset	R2P P.Acc.	ReCoG P.Acc.	Δ
CelebA	76.1%	**94.2%**	**+18.1**

Table A9: Personalized recognition performance on CelebA.

F Sensitivity to the Selected Support Image

The method may be sensitive to the chosen support image. Since relational edges are generated from a single support image per concept, the extracted cues may reflect image-specific artifacts rather than stable concept-level differences. To address this issue, we resampled the support image for every concept using five different random seeds and reconstructed the corresponding concept databases. All other evaluation settings (CUB-Hard) were kept unchanged. ReCoG outperforms R2P under every support-image selection (see Table A10). Although the absolute performance of ReCoG varies across exemplars, its advantage remains consistent. Moreover, we find that the performance variance of ReCoG is even smaller than that of R2P. These results demonstrate that the gain is not tied to one particular set of support images.

Seed	**R2P (P.Acc.)**	**ReCoG (P.Acc.)**
23	59.7%	84.2%
42	65.8%	83.7%
123	61.8%	83.2%
2024	72.4%	90.8%
2025	69.2%	87.6%
Mean±std.	65.8±5.2%	85.9±3.2%

Table A10: Sensitivity to support image selection under different random seeds.

G Qualitative Results

Figure A10 in the next page illustrates why explicit comparison is important in fine-grained personalization. For reference, the answer is `Yes`.

In the relation-agnostic pipeline (top), the model reasons from independently described concept features and therefore tends to focus on salient attributes that are visible in the query, such as the small red cube, and the large purple metallic cylinder. However, many of these cues are not truly discriminative because they are also compatible with competing concepts (*hex* and *nba*). As a result, the prediction is driven by generic overlap rather than by the critical differences among candidates, ultimately leading to the wrong answer: *nba*.

In contrast, ReCoG (bottom) explicitly compares candidates using relation-aware evidence stored between concept pairs. This makes the model attend to how one candidate differs from another, rather than simply checking whether a candidate has some visible attributes. For example, the comparison highlights subtle yet distinguishing cues such as "The green sphere has been replaced" (vs. *nba*) or "Purple cylinder has been added" (vs. *hex*). These separate *kie* from the other retrieved concepts. By grounding the decision on these inter-concept differences, ReCoG correctly resolves the ambiguity and predicts *kie*.

In addition, Fig. A11 compares the captioning results of R2P and ReCoG.

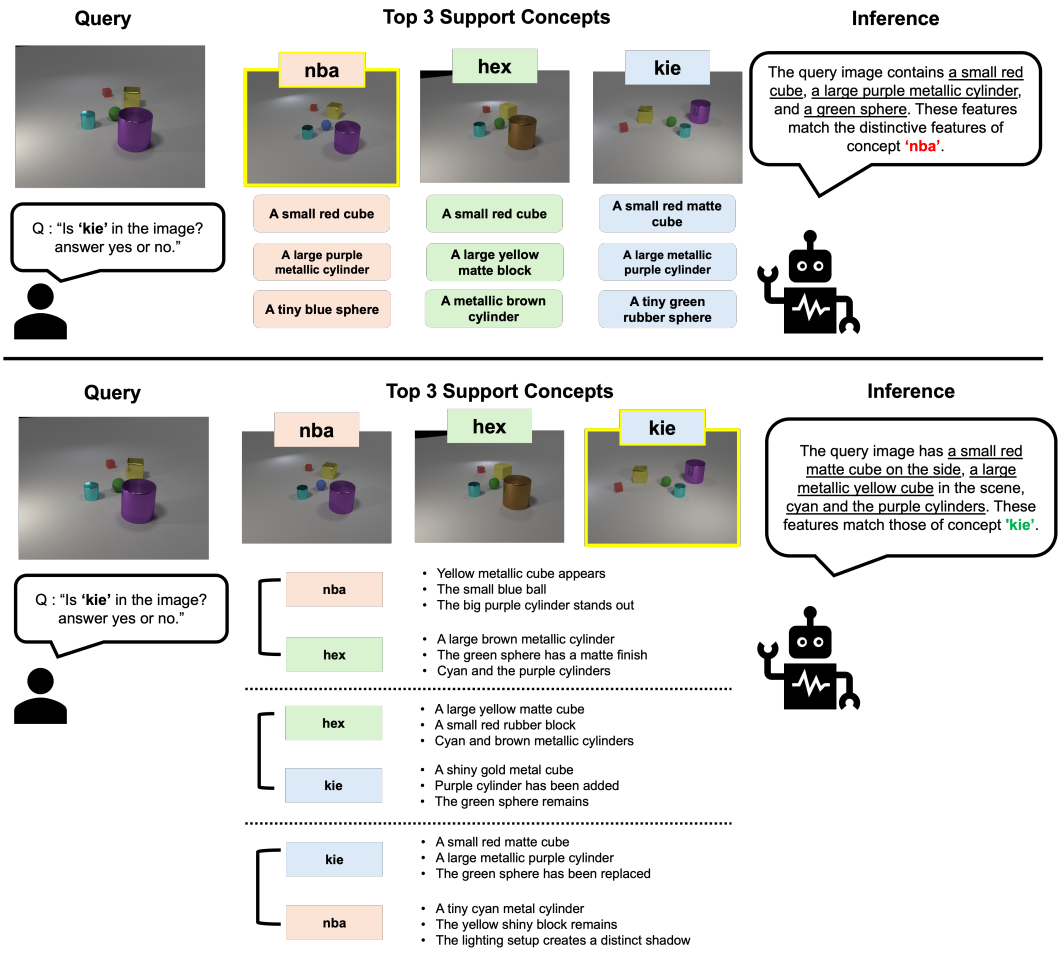

Figure A10: Qualitative example comparing R2P-style retrieval (top) and ReCoG (bottom). The query belongs to *kie*. While the relation-agnostic method is misled by prominent shared attributes and incorrectly favors *nba*, ReCoG resolves the ambiguity through explicit pairwise comparison with relation-aware evidence and correctly predicts *kie*.

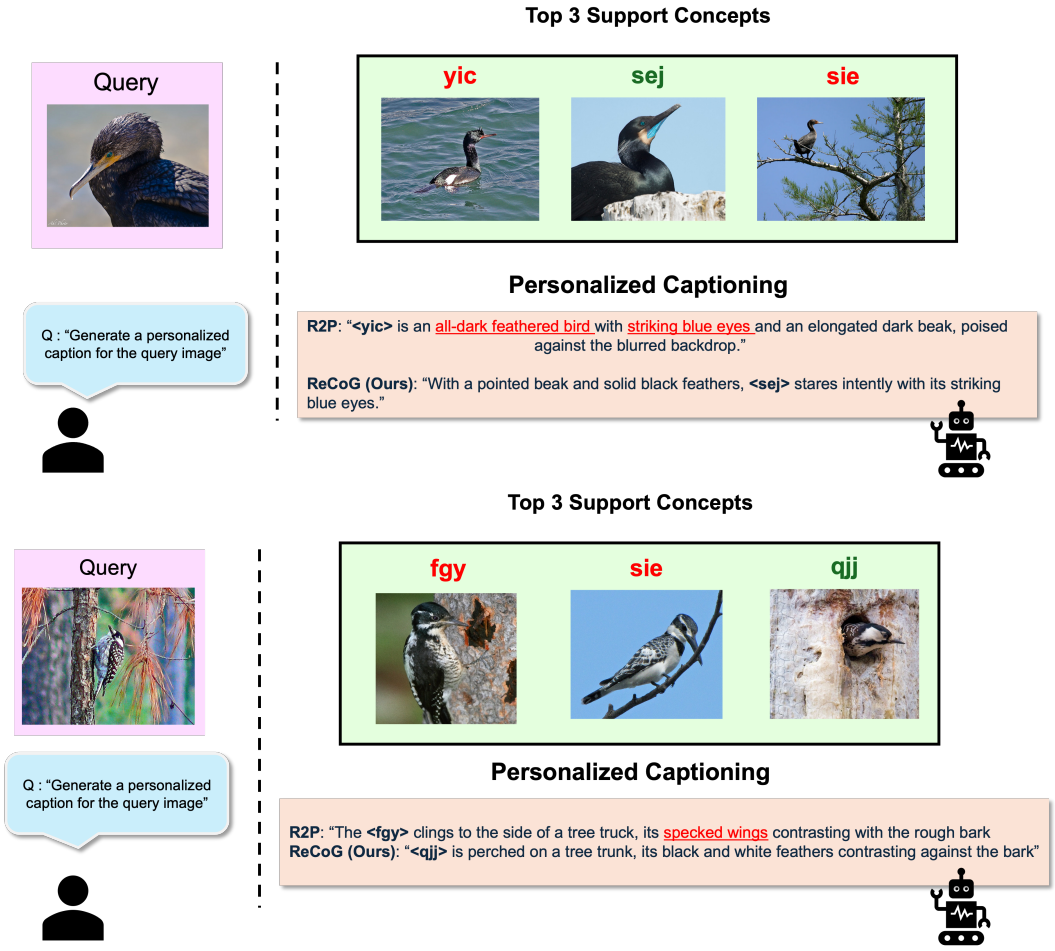

Figure A11: Qualitative examples comparing R2P-style captioning and ReCoG.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We present the contributions and scope of our work in abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations at the end of conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Our work does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We present all experimental settings and the information necessary to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data and code are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: As the proposed method is fully deterministic, reporting statistical significance is less necessary.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the relevant information.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We respect the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We believe that this work does not present any notable societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We explicitly mention them.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The newly proposed benchmark is explained in detail, and corresponding documentation is publicly available.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our work does not include such issue.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not include such issue.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We explicitly use VLM as a core component, and describe its usage in detail.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.