CARLoS: Retrieval via Concise Assessment Representation of LoRAs at Scale

Supplementary Material

A. Extended Qualitative Retrieval Comparison

We provide an extended qualitative evaluation in Figure 8 to give the reader a granular view of the retrieval performance distribution, comparing our CARLoS retrieval method against our strongest textual baseline, Qwen3. The examples showcase the top-5 retrieved LoRAs for a selection of 12 diverse queries, all applied to a fixed base prompt (e.g., 'A cat sitting on a rock') for a clearer visual comparison of the adapter's effect.

The following analysis summarizes the principal qualitative findings from these examples, emphasizing the patterns that explain CARLoS's retrieval advantages.

Semantic and Stylistic Precision: For queries relating to specific artistic or lighting styles, such as '80s retro futurism', 'Blues and purples create a cool calming color scheme', and 'Hard shadows', CARLoS retrieval is highly effective. It consistently returns LoRAs that visually manifest the queried effect across the top-5 ranks. This is a direct result of our approach, which compares the query's semantic shift to the measured generative effect of the LoRAs.

Failure of Textual Baselines: The textual retriever, Qwen3, often fails to match the query to the correct generative effect, particularly for abstract or complex concepts. For example, for '80s retro futurism', the Qwen3 results show minimal or inconsistent stylistic changes. This suggests that the text retrieval is often misled by irrelevant keywords or unreliable user-provided descriptions, as discussed in our main paper (Section 4.3).

Robustness via Filtering: The comparative view illustrates the importance of our Strength and Consistency filtering. CARLoS's generations maintain a better adherence to the base prompt because its retrieval process explicitly filters out LoRAs that are too strong (which can override the prompt) or too inconsistent (unpredictable). Conversely, the textual retriever, Qwen3, includes several LoRAs that would be filtered by CARLoS due to low consistency and/or low strength. For example, Qwen3's result #3 for 'A dense lush forest...' shows a weak effect that fails to consistently apply the style, demonstrating the need to filter out unreliable LoRAs. This filtering ensures the resulting generations are higher quality and more predictable.

In summary, Figure 8 provides visual evidence that retrieving LoRAs based on their unbiased, generative effect yields superior and more consistent results, solidifying the findings of our quantitative and subjective evaluations.

Table 3. Top-7 Retrieval Performance. Scores indicate the quality of retrieved LoRAs as judged by state-of-the-art Vision-Language Models. CARLoS consistently yields results preferred by all evaluators. The scores are normalized in a min-max manner across all queries and retrievers.

Retriever	Evaluator	Top-1	Top-2	Top-3	Top-4	Top-5	Top-6	Top-7
E5	SigLIP2	0.317	0.297	0.289	0.279	0.271	0.267	0.263
	Qwen2.5	0.501	0.488	0.480	0.474	0.470	0.467	0.464
	IR	0.468	0.458	0.449	0.443	0.439	0.435	0.432
	HPS	0.575	0.570	0.565	0.562	0.561	0.559	0.558
GTE	SigLIP2	0.265	0.256	0.258	0.254	0.250	0.246	0.244
	Qwen2.5	0.470	0.461	0.461	0.457	0.454	0.452	0.451
	IR	0.451	0.438	0.440	0.435	0.433	0.432	0.433
	HPS	0.562	0.554	0.556	0.554	0.553	0.553	0.553
BGE	SigLIP2	0.218	0.206	0.199	0.194	0.191	0.189	0.187
	Qwen2.5	0.450	0.434	0.429	0.425	0.421	0.419	0.417
	IR	0.408	0.390	0.387	0.387	0.384	0.381	0.378
	HPS	0.553	0.544	0.543	0.543	0.542	0.540	0.538
Qwen3	SigLIP2	0.343	0.320	0.307	0.298	0.291	0.284	0.278
	Qwen2.5	0.521	0.505	0.495	0.488	0.484	0.480	0.477
	IR	0.509	0.499	0.491	0.484	0.479	0.476	0.473
	HPS	0.599	0.593	0.590	0.587	0.585	0.583	0.582
CARLoS	SigLIP2	**0.385**	**0.368**	**0.350**	**0.342**	**0.334**	**0.328**	**0.324**
	Qwen2.5	**0.561**	**0.547**	**0.532**	**0.524**	**0.520**	**0.515**	**0.512**
	IR	**0.531**	**0.520**	**0.505**	**0.498**	**0.493**	**0.488**	**0.487**
	HPS	**0.607**	**0.603**	**0.596**	**0.594**	**0.591**	**0.590**	**0.589**

B. Extended Quantitative Evaluation

Table 3 extends the main results of the main paper (Table 1) by reporting retrieval performance for $k = 1 \dots 7$ across all evaluators. Whereas Table 1 averages evaluator scores over the top 3 retrieved LoRAs, this table expands the analysis to multiple top-k settings. For each column, we average the scores of the top-k ranked retrieved LoRAs. The scores were min-max linearly normalized per-evaluator, and across all queries, top-7 retrieved images, and retrieval methods. This is the exact same normalization mapping as used in Table 1 for comparability. Specifically, the column **Top-3** in Table 3 matches the measurement reported in the main paper in Table 1. The trends confirm that CARLoS consistently surpasses text-based baselines for all k, indicating that CARLoS retrieves highly relevant LoRAs along multiple ranking spans, which is a desirable property for practical search, aiming for a stable, user-friendly environment.

C. Retrieval Diversity and Non-Bias Analysis

The superior retrieval performance of CARLoS is not solely due to high accuracy but also stems from its ability to select from a **diverse and broad range** of semantically relevant LoRAs, demonstrating a low bias towards highly popular or overly strong adapters. This section provides both qualitative and quantitative evidence for this crucial property.

Figure 8. A qualitative comparison showcasing the top-5 LoRAs retrieved by CARLoS (ours, top row) and the Qwen3 textual baseline (bottom row) for 12 uncurated, randomly selected retrieval queries. The generations all use a fixed base prompt ('Cat sitting on a rock') to better isolate and highlight the stylistic and semantic shifts induced by the retrieved LoRAs. This figure further demonstrates CARLoS's ability to consistently retrieve LoRAs that are visually and semantically relevant to the query, outperforming the textual retriever, especially for abstract concepts (e.g., '80s retro futurism', 'Kimono inspired', and 'Hard shadows'). Zoomed in viewing recommended.

C.1. Qualitative Analysis: Retrieval Frequency

To confirm that CARLoS does not simply rely on a small set of popular LoRAs, we analyzed the retrieval frequency across our comprehensive benchmark of over 700 queries.

The analysis, summarized in Figure 9, plots the number of times each unique LoRA appeared in the top-3 retrieved results, sorted by descending frequency.

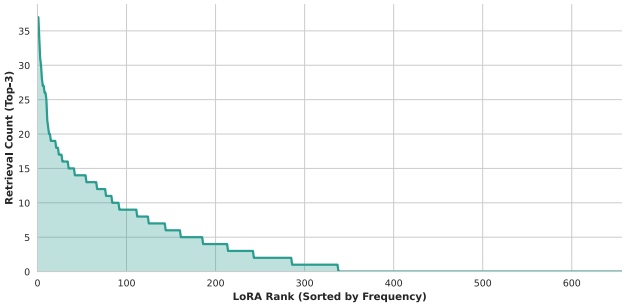

Figure 9. LoRA Retrieval Frequency (Top-3) across the full query set. The graph shows the number of times each unique LoRA appeared in the top-3 results, ordered by frequency, demonstrating that our CARLoS retrieval method utilizes a **diverse and broad range** of the LoRA corpus, rather than relying on a small, popular subset. The long, shallow tail indicates that the majority of LoRAs in the corpus are retrieved at least once.

Broad Coverage: The curve exhibits a long, shallow tail, confirming that the retrieval process is highly diverse. Out of the 656 available LoRAs, the majority are retrieved at least once across the benchmark. This diversity is crucial, as it indicates that the system is not strictly biased towards LoRAs with extreme Strength or by social factors, which often plague popularity-based discovery methods.

This analysis complements the qualitative results shown in Figures 1, 3, and 4 of the main paper, which also depict a variety of effects being retrieved for different queries. The measured diversity underscores the value of our prompt-independent behavioral representation in fostering a non-biased, standardized, and transparent ecosystem for generative components.

C.2. Quantitative Analysis: Diversity Metrics

To quantitatively demonstrate that our method relies on semantic relevance rather than just a small set of "popular" adapters, we rigorously assess the diversity and non-bias of our CARLoS retrieval method, by reporting three complementary measures of distributional diversity (Table 4): Normalized Entropy, Gini Coefficient, and Effective Count. Entropy captures overall uncertainty and sensitivity to rare outcomes, the Gini Coefficient quantifies inequality and concentration of mass, and the Effective Count translates these abstract measures into an interpretable "number of active components". Together, they provide a robust and multi-perspective characterization of the distribution's spread and skewness.

Table 4 shows that CARLoS consistently achieves the highest Normalized Entropy and ELC and the lowest Gini Coefficient across all ranking spans ($k = 1$ to $k = 7$). Moreover, looking at Qwen3, the next best retrieval method after CARLoS as evident in Table 3, seems to fall behind in all diversity measures. This collective evidence asserts the superior performance of CARLoS not only in measures

Table 4. Retrieval distribution metrics across retrievers for each Top-k. Higher is better for Normalized Entropy and Effective LoRA Count; lower is better for Gini Coefficient. **Bold** marks best; underline second best.

Retriever	Normalized Entropy ↑	Gini Coefficient ↓	Effective LoRA Count ↑
Top-1			
E5	0.728	0.861	112.444
GTE	0.609	0.912	52.002
BGE	0.648	0.899	67.080
Qwen3	0.712	0.876	101.578
CARLoS	**0.790**	**0.802**	**167.901**
Top-2			
E5	0.765	0.827	142.817
GTE	0.653	0.881	69.162
BGE	0.663	0.887	73.585
Qwen3	0.738	0.854	119.793
CARLoS	**0.824**	**0.755**	**210.096**
Top-3			
E5	0.780	0.812	157.069
GTE	0.681	0.866	82.915
BGE	0.670	0.881	77.228
Qwen3	0.752	0.842	130.908
CARLoS	**0.843**	**0.723**	**237.598**
Top-4			
E5	0.792	0.796	170.462
GTE	0.704	0.851	96.214
BGE	0.676	0.877	80.008
Qwen3	0.758	0.837	136.115
CARLoS	**0.852**	**0.708**	**250.694**
Top-5			
E5	0.798	0.788	176.911
GTE	0.721	0.839	107.057
BGE	0.677	0.874	80.632
Qwen3	0.761	0.833	139.514
CARLoS	**0.859**	**0.693**	**263.603**
Top-6			
E5	0.803	0.781	183.273
GTE	0.735	0.827	117.830
BGE	0.678	0.873	81.509
Qwen3	0.765	0.830	142.714
CARLoS	**0.865**	**0.682**	**273.499**
Top-7			
E5	0.808	0.775	188.747
GTE	0.747	0.818	126.985
BGE	0.681	0.870	82.737
Qwen3	0.765	0.829	143.187
CARLoS	**0.869**	**0.672**	**281.016**

Table 5. Threshold Sensitivity. Effect of varying Strength and Consistency thresholds (independently) on retrieval performance. The selected operating point is underlined. Consistency shows a clear optimum, while Strength exhibits a broader trend.

Max Strength	Min Consistency	SigLIP2	Qwen2.5	IR	HPS
6.0	∅	**0.298**	0.453	0.456	0.593
7.0	∅	0.293	0.462	0.465	0.592
8.0	∅	0.283	**0.466**	**0.470**	**0.594**
9.0	∅	0.272	0.457	0.463	0.589
9.8	∅	0.267	0.456	0.452	0.584
10.5	∅	0.268	0.457	0.458	0.585
∅	0.005	0.271	0.455	0.453	0.583
∅	0.01	0.274	0.457	0.463	0.588
∅	0.02	0.278	0.455	0.464	0.589
∅	**0.041**	**0.287**	**0.469**	**0.479**	**0.595**
∅	0.08	0.248	0.443	0.453	0.575
∅	0.1	0.249	0.446	0.454	0.573

Table 6. Extended Ablations. Comparison of full CARLoS with alternative indexing backbone and reduced-resource setting, alongside the strongest textual baseline (Qwen3).

Variant	SigLIP2	Qwen2.5	IR	HPS
CARLoS (Full)	**0.350**	**0.532**	**0.505**	**0.596**
Qwen3	0.307	0.495	0.491	0.590
CARLoS (x16 Reduced Compute)	0.324	0.509	0.479	0.575
CARLoS (SigLIP2 Indexing)	0.196	0.427	0.408	0.554

of accuracy but also through genuine and widespread semantic matches across the LoRA corpus, demonstrating the non-biased utility of our behavioral representation.

D. Threshold Sensitivity.

We analyze the effect of varying the Strength (τ_s) and Consistency (τ_c) thresholds independently (Table 5). Consistency shows a clear optimum around $\tau_c = 0.041$, supporting our chosen value. In contrast, Strength exhibits a broader trend without a sharp optimum.

For inclusiveness and diversity, we chose a permissive strength threshold, to qualitatively filter out only very acute behavior. This analysis indicates that Strength may benefit from further tuning, suggesting that its optimal value may depend on the application. Due to computational costs, experiments were conducted on a reduced evaluation subset of approximately 100 queries.

E. Additional Ablations: Backbone and Compute Reduction

We further evaluate two practical aspects of CARLoS: (1) the choice of embedding backbone, and (2) the robustness of the method under reduced computational resources (Table 2).

Alternative Backbone. We replace CLIP with SigLIP2 for both indexing and retrieval. This results in a substantial degradation across all evaluators. The drop is consistent and significant, confirming that CLIP provides a more suitable embedding space for capturing LoRA-induced semantic shifts in our setting. This validates our design choice of CLIP as the underlying representation.

Reduced Compute Setting. To assess scalability, we reduce the indexing process by a factor of $\times 16$, using $4\times$ fewer prompts and $4\times$ fewer seeds (randomly subsampled from the original set). Notably, this configuration is applied without additional tuning or optimization of the reduced prompt set.

Despite this aggressive reduction, CARLoS maintains competitive performance and still surpasses the strongest textual baseline (Qwen3) in two out of four evaluators. This demonstrates that the core signal captured by our representation remains effective even under limited resources.

These findings suggest that CARLoS can be adapted to significantly lower computational budgets. We believe that further engineering efforts, such as optimized prompt selection or adaptive sampling strategies, could close the remaining gap to the full setting, rendering our framework more scalable and applicable to real-world settings.

F. Analysis of Strength and LoRA Scale

To quantitatively evaluate the functional relationship between the "LoRA Scale" hyperparameter and our Strength metric, we repeated the indexing and metric calculation process (described in Section 3 of the main paper). This analysis was performed on 10 different LoRAs across a range of different scale values, as depicted in Figure 10. To showcase existing behaviors, while maintaining a reasonable computation effort, we purposefully sampled 10 LoRAs spanning low/medium/high Strength and Consistency, thus results are illustrative rather than population-level.

While a positive, monotonically increasing connection is generally visible, the results clearly demonstrate that this relationship is complex and not uniform across different LoRAs. We highlight the following key observations:

- **Saturation and Diminishing Returns:** The Strength of some LoRAs (e.g., 180999 and 130162) appears to saturate at a certain scale. Beyond this point, the Strength plateaus or, in some cases, even softly decreases, indicating a non-linear response.
- **Non-Uniform Trajectories:** The functional relationship is not strictly linear. Different LoRAs exhibit distinct initial "biases" (i.e., base Strength at low scales, such as LoRA 159401) and different "incline ratios" or slopes.

This variation can even cause trajectories to cross (e.g., LoRAs 153831 and 180058).

These findings suggest that, due to the observed variability, relying on the LoRA scale setting alone to accurately predict a LoRA's full Strength-to-scale trajectory is unreliable across the ecosystem. Furthermore, extrapolating from a LoRA's Strength at a single, arbitrary scale is unreliable.

Therefore, this analysis confirms that the CARLoS Strength metric (which we calculate at a fixed scale of 1.0) captures intrinsic, valuable information about a LoRA's behavior. This information is distinct from, and cannot be acquired solely by, observing the LoRA scale hyperparameter.

G. Implementation and Curation Details

We provide implementation details to facilitate reproducibility and future extensions.

G.1. Prompt Set Construction

The prompt sets used for indexing and retrieval, as described in Section 3.1, were constructed using ChatGPT 4o. We guided the LLM generation process with the following sequence of prompts to ensure alignment with common community usage on CivitAI:

1. *what are the most popular categories for prompts in the context of text to image ai generations?*
2. *Regarding each category, for each subcategory, provide me with 16 prompts. make sure these prompts are as varied as possible per-category, and that they do not exceed the length of 73 CLIP tokens. keep the semantics safe for work.*

This process yielded a total of 560 unique prompts. This set was then evenly divided into two disjoint subsets:

- **Indexing Set ($\mathcal{P}$):** $N = 280$ prompts, used for generating images to index the LoRA corpus (Section 3.2).
- **Retrieval Set ($\mathcal{P}'$):** $N = 280$ prompts, used as the base for creating the reciprocal textual CLIP-diff query vector (Section 3.3).

$\mathcal{P}$ and $\mathcal{P}'$ are disjoint and match the same category/subcategory taxonomy. Prompts were assigned deterministically (first 8 to $\mathcal{P}$, last 8 to $\mathcal{P}'$) Example prompts from both sets, along with their associated categories and subcategories, are provided in Table 7 and Table 8. The complete set of 560 prompts is included in the supplementary materials as `indexing_and_retrieval_prompts.py` to ensure reproducibility.

G.2. LoRA Corpus Curation

Our LoRA corpus $\mathcal{C}$ was collected from CivitAI, the largest public repository for such models, using their public API. The curation process involved several filtering and validation steps:

- We initially retrieved the metadata for the first 10,000 reachable SDXL LoRAs via the CivitAI API. The API pagination used default ordering, in practice this correlates with downloads.
- This set was filtered to exclude modules that were (a) less than 100 days old, to avoid transient or unstable uploads, and (b) had a file size exceeding 10 GB, which implies a wrong tagging as LoRA, since typical LoRA sizes are much smaller.
- To prioritize testing more stable models, we focused our downloading efforts on the top 1,875 most popular LoRAs from the filtered set, ordered by their 'downloads' attribute.
- We then proceeded to download the weights file for each LoRA, skipping any with missing files.
- Finally, each downloaded LoRA was programmatically validated by attempting to load it into a standard `diffusers` SDXL pipeline.

From an initial pool of 1,875 filtered metadata entries, we successfully validated and loaded 656 LoRAs. This validation success rate ($\frac{656}{1875} \approx 35\%$) highlights the significant portion of non-functional, corrupted, or otherwise unusable modules present in public repositories, underscoring the challenge addressed by our curation process. This final, validated set of 656 LoRAs constitutes our indexed corpus. A detailed list of all indexed LoRAs - containing their names and model IDs as they appear in CivitAI - is attached in a file named `lora_names_with_civit_ids.csv`.

G.3. LoRA Indexing and Embedding

The LoRA indexing process, detailed in Section 3.2, was implemented using the `diffusers` Python library [60].

Image Generation. We used the `StableDiffusionXLPipeline` module with the `stable-diffusion-xl-base-1.0` base model. To enforce uniform conditions and enable parallel processing, each of the 656 LoRAs was indexed in an independent process.

For each LoRA, the process first loaded the base SDXL model and then applied the LoRA weights. We generated $M = 16$ images for each of the $N = 280$ indexing prompts ($\mathcal{P}$). The generation process began with a fixed random seed of 42, which was sequentially incremented for each of the $M = 16$ samples. We used a fixed **LoRA scale of 1.0** for all generations, leaving other hyperparameters at their defaults (e.g., $CFG = 7.5$, Euler Scheduler). A separate, one-time process was run to generate the vanilla (no-LoRA) images for all (prompt, seed) pairs.

CLIP Embedding. Following generation, we embedded all images using the `transformers` library [63]. We employed the `CLIPModel` with the

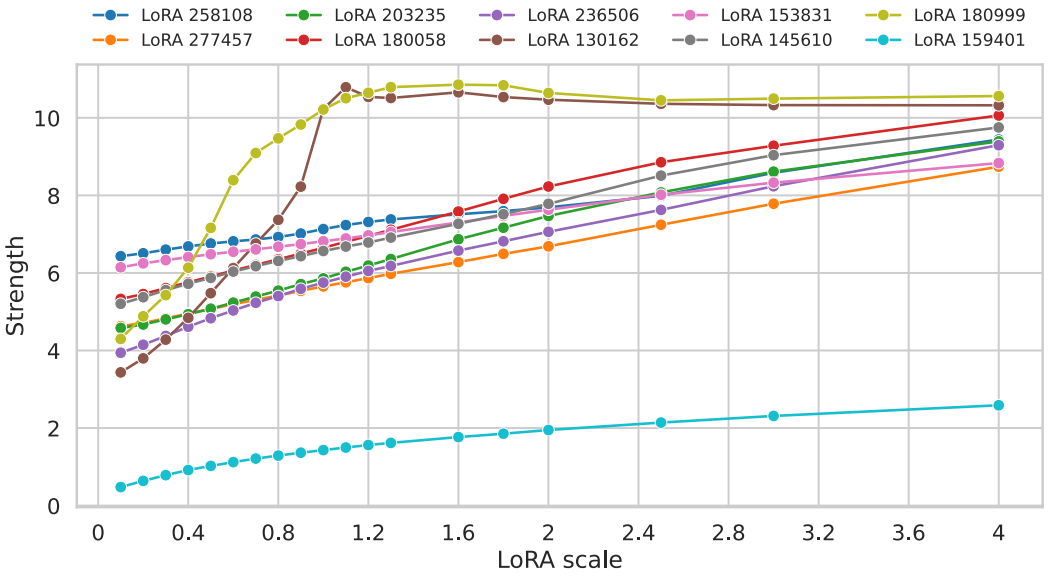

Figure 10. Strength vs. LoRA Scale, for 10 distinct LoRAs selected to represent a range of behaviors in the corpus. The plot shows that while Strength generally increases with scale, the relationship is not strictly linear and highly variable across different LoRAs.

Table 7. A sampled subset of prompts from indexing set $\mathcal{P}$ (full lists in code).

Category	**Sub-Category**	**Example Prompt**
Animals	Dragons, Unicorns, Phoenixes	A colossal fire-breathing dragon perched atop a castle ruin
Animals	Hybrid Creatures	A cat-mermaid lounging on a rock, tail shimmering in the sunlight
Animals	Realistic Wildlife	A majestic snow leopard walking across rocky terrain, snow flurries in the air
Artistic_Styles	Cyberpunk, Synthwave, Vaporwave	A neon-drenched cyberpunk alleyway with holographic signs and flying cars
Artistic_Styles	Oil Painting, Watercolor, Digital Painting	A Baroque-style oil painting of a regal king in ornate armor
Cinematic	Dynamic Action Shots	A warrior leaping through the air, sword drawn, against a fiery backdrop
Conceptual_Arts	Dreamlike or Mind-Bending Scenes	A floating island with a waterfall that cascades into the sky
Fashion	Couture and Avant-Garde	A runway model wearing an avant-garde dress made of reflective shards
Landscapes	Natural Landscapes	A breathtaking sunrise over a vast mountain range, golden mist in the valleys
Portraits	Fantasy and Sci-Fi Characters	A regal elf queen with silver hair, glowing blue eyes, intricate gold armor
Vehicles	Steampunk Machinery and Robots	A massive steampunk airship floating above a Victorian city

Table 8. A sampled subset of prompts from retrieval set $\mathcal{P}'$ (constructed independently of $\mathcal{P}$).

Category	**Sub-Category**	**Example Prompt**
Animals	Dragons, Unicorns, Phoenixes	A unicorn with a mane of galaxies galloping through space
Animals	Realistic Wildlife	A red fox standing alert in a snowy forest, breath visible in the cold air
Artistic_Styles	Cyberpunk, Synthwave, Vaporwave	A vaporwave city square with glitching sculptures and retro grids
Artistic_Styles	Oil Painting, Watercolor, Digital Painting	A luminous digital painting of a cathedral interior with stained glass glow
Cinematic	Scenes That Look Like Movie Frames	A dimly lit interrogation room with a single overhead lamp
Conceptual_Arts	Dreamlike or Mind-Bending Scenes	A corridor of doors opening into different skies and seasons
Fashion	Couture and Avant-Garde	A gown constructed from holographic fabric, refracting rainbow light
Landscapes	Sci-Fi and Futuristic Worlds	A floating island city powered by advanced wind turbines, hovering over the ocean
Portraits	Fantasy and Sci-Fi Characters	A biomechanical cyborg with glowing circuits and synthetic skin
Vehicles	Futuristic Cars, Bikes, Spaceships	A modular space shuttle docking with a rotating ring station

`openai/clip-vit-base-patch32` variant. The **non-normalized, 512-dimensional** CLIP-space vectors were extracted using the `get_image_features` function.

Storage Optimization To optimize storage requirements while preserving sufficient data for future reconstruction validation, the original 1024x1024 generated images were replaced with 64x64 thumbnails (approximately 10 KB each). Importantly, this step was performed after the CLIP

embedding vectors were computed and did not impact our resulting metrics. The 512-dimensional embedding vectors were saved as raw data (approximately 4 KB each). The corpus of CLIP-diff vectors—derived by subtracting the vanilla embedding from the LoRA-modified embedding for each (prompt, seed) pair—forms the core raw data for our analysis.

G.4. Evaluation Metrics

We evaluate the relevance between generated images and text using four complementary Vision-Language Models (VLMs) and aesthetic predictors. Each model produces a scalar score reflecting image-text alignment.

SigLIP2. To avoid using CLIP for both evaluation and inference, we use the SigLIP2 model [55] (`google/siglip2-so400m-patch16-naflex`) for similarity measurement. Given an image x and text t, the model produces joint embeddings, used to compute the alignment score as the sigmoid-normalized image-text matching over the logits:

$$\text{Score}_{\text{SigLIP2}}(x, t) = \sigma\big(s(x, t)\big),$$

where $s(x, t)$ denotes the model's image-text matching logit and $\sigma(\cdot)$ denotes the sigmoid function. This formulation follows the model's default inference behavior and provides a bounded similarity score in $[0, 1]$.

Qwen2.5-VL. As suggested in the literature [15], we use Qwen2.5-VL (`Qwen/Qwen2.5-VL-7B-Instruct`) in a generative evaluation setting. For each image-text pair, we prompt the model with:

> *"Question: Does the image match the description exactly? Answer with exactly one word: yes or no."*

We then compute a continuous alignment score by comparing the probabilities assigned to the tokens "yes" and "no". Specifically, let p_{yes} and p_{no} denote the aggregated probabilities (via log-sum-exp over token variants). The final score is:

$$\text{Score}_{\text{Qwen}}(x, t) = \frac{p_{\text{yes}}}{p_{\text{yes}} + p_{\text{no}}}.$$

This formulation provides a calibrated estimate of semantic agreement between the image and the text.

ImageReward. We use the pretrained ImageReward model (ImageReward-v1.0) [66], which outputs a learned scalar reward score for an image conditioned on text. The score is used directly:

$$\text{Score}_{\text{IR}}(x, t) = \text{ImageReward}(x, t).$$

HPS v2. We use HPS v2.1 [64], a human preference predictor trained to assess perceptual quality and alignment. The score is computed using the model's default scoring function:

$$\text{Score}_{\text{HPS}}(x, t) = \text{HPSv2}(x, t).$$

In all evaluators, the textual input t is constructed as the concatenation of the base prompt and the retrieval query, ensuring that the evaluation jointly reflects prompt fidelity and the desired semantic modification.

G.5. Text Retrieval Prompting

For textual baselines (Qwen3, Multilingual-E5-Instruct, BGE-Reranker-v2-M3, and mGTE-reranker), retrieval is performed over the user-provided metadata of each LoRA, consisting of its *name* and *description* fields obtained from CivitAI and concatenated into a single textual representation per LoRA.

The baselines fall into two groups. Qwen3 and Multilingual-E5-Instruct are embedding-based retrievers. For these models, we encode the query and all corpus entries into a shared embedding space, L2-normalize the resulting embeddings, and rank LoRAs by scaled dot-product similarity. Both models use an instruction prefix for the query as demonstrated in [26, 29]. For Qwen3, we use:

```
Instruct: Given a visual effect or
theme, retrieve relevant passages
that describe a software component
which helps to achieve this effect
or theme in image generation.
Query: <query>
```

For Multilingual-E5-Instruct, we use:

```
Instruct: Given a web search query,
retrieve relevant passages that
answer the query
Query: <query>
```

The corpus documents are encoded directly without additional instruction prefixes.

BGE-Reranker-v2-M3 and mGTE-reranker are cross-encoder rerankers. For these models, we score each query-document pair directly by feeding the pair (q, d) into the model and using the resulting scalar classification logit as the relevance score as shown in [27, 28]. To produce a ranked list over the corpus, we apply a softmax over the per-document scores for the given query and sort accordingly.

In all textual baselines, retrieval relies solely on the metadata text provided by LoRA uploaders, without using any generated images, CARLoS metrics, or additional manual curation beyond the corpus-level filtering described in the main paper.

H. User Study Additional Details

H.1. Study Methodology and Interface

To validate our quantitative Vision-Language Model (VLM) results with human judgment, we conducted a double-blind subjective user study involving 36 unique participants. The

study was deployed using Google Forms. For each comparison, participants were shown the results from two retrieval methods: our proposed CARLoS method and one of the four textual baselines (Qwen3, E5, GTE, or BGE).

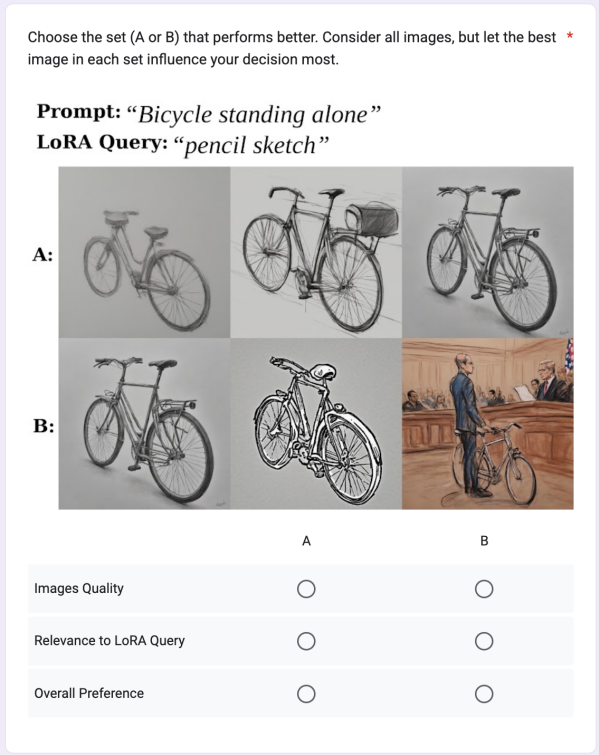

Figure 11. A representative screenshot of the interface used for the double-blind subjective user study. Each question presented two sets of images, labeled 'A' and 'B', generated by the top-3 LoRAs retrieved by CARLoS and a textual baseline for a specific query (e.g., "pencil sketch"). Participants evaluated the sets based on Image Quality, Relevance to the LoRA Query, and Overall Preference. Note that the best image in each set was emphasized as a deciding factor.

H.1.1. Procedure and Randomization

To ensure a rigorous and unbiased evaluation, we implemented the following randomization protocols:

- **Query Selection:** Approximately 100 unique text queries were randomly selected from our comprehensive benchmark set of over 700 queries.
- **Set Labeling:** The results from the CARLoS method were randomly assigned to either "Set A" or "Set B" to maintain the double-blind nature of the study, preventing participants from knowing which set was ours.
- **Baseline Pairing:** CARLoS was randomly paired against one of the four textual retrieval baselines (Qwen3, E5, GTE, BGE) for each question.
- **Image Presentation:** Both sets, A and B, displayed images generated by the respective method's top-3 retrieved LoRAs, applied to a fixed base prompt (e.g., "Bicycle standing alone").

H.1.2. Evaluation Metrics and Aggregation

For each comparison, participants were instructed to evaluate the two image sets based on three criteria (see Figure 11):

1. **Image Quality:** Aesthetic and technical quality of the generations.
2. **Relevance to LoRA Query:** How accurately the overall set reflects the intended semantic or stylistic shift requested by the query.
3. **Overall Preference:** The user's final, holistic choice.

Participants chose between Set A and Set B, for each of the three metrics. A total of approximately 300 unique comparison sub-questions (100 questions $\times$ 3 metrics) were generated. Each sub-question was answered by a minimum of 6 participants, leading to the aggregated preference scores of more than 1800 answered sub-questions reported in Figure 5 of the main paper.

H.2. Participant Details

The study involved 36 unique individual human responders. The participants were recruited from a varied pool, spanning different age groups, genders, and professional occupations. All participants were well-informed of the purpose of the study, which was to assess the quality of generative AI components. This demographic diversity helps ensure that the aggregated results reflect a broad user preference.