StructXLIP: Enhancing Vision-language Models with Multimodal Structural Cues

Supplementary Material

In this supplementary material, we provide a comprehensive analysis and additional details to support the main paper. The content is organized as follows:

- **Dataset details** (Sec. A): We provide visual samples and statistical breakdowns for all datasets (Sec. A). Specifically, we detail the construction and curation process of the specific-domain INSECT dataset (Sec. A.1).
- **Lexicon filtering** (Sec. B): We present the details about lexicon filtering in terms of the used prompts, the appearance-related vocabulary and the filtering process (Sec. B.1). We also provide a statistical analysis on the filtered texts (Sec. B.3).
- **Extended information-theoretic analysis** (Sec. C): We provide detailed theoretical proof (Sec. C.1) and its full supporting empirical analyses (Sec. C.2) across all four datasets.
- **Additional implementation details** (Sec. D): We extend the method description with details on text token extension (Sec. D.1) and the fine-tuning setup (Sec. D.2). Furthermore, we provide computational analysis (Sec. D.3).
- **Additional experimental analyses** (Sec. E): We present additional ablation analysis on the lexicon filtering (Sec. E.1), additional cross-domain evaluation (Sec. E.2), the extended experimental results at deeper ranks (Sec. E.3) and more qualitative results (Sec. E.4) to complement the experimental evaluation in the main paper.

A. Dataset Details

Figs. 1-4 shows sample images with their extracted edge maps, associated original textual descriptions and the filtered textual descriptions from each dataset.

Our experiments are conducted on four datasets spanning different domains and levels of semantic granularity: DCI [36], DOCCI [23], SKETCHY [7], and INSECT [34]. For the general-domain setting, we follow [3], utilize two human-annotated datasets, DCI and DOCCI, which are originally designed for dense image captioning. DCI contains 7,805 natural images, each paired with highly detailed and information-dense descriptions. DOCCI consists of approximately 15k natural scene images collected across diverse geographic regions. Each image is annotated with long, highly compositional and discriminative descriptions, with an average length of 136 words.

In the domain-specific setting, we use SKETCHY and INSECT, two datasets characterized by fine-grained visual concepts and prominent structural properties. SKETCHY is a large-scale fashion dataset containing roughly 46k outfit images, each paired with detailed descriptions covering garment components, fabric, pattern shapes, and spatial relations between parts. Although the average text length is only about 56 words, the descriptions are semantically dense and rich in visual detail. INSECT contains 6k insect images with highly fine-grained and biologically rare categories that are underrepresented in general-purpose pretrained VLMs. Each image is paired with expert-verified biological descriptions covering coloration, wing structures, body-segment proportions, and species-level morphological traits, with an average length of 81 words. Since the original Insect-1M dataset contains substantial redundancy in both images and text, we construct a refined version suitable for fine-tuning on long-text understanding (see in the following).

Regarding dataset splits, we follow the standard protocol for DOCCI and DCI as in [3], with 5,100 and 2,000 images in their test sets, respectively. For SKETCHY, we adopt the official test split of 1.2k images. For INSECT, since the original dataset provides no official split, we construct a split by dividing the curated dataset with an 8:2 train-test ratio.

A.1. Condensed INSECT Dataset Construction

The INSECT dataset is derived from the large-scale insect image repository Insect-1M, which contains over one million images covering approximately 34,000 species, along with hierarchical textual annotations ranging from Phylum down to Species. Despite its scale and richness, the original dataset is not readily suitable for image–text retrieval or cross-modal fine-tuning. First, the dataset does not assign unique descriptions to individual images; instead, it uses an indexed-description mechanism in which large numbers of visually different images share the same short list of text tokens (*e.g.*, descriptionA = [1,2,3]), resulting in *extensive duplication of textual annotations*. Then, the high-level textual descriptions (*e.g.*, Order, Class) are shared across thousands of samples, leading to large textual overhead that are not very meaningful for fine-grained discrimination compared to those description on finer granularity concerning, *e.g.* Genus and Species.

We therefore construct a more suitable version of INSECT for long-text fine-tuning. First, to reduce duplication and increase representational diversity, we design a greedy selection algorithm based on description-ID overlap and select a *core sample set* using an overlap threshold of 6, effectively removing redundant images and repeated descriptions. Second, to create a more challenging cross-family generalization scenario, we apply a strict Family-level out-of-domain split, ensuring that the families in the test set are completely disjoint from those in the training set. Then, to ensure that the textual side focuses on biologically discriminative fea-

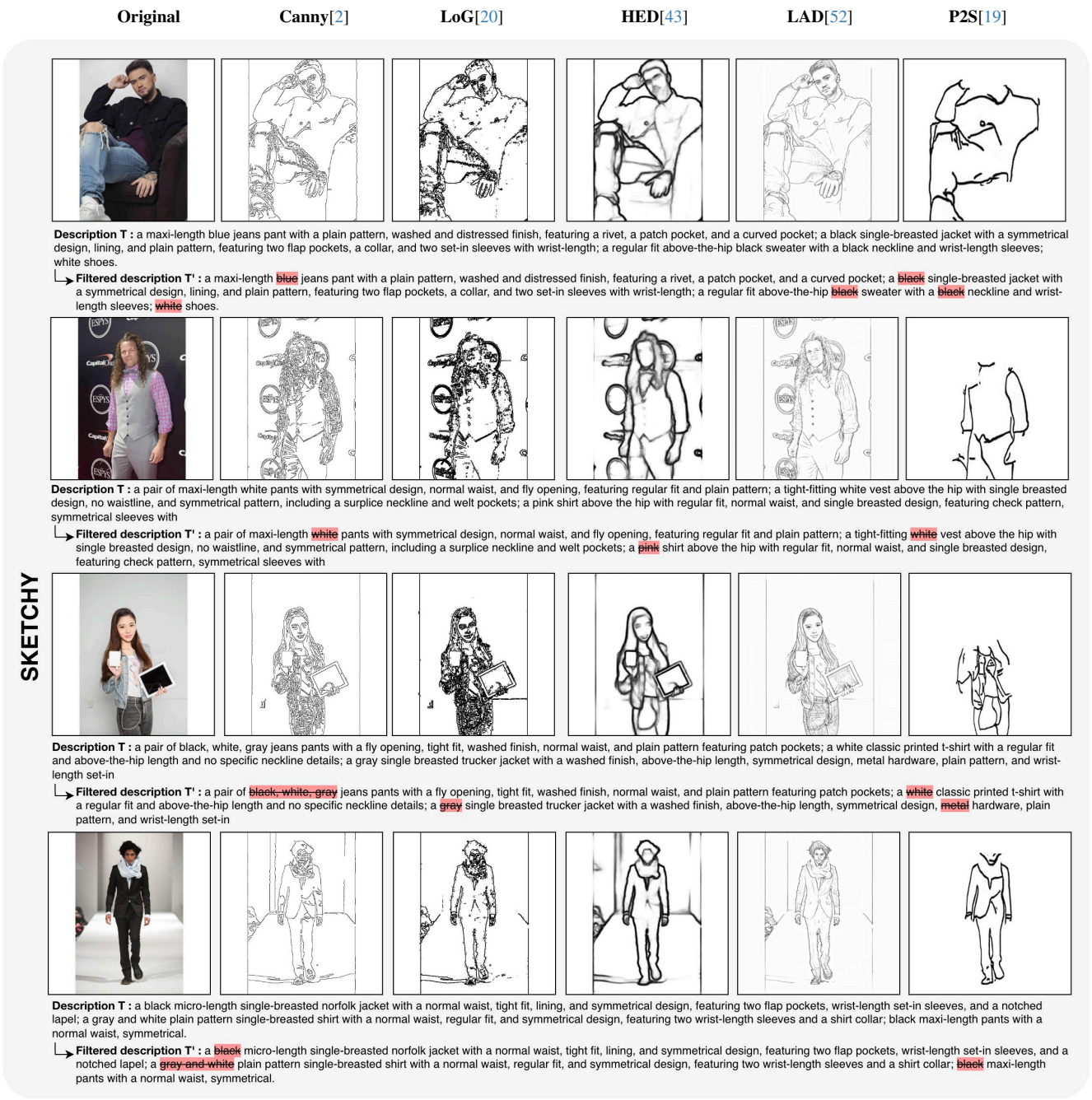


Figure 1. Illustration of SKETCHY dataset samples across different edge-map representations. For each example, we show the original RGB image together with its caption, followed by Canny, LoG, HED, LAD, and P2S edge maps, as well as the filtered caption T′ that preserves only structure-centric information.

tures, we retain only the Genus- and Species-level descriptions and discard higher-level labels, yielding more compact and semantically fine-grained image–text pairs. Through these steps, we obtain a curated 6,000-pair high-quality INSECT dataset. *Both the dataset and the code used for its construction will be released publicly.*

B. Lexicon Filtering

This section provides additional details on the lexicon filtering process described in the main paper (Sec. 3.1), including: i) how we use an LLM to construct the appearance vocabulary; ii) the implementation of our filtering function; and iii) statistical analyses of the filtering effect across the four

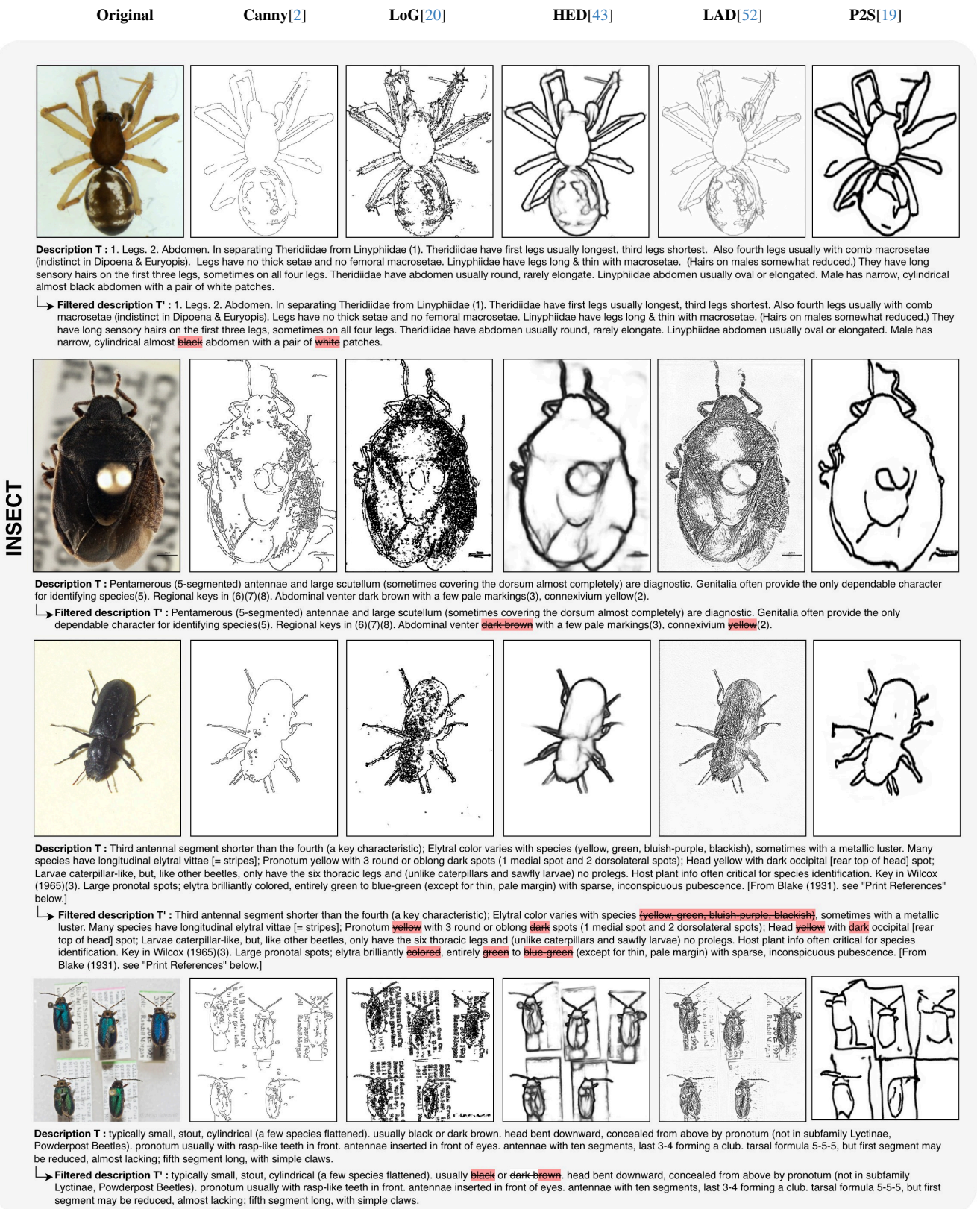


Figure 2. Illustration of INSECT dataset samples across different edge-map representations.

Figure 3. Illustration of DOCCI dataset samples across different edge-map representations.

datasets.

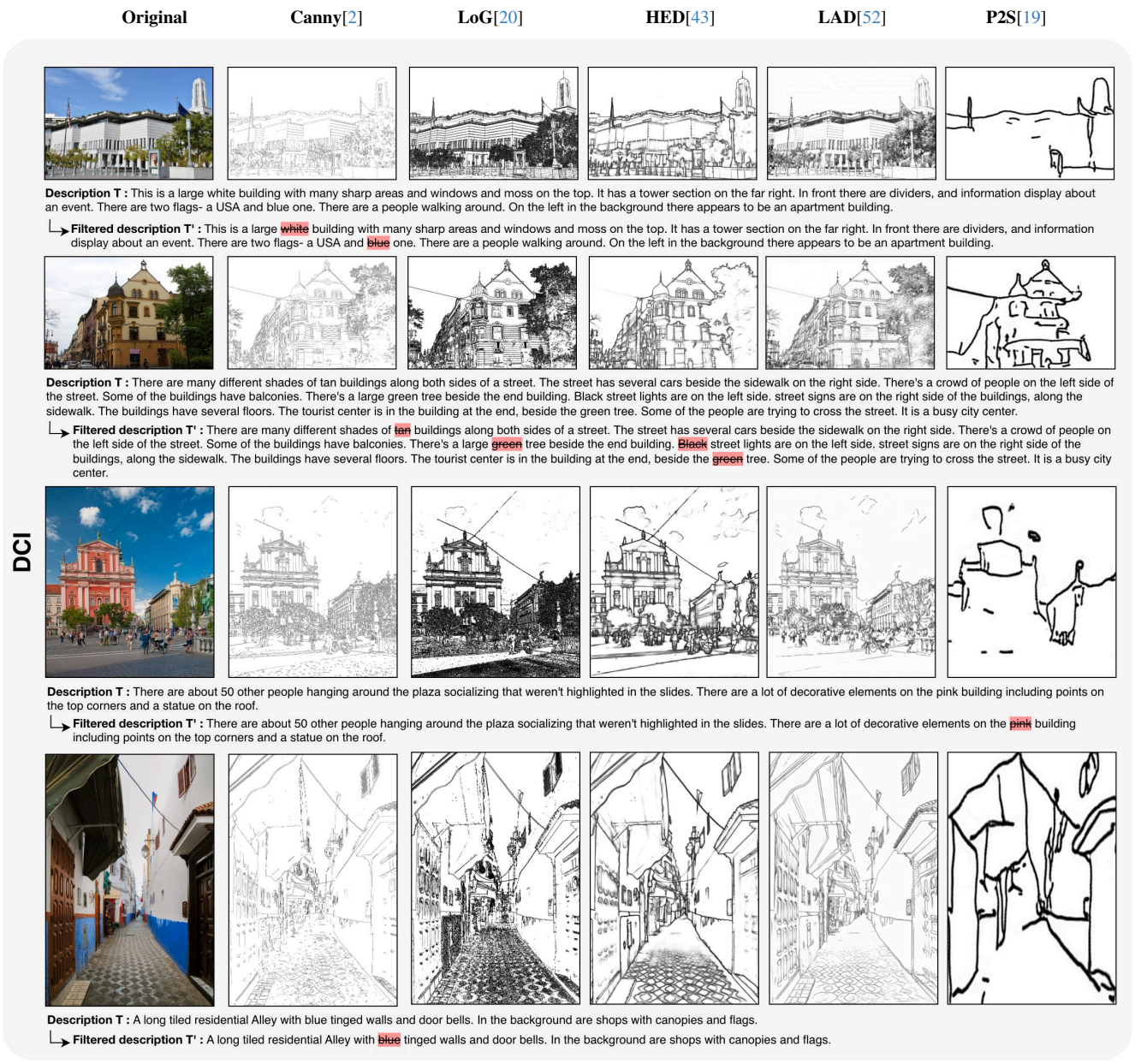


Figure 4. Illustration of DCI dataset samples across different edge-map representations.

B.1. Appearance Vocabulary Generation

To systematically remove appearance-related terms such as color and material from the original text descriptions, we first construct a general-domain appearance lexicon, denoted as $\mathcal{V}^{\mathrm{a}}$, which is later used for our lexicon filtering procedure. In the main paper, this lexicon is defined as $\mathcal{V}^{\mathrm{a}} = \mathcal{LLM}(\mathrm{P})$. Here we provide additional details on how it is obtained.

As shown in Fig. 5, we use a large language model (in practice, ChatGPT-5.1) and provide it with a structured prompt P. The prompt instructs the model to consider a structure-centric representation and to produce a list of appearance-related terms that cannot be inferred from such structural information. The LLM returns a list of 600–800 appearance-only terms, including basic colors, color variants, material categories. The set of returned terms is used to form the final appearance lexicon $\mathcal{V}^{\mathrm{a}}$. The complete vocabulary is listed in Fig. 5. In all experiments, we use the same general-domain appearance lexicon $\mathcal{V}^{\mathrm{a}}$ as the filtering template for both general-domain and domain-specific datasets. The lexicon is built only once during pre-processing and is subsequently kept unchanged, enabling full reproducibility of the pipeline.

(1): General Appearance Lexicon Prompt–Response

USER: You are a vision-language expert. Consider an image representation that contains only edge information (i.e., contours, boundaries, structure, and geometric layout), with all color, shading, material, and surface cues removed. Please produce a comprehensive list of English words and short phrases whose meanings cannot be inferred from such an edge-only representation. Return the list as comma-separated items.

LLM: [red, blue, green, yellow, black, blackish, white, gray, grey, orange, purple, pink, brown, beige, cyan, magenta, turquoise, teal, maroon, navy, violet, indigo, gold, silver, ivory, cream, olive, tan, peach, mint, burgundy, crimson, scarlet, lavender, lilac, azure, teal, aqua, aquamarine, navy blue, sky blue, baby blue, light blue, dark, dark blue, light green, dark green, forest green, lime green, light red, dark red, rose red, wine red, light pink, hot pink, dark gray, light gray, dark grey, light grey, colored, cotton, wool, silk, linen, denim, leather, suede, velvet, satin, chiffon, polyester, nylon, spandex, acrylic, rayon, cashmere, fleece, corduroy, lace, mesh, canvas, tweed, felt, rubber, plastic, metal, steel, iron, aluminum, bronze, brass, ceramic, glass, wood, bamboo, stone, marble, granite, concrete, clay, paper, fur, shearling, down, feather, denier, foam]

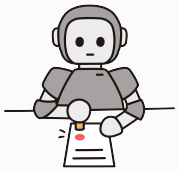


Figure 5. Prompt and response used in our LLM-based construction of the general appearance vocabulary for lexicon filtering. This lexicon is generated only once during preprocessing; the pipeline consistently uses this fixed lexicon thereafter, ensuring full reproducibility.

B.2. Lexicon Filtering Function

In the main paper, we denote the filtering function as $\mathcal{F}(\cdot)$, whose purpose is to remove all appearance-related terms contained in $\mathcal{V}^{\mathrm{a}}$ from the original textual description. Here we provide a detailed explanation of how this function is implemented in practice. We first convert all entries in $\mathcal{V}^{\mathrm{a}}$ into regular-expression matchers. During filtering, these expressions are applied to the input text to identify and remove any appearance terms found in the $\mathcal{V}^{\mathrm{a}}$. The matching process is case-insensitive and respects word boundaries to avoid unintended partial matches. After removing appearance terms, we apply a lightweight grammatical cleanup. Since such removal can produce unnatural or fragmented text, for example, "a blue and white pattern" may temporarily become "a and pattern". We first eliminate redundant spaces and punctuation to prevent repeated whitespace or stray commas. We then remove isolated conjunctions such as "and" or "or," which lose their function once their associated tokens are deleted. Finally, we check whether the filtered sentence still contains sufficient semantic content. If too little meaningful text remains, we revert to the original description to avoid excessive information loss.

B.3. Statistical Analysis on Lexicon Filtering

We provide a statistical analysis of the effect of lexicon filtering across the four datasets. For consistency and ease of comparison, we coarsely categorize $\mathcal{V}^{\mathrm{a}}$ into two classes: color words and material words, and analyze filtering coverage ("*What proportion of captions were modified?*"), target specificity ("*What types of words were removed?*"), and modification intensity ("*How many words were removed on average?*") based on this grouping.

As shown in Fig. 6, A reports the proportion of captions that were modified at least once. SKETCHY and DOCCI exhibit nearly 100% intervention coverage, and DCI reaches 92.8%, indicating that captions in these datasets commonly contain identifiable color or material descriptors. In contrast, INSECT shows only 58.9% modified captions, reflecting that its descriptions inherently emphasize morphology and structure rather than appearance attribute, consistent with the style of biological taxonomic text. B summarizes the composition of removed words (computed over the modified captions). In SKETCHY and INSECT, approximately 97% of removed terms are color-related, with material terms contributing only a negligible fraction. In DOCCI and DCI, material words account for 20% and 17%, respectively, aligning with the fact that captions in these datasets often mention building materials or object surface composition. C shows the average number of removed words per caption and the proportion relative to the original caption length. For the shorter captions of SKETCHY and INSECT, only 2.6-2.9 words are removed on average (corresponding to 3.1% and 10.3% of total caption length). For the longer captions in DOCCI and DCI, 7.1-7.2 words are removed on average, but this corresponds to only 5-6% of the overall text. Our lexicon filtering removes appearance attributes effectively, without disrupting the structural or semantic core of the original caption. Overall, these three analyses demonstrate that lexicon filtering achieves high coverage, strong target specificity, and mild editing intensity, making it both effective and stable across domains.

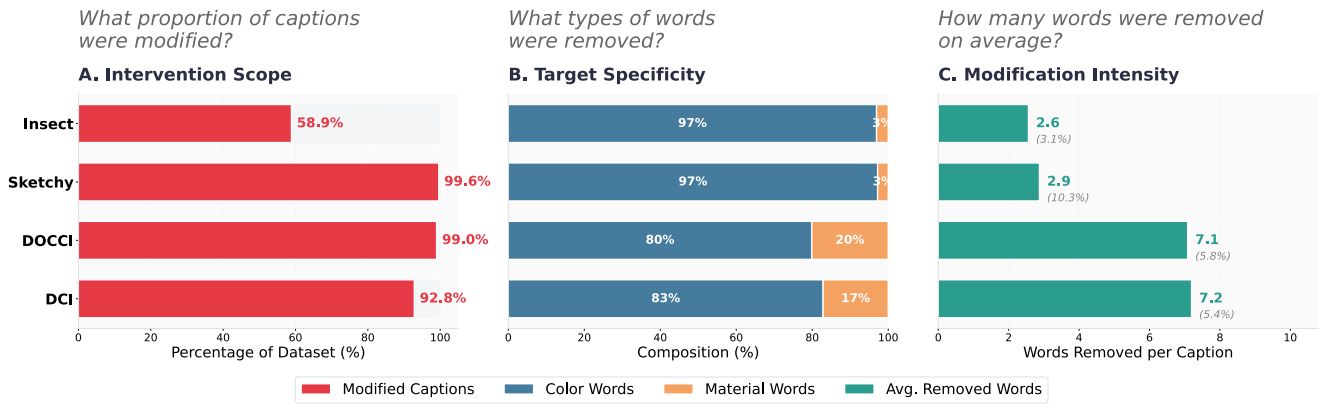


Figure 6. **Statistical analysis of the lexicon filtering effect across the four datasets.** For analysis, removed tokens are coarsely grouped into *color words* and *material words*. (A) **Intervention scope:** percentage of captions in which at least one word was removed. (B) **Target specificity:** composition of removed tokens (computed over the modified captions in Panel A), showing the proportion of color vs. material words. (C) **Modification intensity:** average number of words removed per caption, with the percentage relative to the original caption length shown in parentheses.

C. Extended Information-Theoretic Analysis

C.1. Information-Theoretic Analysis

In this section we expand the theoretical view of Sec. 3.3 of the main paper. Specifically, we analyze the effect of the structure-centric objectives $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ and $\mathcal{L}_{\mathrm{I},\mathrm{I}'}$ under the information-theoretic and optimization lens giving three lemmas and a theorem.

Lemma 1 (Information ordering for structure-centric views)**.** *Let* I *and* T *denote the random variables associated with images and texts, and let* $\mathrm{I}' = \mathcal{E}(\mathrm{I})$ *and* $\mathrm{T}' = \mathcal{F}(\mathrm{T})$ *be their structure-centric counterparts, where* $\mathcal{E}(\cdot)$ *and* $\mathcal{F}(\cdot)$ *are deterministic maps that remove appearance-related information and are not invertible. Then the following inequality holds:*

$$\mathrm{I}_{\mathrm{MI}}(\mathrm{I}', \mathrm{T}') < \mathrm{I}_{\mathrm{MI}}(\mathrm{I}, \mathrm{T}),$$

where $\mathrm{I}_{\mathrm{MI}}(\cdot,\cdot)$ *is the mutual information operator.*

Proof sketch. The pair $(\mathrm{I}', \mathrm{T}')$ is obtained from (I, T) through the deterministic channel $(\mathcal{E}, \mathcal{F})$. By the Data Processing Inequality [33], any such transformation cannot increase mutual information, hence $\mathrm{I}_{\mathrm{MI}}(\mathrm{I}', \mathrm{T}') \leq \mathrm{I}_{\mathrm{MI}}(\mathrm{I}, \mathrm{T})$. The inequality is strict whenever either E or F is not injective, which holds in our setting since edge extraction and lexicon filtering discard appearance cues, rather than introducing new information.

Lemma 2 (InfoNCE lower bounds for the two objectives)**.** *Let* $\mathbf{i} = \mathrm{f}_{\mathtt{img}}(\mathrm{I})$ *and* $\mathbf{t} = \mathrm{f}_{\mathtt{txt}}(\mathrm{T})$ *be the image and text embeddings, and let* $\mathbf{i}' = \mathrm{f}_{\mathtt{img}}(\mathrm{I}')$ *and* $\mathbf{t}' = \mathrm{f}_{\mathtt{txt}}(\mathrm{T}')$ *be the structure-centric embeddings. Assume both* $\mathcal{L}_{\mathrm{I},\mathrm{T}}$ *and* $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ *are symmetric InfoNCE losses with batch size* N*. Then*

$$\mathrm{I}_{\mathrm{MI}}(\mathbf{i}, \mathbf{t}) \geq \log \mathrm{N} - \mathbb{E}[\mathcal{L}_{\mathrm{I},\mathrm{T}}], \quad \mathrm{I}_{\mathrm{MI}}(\mathbf{i}', \mathbf{t}') \geq \log \mathrm{N} - \mathbb{E}[\mathcal{L}_{\mathrm{I}',\mathrm{T}'}].$$

Proof. This is the mere application of [25], stating that the InfoNCE loss is a standard variational lower bound on mutual information. Applying the result to the pairs $(\mathbf{i}, \mathbf{t})$ and $(\mathbf{i}', \mathbf{t}')$ yields the two inequalities. The assumptions on symmetry and batch size match the formulation in Sec. 3.2 of the main paper. In the following, to the sake of clarity, we assume $\mathrm{I}_{\mathrm{MI}}(\mathbf{i}, \mathbf{t}) \approx \mathrm{I}_{\mathrm{MI}}(\mathrm{I}, \mathrm{T})$ and $\mathrm{I}_{\mathrm{MI}}(\mathbf{i}', \mathbf{t}') \approx \mathrm{I}_{\mathrm{MI}}(\mathrm{I}', \mathrm{T}')$.

Lemma 3 (Directional compatibility of gradients)**.** *Let* θ *denote the parameters shared by the vision and text encoders. Consider the gradients* $\nabla_\theta \mathcal{L}_{\mathrm{I},\mathrm{T}}$ *and* $\nabla_\theta \mathcal{L}_{\mathrm{I}',\mathrm{T}'}$*. If the positive pairs in the two losses correspond to the same image–text instances and the encoders are shared, then the expected cosine similarity between the two gradients satisfies*

$$\mathbb{E}\big[\cos(\nabla_\theta \mathcal{L}_{\mathrm{I},\mathrm{T}}, \nabla_\theta \mathcal{L}_{\mathrm{I}',\mathrm{T}'})\big] > 0.$$

Proof sketch. Both losses use the same positive pairs and differ only in the views fed to the encoders (full images and captions vs structure-centric counterparts). The corresponding positive logits are maximized in both objectives, while negatives are pushed apart. This induces aligned update directions for parameters that affect shared features. Under mild regularity assumptions on the encoders, the expected cosine similarity between the two gradients is strictly positive. In practice, this is confirmed by the empirical measurements reported in Fig. 4(c).

Theorem 1 (Effect of the structure-centric auxiliary losses)**.** *Consider the joint objective*

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{I},\mathrm{T}} + \lambda_1 \mathcal{L}_{\mathrm{I}',\mathrm{T}'} + \lambda_2 \mathcal{L}_{\mathrm{I},\mathrm{I}'}, \quad \lambda_1, \lambda_2 > 0.$$

Under the assumptions of the lemmas above, the following properties hold:

1. *The auxiliary alignment task between the pair* $(\mathbf{i}', \mathbf{t}')$ *is information-reduced compared to* $(\mathbf{i}, \mathbf{t})$*, hence* $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ *optimizes a harder objective in the sense of Lemma 1 and 2.*
2. *The gradients of* $\mathcal{L}_{\mathrm{I,T}}$ *and* $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ *are directionally compatible, so the auxiliary loss does not conflict with the main alignment.*
3. *Due to the lower mutual information of* $(\mathbf{i}', \mathbf{t}')$*, the gradient norm of* $\nabla_\theta \mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ *remains non-negligible even when* $\nabla_\theta \mathcal{L}_{\mathrm{I,T}}$ *starts to vanish, which provides persistent optimization signal.*
4. *The consistency term* $\mathcal{L}_{\mathrm{I,I}'}$ *bounds the drift between* $\mathbf{i}$ *and* $\mathbf{i}'$*, so the structure-centric space stays anchored to the semantic manifold of* $\mathbf{i}$ *and fine-tuning remains stable.*

Proof sketch.

Item 1 follows directly from Lemma 1 and the InfoNCE bounds in Lemma 2, which place the pair $(\mathbf{i}', \mathbf{t}')$ at a lower mutual information level than $(\mathbf{i}, \mathbf{t})$. More in the detail, the auxiliary alignment task on $(\mathbf{i}', \mathbf{t}')$ is strictly information-reduced compared to $(\mathbf{i}, \mathbf{t})$, and therefore $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ optimizes a harder objective. More precisely, since $\mathrm{I}' = \mathcal{E}(\mathrm{I})$ and $\mathrm{T}' = \mathcal{F}(\mathrm{T})$ are obtained by applying non-invertible, deterministic maps that remove appearance-related variability, the entropy $\mathrm{H}(\cdot)$ of both variables is reduced:

$$\mathrm{H}(\mathrm{I}') < \mathrm{H}(\mathrm{I}), \qquad \mathrm{H}(\mathrm{T}') < \mathrm{H}(\mathrm{T}).$$

By the Data Processing Inequality, this implies

$$\mathrm{I}_{\mathrm{MI}}(\mathrm{I}', \mathrm{T}') \leq \mathrm{I}_{\mathrm{MI}}(\mathrm{I}, \mathrm{T}).$$

In addition, the structure-centric views induce a contraction of the positive pair distribution: the space of valid matches becomes smaller, the intra-class variability is suppressed, and the negative samples become less separable in the embedding space. These effects lower the effective signal-to-noise ratio of the InfoNCE objective, which increases the difficulty of the optimization landscape associated with $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$. Consequently, the gradients generated by $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ tend to persist longer during fine-tuning, since reaching its minimum requires modeling more subtle, geometry-driven correspondences that remain unresolved after the full-information objective $\mathcal{L}_{\mathrm{I,T}}$ has already saturated.

Item 2 follows from Lemma 3 and concerns the directional compatibility of the gradients. Since both $\mathcal{L}_{\mathrm{I,T}}$ and $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ operate on the same positive image–text instances and update the same encoder parameters, their contrastive objectives induce aligned update rules: both maximize the positive logits and suppress the negative logits associated with the same underlying pairs, even though the views differ. Let the gradients be:

$$\mathrm{g} = \nabla_\theta \mathcal{L}_{\mathrm{I,T}}, \qquad \mathrm{g}' = \nabla_\theta \mathcal{L}_{\mathrm{I}',\mathrm{T}'}.$$

Lemma 3 ensures that their expected cosine similarity satisfies

$$\mathbb{E}\big[\cos(\mathrm{g}, \mathrm{g}')\big] > 0.$$

Therefore, the auxiliary gradient does not conflict with the semantic direction promoted by the main loss. Instead, it expands the set of admissible descent directions within a compatibility cone, enriching the optimization trajectory without introducing destructive interference.

Item 3 relies on the fact that, owing to the information-reduced nature of $(\mathrm{I}', \mathrm{T}')$, the loss $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ converges more slowly than $\mathcal{L}_{\mathrm{I,T}}$ and produces non-vanishing gradients even when the main loss has already flattened. Indeed, $\mathcal{L}_{\mathrm{I,T}}$ optimizes an InfoNCE objective associated with the higher mutual information quantity $\mathrm{I}_{\mathrm{MI}}(\mathrm{I}, \mathrm{T})$, whose landscape typically admits a faster descent: the positive pair $(\mathbf{i}, \mathbf{t})$ carries rich appearance and structural cues, which help discriminate positives from negatives early in fine-tuning. In contrast, $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ maximizes the lower $\mathrm{I}_{\mathrm{MI}}(\mathrm{I}', \mathrm{T}')$, where both $\mathbf{i}'$ and $\mathbf{t}'$ have reduced entropy because appearance information has been removed by $\mathcal{E}(\cdot)$ and $\mathcal{F}(\cdot)$. This contraction of the feature space has two effects:

- the separation between positives and negatives becomes smaller, making the contrastive objective harder to optimize;
- the gradients associated with hard positives and hard negatives decay more slowly, since the model must rely purely on geometric and structural cues to improve the logits.

Formally, since the InfoNCE gradients satisfy

$$\nabla_\theta \mathcal{L}_{\mathrm{I,T}} = \mathbb{E}\big[\mathrm{g}(\mathbf{i}, \mathbf{t})\big], \qquad \nabla_\theta \mathcal{L}_{\mathrm{I}',\mathrm{T}'} = \mathbb{E}\big[\mathrm{g}(\mathbf{i}', \mathbf{t}')\big],$$

and the score function $\mathrm{g}(\cdot)$ for $(\mathbf{i}', \mathbf{t}')$ has larger relative variance due to the reduced mutual information, the expected magnitude of $\nabla_\theta \mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ remains positive for a longer portion of fine-tuning. Empirically, this manifests in a later convergence time for $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ compared to $\mathcal{L}_{\mathrm{I,T}}$, and in a sustained gradient norm that continues to provide meaningful updates even after the main contrastive gradients approach zero. This behavior is consistent with the dynamics shown in Fig. 4(a,b) of the main paper, where $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ displays a slower flattening and a more persistent gradient profile.

Finally, **Item 4** exploits the form of $\mathcal{L}_{\mathrm{I,I}'}$, which penalizes large angular deviations between $\mathbf{i}$ and $\mathbf{i}'$ and therefore constrains the structure-centric representations to remain close to their full-information counterparts. Specifically, since $\mathcal{L}_{\mathrm{I,T}}$ and $\mathcal{L}_{\mathrm{I}',\mathrm{T}'}$ correspond to two correlated but not identical contrastive objectives, their gradients span a larger subspace than either loss alone. Formally, let

$$\mathrm{g} = \nabla_\theta \mathcal{L}_{\mathrm{I,T}}, \qquad \mathrm{g}' = \nabla_\theta \mathcal{L}_{\mathrm{I}',\mathrm{T}'}.$$

From Item 2 we know that $\cos(\mathrm{g}, \mathrm{g}') > 0$ in expectation, which guarantees compatibility. However, because $(\mathbf{i}', \mathbf{t}')$

belongs to an information-reduced space and responds differently to hard negatives and subtle geometric patterns, one has

$$g' \notin \text{span}\{g\},$$

so the matrix $[g,\ g']$ has rank 2 almost everywhere. This implies that the combined update

$$g_{\text{tot}} = g + \lambda_1 g' + \lambda_2 \nabla_\theta \mathcal{L}_{I,I'},$$

explores descent directions that the main loss alone cannot access. This additional set of directions reduces the risk of premature convergence to shallow minima (a common issue in contrastive objectives), improves escape from flat regions of the loss surface, and increases the robustness of SGD trajectories. In the language of multi-objective optimization, $\mathcal{L}_{I',T'}$ introduces a complementary gradient component that expands the effective feasible region of updates while maintaining alignment with the main semantic objective. The consistency loss $\mathcal{L}_{I,I'}$ ensures that this expanded search space remains stable by preventing $\mathbf{i}'$ from drifting too far from $\mathbf{i}$. As a result, the combined gradient retains diversity without diverging from the semantic manifold defined by the original embeddings, yielding a balanced and stable optimization process.

Taken together, these properties explain why the auxiliary structure-centric alignment acts as a beneficial regularizer: ***it introduces correlated but information-reduced gradients that improve convergence stability and lead to more robust alignment.***

C.2. Empirical Analyses

In Sec. 3.3 of the main paper, we reported one representative study of our information-theoretic analysis using the SKETCHY dataset. Here we provide the remaining experiments conducted on the other three datasets used in our evaluation: INSECT, DOCCI and DCI. To verify that the information-theoretic analysis holds consistently across different data scales, we conduct the same experiments on reduced-scale versions of each dataset containing 5%, 20%, and 50% of the original data. We further include experiments performed under the finetuning-agnostic improvement setting combined with the second-best method GOAL [3], to confirm that the same behaviors arise consistently across different fine-tuning configurations. For each experiment, we show the same three quantities presented in Fig. 7 and Fig. 8: (a) the convergence behavior of the two contrastive losses $\mathcal{L}_{I,T}$ and $\mathcal{L}_{I',T'}$, (b) the evolution of their gradient norms and their ratio, and (c) the cosine similarity between the corresponding gradients.

Across all datasets and settings, we observe the same consistent patterns as reported in Sec. 3.3. The auxiliary objective $\mathcal{L}_{I',T'}$ which maximizes the mutual information $I_{MI}(I', T')$ under the information-reduction mappings $\mathcal{E}(\cdot)$ and $\mathcal{F}(\cdot)$, exhibits a later convergence than the main objective $\mathcal{L}_{I,T}$. This is consistent with the Data Processing Inequality [33], which implies $I_{MI}(I', T') \leq I_{MI}(I, T)$, and makes the auxiliary alignment problem inherently more difficult. The gradient of $\mathcal{L}_{I',T'}$ remains informative even after the gradient of $\mathcal{L}_{I,T}$ flattens, providing persistent optimization signals that steer the model toward semantically coherent and structurally consistent minima. This behavior is further reflected in the positive cosine similarity between $\nabla_\theta \mathcal{L}_{I,T}$ and $\nabla_\theta \mathcal{L}_{I',T'}$ throughout the optimization trajectory, confirming that the two tasks pursue compatible optima in the parameter space.

The consistent observation across all datasets and settings provides a strong empirical support to our theoretical interpretation in Sec. 3.3: information-reduced auxiliary objectives act as implicit regularizers that introduce controlled gradient diversity, as suggested by analyses in multitask learning, multi-objective optimization, and contrastive representation learning [28, 40, 58]. Such diversity expands the effective search space and stabilizes convergence, while preserving optimization compatibility through aligned gradient directions. The behavior of StructXLIP is agnostic to the dataset, and the findings drawn from the representative Sketchy experiment generalizes to other evaluated datasets.

D. Additional implementation details

D.1. Text Token Extension

To enable base models like CLIP and SigLIP-2 to handle text sequences longer than their original 77 tokens limit, we fully follow the method proposed in Long-CLIP [51]. Specifically, we implement a positional encoding extension that expands the maximum sequence length of all models to 248 tokens. This process is completed before any fine-tuning and follows a preserve-interpolate-extrapolate strategy. First, we keep the first 20 original position vectors unchanged. Then, we expand the middle part by pairwise linear interpolation. Finally, we use linear extrapolation based on the last two original encodings to fill the remaining positions.

It is worth noting that Long-CLIP is originally designed and optimized for the CLIP architecture. We applied the same text token extension mechanism to all models in this study, including both the CLIP and SigLIP-2 baselines and their variants using our proposed losses. Nonetheless, we do acknowledge that directly applying this extension to SigLIP-2 might not be the optimal way, as SigLIP-2 is trained differently from CLIP. Yet, as there exists no prior work with established techniques for extending SigLIP-2 to long-text scenarios, we opt to this option to ensure experimental consistency and fair comparison conditions.

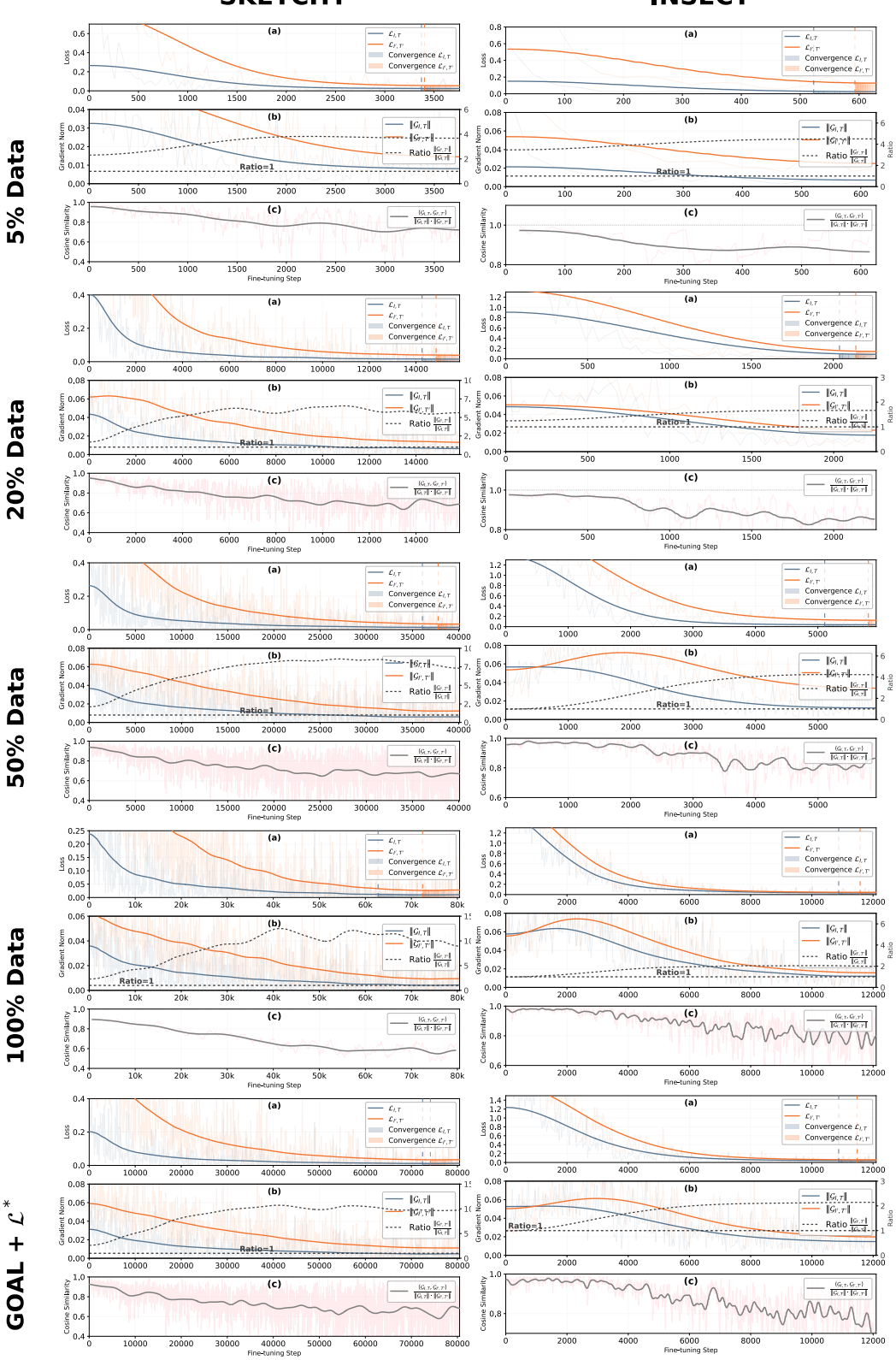


Figure 7. The convergence behavior of the two contrastive losses $\mathcal{L}_{\mathrm{I,T}}$ and $\mathcal{L}_{\mathrm{I',T'}}$ (a), the evolution of their gradient norms and their ratio (b), and the cosine similarity between the corresponding gradients (c), evaluated on all two datasets under different percentages of dataset size.

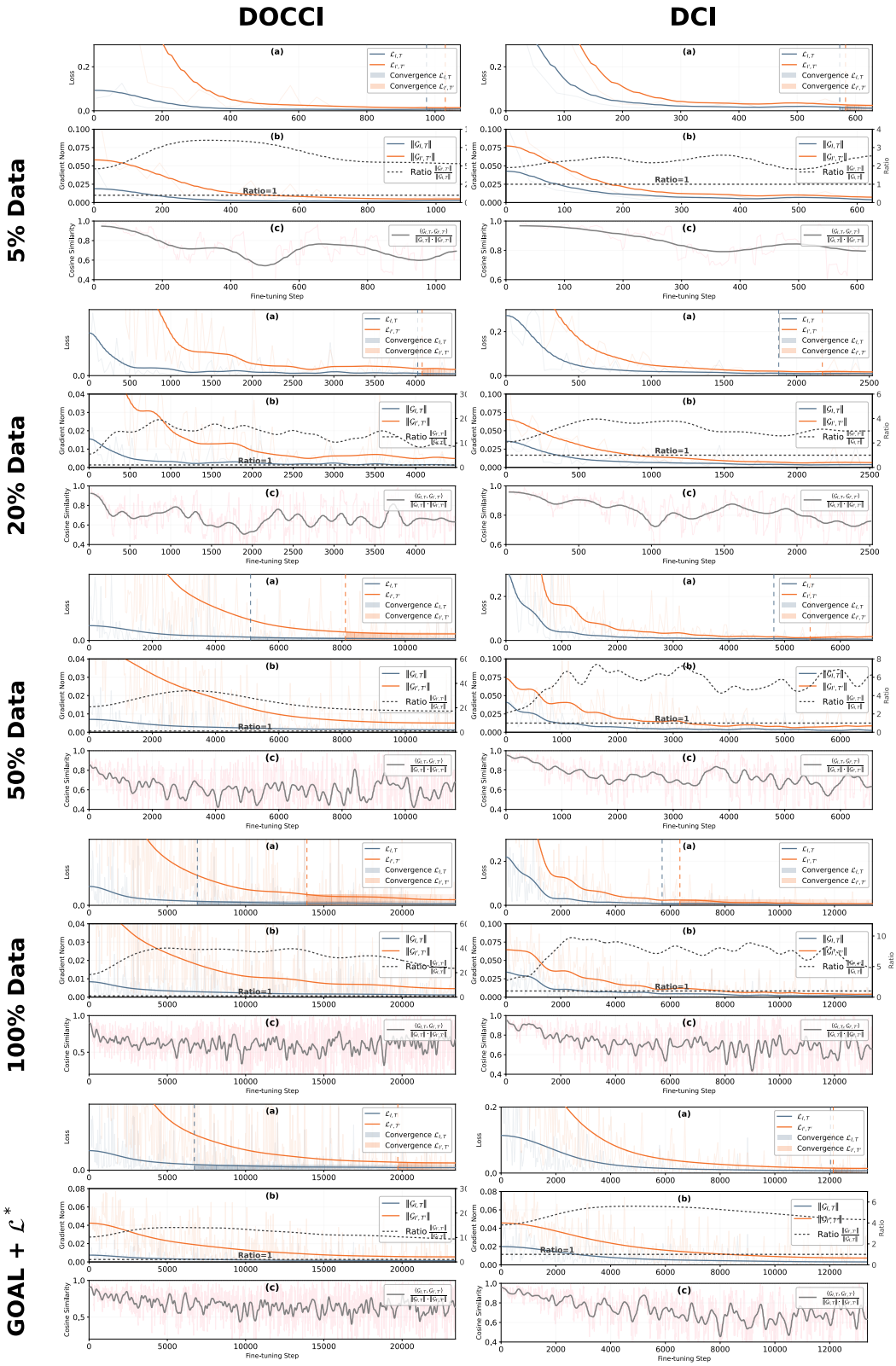


Figure 8. The convergence behavior of the two contrastive losses $\mathcal{L}_{\mathrm{I,T}}$ and $\mathcal{L}_{\mathrm{I',T'}}$ (a), the evolution of their gradient norms and their ratio (b), and the cosine similarity between the corresponding gradients (c), evaluated on all two datasets under different percentages of dataset size.

Table 1. **Comparison of general vs. domain-specific appearance lexicons.** *Text→Image* and *Image→Text* retrieval on SKETCHY and INSECT. **Bold** denotes the best performance.

Setting	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@20	R@50
SKETCHY										
General Lexicon Filter	**69.86**	90.85	95.42	**98.36**	**99.22**	**68.22**	**90.67**	**95.68**	**97.75**	**99.22**
Domain-Specific Lexicon Filter	69.00	**90.87**	**95.60**	98.33	**99.22**	68.14	90.53	95.21	97.40	99.10
INSECT										
General Lexicon Filter	**9.93**	26.60	38.34	**56.99**	**69.34**	**9.50**	26.60	39.64	**54.92**	**68.65**
Domain-Specific Lexicon Filter	9.67	**28.50**	**39.46**	55.09	67.70	9.07	**28.07**	**39.81**	54.32	68.48

D.2. Fine-tuning Details

We fine-tune the model for 10 epochs using the AdamW optimizer with a batch size of 16. The initial learning rate is set to 5×10^{-6}, and we employ a Cosine Annealing scheduler that decays the learning rate to 0 over the course of fine-tuning, without restarts. The weight decay is set to 0.05. For the contrastive objectives, the global temperature parameter τ is learnable (initialized from the pre-trained CLIP and clamped to a maximum logit scale of 3.5), while the local structure alignment temperature γ is fixed at 0.07 to encourage sharp local correspondences.

D.3. Computational Analysis

To evaluate the additional computational overhead introduced by our method, we measure two key preprocessing steps: the local-region segmentation used in $\mathcal{L}^{\text{local}}_{\text{I}',\text{T}'}$ and the edge-map extraction. In practice, we use FastSAM [54] for local segmentation. Due to the large variation in image resolution across datasets, the average per-image inference time also varies: 3.6 ms/image on the lower-resolution Sketchy dataset, 17.1 ms/image on INSECT, and 17.3 ms / 15.9 ms on the higher-resolution DOCCI and DCI datasets, respectively. For edge extraction, we use the Canny detector, which is lightweight, with timings of 3.77 ms/image on Sketchy, 4.59 ms/image on INSECT, and 19.05 ms/16.18 ms on DOCCI and DCI. Higher image resolutions lead to more processing time. Importantly, both pre-processing steps are executed **only once** before fine-tuning.

During fine-tuning, the overall per-batch runtime of our method is approximately 0.17 s, which is comparable to existing CLIP fine-tuning approaches such as Long-CLIP (0.10 s), GOAL (0.18 s), SmartCLIP (0.07 s), and FineLIP (0.05 s). The inference time is the same as as standard CLIP-based methods. Overall, StructXLIP introduces very lightweight (and affordable) additional computation, and maintains the same inference efficiency as standard CLIP-based methods.

E. Additional Experimental Analyses

E.1. Ablation on Appearance Lexicons

We also conducted the ablation investigating *whether constructing dataset-specific appearance vocabularies would lead to better performance, compared to a general lexicon.* To this end, we built two domain-specific appearance lexicons for the fashion dataset SKETCHY and the biological dataset INSECT. For each dataset, examples shown in Fig. 10, instructing the LLM to generate appearance terms particularly relevant to that domain. We then substituted the general $\mathcal{V}^{\text{a}}$ with these domain-specific vocabularies and re-trained our method under exactly the same fine-tuning configuration.

From Tab. 1, we observe that across both the fashion domain (SKETCHY) and the biological domain (INSECT), the domain-specific lexicons yields on-par performance, compared to the general appearance lexicon on most retrieval metrics. This is because the general lexicon already has a *broad coverage* over appearance-related attributes. Instead, the domain-specific lexicon, despite being more specialized, adds only a small number of additional terms into the vocabulary, offering negligible marginal benefit in practice. For this reason, in all our experiments, the appearance vocabulary $\mathcal{V}^{\text{a}}$ used in the lexicon filter is obtained from the general prompt described in Sec. B.1, and the same $\mathcal{V}^{\text{a}}$ is applied across all datasets.

Table 2. **Cross-domain generalization from General (DOCCI) to Specific (Sketchy).** Models are trained on the general dense-captioned dataset (DOCCI) and tested on abstract sketches (Sketchy). Due to the significant domain shift, all models experience a performance drop, but StructXLIP maintains better robustness. Values are Recall@K (%). In-domain results (DOCCI→DOCCI) are in *italic* for reference. Best cross-domain results are in **bold**.

Setting	R@1	R@5	R@10	R@1	R@5	R@10
Fine-tune on DOCCI (General) → Test on Sketchy (Specific)						
Long-CLIP (In-domain)	*64.49*	*87.67*	*93.43*	*63.08*	*87.45*	*93.14*
Long-CLIP (Cross-domain)	8.12	20.03	28.24	10.71	24.27	33.68
GOAL (In-domain)	*79.47*	*96.65*	*98.69*	*79.43*	*96.14*	*97.25*
GOAL (Cross-domain)	8.72	19.60	27.46	9.76	26.86	38.17
StructXLIP (In-domain)	***83.04***	***97.06***	***98.96***	***81.59***	***96.94***	***98.78***
StructXLIP (Cross-domain)	**8.96**	**21.85**	**30.57**	**12.26**	**32.21**	**42.23**

E.2. Additional Cross-domain Evaluation

Complementing to the cross-domain evaluation in the main paper, Table 2 reports the cross-domain evaluation from General (DOCCI) to Specific (SKETCHY). When models are fine-tuned on the general-domain dataset DOCCI and tested on the fashion-focused Sketchy dataset, all methods exhibit a clear performance drop due to the large domain gap. DOCCI contains diverse real-world scenes with dense captions, whereas Sketchy mainly features fashion items. Despite this strong domain shift, StructXLIP consistently achieves the best cross-domain performance across all Recall@K metrics.

E.3. Results at Deeper Ranks

We report the full experimental analyses with Recall@K also at deeper ranks (K = 25, 50) on all four datasets.

Table 3 shows the cross-modal retrieval at full ranks up to K=50. Our method consistently maintains positive margins over the strongest competitors, even at deeper ranks.

Table 4 reports the Recall@K at full ranks for the plug-and-play effectiveness of $\mathcal{L}^*$ onto different finetuning methods. At top ranks (R@1/R@5/R@10), $\mathcal{L}^*$ yields consistent and often substantial improvements, with particularly large gains for lightweight or parameter-constrained methods such as FineLIP, LoRA, and DoRA (improvements ranging from 5% to 18%). On deeper ranks (R@25 and R@50), $\mathcal{L}^*$ continues to provide stable and uniform benefits across nearly all model–dataset combinations. For DOCCI and DCI, the baselines already achieve extremely high R@50 (>97%), yet $\mathcal{L}^*$ still contributes an additional 0.1–1.4% improvement.

Finally, Tab. 5 reports the data efficiency analysis at full ranks. We observe that StructXLIP consistently shows strong data efficiency across all fine-tuning dataset sizes (5%, 20%, 50%). In the most challenging 5% low-data regime, StructXLIP achieves the best overall results. When the data size increases to 20%, all methods improve, yet StructXLIP remains the best performer with noticeable gains on most metrics. At 50% data, although the baselines begin to saturate, StructXLIP still delivers the best results, providing 1–5% improvements on R@1 and R@5, and enhancing deeper-rank performance (R@25 / R@50) on DOCCI and DCI. Overall, StructXLIP demonstrates strong generalization and high data efficiency across all four datasets and all data-scale settings.

E.4. More Qualitative Results

Figure 9 presents additional qualitative results of StructXLIP and the second-best method GOAL on all four datasets. Specifically, we highlight with color some parts of the long texts that are describing some visual objects, and showcase the attention maps between such object-centric texts on the image. It is clear that StructXLIP, compared to the second-best method GOAL, demonstrates a better correspondence between the visual objects and their rich textual descriptions. For instance, in the DOCCI example, given the textual description regarding "the daisy", StructXLIP produces more visible attention map on the daisy petals compared to GOAL, and it also yields more localized attention map regarding the "tree line". Similar patterns can be observed in the DCI samples too, where StructXLIP is more capable in capturing accurate and localized attention map close to the visual object being described. On the specific-domain Sketchy dataset, while StructXLIP exhibits a better coverage on the described visual part, overall both StructXLIP and GOAL are able to capture the correct object. We hypothesize that this might be due to the fact Sketchy dataset contains mostly object-centric, where the model wearing outfits are dominant and mostly centered, which can be a bias that models can easily capture. On the other hand, the INSECT dataset is less represented by pre-trained VLMs, thus their attention map are generally less accurate and localized. Yet, StructXLIP still demonstrates a better alignment between the rich text and the visual counterpart, as evidenced by the more localized attention map on the "antennae of females" and the "thorax" compared to GOAL.

Table 3. **Crossmodal retrieval at full ranks on SKETCHY, INSECT, DOCCI and DCI.** We report mean Recall@K (%) ± standard deviation for K = 1, 5, 10, 25, 50 on both *Text→Image* and *Image→Text*. All results are averaged over three independent runs with random seeds 42, 1337 and 3407. **Bold** indicates the best result, while underline denotes the second-best.

SKETCHY

Method	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50
Long-CLIP[ECCV'24]	54.32±0.46	80.14±2.27	88.43±1.88	95.25±0.78	98.27±0.75	52.76±1.18	80.31±1.89	88.08±1.50	95.16±1.03	97.75±1.00
FineLIP[CVPR'25]	40.59±1.91	71.16±1.28	81.78±0.63	91.27±1.72	95.94±0.57	40.33±1.04	72.11±0.05	82.38±0.11	91.45±1.85	95.77±0.69
SmartCLIP[CVPR'25]	50.73±1.11	81.09±0.93	94.56±1.67	96.11±0.44	99.05±1.28	51.30±0.52	80.83±1.89	94.04±0.73	95.51±1.21	98.96±0.36
GOAL[CVPR'25]	63.21±0.47	87.13±1.58	93.44±0.35	97.67±1.09	99.05±0.19	62.44±1.37	87.82±0.95	92.31±1.44	96.98±0.29	99.00±0.54
StructXLIP	**69.86±0.46**	**90.85±0.09**	**95.42±0.07**	**98.61±0.35**	**99.22±0.00**	**68.22±0.45**	**90.67±0.12**	**95.68±0.13**	**98.03±0.39**	**99.08±0.20**

INSECT

Method	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50
Long-CLIP[ECCV'24]	8.20±0.87	23.83±0.68	34.97±0.25	51.21±0.31	63.30±0.27	9.41±0.84	24.78±0.62	37.31±0.88	53.18±0.47	63.39±0.98
FineLIP[CVPR'25]	8.46±0.59	23.32±0.81	33.59±0.44	51.21±1.93	66.58±0.69	6.86±0.74	23.75±1.41	34.46±0.53	52.42±1.26	65.37±0.88
SmartCLIP[CVPR'25]	4.84±0.66	16.84±1.52	34.63±0.37	39.03±1.18	57.60±0.44	4.66±0.91	15.46±0.33	34.02±1.74	39.72±0.85	58.38±1.29
GOAL[CVPR'25]	8.81±0.07	24.35±0.99	35.84±1.62	55.44±0.72	67.46±0.91	8.55±0.38	25.91±0.36	36.18±0.66	53.02±1.14	66.41±0.52
StructXLIP	**9.93±0.90**	**26.60±1.20**	**38.34±1.02**	**56.99±0.61**	**69.34±0.90**	**9.50±0.80**	**26.60±0.29**	**39.64±0.64**	**54.92±0.46**	**68.65±0.67**

DOCCI

Method	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50
Long-CLIP[ECCV'24]	64.49±0.38	87.67±0.43	93.43±0.38	97.73±0.26	99.14±0.11	63.08±0.32	87.45±0.44	93.14±0.30	97.45±0.17	99.02±0.12
FineLIP[CVPR'25]	67.80±1.28	90.22±0.56	94.84±0.31	98.22±1.49	99.45±0.52	66.39±0.44	89.12±1.22	94.47±0.67	97.90±0.38	99.20±0.93
SmartCLIP[CVPR'25]	74.92±0.66	94.08±0.29	97.31±1.12	99.37±0.44	99.82±0.18	74.91±0.53	94.04±0.72	97.29±0.36	99.32±0.91	99.84±0.27
GOAL[CVPR'25]	79.47±0.41	96.65±1.33	98.69±0.02	99.69±0.57	99.92±0.14	79.43±0.88	96.14±0.38	97.25±0.73	99.61±1.12	99.90±0.19
StructXLIP	**83.04±0.05**	**97.06±0.20**	**98.96±0.04**	**99.84±0.02**	**99.98±0.01**	**81.59±0.34**	**96.94±0.04**	**98.78±0.03**	**99.76±0.03**	**99.92±0.02**

DCI

Method	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50
Long-CLIP[ECCV'24]	59.23±0.72	80.89±0.55	87.04±0.48	92.60±0.31	95.10±0.27	60.13±0.69	81.44±0.52	87.54±0.46	92.85±0.33	95.60±0.30
FineLIP[CVPR'25]	66.13±1.77	85.34±0.44	89.79±1.06	94.14±0.59	96.35±0.52	64.58±0.89	84.59±0.71	89.54±1.68	94.00±0.34	96.40±1.08
SmartCLIP[CVPR'25]	69.88±1.41	86.64±0.63	94.05±1.18	95.00±0.31	97.25±0.77	70.94±0.52	87.04±1.29	92.77±0.84	95.75±0.48	97.05±1.03
GOAL[CVPR'25]	72.64±0.55	89.89±1.22	93.70±0.33	95.75±0.66	97.25±0.41	72.84±1.11	**90.50±0.40**	93.20±0.81	96.60±0.57	97.60±0.22
StructXLIP	**75.90±0.50**	**90.00±0.40**	**95.15±0.39**	**95.95±0.13**	**97.85±0.15**	**74.39±0.16**	89.90±0.05	**94.30±0.23**	**96.75±0.23**	**97.75±0.10**

Table 4. **Plug-and-play enhancement of our $\mathcal{L}^*$ on CLIP-based finetuning.** Results on SKETCHY, INSECT, DOCCI, and DCI for *Text→Image* and *Image→Text* retrieval. We report R@1, R@5, R@10, R@25, and R@50. Upper: SKETCHY and INSECT; Lower: DOCCI and DCI. Best in **bold**, with gain in ↑ green.

Method	SKETCHY										INSECT									
	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50
Long-CLIP	54.32	80.14	88.43	95.25	98.27	52.76	80.31	88.08	95.16	97.75	8.20	23.83	34.97	51.21	63.30	**9.41**	24.78	37.31	53.18	63.39
+our $\mathcal{L}^*$	**59.24**	**85.32**	**91.45**	**96.20**	**98.53**	**59.59**	**84.37**	**91.02**	**96.29**	**98.45**	**9.24**	**25.39**	**36.36**	**52.33**	**66.84**	9.38	**27.29**	**38.60**	**55.35**	**66.15**
Δ	↑ 4.92	↑ 5.18	↑ 3.02	↑ 0.95	↑ 0.26	↑ 6.83	↑ 4.06	↑ 2.94	↑ 1.13	↑ 0.70	↑ 1.04	↑ 1.56	↑ 1.39	↑ 1.12	↑ 3.54	↓ 0.03	↑ 2.51	↑ 1.29	↑ 2.17	↑ 2.76
FineLIP	40.59	71.16	81.78	91.27	95.94	40.33	72.11	82.38	91.45	95.77	8.46	23.32	33.59	51.21	66.58	**6.86**	23.75	34.46	52.42	65.37
+our $\mathcal{L}^*$	**59.15**	**85.23**	**91.28**	**96.63**	**99.40**	**58.55**	**84.54**	**90.07**	**96.20**	**99.05**	**8.89**	**23.83**	**35.75**	**53.63**	**67.18**	6.56	**23.76**	**36.01**	**53.45**	**66.84**
Δ	↑ 18.56	↑ 14.07	↑ 9.50	↑ 5.36	↑ 3.46	↑ 18.22	↑ 12.43	↑ 7.69	↑ 4.75	↑ 3.28	↑ 0.43	↑ 0.51	↑ 2.16	↑ 2.42	↑ 0.60	↓ 0.30	↑ 0.01	↑ 1.55	↑ 1.03	↑ 1.47
SmartCLIP	50.73	81.09	94.56	96.11	99.05	51.30	80.83	**94.04**	95.51	98.96	4.84	16.84	34.63	39.03	57.60	4.66	15.46	34.02	39.72	58.38
+our $\mathcal{L}^*$	**52.94**	**81.26**	**94.77**	**96.13**	**99.20**	**52.33**	**80.92**	**94.04**	**95.60**	**98.96**	**5.61**	**16.89**	**34.80**	**40.16**	**60.61**	**5.18**	**15.80**	**34.07**	**39.72**	**59.20**
Δ	↑ 2.21	↑ 0.17	↑ 0.21	↑ 0.02	↑ 0.15	↑ 1.03	↑ 0.09	0.00	↑ 0.09	0.00	↑ 0.77	↑ 0.05	↑ 0.17	↑ 1.13	↑ 3.01	↑ 0.52	↑ 0.34	↑ 0.05	0.00	↑ 0.82
GOAL	63.21	87.13	93.44	97.67	99.05	62.44	87.82	92.31	96.98	99.00	**8.81**	24.35	35.84	55.44	67.46	8.55	25.91	36.18	53.02	66.41
+our $\mathcal{L}^*$	**67.88**	**90.33**	**95.16**	**98.53**	**99.65**	**68.48**	**89.81**	**94.82**	**98.27**	**99.31**	**8.81**	**28.07**	**38.36**	**56.65**	**68.83**	**8.75**	**27.12**	**39.21**	**54.49**	**68.05**
Δ	↑ 4.67	↑ 3.20	↑ 1.72	↑ 0.86	↑ 0.60	↑ 6.04	↑ 1.99	↑ 2.51	↑ 1.29	↑ 0.31	0.00	↑ 3.72	↑ 2.52	↑ 1.21	↑ 1.37	↑ 0.20	↑ 1.21	↑ 3.03	↑ 1.47	↑ 1.64
SigLIP2	68.91	90.85	95.16	**98.79**	99.57	66.75	90.24	93.96	**98.10**	99.31	6.37	21.07	31.87	48.19	60.36	6.56	21.85	31.95	49.22	60.19
+our $\mathcal{L}^*$	**73.49**	**91.97**	**95.77**	**98.79**	**99.60**	**70.38**	**91.54**	**95.77**	**98.10**	**99.35**	**7.69**	**23.49**	**33.42**	**50.52**	**63.04**	**6.99**	**23.66**	**35.06**	**52.16**	**63.21**
Δ	↑ 4.58	↑ 1.12	↑ 0.61	0.00	↑ 0.03	↑ 3.63	↑ 1.30	↑ 1.81	0.00	↑ 0.04	↑ 1.32	↑ 2.42	↑ 1.55	↑ 2.33	↑ 2.68	↑ 0.43	↑ 1.81	↑ 3.11	↑ 2.94	↑ 3.02
LoRA	57.08	84.72	91.80	96.46	98.70	56.74	85.32	91.71	96.29	98.36	5.79	16.84	25.82	42.49	56.22	4.84	16.49	24.96	40.33	56.30
+our $\mathcal{L}^*$	**62.09**	**86.79**	**93.95**	**97.32**	**99.05**	**59.41**	**85.92**	**92.75**	**97.24**	**98.97**	**5.87**	**18.31**	**27.63**	**44.99**	**59.67**	**5.32**	**18.65**	**28.41**	**44.82**	**56.99**
Δ	↑ 5.01	↑ 2.07	↑ 2.15	↑ 0.86	↑ 0.35	↑ 2.67	↑ 0.60	↑ 1.04	↑ 0.95	↑ 0.61	↑ 0.08	↑ 1.47	↑ 1.81	↑ 2.50	↑ 3.45	↑ 0.48	↑ 2.16	↑ 3.45	↑ 4.49	↑ 0.69
DoRA	61.77	86.18	91.88	97.32	98.88	60.94	87.33	92.31	96.46	98.46	**7.17**	20.21	29.88	46.20	61.92	6.04	20.81	31.52	46.29	60.02
+our $\mathcal{L}^*$	**65.20**	**90.26**	**94.91**	**98.01**	**99.31**	**64.94**	**88.35**	**93.52**	**97.58**	**98.96**	7.08	**24.01**	**35.06**	**52.59**	**67.01**	**8.29**	**24.35**	**35.15**	**52.76**	**66.58**
Δ	↑ 3.43	↑ 4.08	↑ 3.03	↑ 0.69	↑ 0.43	↑ 4.00	↑ 1.02	↑ 1.21	↑ 1.12	↑ 0.50	↓ 0.09	↑ 3.80	↑ 5.18	↑ 6.39	↑ 5.09	↑ 2.25	↑ 3.54	↑ 3.63	↑ 6.47	↑ 6.56

Method	DOCCI										DCI									
	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50
Long-CLIP	64.49	87.67	93.43	97.73	99.14	63.08	87.45	93.14	97.45	99.02	59.23	80.89	87.04	92.60	95.10	60.13	81.44	87.54	92.85	95.60
+our $\mathcal{L}^*$	**67.67**	**90.82**	**95.59**	**98.43**	**99.39**	**67.92**	**90.16**	**95.10**	**98.22**	**99.47**	**63.13**	**84.14**	**89.69**	**94.20**	**96.55**	**64.33**	**86.19**	**89.14**	**94.20**	**96.50**
Δ	↑ 3.18	↑ 3.15	↑ 2.16	↑ 0.70	↑ 0.25	↑ 4.84	↑ 2.71	↑ 1.96	↑ 0.77	↑ 0.45	↑ 3.90	↑ 3.25	↑ 2.65	↑ 1.60	↑ 1.45	↑ 4.20	↑ 4.75	↑ 1.60	↑ 1.35	↑ 0.90
FineLIP	67.80	90.22	94.84	98.22	99.45	66.39	89.12	94.47	97.90	99.20	66.13	85.34	89.79	94.14	96.35	64.58	84.59	89.54	94.00	96.40
+our $\mathcal{L}^*$	**74.06**	**94.24**	**97.35**	**99.23**	**99.80**	**72.94**	**93.27**	**96.55**	**98.78**	**99.61**	**68.88**	**86.64**	**91.10**	**95.24**	**96.90**	**67.33**	**86.69**	**90.75**	**94.85**	**97.00**
Δ	↑ 6.26	↑ 4.02	↑ 2.51	↑ 1.01	↑ 0.35	↑ 6.55	↑ 4.15	↑ 2.08	↑ 0.88	↑ 0.41	↑ 2.75	↑ 1.30	↑ 1.31	↑ 1.10	↑ 0.55	↑ 2.75	↑ 2.10	↑ 1.21	↑ 0.85	↑ 0.60
SmartCLIP	74.92	94.08	97.31	99.37	99.82	74.91	94.04	97.29	99.32	99.84	69.88	86.64	94.05	95.00	97.25	70.94	87.04	92.77	95.75	97.05
+our $\mathcal{L}^*$	**77.39**	**95.57**	**98.66**	**99.47**	**99.94**	**77.10**	**95.49**	**98.34**	**99.86**	**99.89**	**69.93**	**86.94**	**94.35**	**95.10**	**97.30**	**71.14**	**87.64**	**94.10**	**95.80**	**98.60**
Δ	↑ 2.47	↑ 1.49	↑ 1.35	↑ 0.10	↑ 0.12	↑ 2.19	↑ 1.45	↑ 1.05	↑ 0.54	↑ 0.05	↑ 0.05	↑ 0.30	↑ 0.30	↑ 0.10	↑ 0.05	↑ 0.20	↑ 0.60	↑ 1.33	↑ 0.05	↑ 0.55
GOAL	79.47	96.65	98.69	99.69	**99.92**	79.43	96.14	97.25	99.61	99.90	72.64	**89.89**	93.70	95.75	97.25	72.84	**90.50**	93.20	96.60	97.60
+our $\mathcal{L}^*$	**80.96**	**96.90**	**98.96**	**99.76**	**99.92**	**80.31**	**96.73**	**98.84**	**99.75**	**99.94**	**72.89**	89.79	**94.40**	**96.15**	**97.65**	**73.89**	89.93	**93.50**	**96.80**	**98.15**
Δ	↑ 1.49	↑ 0.25	↑ 0.27	↑ 0.07	0.00	↑ 0.88	↑ 0.59	↑ 0.33	↑ 0.14	↑ 0.04	↑ 0.25	↓ 0.10	↑ 0.7	↑ 0.20	↑ 0.40	↑ 1.05	↓ 0.57	↑ 0.30	↑ 0.20	↑ 0.25
SigLIP2	71.80	92.53	95.88	98.78	99.43	71.51	92.41	96.06	98.51	99.39	66.13	84.49	89.34	94.00	96.40	65.08	84.54	89.84	94.65	**96.95**
+our $\mathcal{L}^*$	**75.47**	**94.82**	**97.67**	**99.30**	**99.78**	**73.59**	**94.33**	**97.43**	**99.37**	**99.84**	**67.14**	**86.54**	**90.65**	**94.85**	**96.85**	**66.78**	**86.09**	**90.50**	**94.75**	96.90
Δ	↑ 3.67	↑ 2.29	↑ 1.79	↑ 0.52	↑ 0.35	↑ 2.08	↑ 1.92	↑ 1.37	↑ 0.86	↑ 0.45	↑ 1.01	↑ 2.05	↑ 1.31	↑ 0.85	↑ 0.45	↑ 1.7	↑ 1.55	↑ 0.66	↑ 0.10	↓ 0.05
LoRA	77.80	96.45	98.55	99.11	99.14	77.00	96.02	98.05	99.32	99.14	72.04	88.69	92.35	95.50	97.15	72.04	**89.24**	**93.20**	95.00	97.40
+our $\mathcal{L}^*$	**79.35**	**96.61**	**98.57**	**99.61**	**99.40**	**78.65**	**96.31**	**98.45**	**99.63**	**99.86**	**74.44**	**89.44**	**92.85**	**95.95**	**97.20**	**73.89**	**89.24**	92.60	**96.40**	**97.70**
Δ	↑ 1.55	↑ 0.16	↑ 0.02	↑ 0.50	↑ 0.26	↑ 1.65	↑ 0.29	↑ 0.40	↑ 0.31	↑ 0.72	↑ 2.40	↑ 0.75	↑ 0.50	↑ 0.45	↑ 0.05	↑ 1.85	0.00	↓ 0.60	↑ 1.40	↑ 0.30
DoRA	66.27	90.50	95.18	98.49	99.67	65.65	89.55	94.78	98.41	99.36	60.33	82.19	88.09	93.45	95.75	60.53	81.84	87.79	93.40	95.30
+our $\mathcal{L}^*$	**70.65**	**92.47**	**96.57**	**99.02**	**99.73**	**69.22**	**91.80**	**95.98**	**98.98**	**99.61**	**62.58**	**82.34**	**88.34**	**93.75**	**96.10**	**62.78**	**82.99**	**88.39**	**94.25**	**96.75**
Δ	↑ 4.38	↑ 1.97	↑ 1.39	↑ 0.53	↑ 0.06	↑ 3.57	↑ 2.25	↑ 1.20	↑ 0.57	↑ 0.25	↑ 2.25	↑ 0.15	↑ 0.25	↑ 0.30	↑ 0.35	↑ 2.25	↑ 1.15	↑ 0.60	↑ 0.85	↑ 1.45

Table 5. **Data Efficiency Analysis.** We report mean Recall@K (%) on SKETCHY, INSECT, DOCCI, and DCI using **5%**, **20%**, **50%** respectively of the training data. Results are reported for K = 1, 5, 10, 25, 50 on both *Text→Image* and *Image→Text*. **Bold** indicates the best result, while underline denotes the second-best.

Setting	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50	R@1	R@5	R@10	R@25	R@50
5% Data	**SKETCHY**										**INSECT**									
Long-CLIP[ECCV'24]	21.42	47.58	59.59	76.77	86.18	23.40	49.14	62.44	78.24	87.31	2.50	6.99	10.88	18.83	27.98	2.68	8.03	12.61	20.55	30.66
FineLIP[CVPR'25]	30.92	60.28	71.68	85.49	91.88	31.00	59.50	70.12	84.97	92.57	2.50	8.12	11.87	22.13	32.99	3.28	8.33	13.04	22.31	32.06
SmartCLIP[CVPR'25]	25.22	53.02	77.12	81.35	90.59	27.98	54.15	75.76	82.90	91.97	2.68	7.43	11.66	19.86	30.14	3.37	8.44	13.15	21.66	32.25
GOAL[CVPR'25]	33.42	63.99	74.18	86.23	92.49	35.05	64.08	75.82	88.69	93.53	2.68	7.51	11.83	22.11	31.35	3.34	8.20	12.33	21.24	31.78
StructXLIP	**36.70**	**68.05**	**78.07**	**88.43**	**94.39**	**35.75**	**68.05**	**77.46**	**89.12**	**94.91**	**3.33**	**8.46**	**12.87**	**22.28**	**33.68**	**3.57**	**8.98**	**13.56**	**22.54**	**33.16**
Δ	↑ 3.28	↑ 4.06	↑ 0.95	↑ 2.20	↑ 1.90	↑ 0.70	↑ 3.97	↑ 1.64	↑ 0.43	↑ 1.38	↑ 0.65	↑ 0.34	↑ 1.00	↑ 0.15	↑ 0.69	↑ 0.20	↑ 0.54	↑ 0.41	↑ 0.23	↑ 0.91
5% Data	**DOCCI**										**DCI**									
Long-CLIP[ECCV'24]	61.20	86.33	92.61	97.43	99.08	61.57	86.51	92.51	97.16	98.98	53.73	75.34	82.24	88.79	92.25	53.98	76.69	83.74	90.05	93.85
FineLIP[CVPR'25]	66.00	89.73	94.75	97.92	99.15	63.65	87.39	92.20	97.06	98.20	58.08	79.74	85.79	91.45	**95.15**	60.88	80.29	86.64	93.35	95.80
SmartCLIP[CVPR'25]	65.12	89.33	94.37	98.25	99.31	65.71	89.76	95.20	98.27	99.00	57.78	78.14	87.33	90.60	93.78	56.13	78.84	88.14	91.05	95.30
GOAL[CVPR'25]	66.45	91.29	95.00	98.17	99.57	67.85	91.47	95.44	98.20	99.00	60.88	81.54	87.59	92.25	94.90	63.13	83.14	88.94	93.55	96.15
StructXLIP	**68.92**	**91.84**	**95.84**	**98.49**	**99.61**	**69.76**	**91.61**	**95.76**	**98.43**	**99.49**	**62.53**	**82.99**	**88.20**	**93.15**	95.10	**64.03**	**84.34**	**89.19**	**93.80**	**96.60**
Δ	↑ 2.47	↑ 0.55	↑ 0.84	↑ 0.24	↑ 0.04	↑ 1.91	↑ 0.14	↑ 0.32	↑ 0.16	↑ 0.49	↑ 1.65	↑ 1.45	↑ 0.61	↑ 0.90	↓ 0.05	↑ 0.9	↑ 1.20	↑ 0.25	↑ 0.25	↑ 0.45
20% Data	**SKETCHY**										**INSECT**									
Long-CLIP[ECCV'24]	35.75	64.85	75.73	85.92	93.96	36.01	65.72	76.86	88.08	94.39	4.05	11.05	18.74	30.92	41.54	4.15	**14.68**	20.29	32.82	43.70
FineLIP[CVPR'25]	38.17	69.17	78.84	89.81	95.51	35.75	67.18	78.41	88.32	94.99	3.63	12.44	19.08	30.31	43.09	4.23	13.64	20.81	31.69	43.52
SmartCLIP[CVPR'25]	38.00	68.48	88.08	90.85	97.24	38.95	69.86	88.43	90.85	96.29	3.45	9.54	15.28	28.14	40.39	3.20	12.15	19.39	29.74	40.01
GOAL[CVPR'25]	46.63	77.37	85.06	92.45	96.90	46.46	76.94	86.10	92.66	96.43	3.21	12.80	18.83	33.51	45.37	4.33	13.99	21.16	33.15	46.06
StructXLIP	**53.20**	**80.22**	**88.26**	**94.39**	**97.50**	**49.48**	**79.45**	**88.51**	**94.73**	**97.15**	**4.58**	**13.30**	**19.60**	**34.11**	**47.58**	**4.92**	14.16	**22.12**	**34.22**	**46.72**
Δ	↑ 6.57	↑ 2.85	↑ 0.18	↑ 1.94	↑ 0.26	↑ 3.02	↑ 2.51	↑ 0.08	↑ 2.07	↑ 0.72	↑ 0.53	↑ 0.50	↑ 0.52	↑ 0.60	↑ 2.21	↑ 0.59	↓ 0.52	↑ 0.96	↑ 1.07	↑ 0.66
20% Data	**DOCCI**										**DCI**									
Long-CLIP[ECCV'24]	62.16	86.75	92.61	97.14	99.00	61.61	86.67	92.00	97.37	99.02	56.53	79.64	85.54	91.05	94.00	57.33	79.14	85.84	91.00	94.40
FineLIP[CVPR'25]	67.18	89.92	94.00	98.10	99.22	63.96	87.80	92.40	97.41	98.37	63.98	83.79	88.69	93.05	95.85	62.83	83.79	88.69	93.05	95.85
SmartCLIP[CVPR'25]	70.00	91.69	95.31	98.80	99.41	72.12	92.64	96.24	99.00	99.38	64.78	82.84	89.00	92.89	95.97	62.13	83.09	89.44	93.45	96.15
GOAL[CVPR'25]	71.43	93.12	96.90	99.18	99.75	72.94	93.63	96.84	99.00	99.67	64.63	84.84	90.05	93.90	96.20	65.73	85.29	90.65	94.65	97.00
StructXLIP	**77.18**	**95.61**	**97.82**	**99.47**	**99.82**	**76.47**	**95.31**	**97.86**	**99.39**	**99.82**	**67.03**	**86.24**	**90.60**	**94.59**	**96.50**	**69.03**	**87.34**	**91.35**	**95.45**	**97.50**
Δ	↑ 5.75	↑ 2.49	↑ 0.92	↑ 0.29	↑ 0.07	↑ 3.53	↑ 1.68	↑ 1.02	↑ 0.39	↑ 0.15	↑ 2.25	↑ 1.40	↑ 0.55	↑ 0.69	↑ 0.30	↑ 3.30	↑ 2.05	↑ 0.70	↑ 0.80	↑ 0.50
50% Data	**SKETCHY**										**INSECT**									
Long-CLIP[ECCV'24]	47.93	74.01	82.99	92.23	96.29	43.96	73.58	82.56	91.88	96.03	5.18	15.80	25.47	40.76	53.28	4.75	18.13	27.63	40.59	53.11
FineLIP[CVPR'25]	39.98	69.44	79.97	90.15	95.60	37.65	68.91	79.10	88.95	94.99	5.61	17.10	24.96	41.11	54.75	5.01	17.44	26.34	41.02	54.06
SmartCLIP[CVPR'25]	45.34	76.34	90.93	93.01	97.25	46.46	74.70	90.16	92.06	97.58	3.57	11.92	19.74	33.31	43.71	3.63	13.31	21.67	32.97	42.06
GOAL[CVPR'25]	55.79	82.73	88.35	96.20	98.10	55.27	82.04	89.12	95.42	98.36	5.87	18.13	25.51	41.02	55.22	5.35	18.77	26.84	43.05	56.55
StructXLIP	**60.97**	**87.48**	**92.49**	**96.89**	**98.70**	**59.41**	**85.58**	**91.80**	**95.55**	**98.70**	**6.56**	**20.12**	**29.17**	**43.78**	**57.77**	**6.22**	**20.64**	**29.84**	**44.13**	**57.94**
Δ	↑ 5.18	↑ 4.75	↑ 1.56	↑ 0.69	↑ 0.60	↑ 4.14	↑ 3.54	↑ 1.64	↑ 0.13	↑ 0.34	↑ 0.69	↑ 1.99	↑ 3.66	↑ 2.67	↑ 2.55	↑ 0.87	↑ 1.87	↑ 2.21	↑ 1.08	↑ 1.39
50% Data	**DOCCI**										**DCI**									
Long-CLIP[ECCV'24]	63.49	87.49	93.20	97.33	99.06	62.20	87.41	92.90	97.40	99.02	58.63	80.79	85.99	91.65	94.65	59.04	80.89	85.69	92.20	95.35
FineLIP[CVPR'25]	67.20	90.06	94.75	98.22	99.31	64.29	87.94	93.10	97.72	98.86	65.43	84.44	88.10	94.00	96.05	63.83	84.44	89.19	93.14	95.00
SmartCLIP[CVPR'25]	73.75	94.24	97.41	99.31	99.77	74.00	94.04	97.17	98.84	99.65	66.98	84.79	90.77	93.40	96.00	66.20	84.84	91.15	94.65	97.12
GOAL[CVPR'25]	75.25	94.61	97.59	99.51	**99.88**	75.73	94.53	97.37	99.45	**99.80**	69.03	86.84	91.10	94.95	96.50	67.63	86.69	91.20	95.65	97.05
StructXLIP	**79.78**	**96.37**	**98.31**	**99.61**	**99.88**	**78.45**	**95.98**	**98.43**	**99.57**	**99.80**	**71.19**	**87.84**	**91.30**	**95.55**	**97.05**	**71.64**	**87.39**	**92.65**	**96.90**	**97.50**
Δ	↑ 4.53	↑ 1.76	↑ 0.72	↑ 0.10	0.00	↑ 2.72	↑ 1.45	↑ 1.06	↑ 0.12	0.00	↑ 2.16	↑ 1.00	↑ 0.20	↑ 0.60	↑ 0.55	↑ 4.01	↑ 0.70	↑ 1.45	↑ 1.25	↑ 0.38

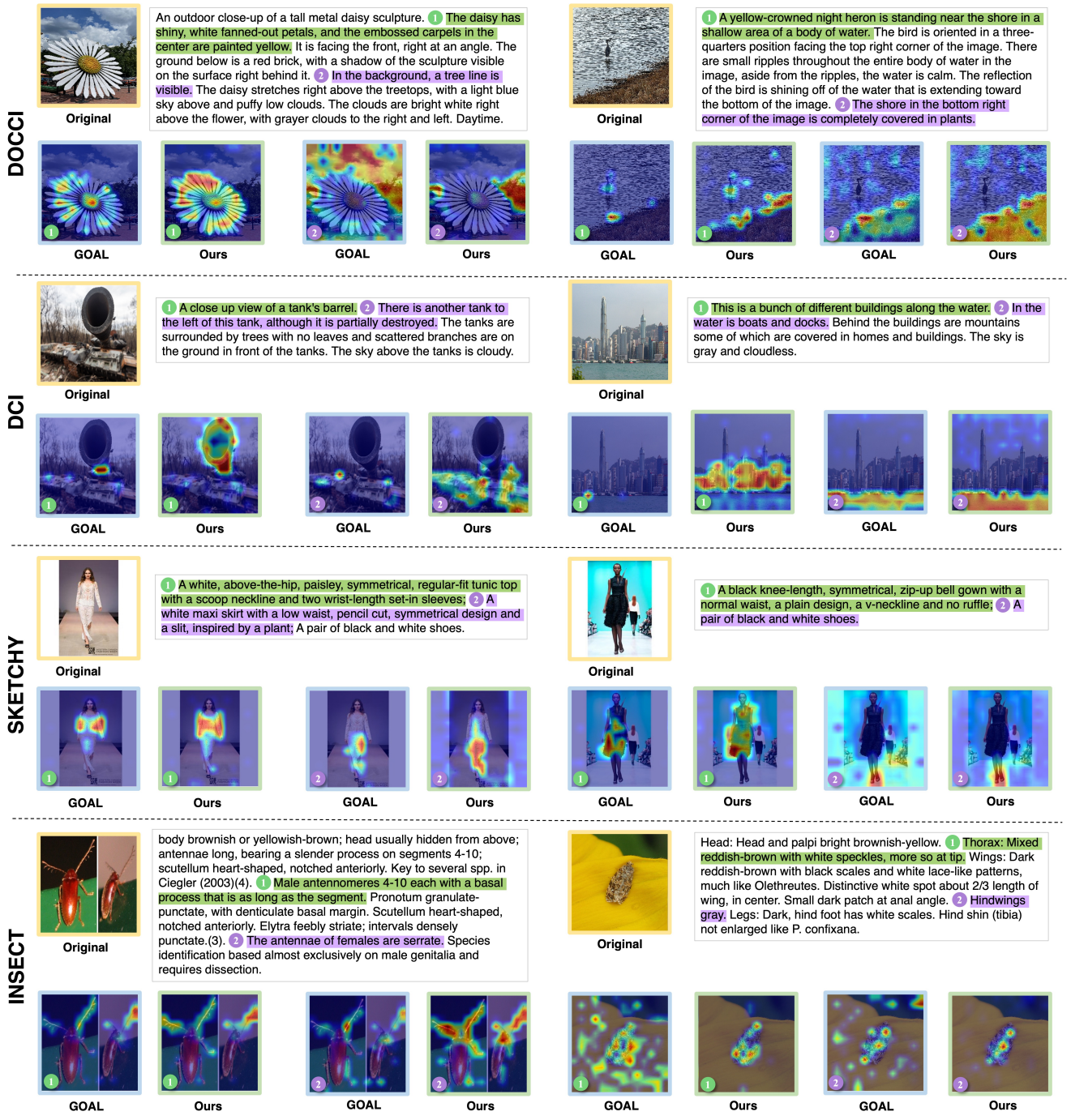
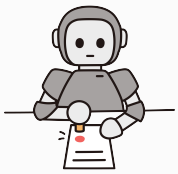


Figure 9. Qualitative results of StructXLIP and the second-best method GOAL on all four datasets. We present the attention maps between the color-highlighted texts on the visual counterpart. Compared to GOAL, StructXLIP overall shows a better correspondence between the rich textual descriptions and the visual objects.

(2): Domain-Specific Appearance Lexicon Prompt–Response

USER: You are a vision-language expert specializing in ***fashion*** imagery. Consider a representation of a garment where only edge information is preserved (i.e., contours, stitching lines, panel boundaries, silhouettes, and geometric layout), with all color, shading, and material cues removed. Please provide a comprehensive list of English words and short phrases used in the fashion domain whose meanings cannot be inferred from such an edge-only representation. Return the list as comma-separated items.

LLM: [red, blue, green, yellow, orange, purple, pink, brown, black, white, grey, gray, navy, sky blue, baby blue, royal blue, midnight blue, teal, turquoise, aqua, mint, forest green, lime, olive, khaki, sage, emerald, moss, maroon, burgundy, wine, cranberry, magenta, fuchsia, rose, blush, salmon, coral, peach, apricot, mustard, gold, lemon, amber, beige, cream, ivory, off-white, tan, camel, sand, taupe, stone, ecru, charcoal, slate, ash, lavender, violet, lilac, plum, eggplant, silver, bronze, copper, champagne, metallic, pastel, neon, bright, dark, light, neutral, multicolor, multi-color, multicolored, multi-colored, cotton, linen, hemp, ramie, jute, wool, merino, cashmere, mohair, alpaca, angora, silk, mulberry silk, tussah silk, denim, chambray, twill, gabardine, canvas, jersey, interlock, french terry, flannel, fleece, felt, velvet, velour, corduroy, tweed, boucle, jacquard, brocade, damask, viscose, rayon, modal, lyocell, tencel, acetate, cupro, polyester, nylon, spandex, elastane, lycra, acrylic, polyamide, microfiber, chiffon, georgette, organza, tulle, mesh, crepe, charmeuse, satin, taffeta, batiste, voile, lawn, leather, suede, nubuck, shearling, sheepskin, faux leather, vegan leather, faux suede, fur, faux fur, mink, fox fur, rabbit fur, knit, rib knit, cable knit, fine knit, softshell, hardshell, neoprene, Gore-Tex, ripstop, mesh, spacer fabric, lamé, bonded fabric, coated fabric, sherpa, denier, pongee, jacquard knit]



USER: You are an ***entomology***-focused vision–language expert. Consider an ***insect*** image represented only by its edge map, that is, a structure-centric depiction containing contours, boundaries, wing venation geometry, and overall body segmentation, but with all color, shading, pigmentation, and optical surface cues removed. Please produce a comprehensive list of words and short phrases used in insect field guides or taxonomic descriptions whose meanings cannot be inferred from such an edge-only representation. These should strictly correspond to appearance attributes such as color terms, pigmentation qualities, optical reflectance (e.g., iridescence, metallic sheen), translucency/opacity, and other non-geometric visual properties. Return the list as comma-separated items.

LLM: [black, brown, dark brown, light brown, tan, beige, cream, white, off-white, gray, grey, charcoal, slate, ash, red, reddish, orange, yellow, green, blue, purple, pink, magenta, violet, rust, russet, chestnut, mahogany, clay, ochre, umber, sienna, tawny, fawn, amber, honey-colored, golden, bronze, coppery, pale, dusky, washed-out, faded, dim, drab, rufous, testaceous, fulvous, ferruginous, castaneous, fuscous, livid, piceous, violaceous, cyaneous, glaucous, mottled, blotchy, flecked, freckled, stained, tinged, tinted, smudged, clouded, diffuse, suffused, irregularly pigmented, unevenly pigmented, faintly pigmented, deeply pigmented, melanized, depigmented, discolored, frosted, pruinose, powdery, mealy, chalky, dusty, granular, velvety, shaded, darkened, lightened, somber, sooty, smoky, smeared, iridescent, metallic, submetallic, opalescent, pearlescent, rainbowlike, prismatic, lustrous, holographic, shiny, glossy, subglossy, dull, matte, satiny, silky, polished, reflective, non-reflective, sheeny, mirrorlike, transparent, translucent, semi-translucent, opaque, hyaline, subhyaline, smoky-hyaline, warm-toned, cool-toned, earthy, vivid, dull-colored, bright, pale-colored, dark-colored, satiny, silken, glassy, resinous, lacquered, oily, greasy, waxy, glistening, gleaming]

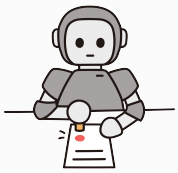


Figure 10. Prompt and response used in our LLM-based construction of the domain-specific appearance vocabulary lists for lexicon filtering.