COVERCLIP: COVERAGE-AWARE VISUAL AND TEXTUAL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive Language–Image Pre-training (CLIP) performs strongly on image–text retrieval and zero-shot classification. However, existing CLIP training objectives treat all semantically correct captions as equally valid, largely ignoring their semantic coverage. Ideally, for a given image, a caption covering more visible semantic information should receive a higher image–text similarity score. To explicitly measure this, we introduce **CoverageBench**, a manually verified benchmark containing 1,300 images derived from Visual Genome. Each image is annotated with four visual semantic atoms (e.g., object, attribute, relation, etc.) and paired with four captions with increasing semantic coverage, denoted as C_1, C_2, C_3, C_4. Models are expected to assign similarity scores satisfying $s(I, C_1) < s(I, C_2) < s(I, C_3) < s(I, C_4)$. Evaluating on CoverageBench, we find that existing CLIP models struggle to consistently distinguish correct captions according to their semantic coverage. To address this limitation, we propose **CoverCLIP**, a coverage-aware training framework built on three components. First, we construct **Coverage-1M**, which consists of 710K samples with graded semantic coverage captions, 228K samples with long-short captions, and 76K samples with highly descriptive captions, providing explicit supervision for graded semantic coverage. Second, we introduce a step-wise mixed training strategy that incorporates semantic coverage-aware supervision into standard contrastive learning. Finally, we design coverage-ranking and hard-negative separation objectives to further enhance coverage-aware supervision. Experiments demonstrate that CoverCLIP improves CoverageBench performance while remaining competitive on image–text retrieval and zero-shot classification. Interestingly, when applied to text-to-image generation by replacing the CLIP text encoder in SDXL, CoverCLIP transfers its coverage awareness to the generation process, enabling the model to progressively generate more detailed images as semantic coverage increases.

1 INTRODUCTION

Contrastive Language–Image Pre-training (CLIP) (Radford et al., 2021) has emerged as a foundational model for a wide range of downstream tasks in vision-language understanding. By learning a shared embedding space for images and text through large-scale contrastive learning, CLIP demonstrates strong performance on tasks such as image–text retrieval and zero-shot classification.

Despite these successes, CLIP and its variants exhibit a notable limitation: their training objectives treat all semantically correct captions as equally valid, without distinguishing their graded semantic coverage. Ideally, for a given image, a caption that covers more visible semantic information (e.g., describing both the main subject and its attributes, actions, and context) should receive a higher similarity score than a caption that only describes a subset of these elements. However, existing CLIP models exhibit limited sensitivity to such fine-grained differences. This limitation restricts their utility in scenarios such as controllable generation and fine-grained retrieval.

To make this problem measurable, we introduce CoverageBench, a manually verified benchmark comprising 1,300 images from the Visual Genome dataset (Krishna et al., 2017). As illustrated in Figure 1, each image is annotated with four visual semantic atoms and paired with four captions with increasing coverage levels (C_1, C_2, C_3, C_4). Models are expected to assign similarity scores satisfying $s(I, C_1) < s(I, C_2) < s(I, C_3) < s(I, C_4)$ (metric: Exact Chain), which measures

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

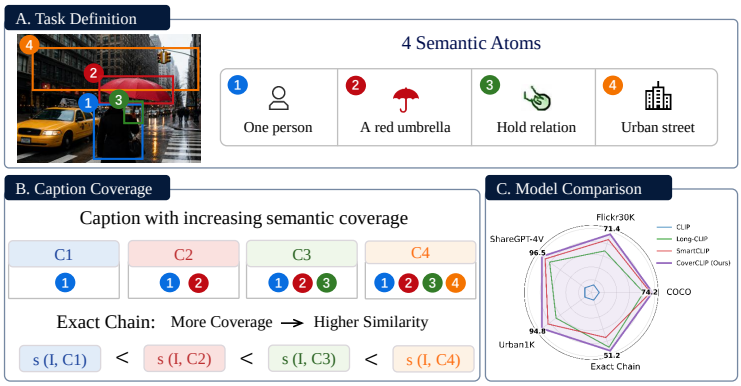

Figure 1: **Overview of CoverageBench.** Four semantic atoms form captions C_1-C_4 with increasing coverage, and Exact Chain evaluates whether the model correctly ranks their similarities to exactly match their coverage. Panel C summarizes ViT-L/14 results; COCO and Flickr30K report the average over I2T/T2I R@1/5/10, while ShareGPT-4V and Urban1K report the average I2T/T2I R@1.

whether the model correctly ranks all four captions in order of increasing coverage. Evaluating on CoverageBench, we find that existing CLIP models exhibit limited ability to consistently rank correct captions by semantic coverage. For example, the original OpenAI CLIP ViT-L/14 achieves only 29.7% accuracy on the Exact Chain.

To address this, we propose CoverCLIP, a coverage-aware training framework built on three components. First, we construct Coverage-1M by deduplicating and re-annotating the FineHARD-12M (Xie et al., 2025b) dataset, retaining 938K FineHARD images. Of these, 710K coverage samples are re-annotated with graded coverage captions, while the remaining 228K general samples retain their original long-short caption pairs. Together with 76K highly descriptive samples from ShareGPT-4V (Chen et al., 2024), these form the Coverage-1M dataset. Second, we design a step-wise mixed training strategy that incorporates semantic coverage-aware supervision into standard contrastive learning. Finally, we design coverage-ranking and hard-negative separation objectives to further enhance coverage-aware supervision. Together, these components improve CoverCLIP's sensitivity to graded coverage while preserving its general capability.

Our contributions can be summarized as follows:

- **CoverageBench**: We introduce a manually verified benchmark of 1,300 images, designed to evaluate models' sensitivity to graded semantic coverage.
- **Coverage-1M**: We construct a large-scale training dataset containing graded coverage captions, general long-short captions, and highly descriptive captions, providing explicit supervision for coverage-aware learning.
- **CoverCLIP**: We propose a novel step-wise mixed training framework that integrates semantic coverage-aware supervision with standard contrastive learning, and further introduces coverage-ranking and hard-negative separation objectives.
- **Empirical results**: CoverCLIP improves performance on CoverageBench while remaining competitive on standard image–text retrieval and zero-shot classification tasks. Furthermore, replacing the CLIP text encoder in SDXL (Podell et al., 2024) with CoverCLIP enables the model to progressively generate more detailed images as additional semantic content is introduced.

2 Related Work

CLIP Models. CLIP (Radford et al., 2021) pioneered the paradigm of learning visual representations from natural language supervision. A line of subsequent works has focused on scaling up training data and improving training efficiency. ALIGN (Jia et al., 2021) scales to 1.8 billion image-text pairs, while SigLIP (Zhai et al., 2023) replaces the softmax-based loss with a sigmoid loss for

better efficiency. Another research thread aims to improve CLIP's ability to handle long and detailed captions (Zhang et al., 2024; Huang et al., 2026). Several works also focus on improving compositional and fine-grained reasoning (Patel et al., 2024; Xie et al., 2025b;c). However, to the best of our knowledge, prior work has not explicitly addressed graded semantic coverage, a setting in which models must distinguish captions according to the amount of visual information they cover.

CLIP Benchmarks. Standard CLIP benchmarks cover a range of tasks. MS-COCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015) are widely used for image–text retrieval, while ImageNet (Russakovsky et al., 2015) serves as a standard benchmark for zero-shot classification. Recent benchmarks such as ARO (Yuksekgonul et al., 2022), Winoground (Thrush et al., 2022), CREPE (Ma et al., 2023), and SugarCrepe (Hsieh et al., 2023) further evaluate compositional and fine-grained image–text understanding through controlled semantic variations. However, these benchmarks mainly focus on distinguishing correct captions from semantically altered or corrupted alternatives. In contrast, CoverageBench evaluates whether models can consistently rank multiple correct captions for the same image according to their graded semantic coverage.

CLIP Training Data. CLIP training data has evolved from early web-crawled datasets (WIT, LAION-5B, COYO-700M) (Radford et al., 2021; Schuhmann et al., 2022; Byeon et al., 2022) to filtered datasets like DataComp-1B (Gadre et al., 2023) and DFN-5B (Fang et al., 2024). Recent works also provide high-quality data: ShareGPT-4V (Chen et al., 2024) provides highly descriptive captions generated with the assistance of large multimodal models, while the FineHARD-12M (Xie et al., 2025b) is specifically designed for fine-grained vision-language understanding. Our Coverage-1M serves as a dedicated training dataset for mixed-objective coverage-aware supervision, enabling models to distinguish captions based on their graded semantic coverage.

3 COVERAGEBENCH

To systematically evaluate models' sensitivity to graded semantic coverage, we construct CoverageBench through a three-stage pipeline: data filtering, vision language model (VLM) based verification and construction, and automated validation with human review.

3.1 STAGE 1: DATA FILTERING

We start from Visual Genome (Krishna et al., 2017) and apply a set of strict filtering criteria to ensure image quality and semantic richness. First, we only consider objects whose bounding box area exceeds 15% of the image area as candidates. Second, an image must contain at least 4 valid objects, 2 objects with attributes, 1 relationship, and 3 region descriptions to proceed in the pipeline. Third, we remove adjectives, colors, relational terms, and other non-entity tokens from candidate object labels, while retaining them separately as attribute or relation annotations. Finally, candidate objects are normalized into standardized `object_categories` through singular/plural normalization (e.g., "dogs" $\rightarrow$ "dog") and synonym merging (e.g., "automobile" $\rightarrow$ "car") to reduce lexical variance and ensure consistent category representation. Then, to quantify the importance of each candidate object, we compute a salience score for each object as follows:

$$\begin{aligned}\text{salience} = {} & 0.55 \times \text{area_score} + 0.20 \times \text{norm_mention_score} \\ & + 0.15 \times \text{norm_relation_score} + 0.10 \times \text{norm_attribute_score}\end{aligned} \tag{1}$$

where `area_score` is the bounding box area ratio, `mention_score` is the frequency of the object's canonical name in region descriptions, `relation_score` is the number of relationships involving the object, and `attribute_score` is the number of associated attributes. The latter three are normalized by their maximum among all candidate objects in the same image. All bounding-box-qualified objects are sorted in descending order of salience, and the top-10 are retained. To select images with broader semantic coverage, we require the 10 retained objects to collectively span at least 6 of the following 7 coarse semantic groups: food, person, animal, vehicle, nature, building_scene, furniture_object. This yields 2,760 candidate images.

3.2 Stage 2: Verification and Construction

However, manual inspection revealed that rule-based filtering alone was insufficient to ensure entity validity. To further improve data quality, the 2,760 candidate images and their selected top-10 objects are verified by `Gemma4-31B` (Team et al., 2026) for entity validity. The VLM checks whether objects actually appear in the image, whether they are real entities, removes non-entity words (e.g., "is", "the", "a"), color/age/preposition/relational words, and retains only specific categories when both generic and specific categories appear (e.g., "pets + dog + cat" → "dog + cat"). The detailed prompt is in Appendix E. After Gemma verification, images with `true_object_count > 5` are retained, resulting in 1,370 images.

For each of the 1,370 images, we feed the image along with its raw VG annotations (objects, attributes, relationships, region descriptions) into GPT-5.6 Sol, which selects four semantic atoms (a semantic atom is a manually verifiable visual proposition represented as an entity, an entity-attribute pair, a subject-relation-object triplet, or a scene-level fact) and constructs four correct captions with increasing coverage levels (C_1, C_2, C_3, C_4) covering 1, 2, 3, and 4 atoms respectively. Motivated by Long-CLIP (Zhang et al., 2024), which shows that the original CLIP has limited effective text length, we constrain all captions to 16–20 English words to keep them concise and reduce length-based shortcuts. The detailed prompt is in Appendix E.

3.3 Stage 3: Automated Validation and Human Review

GPT-generated outputs are automatically validated through a combination of rule-based checks and VLM-based verification. Rule-based checks ensure format correctness: exactly 4 semantic atoms, non-empty text with 16–20 words, correct caption IDs (C_1, C_2, C_3, C_4). A separate VLM, Gemma4-31B, the same model used for entity verification, is then used to verify whether each C_k caption precisely covers k scored atoms without introducing extra visual information. Out of 1,370 input images, 1,357 samples pass automated validation.

All 1,357 automatically validated samples are independently reviewed by all four PhD students specializing in VLM research, with each sample assessed by every annotator. Each annotator independently verifies: consistency between the image and semantic atoms, precise coverage of scored atoms by each caption, absence of explanatory text or metadata, and freedom from hallucination or image-sample misalignment. Only samples receiving unanimous approval from all four annotators are retained; any sample with at least one dissenting vote is excluded. After this rigorous filtering process, 1,300 samples are finalized as CoverageBench (95.8% retention rate). The detailed data distribution of the final dataset is presented in Appendix A.

4 Method: Coverage-Aware CLIP

In this section, we first describe the construction process of Coverage-1M in detail, followed by the corresponding step-wise mixed training strategy. Our goal is to fine-tune the CLIP model with a relatively small amount of data, enabling it to improve sensitivity to graded semantic coverage while preserving its original capability.

4.1 Coverage-1M Training Dataset

To support training, we re-annotate the existing FineHARD-12M dataset, which contains detailed descriptions for multiple bounding boxes within each image, naturally aligning with our need for semantic atoms. Due to broken data links, we downloaded 7.8M images from the original source. For deduplication, we encode images using FG-CLIP2 (Xie et al., 2025a) into 768-dimensional FP16 embeddings and perform FAISS-based similarity search. We experimented with multiple thresholds and manually selected 0.80 after inspection, retaining 3.4M unique images.

From the deduplicated pool, we retain images with at least four bounding boxes, yielding 1.2M candidates. We then employ Gemma4-31B in a two-stage annotation pipeline. In the first stage, because different bounding-box descriptions may contain overlapping semantics, the corresponding visual expressions are decomposed into atomic facts and semantically deduplicated. The retained semantic contents are organized into four nested sets $A_1 \subset A_2 \subset A_3 \subset A_4$, where A_k denotes the seman-

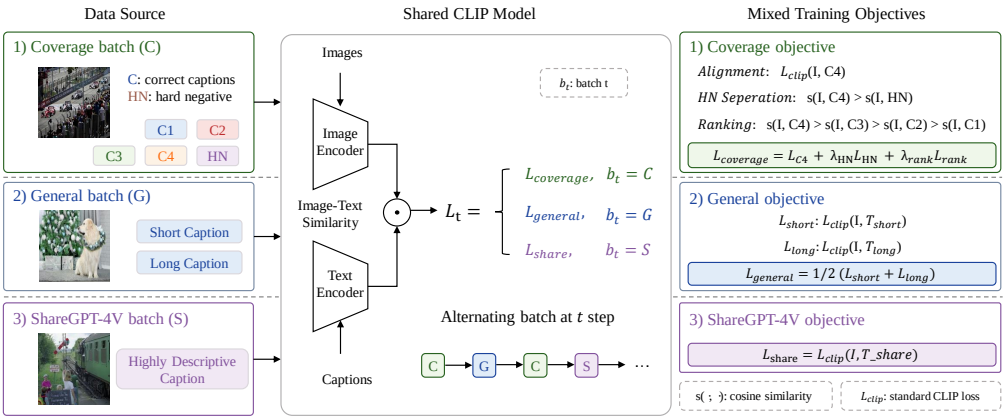

Figure 2: Overview of our step-wise mixed training strategy. Three types of batches—coverage, general, and ShareGPT-4V—are alternated during training, each with its own optimization objective, while jointly updating a single shared CLIP model.

tic content at coverage level k. After semantic deduplication and four-level coverage construction, 938K images are retained. The 938K images are then randomly partitioned into 710K coverage samples and 228K general samples. For the coverage subset, the hierarchical atom sets and the original caption information are provided to Gemma4-31B in the second stage to generate four correct captions $C_1, \ldots, C_4$, where C_k verbalizes the atoms in A_k. We additionally construct a hard negative HN, following prior hard-negative augmentation strategies (Yuksekgonul et al., 2022), which closely resembles C_4 but violates one key visual fact. The HN serves as an auxiliary training signal to discourage high similarity for semantically rich but visually incorrect captions. The 228K general samples retain their original long-short caption pairs. Together with 76K ShareGPT-4V samples containing highly descriptive captions, these data constitute the Coverage-1M dataset. Annotation is accelerated using vLLM (Kwon et al., 2023) across 48 H20 GPUs and is completed in approximately 7 hours. The prompts used in the two annotation stages are provided in Appendix E.

4.2 Training Method

To incorporate semantic coverage-aware supervision into standard contrastive learning, we design a step-wise mixed training strategy with three types of data batches, each optimized with its corresponding objective while sharing the same CLIP model.

Alternating Data Batches. As shown in Figure 2, we employ three distinct types of batches during training. To maintain a precise data distribution ratio, the total 1M training data is partitioned based on the global batch size and the corresponding number of optimization steps per epoch. For instance, with a global batch size of 768, one epoch comprises 1,322 steps, explicitly allocated as 925 steps for the coverage data, 297 steps for the general data, and 100 steps for the ShareGPT-4V data. This proportional step-wise scheduling ensures that, regardless of the specific batch size used, each epoch consistently processes approximately 710K coverage-aware samples, 228K general samples, and 76K ShareGPT-4V samples. At each optimization step, only the objective associated with the sampled batch is activated.

Mixed Training Objective. Different data batches share the same CLIP model but are optimized with their own specific objectives.

Coverage Objective. For the coverage batch, the optimization goal is to enable the model to distinguish captions with varying semantic coverage and hard negatives while preserving its original capabilities. Therefore, we optimize a coverage objective. The overall loss is:

$$\mathcal{L} = \lambda_{\mathrm{c}} \mathcal{L}_{\mathrm{c}}^{C_4} + \lambda_{\mathrm{HN}} \mathcal{L}_{\mathrm{HN}} + \lambda_{\mathrm{rank}} \mathcal{L}_{\mathrm{rank}} \tag{2}$$

It consists of three terms: alignment loss with $\lambda_{\mathrm{c}} = 1.0$ balances the contrastive alignment, HN separation loss with $\lambda_{\mathrm{HN}} = 0.1$ controls the importance of hard-negative separation, and ranking loss with $\lambda_{\mathrm{rank}} = 0.05$ adjusts the contribution of graded coverage ranking.

Alignment loss. As C_4 is the most complete caption, we compute bidirectional CLIP loss between the image and C_4 to preserve the model's original alignment capability:

$$\mathcal{L}_{\mathrm{c}}^{C_4} = \frac{1}{2}\big[\mathrm{CE}(I \to C_4) + \mathrm{CE}(C_4 \to I)\big] \tag{3}$$

where $\mathrm{CE}(\cdot \to \cdot)$ denotes the cross-entropy loss.

Ranking loss. It enforces the graded coverage ordering among correct captions, requiring higher coverage captions to receive higher scores. We investigate two ranking strategies: **all-pair ranking** which enforces ordering constraints over all 6 pairs $\mathcal{P}_{\text{all}} = \{(4,3),(4,2),(4,1),(3,2),(3,1),(2,1)\}$, and **adjacent-pair ranking** which only constrains adjacent coverage levels $\mathcal{P}_{\text{adj}} = \{(4,3),(3,2),(2,1)\}$. Standard ranking loss pushes all lower-level captions away from higher-level ones, which may excessively distort lower-level representations. For example, optimizing the pair (C_4, C_1) would push C_1 far away from C_4, even though C_1 covers partially correct semantics and should still receive a high score to avoid conflicting with the original CLIP alignment objective. To prevent this, we adopt **StopGrad-Low**, which applies stop-gradient to the similarity scalar $s(I, C_j)$, detaching the entire lower-score branch. This prevents the lower branch from updating either the image encoder or the caption embedding, allowing only the higher-score branch to contribute to the loss:

$$\mathcal{L}_{\mathrm{rank}} = \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \max\big(0, \mathtt{stopgrad}(s(I,C_j)) - s(I,C_i) + \gamma_{\mathrm{rank}}\big) \tag{4}$$

where P is either P_{all} or P_{adj}, $\mathrm{stopgrad}(\cdot)$ detaches the entire lower-score similarity branch, and $\gamma_{\mathrm{rank}} = 0.01$ is the default margin.

HN separation loss. While the coverage ranking encourages higher similarity for captions containing more correct visual semantics, it does not explicitly penalize semantically rich captions containing incorrect information. We therefore introduce an HN separation loss that enforces the complete correct caption C_4 to score higher than its minimally perturbed hard negative HN by a margin $\gamma_{\mathrm{HN}} = 0.03$:

$$\mathcal{L}_{\mathrm{HN}} = \max\left(0,\ s(I,\mathrm{HN}) - s(I,C_4) + \gamma_{\mathrm{HN}}\right). \tag{5}$$

where $s(\cdot,\cdot)$ denotes cosine similarity. This prevents the model from rewarding increased semantic content when it contains a visually incorrect fact.

General Objective. Inspired by Long-CLIP, we use general long-short caption pairs to preserve the model's basic alignment capability. The general loss is computed as:

$$\mathcal{L}_{\mathrm{gen}} = \frac{1}{2}\Big[\mathcal{L}_{\mathrm{CLIP}}(I, T_{\mathrm{short}}) + \mathcal{L}_{\mathrm{CLIP}}(I, T_{\mathrm{long}})\Big] \tag{6}$$

This dual supervision ensures the model retains its ability to align images with both concise and detailed captions. The $\mathcal{L}_{\mathrm{CLIP}}$ denotes the standard CLIP loss.

ShareGPT-4V Objective. We use ShareGPT-4V samples with highly descriptive captions to enhance the model's fine-grained understanding capability. For each image paired with its detailed caption T_{share}, we apply the standard bidirectional CLIP loss:

$$\mathcal{L}_{\text{share}} = \frac{1}{2}\big[\mathrm{CE}(I \to T_{\text{share}}) + \mathrm{CE}(T_{\text{share}} \to I)\big] \tag{7}$$

5 Experiments

5.1 Setup

Implementation Details and Baselines. We initialize all models from the original OpenAI CLIP checkpoints (ViT-B/16 and ViT-L/14). We reproduce Long-CLIP and SmartCLIP using their official codebases and the ShareGPT-4V-1M dataset, following their original training objectives and configurations. CoverCLIP is fine-tuned on Coverage-1M using the proposed step-wise mixed training strategy, with the same optimizer, learning rate, and number of training epochs as SmartCLIP. To isolate the contribution of Coverage-1M from that of coverage-aware supervision, additional *Direct Fine-tuning* baseline is adopted, which trains the same CLIP backbone on Coverage-1M using only the standard CLIP loss. Following Long-CLIP, the maximum text length is extended from 77 to 248 tokens via positional-embedding interpolation. Full training details are provided in Appendix B.

		COCO						Flickr30K						CoverageBench
		Image-to-Text			Text-to-Image			Image-to-Text			Text-to-Image			Exact Chain
	Method	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	
B/16	CLIP	51.5	76.1	84.6	32.5	57.6	68.2	43.6	68.0	77.0	25.0	45.4	55.0	26.5
	Direct Fine-tuning	59.9	82.4	89.4	41.8	67.7	77.3	55.3	78.1	85.0	36.3	59.0	68.0	42.2
	Long-CLIP	57.3	81.3	87.7	40.1	65.2	75.1	46.9	72.1	80.6	33.8	56.2	65.4	43.6
	SmartCLIP	60.9	**83.1**	**89.5**	42.1	67.8	77.4	54.8	77.9	85.0	35.6	58.1	67.2	39.2
	CoverCLIP (Ours)	**61.2**	82.6	89.2	**42.5**	**68.2**	**77.8**	**57.5**	**79.5**	**86.5**	**37.4**	**60.1**	**69.0**	**45.5**
L/14	CLIP	55.9	79.1	86.8	35.2	59.9	69.9	47.8	72.1	80.6	28.2	49.5	58.9	29.7
	Direct Fine-tuning	64.9	86.1	91.4	47.4	72.2	81.0	63.1	84.2	90.2	43.6	66.4	74.7	44.7
	Long-CLIP	63.0	84.9	91.1	46.1	70.8	79.9	55.1	78.9	86.4	41.1	64.2	72.6	49.7
	SmartCLIP	64.6	85.9	**91.8**	47.5	72.3	**81.3**	62.5	83.8	89.8	42.8	65.7	74.1	45.6
	CoverCLIP (Ours)	**65.6**	**86.4**	91.7	**47.8**	**72.7**	**81.3**	**65.0**	**85.2**	**90.9**	**44.6**	**67.1**	**75.4**	**51.2**

Table 1: Results of short-caption text-image retrieval on the MS-COCO 2017 validation set, the whole Flickr30K dataset, and exact-chain accuracy on CoverageBench. The best results are **bold**.

Evaluation. We explicitly verify that Coverage-1M and CoverageBench contain no overlapping images, preventing train–test leakage in the coverage evaluation. We then evaluate retrieval performance on the MS-COCO 2017 validation set (Lin et al., 2014) and the full Flickr30K dataset (Plummer et al., 2015). On CoverageBench, we evaluate the model's sensitivity to graded semantic coverage. Long-text retrieval is evaluated on ShareGPT-4V (Chen et al., 2024) and Urban1K (Zhang et al., 2024). Zero-shot classification is evaluated on Country211 (Radford et al., 2021), FER2013 (Goodfellow et al., 2013), FGVC-Aircraft (Maji et al., 2013), GTSRB (Stallkamp et al., 2012), ImageNet (Russakovsky et al., 2015), ImageNet-V2 (Recht et al., 2019), VOC2007 (Everingham et al., 2010), VOC2007-Multi (Everingham et al., 2010), and SUN397 (Xiao et al., 2010). Furthermore, we evaluate generation quality by replacing the CLIP text encoder in SDXL (Podell et al., 2024) with different CLIP models.

5.2 Main results

Short-Caption Retrieval and CoverageBench Results. As shown in Table 1, CoverCLIP achieves the best Exact Chain accuracy (51.2%) on CoverageBench, substantially outperforming the original CLIP with a gain of **21.5** percentage points, while also surpassing Long-CLIP and SmartCLIP. Meanwhile, CoverCLIP remains competitive in image-to-text retrieval and achieves the best text-to-image retrieval performance across both COCO and Flickr30K. These results demonstrate that our approach effectively improves sensitivity to graded semantic coverage without sacrificing retrieval performance.

Long-Caption Retrieval Results. Similarly, as shown in Table 2, CoverCLIP achieves the best performance on long-caption retrieval tasks (ShareGPT-4V validation set and Urban1K), especially on Urban1K where CoverCLIP improves text-to-image retrieval by 6.9 percentage points over SmartCLIP. Notably, among our training data, only 228K general samples and 76K ShareGPT-4V samples contain long captions, yet CoverCLIP still outperforms Long-CLIP and SmartCLIP trained on the full ShareGPT-4V-1M dataset. This demonstrates the effectiveness of our mixed training strategy in enhancing the model's long-caption retrieval ability.

Zero-Shot Classification Results. To further explore the capability of CoverCLIP, we evaluate zero-shot classification on 9 diverse datasets in Table 3. CoverCLIP achieves the best result on VOC2007-Multi (94.0%) and obtains competitive second-best results on GTSRB, ImageNet, ImageNet-V2, and SUN397. On the remaining datasets, performance is mixed, indicating that coverage-aware fine-tuning does not uniformly improve conventional zero-shot classification. Overall, CoverCLIP maintains competitive zero-shot performance while gaining sensitivity to graded semantic coverage.

	Method	ShareGPT-4V I2T	ShareGPT-4V T2I	Urban1K I2T	Urban1K T2I
B/16	CLIP	75.8	67.2	67.5	53.5
	Direct Fine-tuning	90.6	87.4	84.0	84.5
	Long-CLIP	91.1	91.1	77.4	79.6
	SmartCLIP	95.1	**92.7**	89.4	85.7
	CoverCLIP (Ours)	**95.9**	**92.7**	**91.0**	**90.5**
L/14	CLIP	77.1	69.3	68.5	55.9
	Direct Fine-tuning	92.8	90.0	88.3	89.0
	Long-CLIP	92.1	92.5	81.4	86.7
	SmartCLIP	95.5	93.9	92.2	88.2
	CoverCLIP (Ours)	**97.0**	**96.0**	**94.6**	**95.1**

Table 2: R@1 of long-caption retrieval on the 1k ShareGPT-4V validation set and Urban1K dataset. The best results are **bold**. T2I: Text-to-Image, I2T: Image-to-Text.

Dataset	CLIP	Long-CLIP	SmartCLIP	CoverCLIP
Country211	**31.5**	28.6	27.5	28.4
FER2013	45.2	**54.6**	50.8	45.8
FGVC-Aircraft	19.5	**21.4**	20.6	18.9
GTSRB	50.8	49.9	**52.8**	51.6
ImageNet	**80.7**	77.0	77.3	77.5
ImageNet-V2	**69.6**	65.6	67.1	68.0
SUN397	57.2	**62.5**	56.4	60.0
VOC2007	76.8	75.7	**77.6**	76.3
VOC2007-Multi	91.8	90.9	91.9	**94.0**

Table 3: Comparison of zero-shot classification performance across different ViT-L/14-based models. The best results are **bold**, and the second-best results are underlined.

We replace the CLIP text encoder in SDXL with different CLIP models to generate images with progressively increasing semantic coverage:
C_1: bright red vintage bus
C_2: + a large leafy oak tree
C_3: + a white horse
C_4: + a colorful hot air balloon

CoverCLIP properly adds each new visual element while retaining previously requested semantic elements, achieving accurate step-by-step semantic coverage. In contrast, both Long-CLIP and SmartCLIP begin to fail from C_3 onward. SmartCLIP generates an unnaturally shaped bus in C_3 and misses the white horse in C_4. Long-CLIP produces an extra red horse in C_3, and generates multiple red hot air balloons along with a white car in C_4.

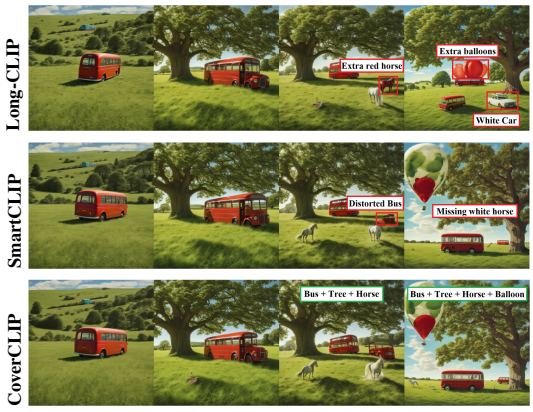

Figure 3: **Comparison of progressive semantic coverage in text-to-image generation.** Each row shows generated images from C_1 (left) to C_4 (right) for Long-CLIP, SmartCLIP, and CoverCLIP.

5.3 Ablation Studies

We conduct ablation studies on the ViT-L/14 backbone to examine the effects of training data composition, coverage-specific objectives, ranking design, and hyperparameter choices in CoverCLIP. Here, we summarize the key findings, while detailed quantitative results and further analyses are provided in Appendix D.

Training data ablation. We first analyze the contribution of different training data sources. The results show that coverage data provides the primary supervision for learning graded semantic completeness, while general image–text data and ShareGPT-4V data help preserve broader retrieval capability and improve robustness to semantically richer captions. Combining the three sources provides the best overall balance between semantic coverage and general image–text alignment.

Coverage supervision and ranking design. We further study the contribution of the HN separation loss and the coverage-ranking loss. Mixed-data training already provides a strong coverage-aware signal, while introducing either objective brings additional improvements in Exact Chain accuracy. Combining the two objectives yields the strongest Exact Chain performance, suggesting that they provide complementary coverage-aware supervision. We also compare different ranking designs. Adjacent-pair ranking performs better than all-pair ranking by directly modeling the progressive order $C_1 < C_2 < C_3 < C_4$. Applying stop-gradient to the lower-coverage caption further improves performance by encouraging the higher-coverage caption to move upward in similarity, while keeping the lower-coverage but still correct caption unchanged. This is preferable because lower-coverage captions remain correct descriptions of the image and should not be explicitly pushed downward in similarity.

Table 4: Text-to-image generation results over 200 paired random seeds. PSC measures the coverage of the weakest required concept throughout progressive generation, while New Atom measures the newly introduced concept at each stage. Semantic scores are computed as 100 times cosine similarity using an independent frozen FG-CLIP2-large evaluator. Higher is better.

Method	PSC ↑	New Atom ↑
Long-CLIP	11.382	11.892
SmartCLIP	12.198	12.711
CoverCLIP (Ours)	**12.391**	**12.759**

Hyperparameter Sensitivity. We additionally evaluate the sensitivity of CoverCLIP to the weights and margins of the coverage-specific objectives. Performance remains generally stable under moderate variations, while overly large ranking margins can reduce Exact Chain accuracy.

5.4 Text-to-Image Generation Evaluation

Qualitative Results. We replace the first CLIP text encoder in SDXL with Long-CLIP, SmartCLIP, or CoverCLIP while keeping all other model components and generation settings unchanged. Figure 3 presents a representative example in which CoverCLIP progressively realizes the requested semantics from C_1 to C_4, whereas Long-CLIP and SmartCLIP omit or incorrectly render some concepts from C_3 onward. Across additional representative examples, CoverCLIP more reliably incorporates weak or late-added concepts. More qualitative examples are provided in Appendix C.

Quantitative Results. To assess whether coverage-aware alignment transfers to generative tasks, we conduct a paired evaluation over 200 random seeds, generating 2,400 images in total (3 text encoders $\times$ 4 captions $\times$ 200 seeds). All methods use identical prompts, initial latent noise, and generation settings, with only the first text encoder replaced. We use FG-CLIP2-large (Xie et al., 2025a) as an independent frozen evaluator because it has strong fine-grained understanding ability. Our primary metric is *Progressive Semantic Coverage (PSC)*. For each generated image at stage C_t, we compute its similarity to every semantic atom required by C_t, take the minimum similarity, and average these minima over stages C_2–C_4. PSC therefore measures whether the least-represented required concept is realized, rather than only measuring alignment with the most salient content. As shown in Table 4, CoverCLIP achieves the highest PSC. To determine whether this advantage is consistent across random seeds rather than being driven by a small number of favorable generations, we perform paired statistical tests using the same seed for all three text encoders. CoverCLIP achieves significantly higher PSC than both Long-CLIP and SmartCLIP after Holm correction ($p = 2.00 \times 10^{-5}$ and $p = 6.40 \times 10^{-4}$, respectively), showing that its improvement in progressive semantic coverage is statistically reliable across the evaluated seeds. We further report the *New-atom score*. At each stage from C_2 to C_4, we compute the similarity between the generated image and the semantic atom newly introduced at that stage, and then average these scores across the three stages. This metric directly measures whether newly added semantics are successfully reflected in the generated image. CoverCLIP also obtains the highest mean New-atom score. Together with its advantages in PSC and the final-stage minimum atom score, these results show that CoverCLIP more reliably realizes newly introduced concepts and prevents weak or late-added semantics from being omitted during progressive generation.

6 Conclusion

In this work, we identify a limitation of existing CLIP models: their insensitivity to graded semantic coverage. To measure this, we propose CoverageBench, a manually verified benchmark of 1,300 images designed to evaluate models' sensitivity to graded semantic coverage. To address this, we propose CoverCLIP, a novel approach that incorporates coverage-aware supervision into standard contrastive learning. We introduce Coverage-1M, a dataset comprising graded coverage captions, long-short captions, and descriptive caption samples to support coverage-aware training. Building upon this data, we propose a step-wise mixed training framework that improves the model's coverage sensitivity while preserving its original alignment capability. Experimental results demonstrate the effectiveness of our approach, showing improvements on CoverageBench while maintaining competitive performance on standard retrieval and zero-shot classification tasks.

AI USE STATEMENT

We use large language and vision-language models as part of the data construction pipeline described in this paper. Specifically, Gemma4-31B is used for entity verification and training-data annotation, while GPT-5.6 Sol is used to assist the construction of CoverageBench captions from Visual Genome annotations. All automatically generated benchmark samples are subsequently validated and manually reviewed as described in Section 3.3. GPT-5.6 Sol is also used to improve the clarity and fluency of the manuscript.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the CoverageBench construction and validation pipeline in Section 3 and Appendix A, the Coverage-1M construction and training objectives in Section 4, and complete training configurations and hyperparameters in Appendix B. The prompts used for VLM- and LLM-based annotation are provided in Appendix E. Detailed evaluation protocols for text-to-image generation, including generation settings, metrics, and statistical testing procedures, are provided in Appendix C. Anonymous code and evaluation scripts are provided in the supplementary material. We will publicly release the benchmark data, training code, evaluation code, and model checkpoints upon publication.

REFERENCES

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pp. 370–387. Springer, 2024.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. In *International Conference on Learning Representations*, volume 2024, pp. 36221–36237, 2024.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.

Ian J. Goodfellow, Dumitru Erhan, Pierre-Luc Carrier, et al. Challenges in representation learning: A report on three machine learning contests. In *NeurIPS Workshop*, 2013.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in neural information processing systems*, 36:31096–31116, 2023.

Weiquan Huang, Aoqi Wu, Yifan Yang, Xufang Luo, Yuqing Yang, Usman Naseem, Chunyu Wang, Qi Dai, Xiyang Dai, Dongdong Chen, et al. Llm2clip: Powerful language model unlocks richer cross-modality representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5131–5139, 2026.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Ranjay Krishna, Yuke Zhu, Oliver Groth, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *IJCV*, 2017.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

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

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. @ crepe: Can vision-language foundation models reason compositionally? In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10910–10921. IEEE, 2023.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Maitreya Patel, Abhiram Kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, and Yezhou Yang. Tripletclip: Improving compositional reasoning of clip via synthetic vision-language negatives. *Advances in neural information processing systems*, 37:32731–32760, 2024.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. In *International Joint Conference on Neural Networks*, 2012.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5228–5238. IEEE, 2022.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

Chunyu Xie, Bin Wang, Fanjing Kong, Jincheng Li, Dawei Liang, Ji Ao, Dawei Leng, and Yuhui Yin. Fg-clip 2: A bilingual fine-grained vision-language alignment model. *arXiv preprint arXiv:2510.10921*, 2025a.

Chunyu Xie, Bin Wang, Fanjing Kong, Jincheng Li, Dawei Liang, Gengshen Zhang, Dawei Leng, and Yuhui Yin. Fg-clip: Fine-grained visual and textual alignment. *arXiv preprint arXiv:2505.05071*, 2025b.

Shaoan Xie, Lingjing Lingjing, Yujia Zheng, Yu Yao, Zeyu Tang, Eric P Xing, Guangyi Chen, and Kun Zhang. Smartclip: Modular vision-language alignment with identification guarantees. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29780–29790, 2025c.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *European conference on computer vision*, pp. 310–325. Springer, 2024.

A COVERAGEBENCH STATISTICS

A.1 FINAL DATASET SCALE

The final benchmark contains 1,300 non-duplicate images sourced from Visual Genome. Each image is annotated with 4 scored semantic atoms and 4 captions with progressively increasing semantic coverage, resulting in a total of 5,200 semantic atom instances and 5,200 image-caption pairs, as summarized in Table 5.

A.2 CAPTION LENGTH DISTRIBUTION

To prevent models from exploiting text length as a shortcut for coverage ranking, the captions C_4, C_3, C_2, and C_1 are carefully controlled to fall within a similar length range (Table 6). Word counts are computed using whitespace-separated English words, with hyphenated or apostrophe-containing compounds treated as single words.

Table 5: CoverageBench dataset statistics.

Statistic	**Value**
Number of images	1,300
Total semantic atoms	5,200
Semantic atoms per image	4
Total captions	5,200
Captions per image	4
Human review pass rate (of 1,357)	95.80%

Table 6: Caption length statistics across coverage levels.

Caption	Mean	Std	Median	Min	Max
C_4	17.78	1.18	18	16	20
C_3	17.18	0.86	17	16	20
C_2	16.80	0.77	17	16	20
C_1	16.48	0.62	16	16	19

For each sample, the length span across coverage levels is defined as:

$$\Delta_{\text{len}} = \max_{k\in\{1,2,3,4\}} \ell(C_k) - \min_{k\in\{1,2,3,4\}} \ell(C_k). \tag{8}$$

The overall length balance statistics are summarized in Table 7, and the detailed distribution of these spans is presented in Table 8.

All samples exhibit a maximum caption length span of no more than 4 words, with an average C_4–C_1 length difference of only 1.30 words. This controlled length distribution reduces the possibility that models can infer semantic coverage primarily from caption length rather than image-text alignment.

Statistic	**Value**
Mean caption length span	1.73 words
Median length span	2 words
Maximum length span	4 words
Samples with span ≤ 4 words	100.00%
Mean C_4 vs. C_1 length difference	1.30 words

Table 7: Length balance statistics.

Max length diff.	Samples	Ratio
0 words	72	5.54%
1 word	510	39.23%
2 words	468	36.00%
3 words	199	15.31%
4 words	51	3.92%

Table 8: Distribution of maximum length differences across coverage levels.

A.3 Image Entity Diversity

All 1,300 images are selected from the entity-verified set with `true_object_count > 5`. The distribution of true entity counts in the final samples is shown in Table 9.

True entity count	Image count	Ratio
6	660	50.77%
7	413	31.77%
8	177	13.62%
9	47	3.62%
10	3	0.23%

Table 9: Distribution of true entity counts per image.

The mean true entity count is 6.71 ± 0.85, with a median of 6 and a range of 6–10. This distribution ensures that the benchmark primarily consists of complex scenes featuring multiple salient entities, rather than single-subject images. Note that `true_object_count` is utilized for candidate image filtering and does not necessarily equal the final scored semantic atom count. Each image consistently selects exactly 4 human-verified atoms to construct the controllable C_1–C_4 coverage hierarchy.

A.4 Semantic Atom Statistics

Each image contains exactly 4 scored semantic atoms, with comprehensive statistics provided in Table 10.

Statistic	**Value**
Total semantic atoms	5,200
Atoms per image	4
Distinct atom texts	3,959
Mean atom length	2.92 words
Atom length std	1.42 words
Median atom length	3 words
Atom length range	1–9 words

Table 10: Semantic atom statistics.

These semantic atoms describe a variety of visual concepts, including objects, attributes, relations, actions or states, counts, and scene information.

B Implementation Details

All experiments are conducted using distributed data parallel training on 8 NVIDIA H20 GPUs with 96 GB of memory per GPU. As shown in Table 11, we train CoverCLIP for three epochs using

Table 11: Implementation details for CoverCLIP-B/16 and CoverCLIP-L/14.

Shared configurations

Setting	Value	Setting	Value
Initialization	OpenAI CLIP	Number of GPUs	8
Text context length	248	GPU memory	96 GB
Preserved positions	20	Epochs	3
Training precision	BF16	Optimizer	AdamW
Learning rate	1×10^{-6}	Weight decay	0.01
Warmup steps	0	Gradient clipping	1.0
Nominal data ratio	70/22.5/7.5	Random seed	42
Data workers/GPU	4	Sample shuffle buffer	8192
Shard shuffle buffer	128	Checkpoint interval	Every epoch

Model-specific configurations

Setting	ViT-B/16	ViT-L/14
Batch size per GPU	128	96
Global batch size	1024	768
Steps per epoch	991	1322
Total optimizer updates	2973	3966
Coverage batches/epoch	694	925
FineHARD general batches/epoch	222	297
ShareGPT-4V batches/epoch	75	100
Actual data ratio	70.03/22.40/7.57	69.97/22.47/7.56

AdamW with a learning rate of 1×10^{-6}, a weight decay of 0.01, and no learning-rate warmup. Gradients are clipped to a maximum norm of 1.0. Training uses bfloat16 precision without gradient accumulation. We use a fixed random seed of 42 and save a checkpoint after every epoch. To maintain a consistent data distribution, we apply the same nominal training-data proportions for both CoverCLIP-B/16 and CoverCLIP-L/14: 70% coverage-aware batches, 22.5% general batches, and 7.5% ShareGPT-4V batches. The three batch types are interleaved across optimization steps, and each step processes only one batch type.

For CoverCLIP-B/16, we use a per-GPU batch size of 128, resulting in a global batch size of 1,024. Each epoch contains 991 optimization steps, consisting of 694 coverage-aware batches, 222 general batches, and 75 ShareGPT-4V batches. After integer scheduling, the actual proportions are 70.03%, 22.40%, and 7.57%, respectively. The model is trained for 2,973 optimizer updates over three epochs. For CoverCLIP-L/14, we use a per-GPU batch size of 96, resulting in a global batch size of 768. Each epoch contains 1,322 optimization steps, consisting of 925 coverage-aware batches, 297 general batches, and 100 ShareGPT-4V batches. The corresponding actual proportions are 69.97%, 22.47%, and 7.56%. The model is trained for 3,966 optimizer updates over three epochs. For Direct Fine-tuning, we adopt the same training configuration and data as our full model, yet we apply only the standard CLIP loss. Moreover, across all batch types—coverage, general, and ShareGPT-4V—we solely utilize the original long captions (i.e., without any coverage-specific or short captions).

Coverage-1M is stored as WebDataset tar shards. We shuffle 128 shards and use a sample-level shuffle buffer of 8,192 samples. Each GPU uses four data-loading workers, persistent workers, and a prefetch factor of two. We use the standard image preprocessing pipeline associated with the corresponding OpenAI CLIP backbone. Images containing more than 50 million pixels are skipped. Despite the different batch sizes and step counts, both configurations process nearly the same total volume of data per epoch.

C SDXL GENERATION EXPERIMENTS

C.1 EXPERIMENTAL SETUP

Generation Model and Compared Methods. We use Stable Diffusion XL Base 1.0 (Podell et al., 2024) as the text-to-image generation model. We compare Long-CLIP (Zhang et al., 2024), Smart-CLIP (Xie et al., 2025c), and CoverCLIP, all based on the ViT-L/14 architecture. Each method replaces the first text encoder of SDXL, while the second OpenCLIP ViT-bigG/14 encoder remains shared across all methods.

Generation Settings. Images are generated at 1024×1024 resolution using 30 sampling steps, a classifier-free guidance scale of 5.0, and FP16 precision. The maximum context length of the ViT-L/14 encoder is 248 tokens. No additional negative prompt is used; the negative-prompt embedding is a zero tensor. All local generation experiments are conducted on a single NVIDIA RTX 4090 GPU with 24 GB of memory.

Fairness Controls. All methods share the same SDXL checkpoint, ViT-bigG/14 encoder, prompts, initial latent noise, sampling parameters, and numerical precision. The only component varied across methods is the ViT-L/14 text encoder.

Progressive Generation Protocol. Each caption chain contains four captions $C_1, \ldots, C_4$ that progressively introduce four semantic atoms. The four images are generated through independent text-to-image runs. Before each run, the random generator is reset to the same seed, so all stages and methods start from identical initial latent noise. The image generated at stage C_{t-1} is not used as input to stage C_t.

Text Encoding. For each caption, we independently extract token-level hidden states from the ViT-L/14 and ViT-bigG/14 encoders and concatenate them along the feature dimension:

$$E_{\text{prompt}} = \text{Concat}\left(E_{\text{ViT-L/14}}, E_{\text{ViT-bigG/14}}\right). \tag{9}$$

The pooled text embedding required by SDXL is produced by the shared ViT-bigG/14 encoder. We remove the original SDXL text encoders and tokenizer and directly provide the precomputed prompt embeddings. The ViT-bigG/14 positional embeddings are linearly interpolated to support the same context length as the ViT-L/14 encoder.

C.2 FG-CLIP2 EVALUATION PROTOCOL

We use FG-CLIP2-large (Xie et al., 2025a) as an independent frozen evaluator. FG-CLIP2 has strong fine-grained image–text understanding capability and is therefore better suited than a conventional CLIP to measuring whether every atomic concept in a progressive caption is visually realized. Each atomic query is formatted as `a photograph of {atom}`, tokenized with `max_length=64` and `walk_type=short`. Following the official adaptive-patch preprocessing, each 1024×1024 image is encoded with `max_num_patches=1024`. Evaluator inference is performed in FP32.

Let $a_1, \ldots, a_T$ denote the semantic atoms progressively introduced by captions $C_1, \ldots, C_T$, where $T = 4$, and let I_t denote the image generated from caption C_t. Let $f_I(\cdot)$ and $f_T(\cdot)$ denote the ℓ_2-normalized FG-CLIP2 image and text features. We define semantic similarity as

$$s(I_t, a_j) = 100 \cdot \cos\left(f_I(I_t), f_T(a_j)\right). \tag{10}$$

The learned FG-CLIP2 logit scale and bias are not applied. Consequently, $s(I_t, a_j)$ is directly interpretable as 100-scaled cosine similarity, with a theoretical range of $[-100, 100]$; it is not a percentage or a probability.

Progressive Semantic Coverage. Our primary metric is Progressive Semantic Coverage (PSC):

$$\text{PSC} = \frac{1}{T-1} \sum_{t=2}^{T} \min_{1 \le j \le t} s(I_t, a_j), \qquad T = 4. \tag{11}$$

Table 12: Progressive Semantic Coverage over 200 paired random seeds, evaluated with FG-CLIP2-large. Semantic scores are 100 times cosine similarity. Values are mean ± standard deviation. Higher is better.

Method	PSC ↑	N
Long-CLIP	11.382 ± 1.267	200
SmartCLIP	12.198 ± 1.154	200
CoverCLIP (Ours)	$\mathbf{12.391 \pm 0.966}$	200

Table 13: Paired statistical comparisons of CoverCLIP PSC. Confidence intervals use 10,000 paired bootstrap resamples. Adjusted p-values use a two-sided paired sign-flip test with 100,000 permutations and Holm correction.

Comparison	95% CI	Cohen's d_z	Adj. p
CoverCLIP − Long-CLIP	[0.846, 1.176]	0.840	2.00×10^{-5}
CoverCLIP − SmartCLIP	[0.084, 0.302]	0.246	6.40×10^{-4}

At each stage, PSC takes the minimum similarity among all semantic atoms required by the current caption and averages it over stages C_2–C_4. PSC therefore evaluates balanced semantic realization rather than alignment with only the most salient concept. For example, a PSC value of 12.391 corresponds to an average minimum cosine similarity of 0.12391.

Secondary Metrics. We additionally report:

- **Expected-atom mean**: average similarity to all atoms required by the current caption.
- **Retained-atom mean**: average similarity to atoms introduced in previous stages.
- **New-atom score**: similarity to the atom newly introduced at the current stage.
- **New-atom gain**: increase in similarity to a newly introduced atom relative to the preceding stage.
- **Retention delta**: change in similarity to previously introduced atoms between consecutive stages.
- C_4 **minimum atom score**: minimum similarity among the four required atoms in the final image.
- C_4 **mean atom score**: average similarity to the four required atoms in the final image.
- **Adjacent-stage continuity**: unscaled cosine similarity between adjacent normalized FG-CLIP2 image features.

All image–text semantic metrics are reported as 100-scaled FG-CLIP2 cosine-similarity points. Adjacent-stage continuity is the only unscaled metric and theoretically ranges from -1 to 1.

Statistical Analysis. For PSC, we report the mean and standard deviation over paired random seeds. Pairwise confidence intervals use 10,000 paired bootstrap resamples. Statistical significance is evaluated using a two-sided paired sign-flip test with 100,000 permutations followed by Holm correction. We additionally report Cohen's d_z.

C.3 Large-Scale Quantitative Evaluation

We use 200 pre-specified random seeds. For each seed, Long-CLIP, SmartCLIP, and CoverCLIP independently generate images for captions C_1 through C_4, resulting in 2,400 images. The generated images are reused across evaluators; only the evaluation model is changed to FG-CLIP2-large.

CoverCLIP significantly improves PSC over both baselines. The effect is large relative to Long-CLIP and smaller but statistically significant relative to SmartCLIP. This result indicates that CoverCLIP more reliably realizes the weakest required concept across progressive captions.

Table 14: Secondary metrics over 200 paired random seeds evaluated with FG-CLIP2-large. Semantic metrics are reported as 100 times cosine similarity; continuity is unscaled image–image cosine similarity. Values are mean ± standard deviation. Higher is better.

Metric	Long-CLIP	SmartCLIP	CoverCLIP (Ours)
PSC ↑	11.382 ± 1.267	12.198 ± 1.154	$\mathbf{12.391 \pm 0.966}$
Expected-atom mean ↑	16.630 ± 0.679	16.862 ± 0.707	$\mathbf{16.865 \pm 0.737}$
Retained-atom mean ↑	$\mathbf{18.827 \pm 0.817}$	18.793 ± 0.849	18.761 ± 0.879
New-atom score ↑	11.892 ± 1.276	12.711 ± 1.023	$\mathbf{12.759 \pm 0.948}$
New-atom gain ↑	7.172 ± 1.381	8.315 ± 1.315	$\mathbf{8.361 \pm 1.196}$
Retention delta ↑	$\mathbf{-1.599 \pm 0.734}$	-1.669 ± 0.624	-1.762 ± 0.606
C_4 minimum atom score ↑	4.156 ± 2.703	6.231 ± 2.329	$\mathbf{6.564 \pm 1.895}$
C_4 mean atom score ↑	12.313 ± 1.060	$\mathbf{12.775 \pm 0.917}$	12.701 ± 0.935
Adjacent-stage continuity ↑	$\mathbf{0.909 \pm 0.020}$	0.899 ± 0.022	0.901 ± 0.020

Table 15: FG-CLIP2-large evaluation of four selected progressive-generation cases. Semantic metrics are reported as 100 times cosine similarity; continuity is unscaled image–image cosine similarity. Higher is better.

Scenario	Text Encoder	Retained	New Atom	Expected	Continuity	C_4 Min	PSC
Forest Campsite	Long-CLIP	**18.552**	10.087	**15.814**	**0.959**	3.309	10.087
	SmartCLIP	18.147	9.823	15.321	0.956	4.471	9.823
	CoverCLIP (Ours)	18.047	**11.010**	15.627	0.952	**5.549**	**11.010**
Food Truck Plaza	Long-CLIP	14.832	8.189	13.229	**0.932**	-1.022	6.830
	SmartCLIP	15.334	8.695	13.704	0.928	-1.221	7.587
	CoverCLIP (Ours)	**15.612**	**10.972**	**14.535**	0.922	**4.945**	**9.593**
Desert Camper	Long-CLIP	20.494	10.671	17.499	0.903	0.835	10.671
	SmartCLIP	**21.081**	11.086	**18.018**	**0.911**	-0.470	11.086
	CoverCLIP (Ours)	20.274	**13.253**	17.952	0.901	**8.316**	**13.253**

CoverCLIP obtains the best mean PSC, Expected-atom score, New-atom score, New-atom gain, and C_4 minimum atom score. Relative to SmartCLIP, the differences in Expected-atom score, New-atom score, and New-atom gain are small and not statistically significant. In contrast, CoverCLIP's improvement in C_4 minimum atom score is significant, with a paired difference of $+0.334$ and a 95% confidence interval of $[0.094, 0.578]$. Thus, the strongest robust advantage is improved coverage of the weakest final-stage semantic atom.

CoverCLIP does not dominate retention or visual continuity. Long-CLIP has the highest Retained-atom mean and adjacent-stage continuity, and both baselines have a less negative retention delta. Our claim is therefore not universal superiority on every metric, but more balanced coverage and fewer omissions of weak or late-added concepts.

C.4 Qualitative Examples and Case-Level Analysis

Figure 3 in the main paper presents the Vintage Bus scenario, while Figure 4 provides three additional scenarios: Forest Campsite, Food Truck Plaza, and Desert Camper. We report FG-CLIP2-based case-level quantitative results for the three scenarios in Figure 4 in Table 15.

Across these three scenarios, CoverCLIP achieves the highest New-atom score, C_4 minimum atom score, and PSC. In Forest Campsite, its C_4 minimum score is 5.549, compared with 3.309 for Long-CLIP and 4.471 for SmartCLIP. In Food Truck Plaza, CoverCLIP raises PSC to 9.593 from 6.830 and 7.587, and increases the C_4 minimum score to 4.945. In Desert Camper, CoverCLIP reaches a PSC of 13.253 and a C_4 minimum score of 8.316. These case-level results are consistent with the 200-seed experiment: CoverCLIP's most consistent advantage lies in better coverage of weak and late-added semantic elements, rather than universally higher retention or image-feature continuity.

Forest Campsite

C_1: a large orange camping tent
C_2: + a tall evergreen pine tree
C_3: + a blue bicycle
C_4: + a white dog

CoverCLIP generates a recognizable white dog at C_4 while retaining the tent, pine tree, and bicycle. Long-CLIP and SmartCLIP produce darker dogs and do not satisfy the requested white-dog attribute as reliably.



Food Truck Plaza

C_1: a red food truck
C_2: + a large yellow market umbrella
C_3: + a blue bicycle
C_4: + a white dog

CoverCLIP retains the food truck, umbrella, and bicycle while adding a recognizable white dog. The baselines omit the late-added dog in the visually crowded plaza.



Desert Camper

C_1: a bright blue camper van
C_2: + a tall green saguaro cactus
C_3: + a brown camel
C_4: + a red hot air balloon

CoverCLIP preserves the camper van, cactus, and camel while adding a red hot air balloon. Long-CLIP omits the balloon, while SmartCLIP produces an implausible cactus-like shape instead of the requested balloon.

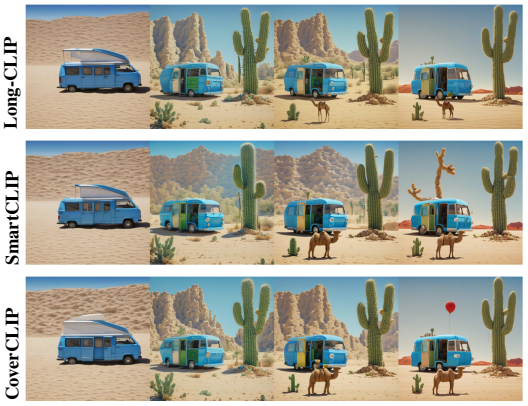

Figure 4: **Additional examples of progressive text-to-image generation.** The three scenarios are Forest Campsite (top), Food Truck Plaza (middle), and Desert Camper (bottom). Each row presents independently generated images for captions C_1 through C_4, using identical initial latent noise across the three text encoders.

D ABLATION RESULTS

We conduct extensive ablation studies on the ViT-L/14 backbone to analyze the contribution of each component in CoverCLIP.

Training data ablation. To understand the role of each data source, we compare different combinations in Table 16 (a). Using coverage data alone achieves 50.2% Exact Chain but degrades retrieval performance. Adding general data improves retrieval but slightly reduces Exact Chain.

Table 16: Ablation studies and hyperparameter sensitivity of CoverCLIP with the ViT-L/14 backbone. The ratios in (a) represent the batch mixing proportions for coverage, general, and ShareGPT-4V objectives. We also report R@1 image-to-text retrieval accuracy on COCO and Flickr30K. The best results are shown in **bold**.

Training Data	Exact Chain	Flickr30K	COCO
Coverage only $(100/0/0)$	50.2	61.6	61.7
Coverage + General $(70/30/0)$	48.0	64.7	65.3
Coverage + ShareGPT-4V $(70/0/30)$	**54.8**	63.9	64.4
Full $(70/22.5/7.5)$	51.2	**65.0**	**65.6**

(a) Training data ablation

Method	HN	Rank	Exact Chain	Flickr30K	COCO
Original CLIP	–	–	29.7	47.8	55.9
Mixed-data CLIP	–	–	44.7	63.1	64.9
Mixed-data + HN Separation Loss	✓	–	49.6	64.9	**66.0**
Mixed-data + Ranking Loss	–	✓	48.7	**65.3**	65.7
Full CoverCLIP	✓	✓	**51.2**	65.0	65.6

(b) Coverage supervision and objective ablation

Ranking Pairs	StopGrad	Exact Chain	Flickr30K	COCO
All-pair	✓	48.7	64.9	65.6
Adjacent	–	50.6	**65.0**	**66.1**
Adjacent	✓	**51.2**	**65.0**	65.6

(c) Coverage-ranking design ablation

Param.	Val.	Exact Chain
λ_{rank}	0.00	49.6
	0.02	49.4
	0.05	**51.2**
	0.10	50.5
λ_{HN}	0.00	48.7
	0.05	50.4
	0.10	**51.2**
	0.20	48.9
γ_{rank}	0.00	49.7
	0.01	**51.2**
	0.03	48.9
	0.05	47.6
γ_{HN}	0.00	48.8
	0.01	48.8
	0.03	**51.2**
	0.05	50.5

(d) Hyperparameter sensitivity

Adding ShareGPT-4V data significantly boosts Exact Chain to 54.8% but still underperforms on retrieval. The full configuration with all three data sources achieves the best trade-off, with 65.0% on Flickr and 65.6% on COCO while maintaining strong coverage sensitivity (51.2% Exact Chain).

Coverage supervision and objective ablation. To isolate the effect of each loss term, we compare different training objectives in Table 16 (b). The mixed-data CLIP baseline already improves Exact Chain accuracy from 29.7% to 44.7%, showing that coverage-aware training data itself provides a strong supervision signal. Adding the HN separation loss further improves Exact Chain accuracy from 44.7% to 49.6%. The ranking loss also improves Exact Chain to 48.7% by explicitly encouraging the desired ordering among correct captions. Combining both objectives achieves the best Exact Chain accuracy of 51.2%, indicating that HN separation and coverage ranking provide complementary supervision beyond mixed-data training. These results suggest that CoverCLIP benefits primarily from the mixed coverage-aware training paradigm, while HN separation and ranking supervision provide complementary signals that further improve Exact Chain performance.

Ranking design ablation. To determine the optimal ranking strategy, we compare different pair selections in Table 16 (c). Adjacent-pair ranking with StopGrad-Low achieves the best Exact Chain (51.2%), outperforming all-pair ranking (48.7%) and adjacent-pair without StopGrad-Low (50.6%). This suggests that focusing on adjacent coverage levels with gradient stopping on lower-level captions is more effective for learning graded coverage ranking.

Hyperparameter Sensitivity. Table 16 (d) reports one-at-a-time hyperparameter sensitivity, where each parameter is varied while all others are fixed to their default values. It shows that CoverCLIP remains relatively stable under moderate variations of the loss weights and margins. Exact Chain is more sensitive to the ranking margin, with performance degrading as γ_{rank} becomes overly large. The best Exact Chain performance is achieved with $\lambda_{\text{rank}} = 0.05$, $\lambda_{\text{HN}} = 0.10$, $\gamma_{\text{rank}} = 0.01$, and $\gamma_{\text{HN}} = 0.03$. Increasing the weight of the HN separation loss or using larger margins does not consistently improve coverage ranking, suggesting that overly strong auxiliary constraints can interfere with learning the desired graded ordering. Overall, the default configuration provides the best Exact Chain performance among the tested settings.

E VLM PROMPTS

Listing 1: Prompt for CoverageBench entity verification

```
You are validating Visual Genome object annotations for a CLIP-style
    image benchmark.

You will receive one image and a list named Selected Top10 Objects.
    Each object comes from VG annotations and has a bbox, salience,
    object_category, and coarse_group.

Your task is to count only TRUE, VISIBLE, CONCRETE entity categories
    in the image.

Important filtering rules:
1. Count only concrete visible entities, object parts, people, animals
    , vehicles, furniture, foods, natural objects, and scene objects
    that are actually visible.
2. Do NOT count function words or non-entities as object categories,
    such as: is, the, an, a, be, of, on, in, into, onto, with, cover,
    covered, covering.
3. Do NOT count colors as objects: red, blue, green, white, black,
    brown, yellow, etc.
4. Do NOT count age or vague descriptors as objects: young, old, adult
    , baby, large, small, left, right, top, bottom, etc.
5. Do NOT count generic parent categories if a more specific true
    category is present in the same image.
   Examples:
   - If dogs or cats are present, do not count pets.
   - If dogs, cats, horses, birds, etc. are present, do not count
     animals.
   - If car, bus, truck, bicycle, etc. are present, do not count
     vehicle.
   - If man, woman, child, person are present, do not count people as
     an extra category.
   - If pizza, cheese, tomato, crust are present, do not count food as
     an extra category.
6. If both a generic category and its real specific category appear,
    keep only the specific category/categories.
7. Count distinct TRUE categories, not duplicate boxes or synonyms.
8. If an annotation is visible but is only a material/texture/pattern
    or too vague, reject it unless it is a concrete visible object
    category.
9. Use the image to verify whether each object candidate is actually
    visible and plausibly corresponds to the annotation.
10. Be conservative: reject categories that are not clear from the
    image or are only abstract/generic labels.

Return valid JSON only, with this exact schema:
{
  "image_id": "...",
  "kept_objects": [
    {"name": "...", "reason": "concrete visible entity"}
  ],
  "rejected_objects": [
    {"name": "...", "reason": "..."}
  ],
  "true_object_count": 0,
  "keep_sample": false,
  "notes": ""
}

Set keep_sample=true only if true_object_count > 5 after applying all
    filtering rules.
Do not include markdown, comments, or explanations outside the JSON.
```

Listing 2: Prompt for Coverage-1M annotation stage 1

```
You are a semantic deduplication assistant.
```

```
You will receive an ordered list of visual expressions extracted from
    image regions. Earlier expressions are more important.

Your task is to remove semantic duplication while preserving all
    unique visual information.

Rules:

1. Process expressions in the original order.
2. Split each expression into atomic visual facts, such as objects,
    attributes, actions, scenes, and relations.
3. Keep the first occurrence of each semantic fact.
4. In later expressions, remove facts already expressed earlier,
    including synonyms, paraphrases, nested phrases, and equivalent
    descriptions.
5. Preserve any genuinely new semantic facts in later expressions.
6. If an expression contains both repeated and new facts, rewrite it
    to contain only the new facts.
7. If an expression contains no new facts after deduplication, remove
    it entirely.
8. Do not add, infer, generalize, or modify visual facts.
9. Keep the remaining expressions concise, natural, and grammatically
    correct.
10. Preserve the original importance order.

Example input:
[
"a light-colored scarf and a pair of glasses",
"a red blazer, a light-colored scarf, and a pair of glasses",
"glasses",
"a voluminous curly hairstyle",
"the subject's face"
]

Example output:
{
"deduplicated_exprs": [
"a light-colored scarf and a pair of glasses",
"a red blazer",
"a voluminous curly hairstyle",
"the subject's face"
]
}

Return exactly one valid JSON object:
{
"deduplicated_exprs": [
"<expression_1>",
"<expression_2>"
]
}

Output JSON only. Do not include markdown, explanations, comments, or
    reasoning.
```

Listing 3: Prompt for Coverage-1M annotation stage 2

```
You are a strict dataset annotation assistant for image-text coverage
    ranking.

Your task is to generate captions for training a CLIP-style model to
    prefer captions with more complete visible visual coverage.

Input:
```

```
* A whole-image caption.
* A short whole-image caption.
* A deduplicated cumulative coverage_plan for C1, C2, C3, and C4.

Output exactly one valid JSON object:
{
"C1": "...",
"C2": "...",
"C3": "...",
"C4": "...",
"HN": "...",
}

Rules:

1. C1, C2, C3, and C4 must all be visually correct captions for the
    same image.
2. Follow the coverage_plan exactly:

  * C1 uses only the contents assigned to C1.
  * C2 uses all C1 contents plus the new C2 contents.
  * C3 uses all C2 contents plus the new C3 contents.
  * C4 uses all contents in the full coverage plan.
3. Do not mention, imply, or paraphrase later-stage contents in
    earlier captions.
4. Prefer the provided expressions and preserve their original meaning
    .
5. Rewrite the expressions into fluent, natural English.
6. Do not add unsupported objects, attributes, actions, scenes, or
    relations.
7. Each caption must contain 16-25 English words.
8. Keep all five captions similar in length.
9. To increase length, use only neutral framing phrases such as:

  * "The image shows..."
  * "In this photograph, we can see..."
  * "This image presents..."
  * "The main subject is shown..."
10. Neutral framing must not introduce new visual information.
11. HN must remain close to C4 in wording and length, but must alter
exactly one independent visual fact. Every other semantic fact in C4
must remain unchanged and correct. If no valid single-error HN can
be constructed, mark the sample as invalid.
12. Prefer one natural sentence for each field.
13. Output JSON only. Do not include markdown, comments, explanations,
     or reasoning.
```

Listing 4: Prompt for CoverageBench caption construction with GPT-5.6 Sol

```
You are constructing **CoverageBench**, a benchmark for **CLIP-style
    image-text similarity ranking**.

You will receive:

1. One image.
2. Visual Genome-derived annotations for that image:

  * object annotations
  * attributes
  * relationships
  * region descriptions
  * scene candidates
```

```
The benchmark evaluates only image-text similarity:

`score(image, caption) = cosine(image_embedding, text_embedding)`

Your task is to generate clean candidate captions with progressively
    increasing semantic coverage for one image.

## Critical principles

### 1. Caption text must be clean

The `text` field of every caption must contain only the candidate
    caption used for CLIP evaluation.

Do not put explanations, evidence, reasoning, labels, comments, or
    annotation notes inside the caption text.

Bad caption:

`A cabbage plant is growing from dark soil, which is visible in the
    image.`

Good caption:

`A cabbage plant is growing from dark soil in a garden bed.`

### 2. Visual Genome object annotations are reliable

Treat the provided VG object annotations as reliable ground truth.

Use the image and other VG annotations to make captions natural, but
    the core visual facts should be derived from the provided
    annotations.

### 3. Coverage is based on scored semantic atoms

Each sample must have exactly 4 scored semantic atoms.

A scored semantic atom is one visually grounded semantic fact, such as
    :

* an object: `broccoli plant`
* an attribute fact: `dark soil`
* an action/state: `plant growing from soil`
* a relation: `plant growing from soil`
* a scene/context fact: `garden bed`
* a count fact: `one plant`

The 4 scored atoms define coverage.

A caption with coverage level 4 must express all 4 atoms.

A caption with coverage level 3 must express exactly 3 atoms.

A caption with coverage level 2 must express exactly 2 atoms.

A caption with coverage level 1 must express exactly 1 atom.

## Caption length control

Every generated caption, including `C4`, `C3`, `C2`, and `C1`, must
    contain between **16 and 20 English words**, inclusive.

Count words using whitespace-separated tokens. Punctuation attached to
     a word does not create an additional word.
```

```
All four captions should have similar lengths. Avoid making lower-
    coverage captions noticeably shorter than higher-coverage captions
    .

Length expansion must not change, strengthen, weaken, or add semantic
    visual information.

To reach the required length, use only semantically neutral framing or
     grammatical expressions, such as:

* `The image shows...`
* `The photograph presents...`
* `We can see that...`
* `In this image, there is...`
* `The scene contains...`
* `Clearly visible in the image is...`
* `The main subject shown here is...`
* `This photograph provides a view of...`
* `Within the image, we can observe...`
* `The ... is shown ... with ...`

These expressions may improve fluency and increase length, but they
    must not introduce additional scored atoms or unsupported visual
    facts.

Do not use omitted atoms in lower-coverage captions for length
    expansion.

For example, if `garden bed` is an omitted atom in `C1`, `C1` must not
     mention a garden, garden bed, outdoor area, soil bed, or an
    equivalent expression.

## Coverage captions

### C4

* Must express all 4 scored semantic atoms.
* Must be visually correct.
* Must contain 16-20 English words.
* Coverage level: `4`

### C3

* Must express exactly 3 of the 4 scored semantic atoms.
* Must be visually correct.
* Must not express or imply the omitted atom.
* Must contain 16-20 English words.
* Coverage level: `3`

### C2

* Must express exactly 2 of the 4 scored semantic atoms.
* Must be visually correct.
* Must not express or imply the omitted atoms.
* Must contain 16-20 English words.
* Coverage level: `2`

### C1

* Must express exactly 1 of the 4 scored semantic atoms.
* Must be visually correct.
* Must not express or imply the omitted atoms.
* Must contain 16-20 English words.
* Coverage level: `1`
```

```
Caption rules:

* All captions must be natural English.
* All captions must be true according to the image and VG annotations.
* Lower-coverage captions should remain positive and meaningful.
* Generic grammatical and framing words are allowed, but extra visual
    facts are not.
* Neutral framing expressions do not count as semantic atoms.
* Paraphrases and synonyms still count as expressing the corresponding
     semantic atom.
* Do not hide an omitted atom through synonyms, hypernyms, pronouns,
    implications, or indirect descriptions.
* Do not repeat the same semantic atom several times merely to reach
    the required length.
* Prefer one complete, natural sentence for each caption.

## Output format

Return valid JSON only.

Do not include Markdown.

Do not include comments.

Do not include explanations outside JSON.

Do not include reasoning.

Use this exact schema:

{
"image_id": "<image_id>",
"semantic_atoms": [
"<atom_1>",
"<atom_2>",
"<atom_3>",
"<atom_4>"
],
"captions": [
{"id": "C4", "text": "...", "coverage_level": 4},
{"id": "C3", "text": "...", "coverage_level": 3},
{"id": "C2", "text": "...", "coverage_level": 2},
{"id": "C1", "text": "...", "coverage_level": 1}
],
"quality": {"needs_manual_review": false}
}

## Final validation before output

Before returning JSON, silently check:

1. There are exactly 4 captions.
2. Required IDs appear exactly once: `C4`, `C3`, `C2`, `C1`.
3. Every caption `text` field contains only a clean candidate caption.
4. No caption `text` field contains explanations, evidence, labels,
    comments, or annotation notes.
5. Every caption contains between 16 and 20 whitespace-separated
    English words, inclusive.
6. The four captions have reasonably similar lengths.
7. There are exactly 4 scored semantic atoms.
8. `C4` expresses all 4 atoms.
9. `C3` expresses exactly 3 atoms.
10. `C2` expresses exactly 2 atoms.
11. `C1` expresses exactly 1 atom.
```

```
12. 'C3', 'C2', and 'C1' do not mention, paraphrase, imply, or
    indirectly reveal omitted atoms.
13. 'C1' is still a visually correct positive caption.
14. Length-expansion phrases do not introduce new visual facts.
15. No caption adds unsupported objects, attributes, actions,
    relations, counts, scenes, colors, lighting, or background
    information.
16. All caption texts are fluent, grammatical, and natural.
17. The output is valid JSON only.
```