SEE MORE FROM LESS: SAMPLE-EFFICIENT GENERATIVE FEEDBACK FOR FINE-GRAINED VISUAL UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative feedback can improve the fine-grained visual understanding of Contrastive Language-Image Pre-training (CLIP), but existing methods require substantial data and computation. We identify two sources of sample inefficiency: first, projected representations can collapse toward a shared constant, weakening gradient feedback to CLIP; second, hierarchical information is underutilized, as supervision from only one or a few levels limits feedback per sample, while directly combining multiple levels can introduce conflicting gradients. We propose **S**ample-efficient **G**enerative **F**eedback (SGF) to extract more useful supervision from each training sample. Margin regularization encourages matched conditions to yield lower reconstruction losses than mismatched ones by a positive margin, preserving image-specific information. Hierarchical boosting progressively integrates coarse-to-fine supervision while retaining earlier objectives, giving them time to stabilize before finer supervision is introduced, thereby reducing cross-level interference. Experiments across six CLIP backbones show that SGF achieves state-of-the-art fine-grained perception using only 84K training samples, at least 82% fewer than prior methods, with a 6.0-7.9$\times$ training speedup over the fastest competitor, demonstrating its improved sample efficiency.

1 INTRODUCTION

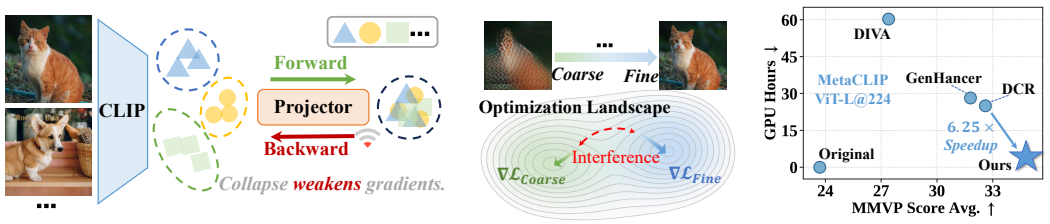
Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) underpins a wide range of visual and multimodal applications, including image classification (Zhang et al., 2022; Zhu et al., 2023; Sun et al., 2024b) and visual understanding in multimodal large language models (MLLMs) (Liu et al., 2023; 2024; Sun et al., 2024a). However, its instance-level contrastive objective can favor global semantics over fine-grained visual details (Zhou et al., 2022; Zhong et al., 2022), such as color and object count. These limitations can propagate to downstream MLLMs, constraining their vision-centric understanding capabilities (Tong et al., 2024).

Existing efforts improve CLIP's fine-grained perception through detailed vision-text supervision (Zhong et al., 2022; Li et al., 2022; Kim et al., 2023; Jing et al., 2024) or additional visual experts (Wysoczańska et al., 2024; Lan et al., 2024), which can incur substantial annotation costs or inference overhead. To avoid these costs, recent methods transfer fine-grained visual knowledge to CLIP through generative feedback (Wang et al., 2025; Ma et al., 2025; Li et al., 2025; Han et al., 2026). Specifically, a learnable projector maps CLIP representations into a pretrained generator's conditioning space, and the generator reconstructs the image from projected conditions and intermediate visual inputs. The reconstruction loss is backpropagated through the projector to refine CLIP. Despite their performance gains, these methods require substantial data and computation, sometimes including additional generator training. This raises a question:

Why do existing generative-feedback methods incur such high computational costs?

To answer this question, we examine how generative supervision is delivered to CLIP and identify two sources of sample inefficiency that increase training costs. **(1) The alignment projector is prone to representation collapse.** During projector training, only a few generation steps strongly rely on conditioning, while much of the image content can be reconstructed from the generator's

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.



(a) (b) (c)

Figure 1: (a) Projector outputs collapse, weakening gradient feedback to CLIP. (b) Supervision from coarse and fine generation levels can induce conflicting gradients on CLIP. (c) SGF achieves state-of-the-art MMVP-VLM performance with substantially lower training cost than prior methods.

learned prior and intermediate visual inputs. Plausible reconstruction is therefore possible with little image-specific conditioning, leaving little incentive to preserve inter-sample distinctions. This allows projected representations to collapse toward a shared constant, potentially weakening gradient feedback to CLIP (Fig. 1a) and increasing the samples needed for refinement. **(2) Existing methods underutilize hierarchical generative information.** Different generation levels provide complementary coarse-to-fine information, yet existing methods typically supervise each image at only one or a few levels per update, limiting the supervision extracted from each sample and resulting in substantial sample inefficiency. Integrating supervision across levels can yield richer feedback per sample; however, direct integration may induce conflicting gradients on CLIP (Fig. 1b), undermining the potential gains in sample efficiency. Together, these limitations increase the number of training samples needed for effective CLIP refinement, thereby increasing computational costs.

We propose **S**ample-efficient **G**enerative **F**eedback (SGF), a framework that extracts more effective supervision from each training sample. To address representation collapse in the projector, we introduce **Margin Regularization** during projector training, encouraging matched conditions to yield lower reconstruction losses than mismatched ones by a positive margin. This encourages image-specific conditioning, alleviating collapse and strengthening feedback from each sample. To address the underutilization of hierarchical information, we propose **Hierarchical Boosting**. Instead of relying on only one or a few generation levels per image and update, it progressively incorporates feedback from coarse to fine while retaining previously activated objectives, giving earlier objectives time to stabilize before finer supervision is introduced. This allows each sample to contribute richer multi-level supervision while reducing cross-level interference, thereby improving sample efficiency.

We primarily implement SGF with a visual autoregressive (VAR) model for its high-quality generation and parallel coarse-to-fine supervision. Across six CLIP backbones, SGF achieves state-of-the-art fine-grained perception using only 84K training samples, at least 82% fewer than prior methods, with a 6.0-7.9× training speedup over the fastest competitor (see Fig. 1c for a representative comparison). These results indicate that SGF extracts more useful supervision from each training sample. We further analyze individual components and validate the framework's applicability to diffusion-based generation.

Our main contributions are summarized as follows:

- We identify and analyze two sources of sample inefficiency in existing generative-feedback methods: representation collapse and underutilized hierarchical information.
- We propose SGF, combining margin regularization and hierarchical boosting to improve supervision from each training sample, with theoretical and empirical support.
- Across six CLIP backbones, SGF achieves state-of-the-art fine-grained perception using at least 82% fewer training samples while substantially reducing training costs. Its effectiveness also extends to diffusion-based generation.

2 Related Work

CLIP Models and MLLMs. CLIP (Radford et al., 2021) has become a foundational vision-language model, inspiring numerous follow-up studies that improve its pre-training techniques and

model architectures (Fang et al., 2024; Sun et al., 2023; Xu et al., 2024; Zhai et al., 2023). Benefiting from its strong semantic alignment, CLIP has been widely used in various vision and vision-language tasks, such as object detection (Subramanian et al., 2022; Mu et al., 2022; Zhong et al., 2022; Kim et al., 2023), image segmentation (Zhou et al., 2022; Xu et al., 2022; Xie et al., 2022; Liang et al., 2023), video understanding (Xu et al., 2021; Tang et al., 2021; Lin et al., 2022; Bose et al., 2023), and text-to-image generation (Rombach et al., 2022; Podell et al., 2023; Esser et al., 2024; Zhang et al., 2024). Moreover, CLIP has promoted the development of MLLMs, where it is commonly adopted as the visual encoder to bridge visual inputs and language models (Liu et al., 2023; 2024; Sun et al., 2024a). However, since CLIP is trained with image-level contrastive objectives, it tends to capture global semantics while overlooking visual details, resulting in limited fine-grained perception abilities (Zhou et al., 2022; Zhong et al., 2022). Such limitations can be inherited by CLIP-based MLLMs and further affect their performance (Tong et al., 2024).

Generative Models for Representation Learning. Generative models have recently been explored to improve visual representation learning. One line of work uses generative models as data augmenters to synthesize additional training data (Shipard et al., 2023; Li et al., 2024b; Tian et al., 2023). Another line leverages generative models for self-supervised reconstruction, encouraging visual encoders to capture fine-grained semantics through generative feedback (Wei et al., 2023; Hudson et al., 2024; Chen et al., 2025). Since CLIP suffers from weak fine-grained visual perception, recent studies further introduce generative reconstruction to enhance CLIP representations (Wang et al., 2025; Ma et al., 2025; Li et al., 2025; Han et al., 2026). Although these methods improve CLIP's fine-grained understanding, they require substantial data and computation.

Visual Autoregressive Models. Visual autoregressive modeling (VAR) (Tian et al., 2024) reformulates autoregressive image generation as coarse-to-fine next-scale prediction rather than next-token prediction. This paradigm has been adopted by conditional generators such as Infinity (Han et al., 2025) and VAR-CLIP (Zhang et al., 2024). We choose a VAR generator for its high-quality generation and natural support for parallel supervision across coarse-to-fine scales during training.

3 PRELIMINARY

We first introduce the generative feedback framework in Sec. 3.1, then describe the conditional generator used in our main implementation in Sec. 3.2.

3.1 GENERATIVE FEEDBACK FOR CLIP ENHANCEMENT

Given an image $\mathbf{I}$, the CLIP visual encoder E_ϕ (Radford et al., 2021) produces visual tokens $\mathbf{v}$, which a learnable projector P_ω maps into the conditioning space of a frozen generative model G:

$$\mathbf{v} = E_\phi(\mathbf{I}) \in \mathbb{R}^{L_v \times d_v}, \qquad \mathbf{c} = P_\omega(\mathbf{v}). \tag{1}$$

Here, ϕ denotes the visual encoder parameters, while L_v and d_v denote the token count and embedding dimension, respectively.

The generator reconstructs the image from $\mathbf{c}$ and intermediate visual inputs, providing feedback at K generation levels with different visual granularities. With G fixed, let $\ell_k(\mathbf{I}, \mathbf{c})$ denote the reconstruction loss at level k, comparing the generator's prediction with the target derived from $\mathbf{I}$. For a mini-batch of B images, let $\mathbf{c}_b = P_\omega(E_\phi(\mathbf{I}_b))$, and let the nonempty set $\mathcal{S}_b \subseteq \{1, \dots, K\}$ contain the levels selected for image $\mathbf{I}_b$ in the current update. The reconstruction objective is

$$\mathcal{L}_{\text{rec}} = \frac{1}{B} \sum_{b=1}^{B} \sum_{k \in \mathcal{S}_b} \lambda_k(\mathcal{S}_b)\, \ell_k(\mathbf{I}_b, \mathbf{c}_b), \qquad \sum_{k \in \mathcal{S}_b} \lambda_k(\mathcal{S}_b) = 1, \tag{2}$$

where the weights are nonnegative. This formulation accommodates both single-level and multi-level supervision per image and update.

Following GenHancer (Ma et al., 2025), we adopt a two-stage training procedure, with the generator frozen throughout.

Stage 1: Projector training. We freeze CLIP and train P_ω with reconstruction feedback to align CLIP features with the generator's conditioning space.

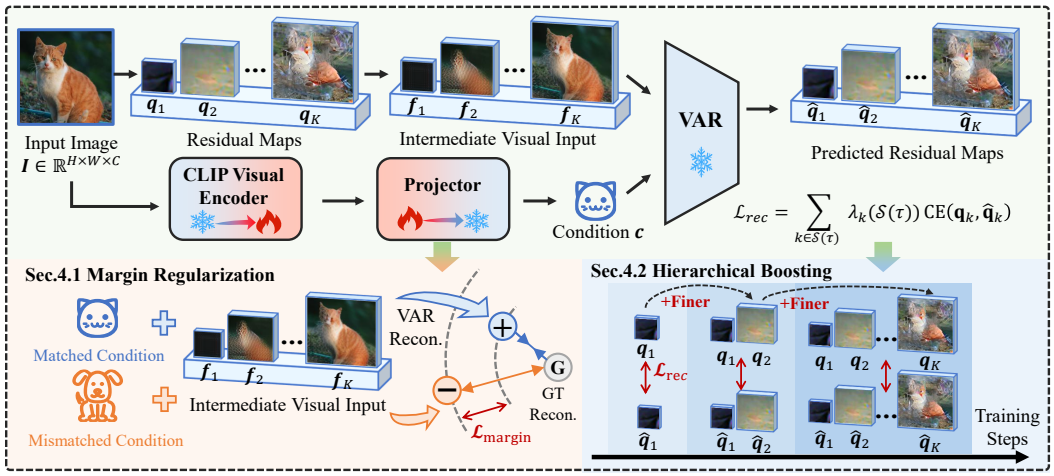


Figure 2: Overview of SGF. **Top:** A frozen VAR generator reconstructs images from projected CLIP features and intermediate visual inputs. **Bottom left:** Margin Regularization encourages matched conditions to yield lower reconstruction losses than mismatched ones by a positive margin. **Bottom right:** Hierarchical Boosting progressively introduces finer supervision.

Stage 2: CLIP refinement. We freeze the projector and fine-tune E_ϕ through LoRA, using reconstruction gradients propagated through the frozen generator and projector.

3.2 VAR Conditional Generator

We primarily instantiate the above feedback framework with Infinity (Han et al., 2025), a visual autoregressive (VAR) (Tian et al., 2024) text-to-image model, for its high-quality generation and parallel coarse-to-fine supervision. The model uses a T5-encoded text condition $\mathbf{c}_{\mathrm{T5}}$. Its visual tokenizer encodes an image $\mathbf{I}$ into K residual maps $(\mathbf{r}_1, \ldots, \mathbf{r}_K) = \mathcal{Q}(\mathbf{I})$ with progressively increasing resolutions (h_k, w_k). At scale k, the corresponding discrete labels $\mathbf{q}_k(\mathbf{I})$ serve as prediction targets.

Under teacher forcing, the scale inputs are

$$\mathbf{f}_1 = P_{\mathrm{sos}}(\mathbf{c}_{\mathrm{T5}}), \qquad \mathbf{f}_k = \sum_{i=1}^{k-1} \mathrm{resize}\big(\mathbf{r}_i, (h_k, w_k)\big), \quad k > 1, \tag{3}$$

where P_{sos} maps $\mathbf{c}_{\mathrm{T5}}$ to the initial feature of the sequence. The condition $\mathbf{c}_{\mathrm{T5}}$ also provides the keys and values for cross-attention. The VAR predicts the label distributions as

$$(\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_K) = G(\mathbf{f}_1, \ldots, \mathbf{f}_K; \mathbf{c}_{\mathrm{T5}}). \tag{4}$$

A block-wise causal mask permits attention only within the current scale and to preceding scales, enabling parallel prediction across all scales in a single teacher-forced forward pass.

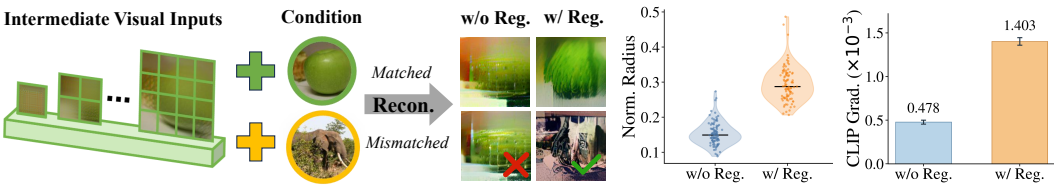
To provide reconstruction feedback to CLIP, we replace $\mathbf{c}_{\mathrm{T5}}$ with the projected condition $\mathbf{c}$ throughout the generator, including the start-of-sequence feature and cross-attention. Denoting the resulting predictions by $\hat{\mathbf{q}}_k(\mathbf{I}, \mathbf{c})$, the per-scale loss and the weights in Eq.(2) are

$$\ell_k(\mathbf{I}, \mathbf{c}) = \mathrm{CE}\big(\mathbf{q}_k(\mathbf{I}), \hat{\mathbf{q}}_k(\mathbf{I}, \mathbf{c})\big), \qquad \lambda_k(\mathcal{S}_b) = \frac{h_k w_k}{\sum_{j\in\mathcal{S}_b} h_j w_j}, \tag{5}$$

where CE denotes the mean cross-entropy at each scale, and the weights yield a token-averaged loss over the selected scales.

4 Method

We propose **S**ample-efficient **G**enerative **F**eedback (SGF) to enhance CLIP's fine-grained visual understanding with greater sample efficiency. Building on the framework in Sec. 3.1, we introduce margin regularization to counter representation collapse (Sec. 4.1), and hierarchical boosting to integrate complementary supervision with reduced cross-level interference (Sec. 4.2).



(a) Visualization of reconstruction (b) Norm. Radius (c) Gradient Strength

Figure 3: Effect of margin regularization. (a) It improves reconstruction quality and increases sensitivity to conditions. (b) It enlarges the normalized RMS radius of projected conditions, alleviating representation collapse. (c) It increases the gradient magnitude at CLIP visual features.

4.1 Representation Collapse and Margin Regularization

Representation Collapse. During projector training, only a few generation steps strongly rely on conditioning, while much of the image content can be reconstructed from the generator's learned prior and intermediate visual inputs. Plausible reconstruction is therefore possible even when projected conditions carry little image-specific information, leaving little incentive to preserve distinctions across samples. This allows the projected conditions $\mathbf{c}$ to collapse toward a shared constant.

When the projector is trained using only $\mathcal{L}_{\text{rec}}$, swapping conditions between images leaves reconstructions nearly unchanged (Fig. 3a), indicating weak reliance on image-specific conditioning. The projected conditions also exhibit a small radius (Fig. 3b), revealing limited variation across images and supporting the presence of representation collapse.

Margin Regularization. To preserve image-specific information for effective feedback, we introduce margin regularization in Stage 1. The design principle is that reconstruction should benefit from the correct image condition: matched conditions should yield lower reconstruction losses than mismatched ones by a positive margin.

We shuffle conditions within each batch, $\mathbf{c}_b^{\text{shuf}} = \mathbf{c}_{\pi(b)}$, where π is a random permutation. Each shuffled condition replaces the matched condition in all conditioning pathways, while condition-independent visual inputs and reconstruction targets remain fixed. Using the same selected levels and weights as in Eq.(2), we obtain

$$\mathcal{L}_{\text{shuf}} = \frac{1}{B}\sum_{b=1}^{B}\sum_{k\in\mathcal{S}_b}\lambda_k(\mathcal{S}_b)\,\ell_k(\mathbf{I}_b, \mathbf{c}_{\pi(b)}). \tag{6}$$

We define the reconstruction loss gap as $\delta = \mathcal{L}_{\text{shuf}} - \mathcal{L}_{\text{rec}}$ and encourage a batch-averaged gap of at least $m > 0$:

$$\mathcal{L}_{\text{margin}} = [m - \delta]_+, \tag{7}$$

where $[x]_+ = \max(x, 0)$. The Stage 1 objective is

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{margin}}\mathcal{L}_{\text{margin}}, \tag{8}$$

where λ_{margin} controls the regularization strength. More details are in Appendix D.2.

Proposition 1 links representation collapse to a reduced upper bound on effective feature gradients (formal definitions are provided in Appendix C.1), explaining how reduced distinctions across images can limit the feedback each sample provides. It also shows that satisfying a positive margin rules out constant projected conditions. Empirically, margin regularization alleviates collapse and strengthens gradient feedback to CLIP (Figs. 3b and 3c), supporting its role in improving sample efficiency.

Proposition 1 (Informal)**.** *Let ε denote the radius of projected conditions around their mean, G_c the gradient magnitude at these conditions, and G_{eff} the feature-gradient magnitude weighted by the covariance of CLIP features. Formal definitions are provided in Appendix C.1. Under a common local affine model of the projector,*

$$G_{\text{eff}} \le \varepsilon G_c. \tag{9}$$

If $L_c > 0$ bounds the sensitivity of each image's weighted reconstruction loss to changes in its condition, then

$$\varepsilon^2 \ge \frac{[\delta]_+^2}{4L_c^2}. \tag{10}$$

4.2 Underutilized Hierarchical Information and Hierarchical Boosting

Underutilized Hierarchical Information. Different generation levels provide complementary information, from coarse structures to fine details. Prior methods often supervise each image at only one or a few levels per update; single-level supervision corresponds to $\mathcal{S}_b = \{k_b\}$ in Eq.(2). This uses only part of the supervision each sample can provide, motivating the integration of multiple levels to improve sample efficiency.

Directly combining these signals, however, can introduce conflicting update directions. Let $\mathcal{L}_k(\phi) = \frac{1}{B}\sum_{b=1}^{B} \ell_k(\mathbf{I}_b, \mathbf{c}_b)$ and $\mathbf{g}_k = \nabla_\phi \mathcal{L}_k(\phi)$ denote the batch-averaged loss at level k and its gradient, where $\mathbf{c}_b = P_\omega(E_\phi(\mathbf{I}_b))$ as defined in Sec. 3.1. Gradients and updates are taken in the vectorized trainable parameter space of the encoder. For nonzero gradients, we measure their interaction by

$$S_{i,j} = \frac{\mathbf{g}_i^\top \mathbf{g}_j}{\|\mathbf{g}_i\|_2 \|\mathbf{g}_j\|_2}. \tag{11}$$

We consider the gradients to conflict when $S_{i,j} < 0$, indicating competing update directions. Fig. 4a reports the frequency of these conflicts. Realizing the benefit of richer supervision therefore also requires limiting interference between levels.

Hierarchical Boosting. To obtain richer feedback from each sample while limiting cross-level interference, we progressively expand the set of supervised levels. Intuitively, training begins with coarse supervision and gradually incorporates finer-grained signals as it progresses, while retaining all previously activated objectives. Indexing the levels from coarse to fine, every image at training step τ uses

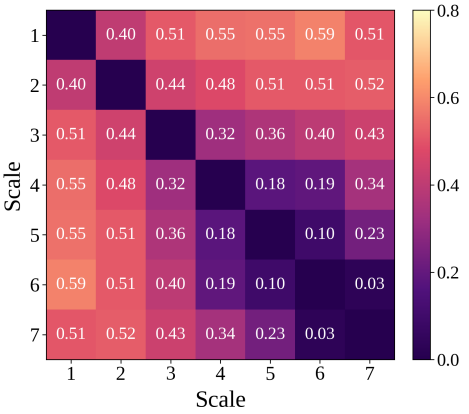
$$\mathcal{S}_b = \mathcal{S}(\tau) = \{1, \ldots, \kappa(\tau)\}, \tag{12}$$

where $\kappa(\tau)$ is nondecreasing. Substituting this shared active set into Eq.(2) gives the reconstruction objective:

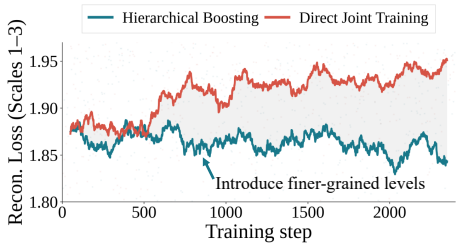
$$\mathcal{L}_{\text{rec}}(\phi; \tau) = \sum_{k \in \mathcal{S}(\tau)} \lambda_k(\mathcal{S}(\tau))\, \mathcal{L}_k(\phi). \tag{13}$$

The weights are normalized over $\mathcal{S}(\tau)$ and, for Infinity, are given by Eq.(5).

To determine when to introduce finer supervision, we use each level's information gain as a reference. We estimate this gain from the reduction in cosine distance between intermediate reconstructions and the original image, using frozen CLIP embeddings. More details are provided in Appendix D.3.



(a) Gradient conflict frequency.



(b) Reconstruction loss at scales 1-3.

Figure 4: (a) Frequency of gradient conflicts between generation levels. Higher values indicate more frequent conflicts. (b) With Hierarchical Boosting, reconstruction loss at scales 1–3 remains consistently lower.

Empirically, Hierarchical Boosting maintains lower coarse-scale losses than direct joint training, and these losses remain stable after finer supervision is introduced (Fig. 4b), suggesting reduced interference with earlier objectives. Proposition 2 provides theoretical support for this interpretation: when the accumulated objective for earlier levels has nearly stabilized, its small gradient limits the effect of a new level's update; before stabilization, the new update may cause stronger interference.

Proposition 2. *Let ϕ denote the current parameters of the CLIP visual encoder. At the activation of level s, let $A_{s-1}(\phi) = \sum_{k=1}^{s-1} \lambda_k \mathcal{L}_k(\phi)$ and $B_s(\phi) = \lambda_s \mathcal{L}_s(\phi)$ denote the weighted losses of the previous levels and the new level, respectively, with weights fixed at their values after activation. Define $\mathbf{a} = \nabla_\phi A_{s-1}(\phi)$ and $\mathbf{b} = \nabla_\phi B_s(\phi)$. Suppose A_{s-1} is β_A-smooth and $\|\mathbf{a}\|_2, \|\mathbf{b}\|_2 \le M$. For fixed $\rho > 0$ and $0 < \eta \le 1$, the marginal interference*

$$\mathcal{I}_s = A_{s-1}\big(\phi - \eta(\mathbf{a}+\mathbf{b})\big) - A_{s-1}\big(\phi - \eta\mathbf{a}\big) \tag{14}$$

Table 1: Training efficiency across six CLIP backbone configurations. We report training sample counts (K), peak GPU memory (GB), and GPU hours (h) for each stage and overall. Our method consistently requires fewer training samples and substantially less training time.

CLIP Backbone	#Params	Method	Stage 1			Stage 2			Overall		
			Data Num.	GPU Mem.	Time	Data Num.	GPU Mem.	Time	Data Num.	GPU Mem.	Time
OpenAI ViT-L/14 @224	427.6M	DIVA	-	-	-	-	-	-	2944K	32.6GB	58.2h
		GenHancer	2560K	15.5GB	20.9h	1280K	13.7GB	6.9h	3840K	15.5GB	27.8h
		un2clip	-	-	-	-	-	-	2944K	56.5GB	125.7h
		DCR	320K	41.5GB	17.0h	147K	58.3GB	8.0h	467K	58.3GB	25.0h
		Ours	60K	20.0GB	2.7h	24K	20.0GB	1.2h	**84K**	**20.0GB**	**3.9h**
OpenAI ViT-L/14 @336	427.9M	DIVA	-	-	-	-	-	-	2944K	40.9GB	70.7h
		GenHancer	2560K	20.8GB	40.0h	1280K	23.5GB	16.6h	3840K	23.5GB	56.6h
		un2clip	-	-	-	-	-	-	2944K	64.2GB	128.2h
		DCR	320K	42.7GB	16.8h	147K	59.1GB	7.9h	467K	59.1GB	24.7h
		Ours	60K	20.0GB	2.9h	24K	20.2GB	1.2h	**84K**	**20.2GB**	**4.1h**
MetaCLIP ViT-L/14 @224	427.6M	DIVA	-	-	-	-	-	-	2944K	32.8GB	60.3h
		GenHancer	2560K	15.5GB	21.3h	1280K	13.7GB	6.9h	3840K	15.5GB	28.2h
		un2clip	-	-	-	-	-	-	-	-	-
		DCR	320K	41.5GB	17.0h	147K	59.1GB	8.0h	467K	59.1GB	25.0h
		Ours	60K	20.0GB	2.8h	24K	19.5GB	1.2h	**84K**	**20.0GB**	**4.0h**
MetaCLIP ViT-H/14 @224	986.1M	DIVA	-	-	-	-	-	-	2944K	43.1GB	67.6h
		GenHancer	2560K	17.7GB	25.9h	1280K	20.6GB	9.3h	3840K	20.6GB	35.2h
		un2clip	-	-	-	-	-	-	-	-	-
		DCR	320K	43.9GB	20.3h	147K	74.6GB	9.6h	467K	74.6GB	29.9h
		Ours	60K	22.1GB	3.1h	24K	22.7GB	1.4h	**84K**	**22.7GB**	**4.5h**
SigLIP ViT-SO/14 @224	877.4M	DIVA	-	-	-	-	-	-	2944K	38.1GB	65.3h
		GenHancer	2560K	17.6GB	22.5h	1280K	16.3GB	7.6h	3840K	17.6GB	30.1h
		un2clip	-	-	-	-	-	-	-	-	-
		DCR	320K	43.6GB	17.7h	147K	65.2GB	8.3h	467K	65.2GB	26.0h
		Ours	60K	21.8GB	2.8h	24K	21.5GB	1.2h	**84K**	**21.8GB**	**4.0h**
SigLIP ViT-SO/14 @384	878.0M	DIVA	-	-	-	-	-	-	2944K	46.4GB	77.8h
		GenHancer	2560K	21.9GB	56.1h	1280K	32.2GB	23.6h	3840K	32.2GB	79.7h
		un2clip	-	-	-	-	-	-	2944K	48.2GB	56.4h
		DCR	320K	45.4GB	21.4h	147K	76.9GB	10.1h	467K	76.9GB	31.5h
		Ours	60K	21.8GB	2.8h	24K	22.6GB	1.2h	**84K**	**22.6GB**	**4.0h**

satisfies

$$|\mathcal{I}_s| \leq \begin{cases} C_2\eta^2, & \|\mathbf{a}\|_2 \leq \rho\eta \quad \textit{(second-order bound)}, \\ C_1\eta, & \|\mathbf{a}\|_2 > \rho\eta \quad \textit{(first-order bound)}, \end{cases} \tag{15}$$

where C_1 and C_2 are independent of η and depend only on the fixed β_A, M, and ρ. Thus, when the accumulated objective for earlier levels is sufficiently close to stationarity, adding a new level has at most an $O(\eta^2)$ effect in the step size; away from this regime, $O(\eta)$ interference may remain.

5 Experiments

Our evaluation of SGF centers on three questions. **Q1: Efficiency and Effectiveness.** Does SGF make more efficient use of generative feedback to improve CLIP's fine-grained visual understanding? **Q2: Ablation.** Do the components of SGF work as intended? **Q3: Scalability.** Can SGF be extended to other conditional generative models? We answer them in Sec.5.1, Sec.5.2, and Sec.5.3.

Implementation details. We use pretrained Infinity-2B (Han et al., 2025) as the default generative model, keeping its parameters frozen throughout training. Following prior work (Wang et al., 2025; Ma et al., 2025; Li et al., 2025; Han et al., 2026), we train SGF on CC3M (Sharma et al., 2018), using 60K samples in Stage 1 and 24K in Stage 2. The main experiments use a single NVIDIA RTX 4090.

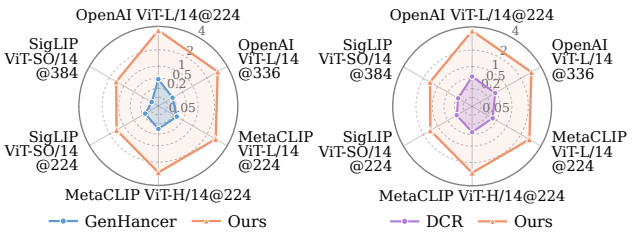


Figure 5: Compute-normalized performance gains across 6 backbones. Each axis reports the MMVP score improvement over the original backbone divided by GPU hours.

Competitors. We compare SGF with the original CLIP models and four recent generative-feedback methods: DIVA (Wang et al., 2025), GenHancer (Ma et al., 2025), un^{2}CLIP (Li et al., 2025), and DCR (Han et al., 2026). DIVA and un^{2}CLIP adopt end-to-end training, whereas others follow a two-stage framework, as does SGF.

Table 2: Performance on the MMVP-VLM benchmark. SGF matches or exceeds the best average MMVP-VLM score among the compared methods across all six backbone configurations. The visual patterns are symbolized as: 🧭: Orientation and Direction, 🔍: Presence of Specific Features, 🔄: State and Condition, 🔢: Quantity and Count, 📍: Positional and Relational Context, 🎨: Color and Appearance, ⚙️: Structural and Physical Characteristics, 🅰️: Texts, 📷: Viewpoint and Perspective.

CLIP Backbone	Resol.	#Params	Method	🧭	🔍	🔄	🔢	📍	🎨	⚙️	🅰️	📷	Avg
OpenAI ViT-L-14	224	427.6M	Original	13.3	13.3	20.0	20.0	13.3	53.3	20.0	6.7	13.3	19.2
			DIVA	13.3	20.0	40.0	6.7	20.0	53.3	46.7	20.0	13.3	25.9
			GenHancer	13.3	33.3	33.3	20.0	6.7	73.3	46.7	20.0	40.0	31.8
			un^{2}CLIP	0.0	33.3	46.7	26.7	13.3	80.0	40.0	20.0	33.3	32.6
			DCR	13.3	33.3	33.3	26.7	6.7	73.3	46.7	20.0	46.7	**33.3**
			Ours	13.3	40.0	60.0	6.7	13.3	60.0	53.3	40.0	13.3	**33.3**
OpenAI ViT-L-14	336	427.9M	Original	0.0	20.0	40.0	20.0	6.7	20.0	33.3	6.7	33.3	20.0
			DIVA	26.7	20.0	33.3	13.3	13.3	46.7	26.7	6.7	40.0	25.2
			GenHancer	6.7	20.0	33.3	20.0	6.7	73.3	53.3	26.7	26.7	29.6
			un^{2}CLIP	6.7	33.3	46.7	13.3	13.3	80.0	40.0	20.0	20.0	30.4
			DCR	13.3	26.7	33.3	20.0	6.7	73.3	46.7	33.3	26.7	31.1
			Ours	6.7	33.3	40.0	20.0	13.3	66.7	53.3	13.3	40.0	**31.9**
MetaCLIP ViT-L-14	224	427.6M	Original	13.3	6.7	66.7	6.7	33.3	46.7	20.0	6.7	13.3	23.7
			DIVA	6.7	6.7	60.0	0.0	26.7	66.7	20.0	20.0	40.0	27.4
			GenHancer	13.3	20.0	53.3	13.3	26.7	80.0	33.3	13.3	33.3	31.8
			un^{2}CLIP	-	-	-	-	-	-	-	-	-	-
			DCR	20.0	20.0	53.3	13.3	26.7	80.0	33.3	13.3	33.3	32.6
			Ours	20.0	26.7	53.3	13.3	46.7	60.0	46.7	13.3	33.3	**34.8**
MetaCLIP ViT-H-14	224	986.1M	Original	6.7	13.3	60.0	13.3	6.7	53.3	26.7	13.3	33.3	25.2
			DIVA	13.3	20.0	53.3	33.3	13.3	66.7	33.3	13.3	40.0	31.8
			GenHancer	20.0	20.0	66.7	26.7	26.7	66.7	33.3	20.0	53.3	37.0
			un^{2}CLIP	-	-	-	-	-	-	-	-	-	-
			DCR	20.0	20.0	66.7	26.7	26.7	66.7	33.3	26.7	53.3	**37.8**
			Ours	13.3	33.3	60.0	26.7	26.7	80.0	33.3	13.3	53.3	**37.8**
SigLIP ViT-SO-14	224	877.4M	Original	26.7	20.0	53.3	40.0	20.0	66.7	40.0	20.0	53.3	37.8
			DIVA	13.3	26.7	60.0	46.7	13.3	73.3	53.3	26.7	53.3	40.7
			GenHancer	20.0	20.0	66.7	60.0	20.0	86.7	40.0	13.0	53.3	42.2
			un^{2}CLIP	-	-	-	-	-	-	-	-	-	-
			DCR	13.3	26.7	73.3	60.0	20.0	80.0	40.0	20.0	53.3	43.0
			Ours	26.7	20.0	66.7	53.3	26.7	73.3	46.7	13.3	66.7	**43.7**
SigLIP ViT-SO-14	384	878.0M	Original	20.0	26.7	60.0	33.3	13.3	66.7	33.3	26.7	53.3	37.0
			DIVA	26.7	33.3	53.3	26.7	13.3	80.0	40.0	26.7	46.7	38.5
			GenHancer	26.7	20.0	66.7	33.3	13.3	86.7	40.0	26.7	46.7	40.0
			un^{2}CLIP	20.0	20.0	60.0	46.7	26.7	73.3	40.0	26.7	60.0	41.5
			DCR	26.7	33.3	60.0	33.3	26.7	80.0	40.0	33.3	46.7	42.2
			Ours	26.7	20.0	73.3	40.0	13.3	73.3	53.3	26.7	60.0	**43.0**

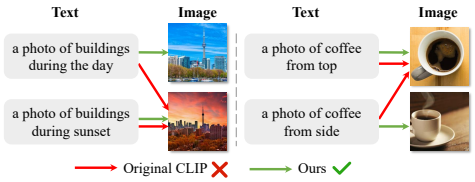


Figure 6: Qualitative results on the MMVP-VLM Benchmark.

Table 3: Performance of MLLMs on various benchmarks. The champion and runner-up results are highlighted in **bold** and underline, respectively.

CLIP	NaturalBench				POPE			Hallusion Avg.
	Acc	Q-Acc	I-Acc	G-Acc	rand	pop	adv	
Original	59.6	33.0	36.1	10.9	88.5	86.8	84.3	21.3
DIVA	65.5	36.4	**41.0**	11.7	**91.0**	88.4	**84.8**	23.6
GenHancer	63.0	28.5	31.8	8.6	90.5	88.2	84.6	24.8
DCR	65.4	36.4	40.2	11.8	90.6	88.1	84.2	24.0
Ours	**65.6**	**36.6**	40.6	**11.9**	90.6	**88.5**	84.5	**24.9**

Evaluation protocol. Following prior work (Wang et al., 2025; Ma et al., 2025; Han et al., 2026), we evaluate six CLIP backbones: OpenAI CLIP ViT-L/14@224/@336 (Radford et al., 2021), MetaCLIP ViT-L/14@224 and ViT-H/14@224 (Xu et al., 2024), and SigLIP ViT-SO/14@224/@384 (Zhai et al., 2023). For efficiency comparisons, we follow each method's official configuration and report peak GPU memory and GPU hours based on direct measurements or extrapolation from runs with smaller batch sizes on a single NVIDIA RTX 4090. The extrapolation procedure is detailed in Appendix E.3. We assess fine-grained visual perception on MMVP-VLM (Tong et al., 2024). To evaluate downstream transfer, we train MLLMs using Vicuna-7B (Zheng et al., 2023) and the enhanced OpenAI CLIP ViT-L/14@336 (Radford et al., 2021) under the same training configuration as the compared methods, and evaluate them on the benchmarks (Guan et al., 2024; Li et al., 2024a; 2023).

5.1 Efficiency and Effectiveness

We jointly assess training cost and fine-grained perception to evaluate how efficiently SGF uses generative feedback. Fig. 5 reports MMVP-VLM score gains over the original backbones per GPU hour of training. Across all six backbones, SGF achieves greater gains per GPU hour than the

compared methods. We first examine its training efficiency, then evaluate the performance achieved with this reduced training budget.

Efficiency. Tab. 1 shows that SGF substantially reduces training costs and sample requirements. Across the six backbone configurations, the full two-stage pipeline requires only 3.9-4.5 GPU hours, achieving a 6.0-7.9× training speedup over the fastest prior method for each configuration. SGF uses a total training budget of 84K samples, compared with 467K-3.84M for prior methods, a reduction of at least 82%.

Fine-grained perception. With this substantially reduced training budget, SGF achieves state-of-the-art performance on MMVP-VLM across all six CLIP backbones (Tab. 2). Specifically, on MetaCLIP ViT-L/14@224, SGF improves the MMVP-VLM score from 23.7 to 34.8. Qualitative comparisons in Fig. 6 further illustrate improved sensitivity to visual details. The MLLM results in Tab. 3 show competitive performance relative to other generative-feedback methods.

Table 4: Ablation of Margin Regularization (MR) and Hierarchical Boosting (HB).

MR	HB	Performance MMVP Score	Efficiency GPU Mem.	Efficiency GPU Hours
–	–	22.2	18.5 GB	3.3 h
✓	–	25.2	20.0 GB	4.7 h
–	✓	28.1	20.1 GB	3.4 h
✓	✓	**31.9**	20.2 GB	4.1 h

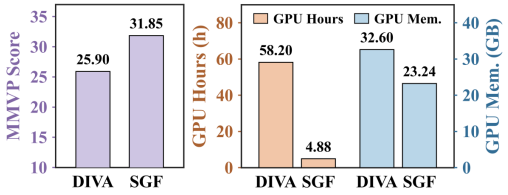


Figure 7: Application of SGF in Diffusion.

5.2 Ablation Study

We conduct all ablations on OpenAI CLIP ViT-L/14@336 (Radford et al., 2021). The naive baseline omits both margin regularization and Hierarchical Boosting, using supervision from a single generation level per image.

Effect of Margin Regularization. To verify the effect of margin regularization, we compare the normalized RMS radius and gradient magnitude propagated to CLIP features with and without it. These quantities increase to 2× (Fig. 3b) and 3× (Fig. 3c) their unregularized values, respectively, indicating reduced collapse and stronger feedback. This improved feedback translates into better fine-grained perception: Tab. 4 shows that regularization raises the MMVP-VLM score from 22.2 to 25.2, with a modest increase in training cost.

Effect of Hierarchical Boosting. We evaluate whether Hierarchical Boosting reduces cross-level interference and improves sample efficiency. Fig. 4b shows that it maintains lower coarse-scale losses than direct joint supervision across levels, even after finer supervision is introduced, suggesting reduced interference with earlier objectives. The benefits also extend to fine-grained perception: With the same number of training samples, adding Hierarchical Boosting to the naive baseline substantially improves the MMVP-VLM score (Tab. 4).

5.3 Scalability

To evaluate its applicability beyond VAR, we implement SGF with Hyper-SD (Ren et al., 2024) to enhance OpenAI CLIP ViT-L/14@224 (Radford et al., 2021). Using the same 84K-sample budget as in our main experiments, SGF requires 97.1% fewer training samples than DIVA's 2.944M, while achieving an 11.9× training speedup and lower GPU memory usage (Fig. 7). Despite this substantially smaller training budget, it outperforms DIVA on MMVP-VLM. These results demonstrate that SGF's sample efficiency and performance benefits extend to diffusion-based generative feedback.

6 Conclusion

We presented SGF to improve CLIP's fine-grained visual understanding through more effective supervision from each sample. Margin regularization alleviates representation collapse, while hierarchical boosting integrates complementary supervision with reduced interference. Experiments across six CLIP backbones demonstrate state-of-the-art performance with substantially fewer training samples and lower computational cost. These benefits also extend to diffusion-based feedback.

AI USE STATEMENT

We used AI tools to polish the manuscript, assist with theoretical derivations, and support parts of the code implementation. The authors reviewed and revised all AI-assisted text, code, and theoretical derivations, and take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

To support reproducibility, Sec. 4 describes the method, while Sec. 5 presents the evaluation benchmarks. The Appendix D and E provide detailed implementation and experimental settings, respectively. We also include the implementation code in the supplementary material to facilitate reproduction of our results.

REFERENCES

Digbalay Bose, Rajat Hebbar, Krishna Somandepalli, Haoyang Zhang, Yin Cui, Kree Cole-McLaughlin, Huisheng Wang, and Shrikanth Narayanan. Movieclip: Visual scene recognition in movies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2083–2092, 2023.

Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. In *International Conference on Learning Representations*, volume 2025, pp. 55458–55472, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385. IEEE, 2024.

Boyu Han, Qianqian Xu, Shilong Bao, Zhiyong Yang, Ruochen Cui, Xilin Zhao, and Qingming Huang. Guiding diffusion-based reconstruction with contrastive signals for balanced visual representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2369–2380, 2026.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15733–15744, 2025.

Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23115–23127, 2024.

Dong Jing, Xiaolong He, Yutian Luo, Nanyi Fei, Guoxing Yang, Wei Wei, Huiwen Zhao, and Zhiwu Lu. Fineclip: Self-distilled region-based clip for better fine-grained understanding. *Advances in Neural Information Processing Systems*, 37:27896–27918, 2024.

Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11144–11154, 2023.

Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. In *European Conference on Computer Vision*, pp. 70–88. Springer, 2024.

Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean De Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. Naturalbench: Evaluating vision-language models on natural adversarial samples. *Advances in Neural Information Processing Systems*, 37:17044–17068, 2024a.

Bohan Li, Xiao Xu, Xinghao Wang, Yutai Hou, Yunlong Feng, Feng Wang, Xuanliang Zhang, Qingfu Zhu, and Wanxiang Che. Semantic-guided generative image augmentation method with diffusion models for image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3018–3027, 2024b.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. IEEE, 2022.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023.

Yinqi Li, Jiahe Zhao, Hong Chang, Ruibing Hou, Shiguang Shan, and Xilin Chen. un2clip: Improving clip's visual detail capturing ability via inverting unclip. In *NeurIPS*, 2025.

Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7061–7070, 2023.

Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard De Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pp. 388–404. Springer, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024.

Shijie Ma, Yuying Ge, Teng Wang, Yuxin Guo, Yixiao Ge, and Ying Shan. Genhancer: Imperfect generative models are secretly strong vision-centric enhancers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24402–24412, 2025.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European conference on computer vision*, pp. 529–544. Springer, 2022.

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. *arXiv preprint arXiv:2307.01952*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *Advances in neural information processing systems*, 37:117340–117362, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 769–778, 2023.

Sanjay Subramanian, William Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. Reclip: A strong zero-shot baseline for referring expression comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5198–5215, 2022.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. In *International conference on learning representations*, volume 2024, pp. 12352–12380, 2024a.

Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13019–13029. IEEE, 2024b.

Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li. Clip4caption: Clip for video caption. In *Proceedings of the 29th ACM international conference on multimedia*, pp. 4858–4862, 2021.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *Advances in Neural Information Processing Systems*, 36:48382–48402, 2023.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9568–9578, 2024.

Wenxuan Wang, Quan Sun, Fan Zhang, Yepeng Tang, Jing Liu, and Xinlong Wang. Diffusion feedback helps clip see better. In *ICLR*, 2025.

Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16284–16294, 2023.

Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks for open-vocabulary semantic segmentation. In *European Conference on Computer Vision*, pp. 320–337. Springer, 2024.

Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Clims: Cross language image matching for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4483–4492, 2022.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6787–6800, 2021.

Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In *International Conference on Learning Representations*, volume 2024, pp. 47812–47831, 2024.

Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European conference on computer vision*, pp. 736–753. Springer, 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.

Qian Zhang, Xiangzi Dai, Ninghua Yang, Xiang An, Ziyong Feng, and Xingyu Ren. Var-clip: Text-to-image generator with visual auto-regressive modeling. *arXiv preprint arXiv:2408.01181*, 2024.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pp. 493–510. Springer, 2022.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16793–16803, 2022.

Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European conference on computer vision*, pp. 696–712. Springer, 2022.

Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2605–2615. IEEE, 2023.

Appendix

CONTENTS

A Symbol Definitions . . . 15
B Algorithm of SGF . . . 16
C Proofs of the Propositions . . . 17
C.1 Proof of Proposition 1 . . . 17
C.2 Proof of Proposition 2 . . . 18
D Additional Implementation Details . . . 20
D.1 Generator Configuration and Two-Stage Training . . . 20
D.2 Margin Regularization . . . 20
D.3 Hierarchical Boosting . . . 20
D.4 Extension to Diffusion . . . 21
D.4.1 Symbol Definitions for SGF with Diffusion . . . 21
D.4.2 Training Implementation . . . 21
D.4.3 Training Algorithm . . . 22
E Additional Experimental Settings . . . 23
E.1 CLIP Backbones . . . 23
E.2 Competitors . . . 23
E.3 Evaluation of Efficiency . . . 23
E.4 Evaluation of Fine-Grained Perception . . . 24
F Additional Experimental Results . . . 25
F.1 Additional Quantitative Results . . . 25
F.2 Additional Qualitative Results . . . 26
G Future Work . . . 27

A SYMBOL DEFINITIONS

Tab. 5 summarizes the main notation used throughout this work.

Table 5: Main notation used throughout this work.

Symbol	Meaning
$\mathbf{I}$	Input image.
B	Batch size.
E_ϕ	CLIP visual encoder with parameters ϕ.
P_ω	Alignment projector with parameters ω.
$\mathbf{v}$	CLIP visual tokens $E_\phi(\mathbf{I})$.
$\mathbf{c}$	Projected condition $P_\omega(\mathbf{v})$.
$\mathbf{c}_{\mathrm{T5}}$	Text condition encoded by T5.
G	Frozen conditional generator.
$\mathcal{Q}$	Frozen visual multi-scale tokenizer.
P_{sos}	Mapping from a condition to the initial sequence feature.
K	Number of generation levels.
(h_k, w_k)	Spatial resolution of scale k.
$\mathbf{r}_1, \ldots, \mathbf{r}_K$	Residual maps produced by $\mathcal{Q}(\mathbf{I})$.
$\mathbf{f}_1, \ldots, \mathbf{f}_K$	Inputs to the VAR generator under teacher forcing.
$\mathbf{q}_k(\mathbf{I})$	Discrete prediction targets at scale k.
$\widehat{\mathbf{q}}_k(\mathbf{I}, \mathbf{c})$	Predicted label distributions at scale k.
$\mathcal{S}(\tau)$	Shared active levels at training step τ.
$\lambda_k(\mathcal{S})$	Normalized supervision weight of level k within $\mathcal{S}$.
$\mathbf{g}_k$	Gradient of the level-k objective with respect to ϕ.
$S_{i,j}$	Cosine similarity between gradients $\mathbf{g}_i$ and $\mathbf{g}_j$.
π	Permutation of image indices within a batch.
m	Required positive margin between matched and mismatched reconstruction losses.
$\lambda_{\mathrm{margin}}$	Weight of margin regularization.
τ	Training step within a stage.
$\kappa(\tau)$	Finest active level at step τ in Hierarchical Boosting.

B ALGORITHM OF SGF

Algorithm 1 presents the training procedure for the VAR implementation of SGF, with notation summarized in Appendix A.

Algorithm 1 Sample-efficient Generative Feedback with a VAR generator.

Input: Dataset $\mathcal{D}$; models $E_\phi, P_\omega, G, \mathcal{Q}, P_{\text{sos}}$; resolutions $\{(h_k, w_k)\}_{k=1}^{K}$; batch size $B \geq 2$; stage lengths T_1, T_2 and schedules κ_1, κ_2; margin $m > 0$ and weight $\lambda_{\text{margin}} > 0$.
Output: Enhanced CLIP visual encoder E_ϕ.

Keep $G, \mathcal{Q}, P_{\text{sos}}$ frozen throughout training.
function RECLOSS($\{\mathbf{r}_{k,b}, \mathbf{q}_{k,b}\}, \{\mathbf{c}_b\}, \mathcal{S}$)
  $\lambda_k(\mathcal{S}) \leftarrow h_k w_k / \sum_{j \in \mathcal{S}} h_j w_j, \quad k \in \mathcal{S}.$
  $\mathbf{f}_{1,b} \leftarrow P_{\text{sos}}(\mathbf{c}_b).$
  $\mathbf{f}_{k,b} \leftarrow \sum_{i=1}^{k-1} \text{resize}(\mathbf{r}_{i,b}, (h_k, w_k)), \quad k \in \mathcal{S} \setminus \{1\}.$
  $(\widehat{\mathbf{q}}_{k,b})_{k \in \mathcal{S}} \leftarrow G((\mathbf{f}_{k,b})_{k \in \mathcal{S}}; \mathbf{c}_b).$
  $\ell_k(\mathbf{I}_b, \mathbf{c}_b) \leftarrow \text{CE}(\mathbf{q}_{k,b}, \widehat{\mathbf{q}}_{k,b}), \quad k \in \mathcal{S}.$
  return $\frac{1}{B} \sum_{b=1}^{B} \sum_{k \in \mathcal{S}} \lambda_k(\mathcal{S})\, \ell_k(\mathbf{I}_b, \mathbf{c}_b).$
end function
▷ **Stage 1: Projector training**
Freeze E_ϕ; train P_ω.
for $\tau = 1, \ldots, T_1$ **do**
  Sample $\{\mathbf{I}_b\}_{b=1}^{B} \sim \mathcal{D}; \quad \mathcal{S} \leftarrow \{1, \ldots, \kappa_1(\tau)\}.$
  $(\mathbf{r}_{1,b}, \ldots, \mathbf{r}_{K,b}) \leftarrow \mathcal{Q}(\mathbf{I}_b); \quad \mathbf{q}_{k,b} \leftarrow \mathbf{q}_k(\mathbf{I}_b).$
  $\mathbf{c}_b \leftarrow P_\omega(E_\phi(\mathbf{I}_b))$; sample a random batch permutation π.
  $\mathcal{L}_{\text{rec}} \leftarrow$ RECLOSS($\{\mathbf{r}_{k,b}, \mathbf{q}_{k,b}\}, \{\mathbf{c}_b\}, \mathcal{S}$); $\quad \mathcal{L}_{\text{stage1}} \leftarrow \mathcal{L}_{\text{rec}}.$
  if MR is enabled **then**
    $\mathcal{L}_{\text{shuf}} \leftarrow$ RECLOSS($\{\mathbf{r}_{k,b}, \mathbf{q}_{k,b}\}, \{\mathbf{c}_{\pi(b)}\}, \mathcal{S}$).
    $\delta \leftarrow \mathcal{L}_{\text{shuf}} - \mathcal{L}_{\text{rec}}; \quad \mathcal{L}_{\text{margin}} \leftarrow [m - \delta]_+.$
    $\mathcal{L}_{\text{stage1}} \leftarrow \mathcal{L}_{\text{rec}} + \lambda_{\text{margin}} \mathcal{L}_{\text{margin}}.$
  end if
  Update ω using $\nabla_\omega \mathcal{L}_{\text{stage1}}$.
end for
▷ **Stage 2: CLIP refinement**
Freeze P_ω; enable LoRA training in E_ϕ.
for $\tau = 1, \ldots, T_2$ **do**
  Sample $\{\mathbf{I}_b\}_{b=1}^{B} \sim \mathcal{D}; \quad \mathcal{S} \leftarrow \{1, \ldots, \kappa_2(\tau)\}.$
  $(\mathbf{r}_{1,b}, \ldots, \mathbf{r}_{K,b}) \leftarrow \mathcal{Q}(\mathbf{I}_b); \quad \mathbf{q}_{k,b} \leftarrow \mathbf{q}_k(\mathbf{I}_b).$
  $\mathbf{c}_b \leftarrow P_\omega(E_\phi(\mathbf{I}_b)).$
  $\mathcal{L}_{\text{rec}} \leftarrow$ RECLOSS($\{\mathbf{r}_{k,b}, \mathbf{q}_{k,b}\}, \{\mathbf{c}_b\}, \mathcal{S}$).
  Update ϕ through LoRA using $\nabla_\phi \mathcal{L}_{\text{rec}}$.
end for
return E_ϕ.

C PROOFS OF THE PROPOSITIONS

C.1 PROOF OF PROPOSITION 1

Notation. For a fixed mini-batch, vectorize the CLIP features $\mathbf{v}_b$ and projected conditions $\mathbf{c}_b = P_\omega(\mathbf{v}_b)$. Using the reconstruction loss defined in Sec. 3.1, write the weighted loss for image $\mathbf{I}_b$ as

$$\ell_b(\mathbf{c}) = \sum_{k \in \mathcal{S}_b} \lambda_k(\mathcal{S}_b)\, \ell_k(\mathbf{I}_b, \mathbf{c}). \tag{16}$$

The image, reconstruction targets, condition-independent visual inputs, selected levels, and weights are fixed when varying $\mathbf{c}$. All condition-dependent computations use the supplied condition, including the initial feature in Infinity. Thus, $\mathcal{L}_{\text{rec}} = \frac{1}{B}\sum_{b=1}^{B} \ell_b(\mathbf{c}_b)$.

Let $\bar{\mathbf{v}}$ and $\bar{\mathbf{c}}$ denote the respective batch means. Define the empirical covariances as

$$\begin{aligned} \mathbf{\Sigma}_v &= \frac{1}{B}\sum_{b=1}^{B} (\mathbf{v}_b - \bar{\mathbf{v}})(\mathbf{v}_b - \bar{\mathbf{v}})^\top, \\ \mathbf{\Sigma}_c &= \frac{1}{B}\sum_{b=1}^{B} (\mathbf{c}_b - \bar{\mathbf{c}})(\mathbf{c}_b - \bar{\mathbf{c}})^\top. \end{aligned} \tag{17}$$

The condition radius satisfies

$$\varepsilon^2 = \frac{1}{B}\sum_{b=1}^{B} \|\mathbf{c}_b - \bar{\mathbf{c}}\|_2^2 = \operatorname{tr}(\mathbf{\Sigma}_c). \tag{18}$$

At differentiable evaluation points, let

$$\begin{aligned} \mathbf{u}_b &= \nabla_{\mathbf{c}} \ell_b(\mathbf{c})|_{\mathbf{c}=\mathbf{c}_b}, \\ \mathbf{h}_b &= \nabla_{\mathbf{v}} \ell_b(P_\omega(\mathbf{v}))|_{\mathbf{v}=\mathbf{v}_b}. \end{aligned}$$

The gradient quantities in Proposition 1 are

$$\begin{aligned} G_c^2 &= \frac{1}{B}\sum_{b=1}^{B} \|\mathbf{u}_b\|_2^2, \\ G_{\text{eff}}^2 &= \frac{1}{B}\sum_{b=1}^{B} \mathbf{h}_b^\top \mathbf{\Sigma}_v \mathbf{h}_b. \end{aligned} \tag{19}$$

For a permutation π of the batch indices, the reconstruction loss gap is

$$\delta = \frac{1}{B}\sum_{b=1}^{B} \big[\ell_b(\mathbf{c}_{\pi(b)}) - \ell_b(\mathbf{c}_b)\big] = \mathcal{L}_{\text{shuf}} - \mathcal{L}_{\text{rec}}. \tag{20}$$

Proposition 1 (Formal). *With the definitions above, suppose the projector admits a common local affine representation $P_\omega(\mathbf{v}) = \mathbf{W}\mathbf{v} + \mathbf{d}$ on a neighborhood of the batch features. Then*

$$G_{\text{eff}} \le \varepsilon G_c.$$

If each weighted reconstruction loss ℓ_b is L_c-Lipschitz in its condition on a domain containing the batch conditions, with $L_c > 0$, then

$$\varepsilon^2 \ge \frac{[\delta]_+^2}{4L_c^2}.$$

The radius bound holds for every batch permutation and does not require the affine assumption. In particular, satisfying a positive margin $\delta \ge m > 0$ implies $\varepsilon \ge m/(2L_c) > 0$.

Proof. For any constant vector $\mathbf{z}$, the centered cross term vanishes, giving

$$\frac{1}{B}\sum_{b=1}^{B} \|\mathbf{c}_b - \mathbf{z}\|_2^2 = \varepsilon^2 + \|\bar{\mathbf{c}} - \mathbf{z}\|_2^2. \tag{21}$$

Consequently, ε is the root-mean-square distance to the best shared constant, attained at $\mathbf{z} = \bar{\mathbf{c}}$.

Under the common affine assumption, the same matrix $\mathbf{W}$ describes the mapping and its Jacobian at every sample. Centering and the chain rule therefore yield

$$\begin{aligned} \mathbf{\Sigma}_c &= \mathbf{W}\mathbf{\Sigma}_v\mathbf{W}^\top, \\ \mathbf{h}_b &= \mathbf{W}^\top\mathbf{u}_b. \end{aligned} \tag{22}$$

Since $\mathbf{\Sigma}_c$ is positive semidefinite,

$$\begin{aligned} G_{\text{eff}}^2 &= \frac{1}{B}\sum_{b=1}^{B}\mathbf{u}_b^\top\mathbf{\Sigma}_c\mathbf{u}_b \\ &\le \operatorname{tr}(\mathbf{\Sigma}_c)\frac{1}{B}\sum_{b=1}^{B}\|\mathbf{u}_b\|_2^2 = \varepsilon^2 G_c^2. \end{aligned} \tag{23}$$

Taking square roots proves Eq.(9). In particular, exact collapse gives $G_{\text{eff}} = 0$.

For the loss gap in Eq.(20), the Lipschitz and triangle inequalities imply

$$\begin{aligned} |\delta| &\le \frac{L_c}{B}\sum_{b=1}^{B}\|\mathbf{c}_{\pi(b)} - \mathbf{c}_b\|_2 \\ &\le \frac{L_c}{B}\sum_{b=1}^{B}\big(\|\mathbf{c}_{\pi(b)} - \bar{\mathbf{c}}\|_2 + \|\mathbf{c}_b - \bar{\mathbf{c}}\|_2\big) \\ &= \frac{2L_c}{B}\sum_{b=1}^{B}\|\mathbf{c}_b - \bar{\mathbf{c}}\|_2 \le 2L_c\varepsilon. \end{aligned} \tag{24}$$

The equality uses that π is a permutation, and the last inequality is Cauchy-Schwarz. Thus $[\delta]_+ \le 2L_c\varepsilon$, and squaring proves Eq.(10). If the margin is satisfied, then $\mathcal{L}_{\text{margin}} = 0$ and $\delta \ge m > 0$, so

$$\varepsilon \ge \frac{m}{2L_c} > 0. \tag{25}$$

Hence, constant conditions cannot satisfy a positive margin. □

Interpretation. G_{eff} weights feature gradients by the feature variation within the batch; it is not the unweighted feature-gradient norm. For bounded G_c, its upper bound decreases with the condition radius. The margin bound excludes exact collapse when the margin is satisfied, but does not by itself guarantee a positive gradient magnitude.

C.2 Proof of Proposition 2

Notation. During encoder refinement (Stage 2), fix the mini-batch and projector and write the per-level loss as

$$\mathcal{L}_k(\phi) = \frac{1}{B}\sum_{b=1}^{B}\ell_k\big(\mathbf{I}_b, P_\omega(E_\phi(\mathbf{I}_b))\big).$$

Gradients and updates are taken with respect to the trainable encoder parameters. Below, λ_k abbreviates $\lambda_k(\{1,\dots,s\})$, the normalized weight after level s is activated.

Proposition 2. *Let ϕ denote the current parameters of the CLIP visual encoder. At the activation of level s, let*

$$\begin{aligned} A_{s-1}(\phi) &= \sum_{k=1}^{s-1}\lambda_k\mathcal{L}_k(\phi), \\ B_s(\phi) &= \lambda_s\mathcal{L}_s(\phi) \end{aligned}$$

denote the weighted losses of the previous levels and the new level, respectively, with weights fixed at their values after activation. Define

$$\begin{aligned} \mathbf{a} &= \nabla_\phi A_{s-1}(\phi), \\ \mathbf{b} &= \nabla_\phi B_s(\phi). \end{aligned}$$

Suppose A_{s-1} is β_A-smooth and $\|\mathbf{a}\|_2, \|\mathbf{b}\|_2 \le M$. For fixed $\rho > 0$ and $0 < \eta \le 1$, the marginal interference

$$\mathcal{I}_s = A_{s-1}\big(\phi - \eta(\mathbf{a}+\mathbf{b})\big) - A_{s-1}\big(\phi - \eta\mathbf{a}\big)$$

satisfies

$$|\mathcal{I}_s| \le \begin{cases} C_2\eta^2, & \|\mathbf{a}\|_2 \le \rho\eta \quad \textit{(second-order bound)}, \\ C_1\eta, & \|\mathbf{a}\|_2 > \rho\eta \quad \textit{(first-order bound)}, \end{cases}$$

where C_1 and C_2 are independent of η and depend only on the fixed β_A, M, and ρ. Thus, when the accumulated objective for earlier levels is sufficiently close to stationarity, adding a new level has at most an $O(\eta^2)$ effect in the step size; away from this regime, $O(\eta)$ interference may remain.

Proof. For brevity, write $A = A_{s-1}$. By integration along the segment joining the two updated parameters,

$$\begin{aligned} \mathcal{I}_s &= -\eta \int_0^1 \nabla A\big(\phi - \eta\mathbf{a} - t\eta\mathbf{b}\big)^\top \mathbf{b}\, dt \\ &= -\eta\mathbf{a}^\top\mathbf{b} + r_s. \end{aligned} \tag{26}$$

Smoothness gives the remainder bound

$$\begin{aligned} |r_s| &\le \beta_A\eta^2\|\mathbf{b}\|_2 \int_0^1 \|\mathbf{a} + t\mathbf{b}\|_2\, dt \\ &\le \beta_A\eta^2 \left(\|\mathbf{a}\|_2\|\mathbf{b}\|_2 + \frac{1}{2}\|\mathbf{b}\|_2^2\right) \\ &\le \frac{3}{2}\beta_A M^2\eta^2. \end{aligned} \tag{27}$$

Therefore,

$$|\mathcal{I}_s| \le \eta\|\mathbf{a}\|_2\|\mathbf{b}\|_2 + \frac{3}{2}\beta_A M^2\eta^2. \tag{28}$$

If $\|\mathbf{a}\|_2 \le \rho\eta$, then

$$|\mathcal{I}_s| \le \left(\rho M + \frac{3}{2}\beta_A M^2\right)\eta^2 = C_2\eta^2. \tag{29}$$

Otherwise, using $\|\mathbf{a}\|_2, \|\mathbf{b}\|_2 \le M$ and $0 < \eta \le 1$ gives

$$\begin{aligned} |\mathcal{I}_s| &\le M^2\eta + \frac{3}{2}\beta_A M^2\eta^2 \\ &\le \left(1 + \frac{3}{2}\beta_A\right) M^2\eta = C_1\eta. \end{aligned} \tag{30}$$

The resulting constants C_1 and C_2 depend only on the fixed quantities β_A, M, and ρ, and are independent of η. This proves Eq.(15). □

Interpretation. $\mathcal{I}_s$ isolates the additional effect of the new level's update on the previous objectives: positive values indicate a larger previous-level loss than under their own update alone. If $\mathbf{a}^\top\mathbf{b} < 0$ at a fixed parameter point, Eq.(26) has a positive first-order term, which dominates for sufficiently small η. Near stationarity, the condition $\|\mathbf{a}\|_2 \le \rho\eta$ instead bounds the additional effect at second order; outside this regime, the general bound is first order, without asserting that interference must be harmful or attain that order. The result supports allowing earlier objectives to stabilize, but does not assert that the activation schedule necessarily reaches stationarity.

D Additional Implementation Details

D.1 Generator Configuration and Two-Stage Training

Both training stages obtain reconstruction supervision from a frozen Infinity-2B (Han et al., 2025) generator. We use its 512×512 configuration, which comprises ten scales: $(1,1)$, $(2,2)$, $(4,4)$, $(6,6)$, $(8,8)$, $(12,12)$, $(16,16)$, $(20,20)$, $(24,24)$, and $(32,32)$. We find that the first seven scales already provide effective supervision for CLIP, while subsequent scales mainly refine textures with limited additional semantic information. We therefore restrict supervision to the first seven scales in both stages, with $\mathcal{S}(\tau) \subseteq \{1,\dots,7\}$. Tab. 6 summarizes the training configurations.

Stage 1: Projector training. Following GenHancer (Ma et al., 2025) and DCR (Han et al., 2026), we use only the global visual token, corresponding to the `[CLS]` token when available. We repeat this token to the conditioning sequence length L before passing it to P_ω. The projector is a two-layer MLP with GELU between the linear layers and LayerNorm applied before the first and after the second linear layer. It maps each token to the condition dimension C, producing $\mathbf{c} \in \mathbb{R}^{L\times C}$. Hierarchical Boosting is also enabled during projector training. We train for 15,000 parameter updates with 4 samples per update, processing 60K training samples.

Stage 2: CLIP refinement. We use LoRA with rank 16 to adapt the CLIP visual encoder. We train for 6,000 parameter updates with 4 samples per update, processing 24K training samples.

Table 6: Training configuration of SGF with Infinity-2B.

Setting	Stage 1	Stage 2
Learning rate	4.0×10^{-5}	1.5×10^{-5}
Batch size	4	4
Training steps	15000	6000
Training samples	$60K$	$24K$
Optimizer	Adam	Adam
Weight decay	0.01	0.01
Margin regularization	✓	–
Hierarchical Boosting	✓	✓

D.2 Margin Regularization

We detail the implementation of margin regularization during projector training. Matched and shuffled conditions are evaluated on the same images, active scales, and normalized weights. Reconstruction targets and visual inputs independent of the condition remain fixed, while all conditioning pathways, including the initial feature and cross-attention, use the condition supplied to each branch. Gradients from both branches update the projector.

In practice, we activate margin regularization only when the finest active scale is at most 5. By the time finer scales are introduced, the projector has already learned to encode image-specific information. We therefore disable regularization and omit the shuffled forward pass thereafter, avoiding its additional computational cost at finer scales.

We apply the hinge penalty to the batch-averaged loss gap, with a default margin of $m = 0.05$ and regularization weight of $\lambda_{\text{margin}} = 0.5$.

D.3 Hierarchical Boosting

We detail how the activation schedule for Hierarchical Boosting is constructed. We first estimate the additional visual information contributed by each scale to guide the allocation of training intervals. Let $\widehat{\mathbf{I}}_k$ denote the image decoded from the accumulated residuals after adding scale k. Using E_ϕ with its pretrained parameters frozen during calibration, we compute

$$\begin{aligned} d_k &= 1 - \cos\Big(E_\phi(\widehat{\mathbf{I}}_k), E_\phi(\mathbf{I})\Big), \\ \Delta_k &= d_{k-1} - d_k, \qquad k = 2,\dots,K, \end{aligned} \tag{31}$$

where cosine similarity is computed between the encoder's global image embeddings. A larger Δ_k indicates a greater contribution from scale k in CLIP's embedding space. We average these gains

over calibration images to guide the initial allocation, assigning longer training intervals to scales with larger gains.

Considering that the pretrained CLIP encoder may overlook finer visual details, we further allocate additional updates to intervals with finer supervision while keeping each stage's total training budget fixed. New scales are activated at the boundaries of the resulting intervals. Tab. 7 reports the schedule used in Stage 2.

Table 7: Hierarchical Boosting schedule. Each active set is retained until the next activation.

Training step τ	Active scales $\mathcal{S}(\tau)$
0	$\{1, 2, 3\}$
800	$\{1, 2, 3, 4, 5\}$
2,400	$\{1, 2, 3, 4, 5, 6\}$
3,600	$\{1, 2, 3, 4, 5, 6, 7\}$

D.4 Extension to Diffusion

We extend SGF to Stable Diffusion v1.5 (Rombach et al., 2022) with the Hyper-SD weights (Ren et al., 2024), enhancing OpenAI CLIP ViT-L/14@224. Its compact set of generation timesteps makes it practical to combine supervision from multiple noise levels within our GPU memory budget. We first summarize the additional notation in Sec. D.4.1, then describe the implementation in Sec. D.4.2 and provide the training algorithm in Sec. D.4.3.

D.4.1 Symbol Definitions for SGF with Diffusion

We retain the notation in Appendix A, including the image $\mathbf{I}$, visual features $\mathbf{v}$, projected condition $\mathbf{c}$, encoder E_ϕ, projector P_ω, and generator G. The dataset $\mathcal{D}$, batch size B, and training step τ also retain their meanings.

Table 8: Additional notation for the diffusion implementation of SGF.

Symbol	Meaning
$\mathbf{z}_0$	Clean latent sampled from the VAE posterior and scaled.
t	A diffusion timestep from the candidate sequence.
$\bar{\alpha}_t$	Cumulative signal coefficient at timestep t.
$\boldsymbol{\epsilon}_t$	Standard Gaussian noise sampled for timestep t.
$\mathbf{z}_t$	Noisy latent at timestep t.
$\widehat{\boldsymbol{\epsilon}}_t$	Noise predicted by the conditional UNet.

D.4.2 Training Implementation

We use Stable Diffusion v1.5 with Hyper-SD weights for 8 steps sampling. Its sampling timesteps define the candidate supervision levels:

$$(t_1, \ldots, t_K) = (999, 874, 749, 624, 499, 374, 249, 124). \tag{32}$$

Given an image $\mathbf{I}_b$, the VAE produces its clean latent $\mathbf{z}_{b,0}$ after latent scaling, while the projector maps its CLIP features to the condition $\mathbf{c}_b = P_\omega(E_\phi(\mathbf{I}_b))$. For each selected timestep t, we independently sample standard Gaussian noise $\boldsymbol{\epsilon}_{b,t}$ and construct the noisy input

$$\mathbf{z}_{b,t} = \sqrt{\bar{\alpha}_t}\, \mathbf{z}_{b,0} + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon}_{b,t}. \tag{33}$$

The conditional UNet predicts the added noise from $\mathbf{z}_{b,t}$, t, and $\mathbf{c}_b$, yielding the feedback loss

$$\ell_t(\mathbf{I}_b, \mathbf{c}_b) = \frac{1}{D_z} \left\| G(\mathbf{z}_{b,t}, t; \mathbf{c}_b) - \boldsymbol{\epsilon}_{b,t} \right\|_2^2, \tag{34}$$

where D_z is the number of latent elements. Following Sec. 3.1, reconstruction gradients first train the projector with CLIP frozen in Stage 1, then refine CLIP with the projector frozen in Stage 2.

Margin Regularization. We apply the same regularization as in the VAR implementation.

Hierarchical Boosting. In both stages, we progressively introduce supervision from higher to lower noise levels while retaining previously activated levels. At step τ of stage j, the active timesteps are $\mathcal{T}_j(\tau) = \{t_1, \ldots, t_{\kappa_j(\tau)}\} \subseteq \{t_1, \ldots, t_K\}$. All active timesteps receive equal weights, giving

$$\mathcal{L}_{\text{rec}} = \frac{1}{B|\mathcal{T}_j(\tau)|} \sum_{b=1}^{B} \sum_{t \in \mathcal{T}_j(\tau)} \ell_t(\mathbf{I}_b, \mathbf{c}_b). \tag{35}$$

The noisy inputs at different timesteps are constructed independently from the same clean latent and processed in parallel along the batch dimension. Their losses jointly contribute to each parameter update.

D.4.3 Training Algorithm

For clarity, Algorithm 2 provides the pseudocode for the diffusion implementation of SGF.

Algorithm 2 Sample-efficient Generative Feedback with a diffusion generator.

Input: Dataset $\mathcal{D}$; models E_ϕ, P_ω, G and a pretrained VAE; candidate timesteps $(t_1, \ldots, t_K)$ and coefficients $\{\bar{\alpha}_{t_k}\}_{k=1}^K$; batch size $B \geq 2$; stage lengths T_1, T_2; schedules κ_1, κ_2; margin m and weight λ_{margin}.
Output: Enhanced CLIP visual encoder E_ϕ.

Keep G and the VAE frozen throughout training.
function BuildInputs($\{\mathbf{I}_b\}_{b=1}^B, \mathcal{T}$)
  Encode each image $\mathbf{I}_b$ with the frozen VAE to obtain its clean latent $\mathbf{z}_{b,0}$.
  for all $b \in \{1, \ldots, B\}$ and $t \in \mathcal{T}$ **do**
    Independently sample $\boldsymbol{\epsilon}_{b,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D_z})$.
    $\mathbf{z}_{b,t} \leftarrow \sqrt{\bar{\alpha}_t}\, \mathbf{z}_{b,0} + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon}_{b,t}$.
  end for
  return $\mathcal{U} = \{(\mathbf{z}_{b,t}, t, \boldsymbol{\epsilon}_{b,t})\}_{b=1,\ldots,B;\, t \in \mathcal{T}}$.
end function
function RecLoss($\mathcal{U}, \{\mathbf{c}_b\}_{b=1}^B, \mathcal{T}$)
  $\widehat{\boldsymbol{\epsilon}}_{b,t} \leftarrow G(\mathbf{z}_{b,t}, t; \mathbf{c}_b)$ for all b and $t \in \mathcal{T}$, in one batched evaluation.
  return $\frac{1}{B|\mathcal{T}|D_z} \sum_{b=1}^B \sum_{t \in \mathcal{T}} \|\widehat{\boldsymbol{\epsilon}}_{b,t} - \boldsymbol{\epsilon}_{b,t}\|_2^2$.
end function
▷ **Stage 1: Projector training**
Freeze E_ϕ; train P_ω.
for $\tau = 1, \ldots, T_1$ **do**
  Sample $\{\mathbf{I}_b\}_{b=1}^B \sim \mathcal{D}$; $\mathcal{T} \leftarrow \{t_1, \ldots, t_{\kappa_1(\tau)}\}$.
  $\mathcal{U} \leftarrow$ BuildInputs($\{\mathbf{I}_b\}_{b=1}^B, \mathcal{T}$); $\mathbf{c}_b \leftarrow P_\omega(E_\phi(\mathbf{I}_b))$.
  $\mathcal{L}_{\text{rec}} \leftarrow$ RecLoss($\mathcal{U}, \{\mathbf{c}_b\}, \mathcal{T}$); $\mathcal{L}_{\text{stage1}} \leftarrow \mathcal{L}_{\text{rec}}$.
  if MR is enabled **then**
    Sample a random batch permutation π.
    $\mathcal{L}_{\text{shuf}} \leftarrow$ RecLoss($\mathcal{U}, \{\mathbf{c}_{\pi(b)}\}, \mathcal{T}$).
    $\mathcal{L}_{\text{margin}} \leftarrow [m - (\mathcal{L}_{\text{shuf}} - \mathcal{L}_{\text{rec}})]_+$.
    $\mathcal{L}_{\text{stage1}} \leftarrow \mathcal{L}_{\text{rec}} + \lambda_{\text{margin}} \mathcal{L}_{\text{margin}}$.
  end if
  Update ω using $\nabla_\omega \mathcal{L}_{\text{stage1}}$.
end for
▷ **Stage 2: CLIP refinement**
Freeze P_ω; enable LoRA training in E_ϕ.
for $\tau = 1, \ldots, T_2$ **do**
  Sample $\{\mathbf{I}_b\}_{b=1}^B \sim \mathcal{D}$; $\mathcal{T} \leftarrow \{t_1, \ldots, t_{\kappa_2(\tau)}\}$.
  $\mathcal{U} \leftarrow$ BuildInputs($\{\mathbf{I}_b\}_{b=1}^B, \mathcal{T}$); $\mathbf{c}_b \leftarrow P_\omega(E_\phi(\mathbf{I}_b))$.
  $\mathcal{L}_{\text{rec}} \leftarrow$ RecLoss($\mathcal{U}, \{\mathbf{c}_b\}, \mathcal{T}$).
  Update ϕ through LoRA using $\nabla_\phi \mathcal{L}_{\text{rec}}$.
end for
return E_ϕ.

E Additional Experimental Settings

E.1 CLIP Backbones

We evaluate six backbone configurations spanning three families of pretrained vision-language models, covering different model sizes, input resolutions, and pretraining objectives:

- **OpenAI CLIP ViT-L@224 and OpenAI CLIP ViT-L@336** (Radford et al., 2021). OpenAI CLIP jointly trains image and text encoders with a contrastive objective that aligns matched image-text pairs and separates mismatched pairs. This alignment supports zero-shot recognition through similarity between image and text embeddings.
- **MetaCLIP ViT-L@224 and MetaCLIP ViT-H@224** (Xu et al., 2024). MetaCLIP focuses on the role of data curation in CLIP pretraining. It uses metadata to filter image-text pairs and balance their distribution across visual concepts, while retaining the contrastive learning framework.
- **SigLIP ViT-SO@224 and SigLIP ViT-SO@384** (Zhai et al., 2023). SigLIP replaces the softmax-based contrastive objective with a pairwise sigmoid loss that distinguishes matched from mismatched image-text pairs without global normalization over pairwise similarities.

E.2 Competitors

We compare SGF with the original pretrained encoders and four generative-feedback methods that differ in their conditioning mechanisms, reconstruction objectives, and training procedures:

- **Original CLIPs** denotes each pretrained backbone without additional refinement. Its original image and text encoders provide the reference performance for measuring improvements from generative feedback.
- **DIVA** (Wang et al., 2025) uses a pretrained text-to-image diffusion model as a visual assistant for CLIP. Its visual dense recap strategy incorporates local patch information alongside global visual features to condition image reconstruction. The denoising loss is backpropagated to refine CLIP, while the diffusion model remains frozen. This refinement uses images without requiring paired text and is performed through end-to-end optimization.
- **GenHancer** (Ma et al., 2025) studies how conditioning and denoising configurations affect visual representation enhancement. It uses global CLIP features as conditions and trains lightweight generative models from scratch. In its two-stage procedure, the projector and denoiser are first trained with CLIP frozen; the visual encoder is then refined using reconstruction feedback. The framework supports both continuous and discrete denoising models.
- **un^{2}CLIP** (Li et al., 2025) refines CLIP by inverting an unCLIP generator, which reconstructs images directly from CLIP image embeddings. Because the generator already accepts embeddings in the corresponding CLIP space, the framework requires no additional alignment projector. During refinement, the generator remains frozen and the image encoder is optimized through the denoising objective, aiming to capture more visual details while preserving compatibility with the original text encoder.
- **DCR** (Han et al., 2026) introduces contrastive supervision into diffusion-based reconstruction to balance fine-grained perception and discriminative ability. It contrasts noise predictions obtained under conditions from the original image, an augmented view, and other images, while also incorporating the ground-truth noise as a positive target. With the diffusion model frozen, DCR first trains the alignment projector and then fixes it to refine the CLIP visual encoder.

E.3 Evaluation of Efficiency

We report training sample counts, GPU hours, and peak GPU memory. All profiling is performed on a single NVIDIA RTX 4090 with 24 GB of memory. Since some baseline methods exceed this capacity under their official configurations, we profile them at several smaller, feasible batch sizes

and extrapolate to the official batch size. Profiling and fitting are performed separately for each method, backbone, and training stage. Configurations that fit within memory are measured directly.

- **Training samples.** We report the number of training samples specified by each method's training configuration, following the official settings for the baselines. For methods with two training stages, we report the sample count for each stage and their sum.
- **GPU hours.** We measure the runtime per training step in hours, $t_{\text{step}}(B)$, at several feasible batch sizes B. We fit a linear model $\widehat{t}_{\text{step}}(B) = a_t + b_t B$, where a_t captures the fixed overhead and b_t estimates the additional runtime per sample. For the official batch size B_{off} and corresponding number of training steps U, the estimated training cost is

$$\widehat{H} = U\,\widehat{t}_{\text{step}}(B_{\text{off}}). \tag{36}$$

  This gives the estimated GPU hours on a single RTX 4090. For two-stage methods, the total cost is the sum over both stages.
- **GPU memory.** We record peak GPU memory, $M_{\text{peak}}(B)$, at the same feasible batch sizes and fit $\widehat{M}_{\text{peak}}(B) = a_M + b_M B$. The intercept captures memory usage that is approximately independent of batch size, while the slope estimates the additional memory required per sample. We evaluate this model at B_{off} to estimate peak memory under the official configuration. For two-stage methods, the overall peak is the larger of the two stage-specific values.

DCR (Han et al., 2026) constructs all image-condition pairs within each batch in both training stages. Its official implementation therefore includes a quadratic number of conditional denoising evaluations, alongside fixed overhead and operations that scale approximately linearly with batch size. To accommodate these different contributions, we fit a second-order polynomial separately for runtime and peak memory in each stage:

$$\widehat{X}_j(B) = a_{X,j} + b_{X,j} B + c_{X,j} B^2, \qquad X \in \{t_{\text{step}}, M_{\text{peak}}\}, \quad j \in \{1, 2\}. \tag{37}$$

Here, B denotes the batch size processed per GPU, and j indexes the training stage. The constant, linear, and quadratic terms accommodate the different sources of cost without assuming that the entire training process scales quadratically. The coefficients are fitted independently from the measurements at feasible batch sizes.

We further cross-check the extrapolated statistics against those reported in the original papers and find broad agreement. These comparisons serve as approximate consistency checks, with runtime differences interpreted in light of the different hardware used.

E.4 Evaluation of Fine-Grained Perception

We use MMVP-VLM (Tong et al., 2024) to assess fine-grained image-text discrimination. A pair is correct only if both image-text matches are correct. We report pair accuracy for nine visual patterns: orientation and direction; presence of specific features; state and condition; quantity and count; positional and relational context; color and appearance; structural and physical characteristics; texts; and viewpoint and perspective. The overall score is the average across these categories. Original and enhanced versions of each backbone follow the same evaluation protocol.

To assess downstream performance, we further evaluate MLLMs built with Vicuna-7B and enhanced CLIP visual encoders on the following benchmarks:

- **NaturalBench** (Li et al., 2024a) evaluates visual understanding using groups of two images and two questions whose answers differ across images, requiring attention to visual details. We report four metrics: *Acc* measures accuracy over individual image-question pairs; *Q-Acc* requires a question to be answered correctly for both images; *I-Acc* requires both questions to be answered correctly for an image; and *G-Acc* requires all four answers in a group to be correct.
- **POPE** (Li et al., 2023) evaluates object hallucination through yes-or-no questions about object presence. Negative questions concern absent objects selected randomly, by their frequency in the dataset, or by their co-occurrence with objects present in the image. We

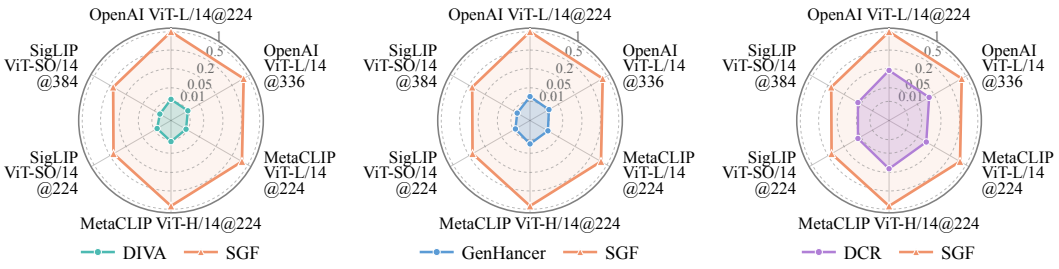


Figure 8: MMVP score gains over the original encoders per 1K training samples across six CLIP backbones. Values are normalized by the global maximum across all compared methods and backbones. Higher values indicate greater sample efficiency.

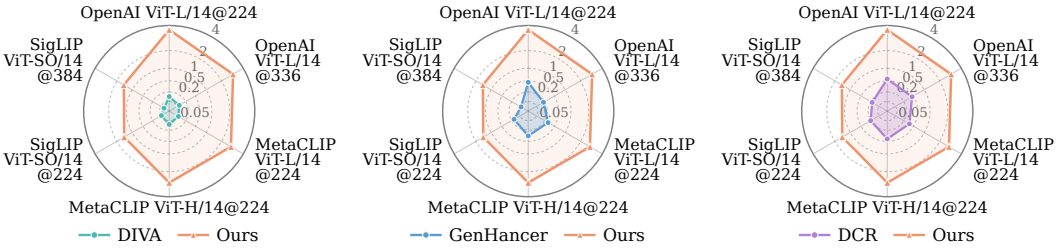


Figure 9: MMVP-VLM score gains over the original encoders per GPU hour across six CLIP backbones. Higher values indicate greater computational efficiency.

report scores for these three settings: random, popular, and adversarial, abbreviated as *rand*, *pop*, and *adv*, respectively.

- **HallusionBench** (Guan et al., 2024) evaluates visual reasoning failures involving language hallucination and visual illusion. Its image-question groups assess whether models interpret visual content correctly and respond consistently when visual evidence changes. We report the score denoted by *Hallusion Avg.* in the main paper.

F Additional Experimental Results

F.1 Additional Quantitative Results

Here, we provide additional quantitative results about SGF.

Sample efficiency. Fig. 8 compares the MMVP-VLM score gain per 1K training samples, defined as

$$\text{Gain per 1K samples} = \frac{\text{MMVP score after training} - \text{Original MMVP score}}{\text{Total training samples (K)}}. \tag{38}$$

The sample count covers the full training process, we divide these ratios by the global maximum across all compared methods and backbones. SGF achieves greater gains per training sample, demonstrating more effective use of the training data.

Computational efficiency. Fig. 9 compares the MMVP-VLM score gain per GPU hour, defined as

$$\text{Gain per GPU hour} = \frac{\text{MMVP score after training} - \text{Original MMVP score}}{\text{Total training cost (GPU hours)}}. \tag{39}$$

The training cost includes all stages. SGF achieves greater improvements per GPU hour, showing that its sample efficiency is accompanied by higher computational efficiency.

We omit un^{2}CLIP (Li et al., 2025) from both comparisons because it does not report results for MetaCLIP ViT-L/H@224 or SigLIP ViT-SO@224.

F.2 Additional Qualitative Results

Fig. 10 presents some qualitative examples from the MMVP-VLM benchmark. The results show that SGF enhances fine-grained visual perception.

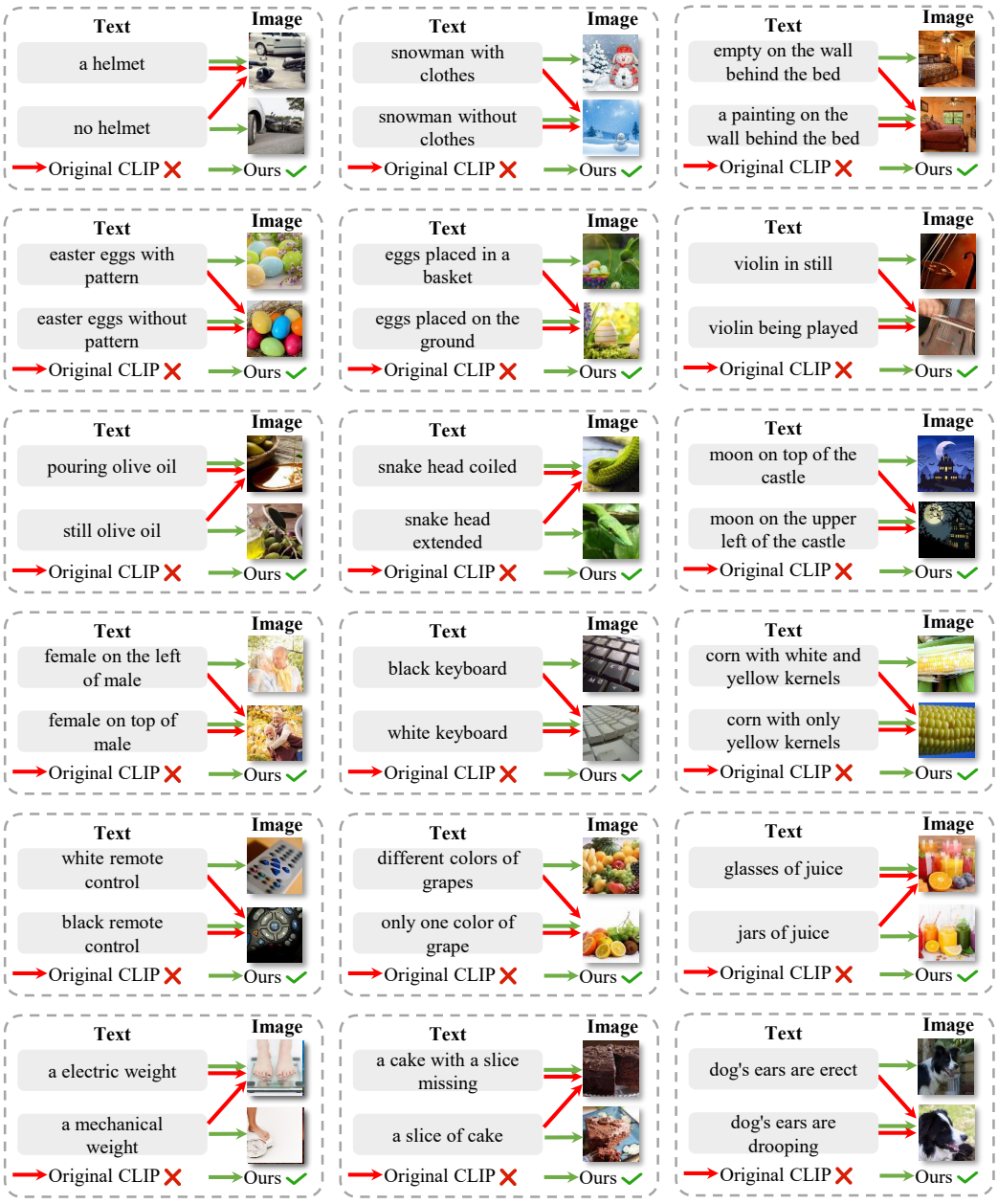


Figure 10: Expanded version of qualitative results on the MMVP-VLM benchmark. The predictions from the original CLIP and SGF improved version are indicated by red and green arrows, respectively. The improved CLIP effectively addresses the original model's limitations in capturing fine-grained visual details.

G FUTURE WORK

We plan to extend SGF to other tasks that use generative feedback to improve visual and multimodal understanding, broadening its applicability beyond CLIP refinement. We also aim to investigate whether supervision from different generation levels strengthens distinct capabilities, providing deeper insights into how generation enhances understanding.