Unifying Contrastive and Generative Objectives for Visual Understanding and Text-to-Image Generation

Chao Li[1*] **Tianhong Li**[1] **Sai Vidyaranya Nuthalapati**[2] **Hong-You Chen**[2]
Satya Narayan Shukla[2] **Jianpeng Cheng**[2] **Yonghuan Yang**[2] **Jun Xiao**[2]
Xiangjun Fan[2] **Aashu Singh**[2] **Dina Katabi**[1] **Shlok Kumar Mishra**[2]

[1]MIT Computer Science & Artificial Intelligence Laboratory [2]Meta AI

Abstract

Unifying text-image contrastive learning and text-to-image (T2I) generation in a single end-to-end model is challenging because the two objectives demand opposing masking regimes: contrastive alignment needs near-complete visible tokens, while masked generative modeling needs heavy corruption. We introduce DREAM, a unified framework that resolves this conflict through *Masking Warmup*, a schedule that shifts the center of the masking distribution over training, so low and high masking ratios coexist at every step. This co-exposure lets a single jointly trained encoder serve both objectives. The resulting stable optimization unlocks *Semantically Aligned Decoding* at inference: the text encoder, trained against visual embeddings at all masking ratios, can score partially generated images and select the best trajectory with as little as 12.5% of the image decoded, improving both FID and throughput. DREAM outperforms its single-objective baselines, CLIP and FLUID: on ImageNet linear probing (+1.1%), 5-shot transfer (+4.1%), ADE20K segmentation (+1.9%), and NYU depth estimation (+6.25%) over CLIP, and, with Semantically Aligned Decoding, on CC12M FID (+6.2%) over FLUID while maintaining CLIP Score. Together, these gains show that text-image contrastive and generative objectives, when properly unified, are synergistic rather than competing.

Code: https://github.com/chaoli-charlie/dream

1 Introduction

Multimodal learning has long been split between **models that represent** and **models that generate**. Discriminative vision–language systems such as CLIP [44] learn semantically rich features through contrastive alignment, while text-to-image (T2I) generative models, including diffusion-based [46, 49] and masked autoregressive (MAR) approaches [34, 21], learn conditional pixel distributions through aggressive corruption and reconstruction. Unifying them inside a single, end-to-end trainable network has proved persistently difficult [10, 62]. Existing unified models sidestep rather than resolve this difficulty by freezing one pathway [73], relying on external teachers [69], or decoupling the two objectives across separate training stages.

The root cause is structural: the two losses require contradictory masking regimes. Contrastive alignment needs the visible token set to carry near-complete semantics, since computing InfoNCE over heavily masked images degrades text-aligned structure [35]; masked generative modeling needs the opposite, since high masking ratios are necessary for the decoder to learn a useful conditional distribution over latent tokens [34]. In our controlled single-stage, unfrozen-encoder setting, the conflict is severe: naive joint training collapses linear probing accuracy to **4.6%** while also failing to converge on generation.

*Work done at Meta. Correspondence: chaoli@mit.edu.

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

Because the conflict originates in the masking-ratio requirements of the two objectives, the tension should be addressable at the level of the masking distribution itself, rather than through architectural workarounds that avoid co-optimization. Building on this insight, we introduce **DREAM**, a unified framework centered on **Masking Warmup**, a schedule that shifts the center of the masking distribution over training. Early on, the distribution is centered low, letting the encoder establish a stable, text-aligned representation in the regime that contrastive alignment requires. The center then shifts toward the high-mask regime the decoder needs, so these warmed-up features anchor the generative objective rather than being overwritten by it. Crucially, the distribution stays wide throughout: low and high masking ratios coexist at every step, so the encoder keeps receiving a text-alignment signal even as the decoder sees heavy corruption. This sustained co-exposure is what allows both representation and generation quality to keep improving after the schedule ends (Fig. 5).

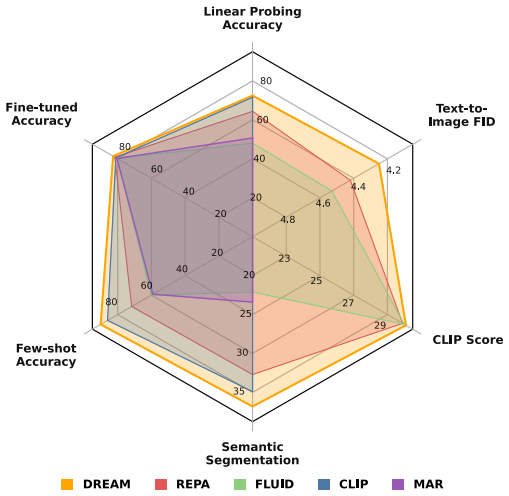


Figure 1: **Comparison of representation and T2I generation models on CC12M.** DREAM (yellow) outperforms the baselines on every discriminative and generative axis, unifying visual understanding and T2I generation in a single-stage model. Generation axes use DREAM with Semantically Aligned Decoding.

Joint optimization is not merely a training convenience; it unlocks an inference-time capability that decoupled models structurally cannot replicate. Following MAR [34], DREAM's encoder drops masked positions and operates on a sparse subset of visible tokens. Because the text encoder is jointly trained against these visual embeddings across the full range of visibility levels, it learns to align text with images that are only partially decoded. We exploit this with **Semantically Aligned Decoding (SD)**: with as little as 12.5% of the image generated, the text encoder can discriminate between candidate trajectories and select the one most aligned with the prompt. SD improves generation FID by 7.0%, evidence of the synergy that joint optimization creates between the contrastive and generative objectives. External CLIP rerankers [46, 8], trained only on fully visible images, cannot reliably score such sparse embeddings, so every candidate must be fully generated before selection. Compared with external reranking, SD improves both FID (+5.6%) and throughput (+10.1%).

Fig. 1 summarizes the gains. DREAM improves discriminative representations over CLIP, the strongest representation baseline: +1.1% linear probing on ImageNet-1K, +4.1% in 5-way 5-shot transfer, +1.9% on ADE20K, and 6.25% lower RMSE on NYU Depth v2. On generation, DREAM with SD improves T2I FID by 6.2% over FLUID on CC12M while maintaining a competitive CLIP Score, and on zero-shot MS-COCO it achieves the highest CLIP Score among all baselines. The pattern reveals two concrete forms of synergy. *Generative → spatial*: gains on dense prediction trace to the reconstruction loss, which encourages pixel-aligned features [24]. *Contrastive → selection*: the jointly trained text encoder scores partial generations far better than an external CLIP (70.8 vs. 22.2 R@1 at 12.5% visibility), which SD converts into generation gains (Appendix D.4). We summarize our contributions as follows:

- **Diagnosing the Unification Conflict at Its Source.** We identify the root cause of unstable joint training as a mismatch in the masking-ratio requirements of the two objectives.
- **Masking Warmup for Stable Single-Stage Joint Training.** We introduce *Masking Warmup*, a schedule on the masking distribution that resolves this conflict and, to our knowledge, enables the first stable single-stage end-to-end joint training of text-image contrastive and T2I generative objectives with a single architecture.
- **Discriminative and Generative Objectives Are Synergistic.** Leveraging this stable training regime, we show through extensive evaluation on classical discriminative and generation benchmarks that the two objectives are mutually reinforcing rather than competing.

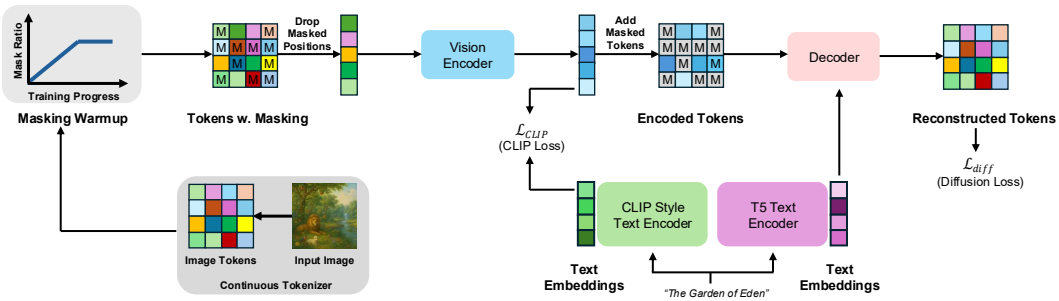


Figure 2: **DREAM training framework.** Images are encoded into continuous tokens by the Stable Diffusion VAE, then randomly masked and dropped following the Masking Warmup schedule. The vision encoder is trained contrastively with text, and the decoder conditions on text to predict the masked tokens with a diffusion-based reconstruction loss. Text conditioning is introduced only in the decoder, ensuring the encoder learns visual representations without a text shortcut.

- **Semantically Aligned Decoding.** We introduce an efficient and effective self-guided inference strategy that scores partially generated images using the model's own contrastive representations, a capability that only the unification of the two objectives enables.

2 Related Work

Unifying Representation and T2I Generation. Prior efforts fall into four groups. *Joint generative modeling without contrastive alignment.* Transfusion [75] jointly trains next-token prediction on text and diffusion on images inside one transformer, and 4M [39] performs masked token prediction across multiple modalities. These models unify generation across modalities but include no contrastive alignment objective, so the contrastive–generative conflict we address does not arise. *Multi-stage pipelines that avoid co-optimization.* Show-o [64], Janus [61], and Harmon [62] split training into multiple stages. Janus also uses separate understanding and generation encoders. *Single-stage training with intra-modal contrast.* MAGE [33] and ST-AR [70] add image–image contrastive losses to unconditional and class-conditional generation. Because their contrast is intra-modal and neither conditions on text, the text-image conflict we address does not arise. *Representation alignment to a frozen external encoder.* REPA [69] regularizes diffusion-transformer features to match a frozen pretrained encoder; RAE [73] runs diffusion directly in the latent space of a frozen pretrained encoder. In both, the alignment target is a fixed external vision model rather than a jointly trained text encoder.

Inference-Time Decoding. Scoring partial generations to prune unpromising trajectories early has emerged as an effective alternative to decoding all candidates to completion, as in future discriminators for language modeling [65], hand-designed early-timestep scores for image editing [30], and auxiliary selectors for 3D Gaussian diffusion [67]. In T2I generation, the common practice is post-hoc CLIP reranking [8, 46], which scores fully decoded candidates with an external pretrained model. All of these approaches rely on a discriminator external to the generator, and none provides a model-internal, text-aligned scorer for text-to-image generation.

3 Method

DREAM adopts a ViT-based encoder–decoder architecture over continuous image latents. The encoder learns language-aligned visual features, and the decoder generates images conditioned on text through a diffusion-based reconstruction loss. Importantly, text conditioning is applied only in the decoder, ensuring the encoder learns visual representations without relying on language shortcuts (see Fig. 2).

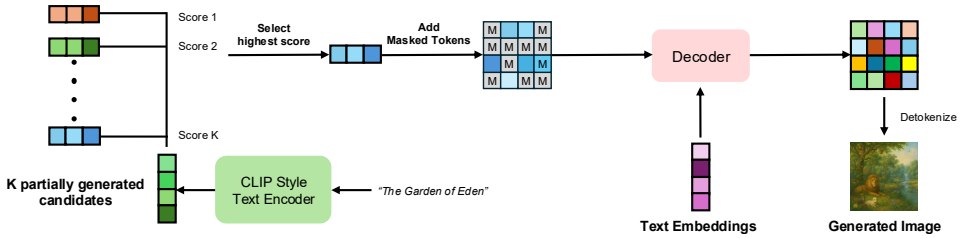


Figure 3: **Semantically Aligned Decoding.** The model spawns K parallel candidates, each partially decoded to an intermediate timestep t. The encoder scores each candidate by comparing its visual embedding to the prompt embedding, and only the top-scoring candidate is fully decoded.

3.1 Architecture

Continuous Tokenization. We encode images into continuous latent representations using the pretrained VAE encoder from Stable Diffusion [48]. This continuous tokenization preserves fine-grained spatial information while maintaining computational efficiency.

Vision Encoder. Our encoder follows MAR's architecture [34]. We prepend learnable buffer tokens to the encoder input to enhance representation capacity and training stability. The encoder processes only the buffer tokens and the unmasked tokens.

Text Encoders. We employ two text encoders with distinct roles in training. For **contrastive alignment**, captions are tokenized using the OpenAI CLIP tokenizer [44] (77 tokens) and encoded by a CLIP text transformer [57] whose features are projected into the vision encoder's latent space. For **generation**, captions are tokenized with SentencePiece [32] (128 tokens) and encoded by a frozen T5-Large [45]. A lightweight 6-layer text aligner then projects the T5 embeddings into the decoder's latent space, providing conditioning for masked autoregressive generation.

Text-to-Image Decoder. Our decoder adopts FLUID's architecture [21] and is paired with the encoder above to form a unified encoder–decoder model. Each block comprises self-attention, cross-attention, and feed-forward layers. Self-attention operates on visual tokens, while cross-attention conditions generation on text features. During training, masked visual tokens are replaced with a learnable token, and the decoder predicts these masked tokens using bidirectional attention, as in BERT [16]. The encoder and decoder each contain half of the transformer blocks. Our Large (L) configuration uses 32 layers (16 per module) and a hidden width of 1024, totaling 570M parameters in the encoder and decoder (636M including the CLIP text encoder; Table 19).

Output Head. To predict continuous latent tokens, we attach a lightweight six-layer MLP diffusion head [34] that matches the transformer's embedding dimension. The diffusion process follows the improved DDPM formulation [40, 34], using a cosine noise schedule with 1000 steps during training and 100 resampled steps during inference.

3.2 Training Objectives

Diffusion Reconstruction Loss. We adopt the diffusion loss of [34, 21], which models the conditional distribution $p(x \mid z)$ through denoising:

$$\mathcal{L}_{\text{diff}}(z, x) = \mathbb{E}_{\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t \mid t, z)\|^2\right] \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is Gaussian noise, and $x_t = \sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon$ represents the noisy latent at timestep t under variance schedule $\bar{\alpha}_t$. The denoising network ϵ_θ predicts the noise conditioned on the embeddings z, so gradients flow end to end. We sample four independent noise levels per image to improve gradient estimates without recomputing z.

Only samples with at least 50% of their tokens masked enter the diffusion loss, since high masking ratios are essential for effective generative modeling [34].

Contrastive Learning. We further incorporate CLIP-style contrastive learning [44] to align image and text representations. Given N paired samples (x_i^I, x_i^T), we compute the InfoNCE loss as:

$$\mathcal{L}_{\text{I}} = -\sum_{i=1}^{N} \log \frac{e^{\text{sim}(f_I(\text{aug}_I(x_i^I)), f_T(x_i^T))/\tau}}{\sum_{k=1}^{N} e^{\text{sim}(f_I(\text{aug}_I(x_i^I)), f_T(x_k^T))/\tau}}, \tag{2}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and τ is a learnable temperature parameter. A symmetric text-to-image term $\mathcal{L}_T$ is computed analogously, and the final contrastive objective is $\mathcal{L}_{\text{clip}} = (\mathcal{L}_I + \mathcal{L}_T)/2$.

Only samples with at most 75% of their tokens masked enter the CLIP loss, since excessive masking degrades discriminative learning [35, 33].

Joint Optimization. The total loss jointly optimizes generation and alignment:

$$\mathcal{L} = \mathcal{L}_{\text{diff}} + \lambda \cdot \mathcal{L}_{\text{clip}}, \tag{3}$$

where λ balances the diffusion and contrastive terms.

3.3 Inference

Representation Extraction. For linear probing and fine-tuning, encoder output features are globally average-pooled and fed into task-specific heads (linear probes or fine-tuned classifiers). For semantic segmentation and depth estimation, we freeze the vision encoder and follow the linear-probe protocols of DINOv3 [51] and DINOv2 [42], respectively.

Image Generation. DREAM follows MAR's next-set-of-tokens prediction strategy [34]: starting from a fully masked sequence, the masking ratio is annealed from 1.0 to 0 using a cosine schedule [7, 33] over 64 steps by default. Decoding uses temperature sampling and fully randomized token orders at every step.

3.4 Semantically Aligned Decoding

We introduce a zero-shot decoding strategy that selects the most text-aligned candidate during decoding. As illustrated in Fig. 3, given a prompt, DREAM spawns K candidates, i.e., independent decoding trajectories initialized with different random seeds. After a small fraction of the 64 decoding steps ($t \ll 64$), DREAM produces partially decoded latents for all K candidates. The vision encoder embeds these partial latents, and DREAM's contrastive text encoder scores each embedding for alignment with the prompt; the jointly trained encoder retrieves partially decoded images far more reliably than an external CLIP (Appendix E.3). Only the top-scoring candidate continues for the remaining decoding steps to reach the full image.

This approach differs from prior image selection methods [8, 46] in three ways: (1) selection occurs at the latent level; (2) only a fraction of the image is decoded, avoiding the cost of generating complete images; and (3) the model uses its own learned alignment rather than an external vision-language model.

4 Implementation Details

Data. We train on Conceptual 12M (CC12M) [9], which contains 11.3M image–text pairs. Images are center-cropped to 256×256 with horizontal flipping as augmentation [34]. We evaluate representations on ImageNet-1K [15] and generation quality on CC12M and MS-COCO [36].

Training. All models are trained for 49 epochs with AdamW [38] ($\beta_1 = 0.9$, $\beta_2 = 0.95$), a batch size of 2048, and a constant learning rate of 8×10^{-4} after a 12-epoch linear warmup. DREAM additionally employs the 36-epoch Masking Warmup described below. Evaluation uses the exponential moving average (EMA) of model weights with a decay rate of 0.9999.

Masking Warmup. The masking ratio is sampled from a truncated Gaussian distribution with a fixed standard deviation. The mean of the distribution increases linearly from 0 to 1.0 over the first 36 epochs. It is then held at 1.0, corresponding to a fully masked image.

5 Experiments

We evaluate representation quality by linear probing (LP) accuracy on ImageNet-1K [15] and text-to-image generation quality by Fréchet Inception Distance (FID) on CC12M [9]. All models are trained for 49 epochs on CC12M.

5.1 Masking Warmup Resolves the Contrastive–Generative Conflict

We first investigate how the masking distribution governs the joint optimization of the text-image contrastive and generative objectives. Across all variants, we hold architecture, training budget, and contrastive loss weight ($\lambda = 0.005$) fixed, varying only the per-step masking distribution. For this controlled study, we set $\sigma = 0.45$, which yields stable convergence (Appendix C.2).

Finding 1: Naive joint training collapses under fixed high masking. We consider a naive co-optimization implementation, **FX**, which follows FLUID's [21] fixed truncated Gaussian centered at a 1.0 masking ratio. As Table 1 shows, FX collapses the representation space relative to the single-objective models (CLIP and FLUID) and performs poorly on both representation and generation. *Under sustained high masking, the visible token set carries too little global semantics to be aligned and contrasted with text*, so the encoder–decoder pair fails to converge.

Table 1: **Component and schedule ablations.** DREAM (WM + reconstruction) achieves the best balance between representation (LP) and generation (FID). N/A: no generator (CLIP, no recon.) or generation failed to converge (FX).

	LP ↑	FID ↓
Baselines		
FLUID	48.1	**4.53**
CLIP	71.6	N/A
DREAM (varying masking schedule)		
FX	4.6	N/A
CD	**73.1**	5.80
WM (ours)	72.5	4.57
DREAM (no recon.)	70.8	N/A

Finding 2: Stable schedules trade generation for representation. We compare two alternatives: **CD**, a cooldown schedule whose mean linearly decreases from 1.0 to 0.0 over the first 36 epochs; and **WM**, a warmup schedule whose mean linearly increases from 0.0 to 1.0 over the same horizon and is held fixed afterward. **CD** and **WM** share the same distributional shape and duration and differ *only* in the direction of the schedule.

While **CD** outperforms CLIP on linear probing, it underperforms the FLUID baseline on FID (5.80 vs. 4.53). **CD** concentrates high-mask exposure early, before the encoder has established a stable text-aligned representation, and ends in a low-mask regime that provides almost no generative signal, sacrificing generation quality for representation gains.

Finding 3: Masking Warmup is synergistic. **WM** demonstrates that stable co-optimization is not only possible but synergistic: within a single jointly trained model, it surpasses CLIP on representation (72.5 vs. 71.6 LP) while matching FLUID on generation (4.57 vs. 4.53 FID). The contrast with **CD** is particularly informative: because the two schedules differ only in direction, the gap between them isolates the importance of *progression*: the contrastive objective must establish a stable semantic basin under low masking *before* the high-mask generative signal arrives. In addition, the distribution remains wide ($\sigma = 0.45$) at every step, so low- and high-mask samples coexist throughout; this preserves stable joint optimization even after warmup ends and the mean is held high (Fig. 5), since the contrastive objective continues to receive sufficiently visible image context. Appendix C.3 separates the two factors: width provides stability, while progression determines FID.

In Appendix C.5, we further distinguish Masking Warmup from explicit staged optimization: a two-stage baseline that pretrains the encoder with CLIP and then trains the generative objective with an unfrozen encoder exhibits a similar collapse regardless of stage-1 duration, confirming that the failure is structural and is resolved only by a sustained contrastive signal during high-masking generative training.

5.2 Effect of Diffusion Reconstruction

Next, we show that DREAM's linear probing gains come not from the Masking Warmup schedule alone but from the synergy between the contrastive and generative objectives that it enables. To

Table 2: **Unified evaluation of visual understanding and text-to-image generation.** All models are trained on CC12M with the same encoder–decoder architecture. *Visual understanding*: top-1 accuracy (%) of linear probing (LP) and fine-tuning (FT) on ImageNet-1K, and of the fine-tuned models on ImageNet-A/R/Sketch/Hard, ImageNetV2 (MF: MatchedFrequency, T7: Threshold0.7, Top: TopImages), and ObjectNet (ON). *Text-to-image generation*: FID and CLIP Score (CS) on CC12M (50K samples) and zero-shot MS-COCO (30K samples). ✗: no text-to-image capability. Results are from a single representative run; 3-seed means are in Table 5.

	Visual Understanding											Text-to-Image Generation			
	LP ↑	FT ↑										CC12M		MS-COCO	
Model	IN-1K	IN-1K	IN-A	IN-R	IN-S	IN-H	MF	T7	Top	ON	Avg.	FID ↓	CS ↑	FID ↓	CS ↑
MAR	50.7	80.4	20.6	44.6	29.4	21.1	69.0	77.0	81.7	37.4	51.2	✗	✗	✗	✗
FLUID	48.1	80.3	20.2	44.2	28.6	21.1	68.9	77.3	81.6	37.6	51.1	4.53	30.0	**9.62**	30.7
CLIP	71.6	81.1	24.3	53.9	40.8	23.3	69.3	77.1	81.7	37.8	54.4	✗	✗	✗	✗
REPA	62.5†	81.7	27.7	51.2	37.2	23.1	71.0	79.1	83.1	39.1	54.8	4.42	29.9	10.0	30.7
DREAM	**72.7**	**82.7**	**32.8**	**55.3**	**42.0**	**26.0**	**71.7**	**79.4**	**83.5**	**41.3**	**57.2**	4.57	29.1	10.0	30.5
+ SD	*same encoder as DREAM*											**4.25**	**30.1**	10.4	**31.5**

† Linear probing on REPA's CLIP-aligned encoder layer instead gives 64.4.

isolate this, we compare DREAM against an ablation that removes the diffusion reconstruction loss, leaving only the contrastive objective trained under the same warmup schedule.

As shown in Table 1, removing the reconstruction loss *degrades* linear probing from 72.5% to 70.8%—below even the CLIP baseline (71.6%). Without a complementary objective in the high-mask regime, the visible token set carries too little global semantics for InfoNCE to compute meaningful alignments, and the encoder's representations deteriorate as training progresses. DREAM avoids this collapse: in the high-masking regime at the end of Masking Warmup, the diffusion reconstruction loss provides spatial supervision precisely where InfoNCE's signal weakens, while contrastive alignment anchors the encoder against the representational drift that pure generative training induces.

5.3 Unified Performance

We now evaluate the synergy enabled by Masking Warmup on standard discriminative and generative benchmarks. Our comparisons target DREAM's problem formulation: single-stage, end-to-end joint training with an unfrozen encoder. Within this formulation, we compare three baseline families that share the same encoder–decoder architecture and training schedule and differ in their objective: (1) contrastive-only (CLIP); (2) generative-only (MAR, reimplemented without class conditioning; FLUID, adapted from decoder-only to encoder–decoder by adding text conditioning to the decoder); and (3) representation-aligned generative (REPA), which extends FLUID by aligning unmasked tokens at encoder layer 6 with a pretrained CLIP-L encoder [44]. REPA was originally proposed for class-conditioned diffusion [69]; we adapt it with text conditioning and use CLIP-L instead of DINOv2, which yields stronger T2I fidelity (Appendix F.1). DREAM applies Masking Warmup to jointly train FLUID's generative objective with a CLIP contrastive loss on the encoder's final layer. For these results, we report DREAM trained with $\sigma = 0.55$; ablations are in Appendix C.6.

5.3.1 Discriminative Representations

Linear Probing. Linear probing on frozen features reflects representation quality directly. As shown in Table 2, generative models, both unconditional (MAR) and text-conditioned (FLUID), learn substantially weaker representations than CLIP, while CLIP-aligned models (REPA, DREAM) close this gap. DREAM attains 72.7% accuracy, surpassing FLUID by 24.6% and exceeding CLIP by 1.1% despite sharing the same contrastive objective, indicating that coupling language-aligned contrastive learning with diffusion-based reconstruction yields more transferable representations. Our CLIP baseline on SD-VAE tokens matches the linear probing accuracy of pixel-based CLIP reported in StableRep [57], confirming that the tokenization does not degrade representation quality.

Fine-Tuning. Following [33], we fine-tune on ImageNet-1K and evaluate on out-of-domain ImageNet variants (Table 2). DREAM achieves the strongest in-domain accuracy, surpassing CLIP

Table 3: **System-level comparison with T2I and unified models.** Linear probing (LP) and fine-tuning (FT) accuracy on ImageNet-1K, and FID and CLIP Score (CS) on CC12M and zero-shot MS-COCO. "–" denotes unreported results. DREAM maintains competitive representation and generation quality against methods that use additional supervision.

	IN-1K		CC12M-50K		MS-COCO-30K	
Model	LP ↑	FT ↑	FID ↓	CS ↑	FID ↓	CS ↑
Text-to-image models						
LPL [4]	–	–	6.22	25.1	–	–
SDXL [3]	–	–	8.53	–	–	25.4[§]
mmDiT-SD3 [3]	–	–	7.54	–	–	24.8[§]
mmDiT-Imp [3]	–	–	6.79	–	–	26.6[§]
GALIP [54]	–	–	–	–	14.6	29.3
SCAD [31]	–	–	–	–	12.3	27.6
FLUID	48.1	80.3	4.53	30.0	9.62	30.7
Unified models						
Show-o-1.3B[†] [64]	–	–	–	–	9.24	–
Janus-1.3B[†] [61]	–	–	–	–	8.53	–
Transfusion-7B[†] [75]	–	–	–	–	6.78	–
4M-L[‡] [39]	–	**86.6**	(11.9[*])	18.9	30.1	23.1
REPA	62.5	81.7	4.42 (5.40[*])	29.9	10.0	30.7
DREAM + SD	**72.7**	82.7	**4.25** (**5.16**[*])	**30.1**	10.4	**31.5**

[†] Trained with extra labels or additional supervision (e.g., pretrained LLMs, larger corpora). [‡] Trained on CC12M with multimodal pseudo-labels distilled from frozen pretrained networks. [*] FID computed over 30K samples. [§] Computed over 40K samples.

by +1.6% and REPA by +1.0% on ImageNet-1K, and the highest accuracy on every out-of-domain variant, exceeding CLIP by +2.8% and REPA by +2.4% on average over all nine evaluation sets. On average, CLIP-aligned models (REPA, DREAM) outperform both CLIP and their generative-only counterparts after fine-tuning, confirming that integrating a generative objective is synergistic with learning transferable features. DREAM's gains are especially pronounced on the hardest variants (IN-A, IN-H), indicating stronger out-of-distribution generalization.

Few-Shot Learning. Prior work has established that high-quality representations are critical for few-shot image classification [60, 56, 17]. DREAM achieves the best or tied-best accuracy on 13 of 14 datasets and exceeds the next-best model (CLIP) by +4.1% on average (Appendix Table 10).

Dense Prediction. Appendix Table 11 reports dense prediction results. DREAM achieves 36.8% mIoU on ADE20K (+1.9% over CLIP) and 0.60 RMSE on NYU Depth v2, matching REPA and reducing CLIP's error by 6.25%. These gains indicate that the diffusion reconstruction objective enhances spatial grounding by encouraging pixel-aligned features that transfer to dense prediction.

Zero-shot Robustness. Because DREAM is trained with variable masking ratios, it is more robust when only a sparse subset of tokens is visible. We compare CLIP, CLIP with Masking Warmup (CLIP-M), and DREAM on zero-shot accuracy across masking levels (Appendix D.3). Masking Warmup improves robustness for both CLIP-M and DREAM over CLIP once a substantial fraction of the image is dropped. DREAM consistently outperforms CLIP-M at all levels, reflecting the benefit of locally grounded representations from diffusion-based reconstruction. At a masking ratio of 0.8, DREAM achieves about 6.2× the zero-shot accuracy of CLIP, highlighting the robustness conferred by joint contrastive–generative training.

5.3.2 T2I Generation

We evaluate DREAM on CC12M and zero-shot MS-COCO using FID and CLIP Score (CS), following Chang et al. [8]. As shown in Table 2, DREAM matches FLUID on CC12M (FID 4.57 vs. 4.53), and with SD it achieves the best FID and CS, reducing FLUID's FID by 6.2% while slightly improving CS. This indicates that coupling contrastive alignment with generation improves fidelity without compromising semantic alignment. REPA also improves over FLUID (FID 4.42) but trails DREAM + SD (4.25). On zero-shot MS-COCO, DREAM attains the highest CS (31.5) but a slightly worse FID than FLUID (10.4 vs. 9.62), a trade-off also observed for REPA. This gap reverses when training

on 100M pairs (10.24 vs. 10.54; Appendix B.2), pointing to CC12M's limited caption coverage. Over three seeds, the margins our claims rest on exceed run-to-run noise by 9–16× (Appendix B.1), and on the compositional GenEval benchmark DREAM + SD matches FLUID (0.807 vs. 0.809; Appendix E.2).

Semantically Aligned Decoding improves both fidelity and alignment. We isolate the contribution of SD by comparing DREAM with and without it under a fixed decoding budget ($T = 64$). SD improves FID by 7.0% and CS by 3.4% on CC12M, and CS by 3.3% on MS-COCO at a small cost in FID (10.0 vs. 10.4). This is direct evidence of synergy: the jointly trained contrastive encoder selects better decoding trajectories for the same generative process, actively benefiting generation rather than merely coexisting with it. Appendix E.7 shows that SD does not reduce diversity relative to external CLIP reranking.

5.3.3 Efficiency Analysis

Table 4 reports throughput, FID, and CLIP Score under matched NFE budgets. The benefit of SD decomposes into two parts:

First, scoring with the internal text encoder beats external CLIP reranking even without partial-latent selection. At a matched budget of 128 NFE, DREAM with $K = 1$ and $T = 128$ reaches FID 4.34, lower than DREAM with $K = 2$ reranked by an external CLIP (4.50). Spending the budget on more decoding steps for a single trajectory outperforms spending it on a second trajectory that an external reranker has to choose between, because the external reranker is trained only on fully visible images and provides a noisier selection signal than the model's own jointly trained text encoder.

Table 4: **Efficiency of DREAM decoding under fixed compute budgets (A100 GPU).** Throughput (images/sec), FID, and CLIP Score (CS) for varying candidate counts K, against a pretrained CLIP used as external reranker. **SD**: Semantically Aligned Decoding.

Method	Thr. ↑	FID ↓	CS ↑
64 NFE budget			
DREAM (K=1, T=64)	3.39	4.57	29.1
128 NFE budget			
DREAM (K=1, T=128)	1.77	4.34	29.4
DREAM (K=2) + Ext. CLIP†	1.69	4.50	29.6
DREAM (K=9) + SD	1.86	**4.25**	30.1
DREAM (K=17) + SD	1.85	4.37	**30.3**

† Uses an external model to select image candidates. All $K > 1$ rows use T=64.

Second, scoring at intermediate timesteps converts that advantage into a throughput win. External reranking must decode every candidate to completion before scoring, so doubling K roughly doubles compute. SD scores at $t_s = 8$ of 64 steps (12.5% decoded), so the remaining 87.5% is spent only on the surviving candidate. At $K = 9$, this yields FID 4.25 and CS 30.1, improving FID by 5.6% and CS by 1.7% over external reranking with $K = 2$ at higher throughput (1.86 vs. 1.69 img/s). Under fixed NFE, increasing K from 1 to 17 changes throughput by at most about 5%.

5.3.4 Scaling Behavior

Fig. 4 shows that DREAM scales effectively for both representation and generation. Linear probing accuracy increases monotonically with model size, following standard scaling trends. Generation quality likewise improves: FID drops from 5.67 (DREAM-B) to 3.89 (DREAM-G), matching purely generative baselines such as FLUID. From **L** upward, Semantically Aligned Decoding lets DREAM surpass FLUID by clear margins (+5.1% on **L**, +5.1% on **H**, and +6.0% on **G**); at the smallest scale (**B**), DREAM + SD only matches FLUID, and DREAM trails CLIP on linear probing.

5.3.5 Comparison with Previous Systems

We compare DREAM against recent T2I and unified multimodal models in Table 3, which groups T2I and unified models; several of them are trained on substantially larger datasets. Most prior T2I models report only generation quality. Our FLUID re-implementation on CC12M illustrates this: it achieves competitive generation (FID 4.53 on CC12M, 9.62 on MS-COCO) yet poor ImageNet-1K transfer (LP 48.1%), confirming that T2I objectives alone are insufficient for strong discriminative features.

Among unified models, 4M-L underperforms DREAM on in-domain CC12M generation (30K-sample FID 11.9 vs. 5.16) while leveraging multimodal pseudo-labels distilled from multiple frozen

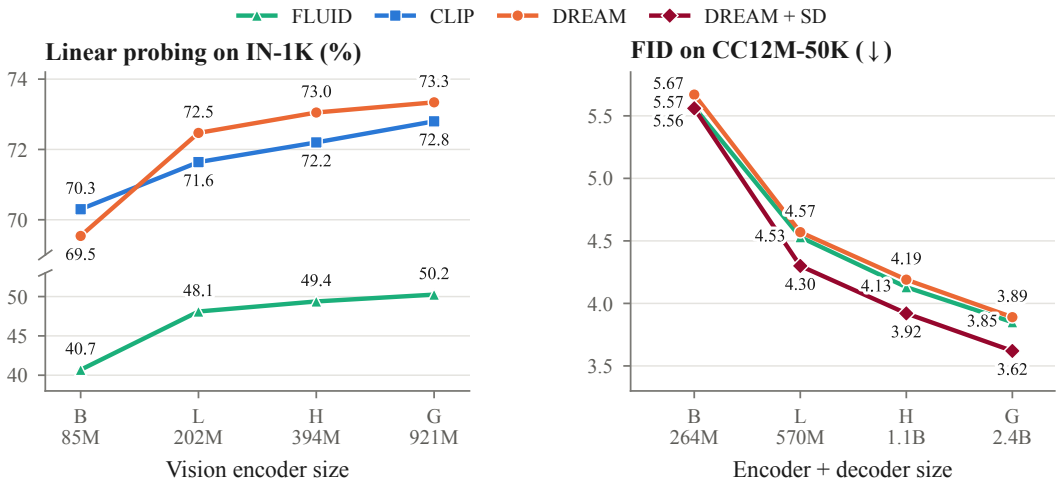


Figure 4: **Scaling DREAM from B to G** (σ=0.45)**. Left:** linear probing on IN-1K versus vision encoder size. **Right:** FID on CC12M versus encoder + decoder size; DREAM adds a 66M-parameter CLIP text encoder at every scale.

pretrained networks; REPA improves FID over FLUID by aligning with a pretrained encoder but sacrifices discriminative quality (LP 62.5% vs. 72.7%). Larger LLM-based unified models trained on substantially more data, namely Show-o-1.3B (35M images), Janus-1.3B (multi-stage, with tens of millions of generation samples), and Transfusion-7B (~3.5B images), achieve stronger MS-COCO FID than DREAM (9.24, 8.53, and 6.78 vs. 10.4), but none of them reports results on the standard discriminative transfer suite (linear probing, few-shot, segmentation, depth). The pattern is clear: existing unified models either trade one axis for the other (REPA sacrifices discriminative quality, while 4M-L lags on generation) or are evaluated only on generation (Show-o, Janus, Transfusion), leaving the trade-off implicit. DREAM is the only entry in this comparison that reports and improves on both axes simultaneously, achieved with a single-stage training recipe and a CC12M-scale corpus.

6 Conclusion

We introduced DREAM, a unified multimodal framework for visual understanding and text-to-image generation. By reconciling text-image contrastive and generative objectives through Masking Warmup during training and exploiting the resulting joint representations via Semantically Aligned Decoding at inference, we show that the two objectives are synergistic within a single end-to-end trainable architecture. We discuss limitations, including data scale and the conditions under which Masking Warmup is known to work, in Appendix A.1. We hope our work will motivate the research community to pursue end-to-end unification of representation and generation across other modalities.

Acknowledgments and Disclosure of Funding

We thank Saining Xie (NYU), Huiwen Chang (Meta), and Xuanming Xui (UCF) for their valuable discussions.

Funding. This work was funded by Meta Platforms, Inc.

Competing interests. Authors affiliated with Meta are employees of Meta, which funded this work. The authors declare no other competing interests.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[2] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

[3] Tariq Berrada, Pietro Astolfi, Melissa Hall, Reyhane Askari-Hemmat, Yohann Benchetrit, Marton Havasi, Matthew Muckley, Karteek Alahari, Adriana Romero-Soriano, Jakob Verbeek, and Michal Drozdzal. On improved conditioning mechanisms and pre-training strategies for diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[4] Tariq Berrada, Pietro Astolfi, Melissa Hall, Marton Havasi, Yohann Benchetrit, Adriana Romero-Soriano, Karteek Alahari, Michal Drozdzal, and Jakob Verbeek. Boosting latent diffusion with perceptual objectives. In *The Thirteenth International Conference on Learning Representations*, 2025.

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[6] Keshigeyan Chandrasegaran, Kyle Sargent, Suchir Agarwal, Michael Jang, Michael Poli, Juan Carlos Niebles, Justin Johnson, Jiajun Wu, and Li Fei-Fei. GPIC: A giant permissive image corpus for visual generation. *arXiv preprint arXiv:2605.30341*, 2026.

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[8] Huiwen Chang, Han Zhang, Jarred Barber, Aaron Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *PMLR*, pages 4055–4075, 2023.

[9] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

[10] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. VLP: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023.

[11] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, Oct 2017. ISSN 1558-2256. doi: 10.1109/jproc.2017.2675998. URL http://dx.doi.org/10.1109/JPROC.2017.2675998.

[12] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[13] Adam Coates, Andrew Ng, and Honglak Lee. An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *AISTATS*, 2011. https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf.

[14] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624, 2020.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. doi: 10.18653/v1/N19-1423.

[17] Guneet S. Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2020.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[19] Mohamed El Banani, Karan Desai, and Justin Johnson. Learning visual representations via language-guided sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19208–19220, 2023.

[20] 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. doi: 10.1007/s11263-009-0275-4.

[21] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. In *The Thirteenth International Conference on Learning Representations*, 2025.

[22] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 52132–52152, 2023.

[23] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech 256. CaltechDATA, Version 1.0, 2022.

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.

[25] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[27] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, 2021.

[28] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[29] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[30] Joowon Kim, Ziseok Lee, Donghyeon Cho, Sanghyun Jo, Yeonsung Jung, Kyungsu Kim, and Eunho Yang. Early timestep zero-shot candidate selection for instruction-guided image editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18844–18854, 2025.

[31] Yuya Kobayashi, Yuhta Takida, Takashi Shibuya, and Yuki Mitsufuji. Efficiency without compromise: CLIP-aided text-to-image GANs with increased diversity. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, 2025. doi: 10.1109/IJCNN64981.2025.11227151.

[32] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, 2018. doi: 10.18653/v1/D18-2012.

[33] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. MAGE: MAsked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2142–2152, 2023.

[34] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

[35] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23390–23400, 2023.

[36] 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*, pages 740–755. Springer, 2014.

[37] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[39] David Mizrahi, Roman Bachmann, Oguzhan Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4M: Massively multimodal masked modeling. *Advances in Neural Information Processing Systems*, 36:58363–58408, 2023.

[40] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[41] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[43] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pages 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a/radford21a.pdf.

[45] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[46] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

[47] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[48] 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*, pages 10684–10695, 2022.

[49] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494, 2022.

[50] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[51] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

[52] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[53] Mohammad Reza Taesiri, Giang Nguyen, Sarra Habchi, Cor-Paul Bezemer, and Anh Nguyen. ImageNet-Hard: The hardest images remaining from a study of the power of zoom and spatial biases in image classification. In *Advances in Neural Information Processing Systems*, volume 36, pages 35878–35953, 2023.

[54] Ming Tao, Bing-Kun Bao, Hao Tang, and Changsheng Xu. GALIP: Generative adversarial CLIPs for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14214–14223, 2023.

[55] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[56] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European conference on computer vision*, pages 266–282. Springer, 2020.

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

[58] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, volume 32, pages 10506–10518, 2019.

[59] Xianbang Wang, Hanhong Zhao, Yiyang Lu, Kangyang Zhou, Linrui Ma, and Kaiming He. MiniT2I: A minimalist baseline for text-to-image generation. https://peppaking8.github.io/minit2i/, 2026. Technical blog post.

[60] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens Van Der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[61] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025.

[62] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17739–17750, 2025.

[63] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, June 2010. doi: 10.1109/CVPR.2010.5539970.

[64] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[65] Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.276. URL https://aclanthology.org/2021.naacl-main.276/.

[66] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010.

[67] Zeyuan Yin and Xiaoming Liu. TRIM: Scalable 3D gaussian diffusion inference with temporal and spatial trimming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[68] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[69] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

[70] Xiaoyu Yue, Zidong Wang, Yuqing Wang, Wenlong Zhang, Xihui Liu, Wanli Ouyang, Lei Bai, and Luping Zhou. Understand before you generate: Self-guided training for autoregressive image generation. In *Advances in Neural Information Processing Systems*, volume 38, pages 143447–143466, 2025.

[71] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[72] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[73] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026.

[74] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[75] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations (ICLR)*, 2025.

A Limitations and Broader Impacts

A.1 Limitations

Data scale. Our main experiments use CC12M (∼11.3M pairs), a controlled setting chosen to isolate the effect of single-stage joint optimization with an unfrozen encoder; comparisons with state-of-the-art unified models trained on hundreds of millions of pairs (e.g., Janus, Harmon) would conflate training objective with data scale. We extend the data axis by roughly an order of magnitude with a 100M-pair run on GPIC (Appendix B.2), and the parameter axis from ∼264M to ∼2.4B parameters (Fig. 4). At 100M pairs, both conclusions hold without re-tuning; along the parameter axis, they hold from the Large scale upward, whereas at the Base scale DREAM trails CLIP on linear probing and DREAM + SD only matches FLUID on FID. The 100M-pair run covers DREAM-L only and is evaluated on linear probing and zero-shot MS-COCO generation; we did not rerun the fine-tuning, few-shot, and dense-prediction suite at that scale. Verification at the billion-pair scale remains future work.

Conditions for Masking Warmup. Our ablations identify the conditions under which the schedule is known to work: a sufficiently wide masking distribution ($\sigma \geq 0.45$; at $\sigma = 0.35$, linear probing stops improving once warmup ends and the final FID is worse, Fig. 5), and a warmup spanning a substantial fraction of training (30–42 of 49 epochs are stable, Table 9b). We have not tested regimes in which it may fail: training budgets too short for the warmup to complete, high-resolution token grids where the semantics of a given masking ratio shift, and scales at which contrastive saturation may change the appropriate loss weight λ.

Fixed components. The SD-VAE tokenizer, the diffusion head, and the frozen T5-Large conditioning encoder are shared across all compared models, so they are differenced out of our comparisons (Appendix F.2). Our design does not establish whether the conclusions hold with a different tokenizer or text encoder; that study is beyond our compute budget.

Alignment–fidelity trade-off. At CC12M scale, DREAM trails FLUID on zero-shot MS-COCO FID, and Semantically Aligned Decoding trades MS-COCO FID (+0.36) for CLIP Score (+0.97). Both effects disappear at 100M pairs (Appendix B.2), which we attribute to caption coverage, but the trade-off is not resolved at CC12M scale. On GenEval, DREAM matches FLUID overall only with Semantically Aligned Decoding (plain DREAM trails by 0.010), remains weaker on counting, position, and color attribution, and is evaluated after fine-tuning the decoder and text encoder on an external alignment mixture (Appendix E.2).

Scope of Semantically Aligned Decoding. Semantically Aligned Decoding helps when the scorer must act on partial generations under a budget too small to complete all candidates. It should not help when the budget allows completing every candidate, or when the text encoder has not been trained on partially visible inputs (Appendix E).

Statistical coverage. The headline CC12M, GPIC, and GenEval results are averaged over 3 seeds, but the ablations in Appendix C are single runs. Orderings that differ by less than about twice the run-to-run noise (Appendix B.1) should therefore be read as ties.

Computational overhead. DREAM-L incurs ∼4% additional training time and GPU memory and 13.5% additional training compute over FLUID-L, and an 11.6% parameter increase that comes entirely from the CLIP text encoder (Table 19). While this overhead is modest given the joint discriminative–generative gains, scaling down the CLIP text encoder is a natural direction for future work.

A.2 Broader Impacts

Positive Societal Impacts. By enabling end-to-end joint training without frozen encoders, DREAM reduces the engineering and compute overhead typically required to obtain both a strong vision encoder and a competitive text-to-image generator. The training-time cost over a generation-only baseline is modest (∼4% additional time and memory), which lowers the barrier for academic and

resource-constrained groups to study unified multimodal models. Improved visual representations—reflected in gains on classification, semantic segmentation, and depth estimation—can also support beneficial downstream applications such as accessibility tools, scientific image analysis, and assistive technologies.

Negative Societal Impacts. DREAM produces a high-quality text-to-image generator and therefore inherits risks shared by this class of models: generation of non-consensual or misleading imagery for disinformation, and amplification of biases present in web-scale image–text training data, which can manifest as stereotyped or uneven depictions across demographic groups. Failure modes such as hallucinated objects or attribute-binding errors may also lead users to over-trust incorrect outputs in sensitive settings (e.g., journalism, education).

B Seed Variance and Data Scale

Unless otherwise noted, every entry in this section is the mean ± sample standard deviation over 3 independent runs that vary initialization, data order, and generation noise. We quote margins as multiples (×) of the combined standard deviation of the two runs being compared; σ keeps its meaning from the main text, the width of the masking distribution.

B.1 Seed Variance

Table 5: **CC12M seed variance.** Table 2 reports a single representative run; the means below are over 3 seeds and therefore differ from it in the final digit. Semantically Aligned Decoding (SD) is inference-only and leaves the encoder unchanged, so its LP is identical to DREAM by construction.

	LP ↑	**CC12M-50K**		**MS-COCO-30K**	
Model	IN-1K	FID ↓	CS ↑	FID ↓	CS ↑
CLIP	71.53 ± 0.06	–	–	–	–
FLUID	48.15 ± 0.04	4.52 ± 0.02	30.01 ± 0.01	**9.64 ± 0.14**	30.73 ± 0.04
DREAM	**72.67 ± 0.11**	4.56 ± 0.03	29.12 ± 0.02	10.02 ± 0.11	30.49 ± 0.02
DREAM + SD	= DREAM	**4.25 ± 0.02**	**30.10 ± 0.02**	10.38 ± 0.08	**31.46 ± 0.02**

Run-to-run noise is ≈ ±0.11 on LP and ≈ ±0.03 on CC12M FID (Table 5). The comparisons that our claims rest on are separated by many standard deviations, and their order holds in all three seeds: DREAM over CLIP on LP (+1.14, ≈ 9×), DREAM + SD over FLUID on CC12M FID (−0.27, ≈ 10×), and DREAM + SD over FLUID on MS-COCO CS (+0.73, ≈ 16×). The one exception is zero-shot MS-COCO FID at CC12M scale, where DREAM trails FLUID by 0.38 (≈ 2×) and DREAM + SD by 0.74 (≈ 4.6×), consistently across seeds. We do not claim an advantage there; Appendix B.2 shows that this gap reverses with a broader training corpus.

Ordering stability of ablations. Our ablations are of two kinds. Those that establish a design choice are separated by many multiples of run-to-run noise: fixed high masking (FX) collapses LP to 4.6 against 72.5 for WM; the cooldown schedule costs 1.23 FID (5.80 vs. 4.57, ≈ 29×); removing diffusion reconstruction costs 1.7 LP (≈ 11×); applying the CLIP loss at layer 16 rather than layer 6 gains 5.9 LP (≈ 38×, Table 9g); and UNI trades 0.6 LP for 0.31 FID against $\sigma = 0.45$ (≈ 4× and ≈ 7×), a trade-off along two axes rather than a contested ranking. Sweeps over hyperparameters within the chosen design (Appendix C) instead form plateaus, which is the intended finding: they establish that Masking Warmup is stable across settings, not that one setting is optimal. Where configurations differ by less than 2× the noise (e.g., $\sigma = 0.45$ vs. 0.55: identical FID, 0.2 LP apart), no conclusion depends on which is nominally ahead.

B.2 Training at 100M Pairs

To test whether our conclusions extend beyond CC12M, we train DREAM-L and its baselines on GPIC [6], a 100M-pair corpus with more diverse captions. We train for 6 epochs (≈293K steps at batch size 2048, against ≈270K for the 49-epoch CC12M runs), so that corpus composition rather

than compute is what changes. All schedule hyperparameters are carried over from CC12M without re-tuning, with step-dependent ones expressed as a fraction of total training steps.

Table 6: **Models trained on GPIC (100M pairs).** ImageNet linear probing and zero-shot MS-COCO-30K, mean ± std over 3 seeds.

Model	LP (IN-1K) ↑	FID (COCO-30K) ↓	CS (COCO-30K) ↑
CLIP	67.58 ± 0.03	–	–
FLUID	48.90 ± 0.05	10.54 ± 0.07	31.13 ± 0.03
DREAM	**68.88 ± 0.05**	10.24 ± 0.05	31.12 ± 0.02
DREAM + SD	= DREAM	**10.22 ± 0.04**	**31.33 ± 0.01**

Both claims hold at roughly 9× the data (Table 6). DREAM exceeds CLIP on LP by 1.30, comparable to its +1.14 margin at CC12M scale. The zero-shot MS-COCO FID ordering against FLUID reverses in DREAM's favor (10.24 vs. 10.54, ≈ 3.5×) at matched CLIP Score (31.12 vs. 31.13), a swing of 0.68 FID relative to CC12M produced by changing the training corpus alone. Adding SD leaves FID unchanged within noise while raising CS by 0.21, so at this scale SD is a CS gain at no fidelity cost.

We attribute the CC12M-scale MS-COCO FID deficit to the limited caption coverage of an 11.3M-pair corpus rather than to a structural property of joint training. The deficit appears only out of domain: in domain, DREAM matches FLUID (4.56 vs. 4.52) and improves on it with SD (4.25). A structural cost of adding the contrastive objective would be visible in domain as well. Moreover, the corpus that closes the gap is the one whose captions resemble COCO's: GPIC's captions are VLM-generated scene descriptions rather than CC12M's short, entity-centric alt-text. The same substitution lowers absolute LP for CLIP and DREAM (e.g., CLIP 71.53 → 67.58; FLUID is essentially unchanged), as expected if alt-text transfers more directly to entity-level classification; the relative ordering, which is what our claims concern, is preserved.

C Masking Warmup: Analysis and Ablations

Unless otherwise noted, the ablations in this section use DREAM-L trained for 49 epochs on CC12M with masking standard deviation $\sigma = 0.45$, CLIP loss weight $\lambda = 0.005$, and generation without Semantically Aligned Decoding, matching the controlled study of Section 5.1. Default settings are highlighted in gray and the best value in each ablation is in bold. Ablations are single runs; Appendix B.1 discusses which differences exceed run-to-run noise.

C.1 Schedule Definitions

The masking schedules compared in Table 1 are defined as follows. Unless otherwise noted, all schedules use a truncated Gaussian distribution with a standard deviation of 0.45, a maximum masking ratio of 0.75 for computing the CLIP loss (i.e., 75% of tokens masked), and a minimum masking ratio of 0.5 for computing the MAR loss. The global minimum and maximum masking ratios are fixed at 0 and 1.0, respectively.

- **FX**: Fixed truncated Gaussian distribution centered at 1.0.
- **CD**: A Gaussian distribution whose mean starts at 1.0 and linearly decreases to 0.0 over the first 36 epochs, after which it remains fixed at 0.0.
- **WM**: A Gaussian distribution whose mean starts at 0.0 and linearly increases to 1.0 over the first 36 epochs, after which it remains fixed at 1.0.

C.2 Stability Across Masking Widths

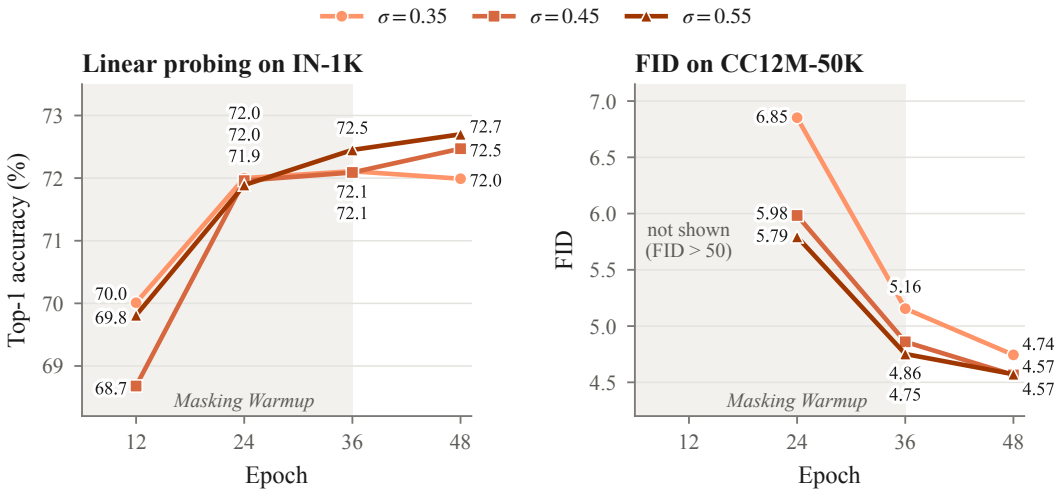


Figure 5: **Performance of DREAM across different masking standard deviations. Left:** Linear probing on IN-1K. **Right:** FID on CC12M-50K without Semantically Aligned Decoding. The x-axis gives the 0-indexed epoch of each checkpoint, so 48 is the final epoch of the 49-epoch schedule. We do not plot the FID values for epoch 12 since the training has not converged and the FID values are too high (FID > 50). The curves show that when $\sigma > 0.35$, both Linear Probing and FID performance are stable with increasing training duration even after Masking Warmup ends.

In Fig. 5, we plot Linear Probing accuracy and FID (on CC12M) across training for three masking standard deviations ($\sigma \in \{0.35, 0.45, 0.55\}$). For $\sigma \in \{0.45, 0.55\}$, both metrics improve monotonically throughout training. In contrast, with $\sigma = 0.35$, linear probing stops improving once Masking Warmup ends (72.1 → 72.0) and the final FID is worse (4.74 vs. 4.57). This suggests that stable unification requires the masking distribution to exceed a certain variance threshold. A larger standard deviation retains more lightly masked samples after Masking Warmup ends, ensuring that enough unmasked visual context is available to compute the CLIP loss and maintain discriminative performance.

Overall, these results indicate that given a well-defined masking standard deviation, Masking Warmup enables stable joint optimization of discriminative and generative objectives.

C.3 Progression versus Width of the Masking Distribution

Masking Warmup combines two factors: the *progression* of the distribution's mean from low to high masking, and the *width* of the distribution, which keeps low- and high-masked samples coexisting at every step. Read as a factorial over these two factors, our ablations separate their roles (Table 7).

Table 7: **Masking schedules as a factorial over progression and width.** Collected from Tables 1 and 9a.

Schedule	Progression	Width	LP ↑	FID ↓
UNI	none (fixed)	maximal	**73.1**	4.88
CD	high → low	$\sigma = 0.45$	**73.1**	5.80
WM	low → high	$\sigma = 0.35$	72.0	4.74
WM (ours)	low → high	$\sigma = 0.45$	72.5	**4.57**

Width gives stability. The two WM rows differ only in width; at $\sigma = 0.35$, linear probing stops improving once warmup ends and FID is worse (Fig. 5), while every $\sigma > 0.35$ keeps improving. *Progression determines FID.* CD and WM share shape, width, and duration, so their 1.23 FID gap ($\approx 29\times$ the seed noise, Appendix B.1) isolates progression alone. Neither factor subsumes the other:

UNI has maximal coexistence but no progression, giving the best LP (tied with CD) at worse FID than WM (4.88 vs. 4.57). Coexistence is needed for stability, and progression for FID.

C.4 Evidence for the Masking-Ratio Conflict

Two premises are inherited from prior work: masked generation needs heavy masking, and contrastive alignment degrades as masking rises. MAR [34] establishes the first, with a masking schedule spanning $[0.70, 1.00]$. For the second, FLIP [35] finds that at fixed batch size 50% masking is neutral for zero-shot IN-1K (68.5 vs. 68.6 unmasked) while 75% costs 2.8 points. FLIP stops at 75% and MAR starts at 70%, so neither covers the 75–100% band where a generative objective places nearly all of its mass.

Our direct evidence is a dose–response on each objective with the schedule fixed at its default, so the schedule is not the source of variation. Narrowing ϕ, the maximum masking ratio at which a sample enters the CLIP loss, costs LP monotonically: 72.5 / 70.0 / 63.4 at $\phi = 0.75/0.5/0.25$ (Table 9d), a 9.1-point swing ($\approx 58\times$ the seed noise) from changing only which samples reach the loss. The minimum masking ratio γ at which a sample enters the diffusion loss mirrors it: admitting fully visible samples costs 0.38 FID (4.95 at $\gamma = 0$ vs. 4.57 at $\gamma = 0.5$, Table 9c). The diffusion loss thus needs samples above 50% masking and the contrastive loss below 75%: the two admission windows sit on opposite halves of one axis, overlapping in a 25-point band, and each objective degrades as its window is squeezed toward the other's half. Table 8 rules out the alternatives: with CLIP pre-training of 12, 24, or 36 epochs and the contrastive loss active in stage 2, LP still collapses (≤ 9.1) under fixed high masking.

C.5 Two-Stage Baselines

To isolate the effect of staged initialization from the effect of masking warmup, we ablate two-stage pipelines along two axes: the CLIP pretraining duration (epochs 12, 24, and 36), and whether the second stage applies joint fine-tuning (CLIP contrastive loss and MAR diffusion loss together) or generation-only training (MAR diffusion loss alone). We used the same training setting defined in Section 4.

For the generation-only second stage, we follow FLUID's masking design. For the joint fine-tuning second stage, we additionally introduce the CLIP loss with the same weight used in DREAM's configuration.

Table 8: **Two-stage baselines.** Stage 2 is either generation-only (diffusion loss) or joint training (diffusion + CLIP loss). N/A denotes FID > 50 (generation collapse).

Stage 2 Training	S1 Epochs	S2 Epochs	LP ↑	FID ↓
Generation-Only	12	37	1.2	N/A
	24	25	2.0	N/A
	36	13	8.5	N/A
Joint	12	37	1.8	N/A
	24	25	2.5	N/A
	36	13	9.1	N/A

As shown in Table 8, varying the Stage 1 duration from 12 to 36 epochs changes the vision encoder's initialization but does not change the outcome. The joint training rows are particularly insightful: even when the CLIP contrastive loss is reintroduced during Stage 2 joint training, both LP and FID collapse across all three initialization strengths. This eliminates the possibility that the failure in the generation-only rows is simply a consequence of missing contrastive regularization in Stage 2.

The failure persists because the root cause lies in the masking distribution design. In the generation-only rows, the high-masking objective overwrites CLIP representations because there is no contrastive signal to anchor them. In the joint training rows, the contrastive signal is present but cannot function effectively under fixed high masking, producing the same collapse. A fixed high-masking regime is structurally incompatible with effective contrastive alignment regardless of initialization strength—consistent with the FX ablation in Table 1.

C.6 Hyperparameter Robustness

The remaining ablations vary one hyperparameter of Masking Warmup at a time. Performance is flat within a plateau for the width σ (Table 9a, Fig. 5), the warmup length (Table 9b), the admission thresholds γ and ϕ (Tables 9c and 9d), the loss weight λ (Table 9e), and the token set of the CLIP loss (Table 9f). The plateaus end where the analysis of Appendix C.4 predicts: for $\sigma \leq 0.35$, for $\phi \leq 0.5$, and for γ above 0.5. The layer at which the CLIP loss is applied is a design choice rather than a plateau (Table 9g). Note that γ and ϕ *filter* samples out of the respective loss; they do not clip the sampled masking ratio (Algorithm 1).

Table 9: **Hyperparameter ablations of Masking Warmup.** DREAM-L on CC12M; the default setting is highlighted and the best value in each panel is in bold.

(a) Masking std σ

	0.35	0.45	0.55	UNI
LP ↑	72.0	72.5	72.7	**73.1**
FID ↓	4.74	**4.57**	**4.57**	4.88

(b) Masking warmup duration (epochs)

	30	36	42
LP ↑	**72.5**	**72.5**	72.4
FID ↓	4.68	**4.57**	4.71

(c) Min. masking ratio for diffusion loss γ

	0	0.25	0.5	0.75
LP ↑	72.1	**72.7**	72.5	71.4
FID ↓	4.95	4.73	**4.57**	7.25

(d) Max. masking ratio for CLIP loss ϕ

	0.25	0.5	0.75	1.0
LP ↑	63.4	70.0	**72.5**	**72.5**
FID ↓	8.65	4.81	**4.57**	4.71

(e) CLIP loss weight λ

	0.002	0.005	0.01
LP ↑	72.1	**72.5**	72.0
FID ↓	4.76	**4.57**	4.64

(f) Tokens in CLIP loss

	All	Buffer
LP ↑	**72.5**	72.1
FID ↓	**4.57**	**4.57**

(g) Encoder layer for CLIP loss

	6	8	10	16
LP ↑	66.6	68.9	70.7	**72.5**
FID ↓	**4.51**	4.71	4.53	4.57

Standard deviation of masking. Our masking warmup samples masking ratios from $\mathcal{N}(\mu, \sigma^2)$ clipped to $[0, 1]$. To study the effect of variance, we vary σ and additionally include UNI, a schedule that approximates the high-variance limit with a uniform masking distribution. As shown in Table 9a, increasing σ from 0.35 to 0.45–0.55 improves both representation quality and generative performance: Linear Probing increases from 72.0% to 72.7%, while FID decreases from 4.74 to 4.57. Pushing variance to the extreme (UNI) further boosts Linear Probing to 73.1% but degrades FID to 4.88, improving representation at the cost of generation rather than both. Moderate variance, $\sigma \in [0.45, 0.55]$, thus yields the best generative quality while remaining close to the best representation performance. The unified performance analysis in Section 5.3 adopts $\sigma = 0.55$, which matches the best FID (4.57) and provides slightly stronger Linear Probing than $\sigma = 0.45$.

Duration of masking warmup. Linear Probing accuracy remains largely stable across warmup durations of 30–42 epochs (Table 9b). In contrast, generative quality exhibits a clear optimum: a 36-epoch warmup achieves the lowest FID (4.57), outperforming both shorter and longer schedules. Warmup duration therefore primarily influences generation fidelity rather than discriminative performance.

Minimum masking ratio for the diffusion loss. As discussed in Section 3.2, generative training benefits from sufficiently high masking ratios. Setting $\gamma = 0.5$ (i.e., only samples with at least 50% of tokens masked enter the diffusion loss) yields the lowest FID (4.57) among the tested configurations (Table 9c). Linear Probing accuracy decreases as γ increases from 0.25 to 0.75, suggesting that very high thresholds remove too many lightly masked samples from the reconstruction objective and limit its contribution to representation learning.

Maximum masking ratio for the CLIP loss. Increasing ϕ from 0.25 to 0.75 leads to consistent gains in Linear Probing accuracy (Table 9d): a larger upper bound lets more samples participate in

the contrastive objective. Raising ϕ further to 1.0 leaves Linear Probing unchanged but worsens FID (4.57 → 4.71). Together, γ and ϕ control the overlap between the samples that participate in each objective, and thereby the balance between generative reconstruction and discriminative alignment.

CLIP loss weight. LP and FID change little across $\lambda \in \{0.002, 0.005, 0.01\}$ (Table 9e), indicating that DREAM is not sensitive to the exact balance between the two objectives.

CLIP loss on buffer tokens. Restricting the CLIP loss to the buffer tokens, rather than applying it across both buffer and image tokens, slightly reduces Linear Probing accuracy while leaving FID unchanged (Table 9f).

Layer to apply the CLIP loss. Similar to REPA, we ablate the encoder layer at which the CLIP loss is applied, with a linear probe always on the final layer of the frozen vision encoder (layer 16). Deeper layers monotonically improve Linear Probing (66.6% → 72.5%) while FID remains stable (4.51–4.71; Table 9g). Aligning with language at the final encoder layer, where the most abstract semantic features reside, yields semantically grounded representations without interfering with the decoder's generative capacity. Our framework also achieves higher Linear Probing than REPA at the same depth (layer 6 in Table 15), since the CLIP loss provides stronger text supervision to the visual encoder than feature alignment does.

D Additional Representation Results

D.1 Few-Shot Transfer

Table 10: **Few-shot transfer.** 5-way, 5-shot classification accuracy across 14 datasets. The best result for each dataset is in bold.

Model	DTD [12]	Caltech-256 [23]	SUN397 [63]	Food-101 [5]	VOC2007 [20]	STL-10 [13]	Flowers [41]	UC Merced [66]	EuroSAT [25]	Country211 [55]	RESISC45 [11]	Dogs [29]	MIT Indoors [43]	IN-1K [15]	**Average**
MAR	64.9	56.1	73.0	42.3	47.2	53.1	74.6	77.3	74.9	28.9	73.8	37.7	61.2	62.3	59.1
FLUID	66.3	58.0	73.7	43.7	47.9	53.7	76.1	77.0	75.8	29.4	74.4	38.9	62.7	64.1	60.1
CLIP	81.0	95.1	95.8	86.0	79.6	96.1	**96.2**	92.6	82.1	**42.9**	90.4	78.2	93.7	94.3	86.0
REPA	73.2	73.6	88.1	60.2	61.2	74.6	86.5	82.4	79.2	33.8	80.4	50.2	81.2	78.9	71.7
DREAM	**88.6**	**96.4**	**97.3**	**86.9**	**95.0**	**99.8**	95.9	**98.5**	**95.0**	**42.9**	**93.4**	**81.2**	**94.5**	**96.1**	**90.1**

Following the standard 5-way, 5-shot protocol [57, 60, 19], we evaluate DREAM across 14 benchmark datasets compared to the other single-stage baselines. DREAM achieves the best or tied-best accuracy on 13 of the 14 datasets and exceeds CLIP by +4.1% on average (Table 10).

D.2 Dense Prediction

Table 11: **Dense prediction with frozen backbones.** Results for semantic segmentation (on ADE20K [74]) and depth estimation (on NYU Depth V2 [50]) with linear probes on frozen features, following DINOv3 [51] for segmentation and DINOv2 [42] for depth. mIoU: mean Intersection-over-Union; PixAcc: pixel accuracy; RMSE: root mean squared error; ARel: absolute relative error.

	Sem. Seg. (ADE20K)		Depth (NYU Depth V2)	
Model	mIoU↑	PixAcc↑	RMSE↓	ARel↓
MAR	23.4	70.6	0.75	0.25
FLUID	22.1	69.5	0.76	0.26
CLIP	34.9	75.3	0.64	0.20
REPA	32.7	75.9	**0.60**	**0.19**
DREAM	**36.8**	**76.7**	**0.60**	**0.19**

Table 11 reports semantic segmentation on ADE20K [74] and depth estimation on NYU Depth V2 [50] with linear probes on frozen features, following the DINOv3 [51] and DINOv2 [42] protocols, respectively.

D.3 Zero-Shot Robustness Under Masking

Since DREAM is trained with variable masking ratios, it exhibits improved resilience in highly masked settings. We compare three models—(1) standard CLIP trained without masking, (2) CLIP trained with our Masking Warmup schedule (CLIP-M), and (3) DREAM trained with variable-ratio masking—by plotting zero-shot accuracy across varying masking levels.

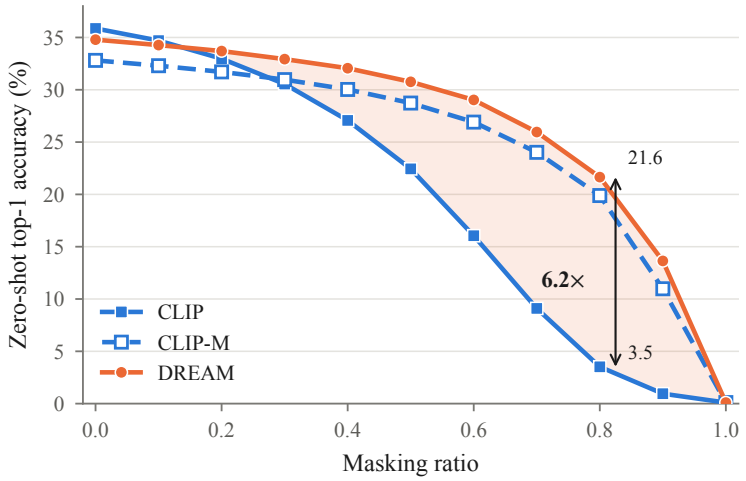


Figure 6: **Zero-shot accuracy under masking** for CLIP, CLIP with Masking Warmup (CLIP-M), and DREAM. The shaded region marks the masking levels at which DREAM surpasses CLIP; the arrow marks the 6.2× gap at a masking ratio of 0.8.

As shown in Fig. 6, progressive Masking Warmup improves robustness for both CLIP-M and DREAM over CLIP, particularly at moderate masking levels. DREAM exceeds CLIP already at 20% masking (33.7 vs. 33.0), and CLIP-M overtakes CLIP shortly after; both margins widen as more of the image is masked. Across all masking levels, DREAM further surpasses CLIP-M, demonstrating the additional benefit of its locally grounded representations learned through the reconstruction-based diffusion objective. At extreme occlusion (masking ratio > 0.8), DREAM's advantage becomes pronounced: it achieves about 6.2× the zero-shot accuracy of CLIP at a masking ratio of 0.8 (21.6 vs. 3.5), highlighting the strong resilience conferred by joint contrastive–generative training.

D.4 Forms of the Objective Synergy

The synergy between the two objectives takes two concrete forms in our results. *Generative* → *spatial*: the per-patch probes are where reconstruction shows up, with 36.8 mIoU on ADE20K and 0.60 RMSE on NYU Depth v2 against CLIP's 34.9 and 0.64, beating REPA's 32.7 mIoU despite its frozen teacher (Table 11). *Contrastive* → *selection*: the jointly trained text encoder scores partial generations far better than an external one (70.8 vs. 22.2 R@1 at 12.5% visibility, Table 13), and SD converts this into generation gains through trajectory selection alone (FID 4.56 → 4.25, CS 29.12 → 30.10, Table 5).

E Generation and Semantically Aligned Decoding

E.1 Qualitative Results

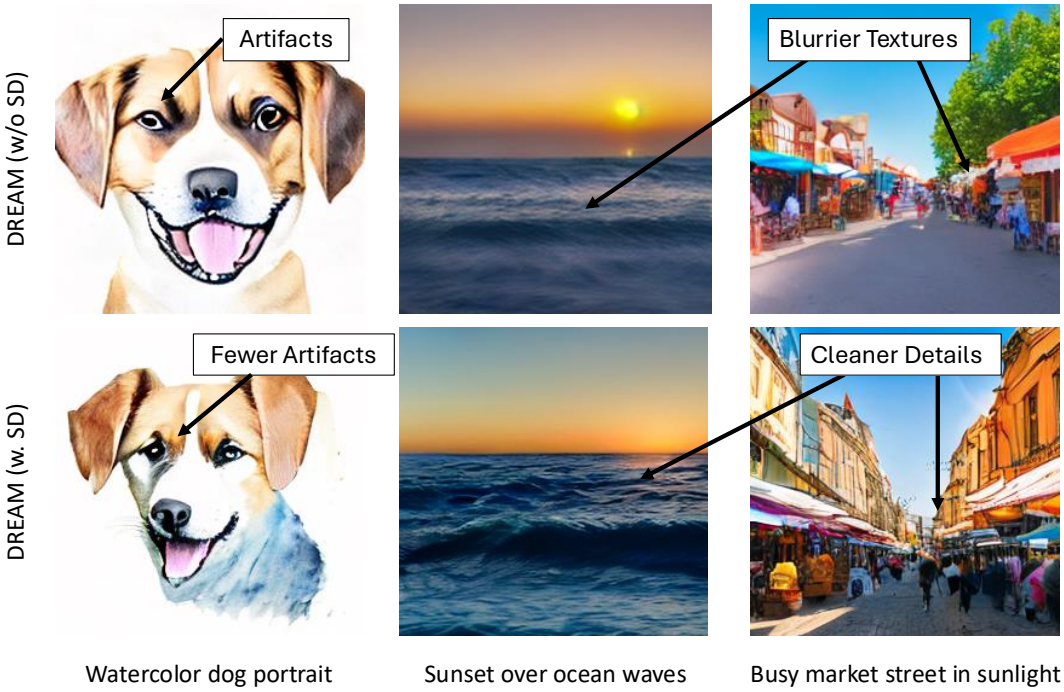


Figure 7: **Examples of images generated by DREAM with and without Semantically Aligned Decoding (SD).** Without Semantically Aligned Decoding, the outputs exhibit less coherent structure and more low-level blur. Applying Semantically Aligned Decoding produces images with clearer details and improved consistency with the prompt, in line with the gains observed in FID and CLIP scores.

We visualize the generated images to understand the effect of Semantically Aligned Decoding. Fig. 7 shows that with Semantically Aligned Decoding, the generated images tend to exhibit more stable structure and fewer low-level artifacts, consistent with the improvements observed in FID.

This effect aligns with the role of the text-guided retrieval step: since intermediate latents are trained to be semantically aligned with the text encoder, retrieving candidates conditioned on the prompt encourages selecting latents that both match the intended content and reconstruct cleanly. Our semantic segmentation results further indicate that the vision encoder encodes strong object-level cues, suggesting that Semantically Aligned Decoding primarily reinforces this semantic grounding during inference.

As a result, the decoding process naturally favors samples that are semantically coherent with the prompt while maintaining stronger distribution-level realism. Additional qualitative examples across different model scales are included in Fig. 8, 9, and 10.

E.2 Compositional Generation

FID and CLIP Score do not probe fine-grained compositionality, so we evaluate on GenEval [22]. Following the protocol of MiniT2I [59], we fine-tune the decoder and the CLIP-style text encoder on MiniT2I's public fine-tuning mixture (≈67.5K steps, 8×H200), applying an identical recipe to FLUID and DREAM. The vision encoder is frozen throughout, so every vision-encoder result in the paper carries over unchanged, and the comparison isolates whether the jointly trained representations support compositional generation once the decoder sees alignment data.

Table 12: **GenEval.** Decoder and CLIP-style text encoder fine-tuned on MiniT2I's public mixture under an identical recipe; vision encoder frozen. Mean ± std over 3 seeds; all metrics ↑.

Category	FLUID	DREAM	DREAM + SD
Single object	0.984 ± 0.003	0.983 ± 0.002	**0.996 ± 0.004**
Two objects	0.892 ± 0.009	0.886 ± 0.003	**0.893 ± 0.004**
Colors	0.899 ± 0.010	0.909 ± 0.004	**0.923 ± 0.005**
Counting	**0.789 ± 0.010**	0.772 ± 0.006	0.770 ± 0.024
Position	0.606 ± 0.001	**0.624 ± 0.010**	0.588 ± 0.005
Color attribution	**0.682 ± 0.007**	0.619 ± 0.004	0.672 ± 0.007
Overall	**0.809 ± 0.002**	0.799 ± 0.002	0.807 ± 0.005

DREAM + SD is statistically indistinguishable from FLUID overall (0.807 ± 0.005 vs. 0.809 ± 0.002), while plain DREAM trails by 0.010 (≈ 3.5×; Table 12). Per category, DREAM's joint representation is stronger on spatial position (0.624 vs. 0.606) but weaker on color attribution (0.619 vs. 0.682), suggesting that it captures which concepts appear and their broad layout, while FLUID's purely generative training binds attributes more precisely. SD recovers most of the attribution deficit (0.619 → 0.672) and lifts single-object presence and colors above FLUID, at the cost of position (0.588). DREAM therefore matches FLUID on compositional generation while being the only one of the two with competitive discriminative transfer.

E.3 Why the Encoder Discriminates at 12.5% Decoded

Table 13: **Retrieval by visibility.** Text-to-image retrieval over 1,000 CC12M pairs. DREAM scores the unmasked latent tokens; an external CLIP scores the correspondingly masked RGB images.

Visibility	DREAM R@1 / R@5 / R@10	External CLIP R@1 / R@5 / R@10
12.5%	**70.8 / 90.8 / 93.5**	22.2 / 40.3 / 49.0
25%	**83.9 / 96.2 / 97.7**	39.4 / 62.7 / 72.3
75%	**90.5 / 97.2 / 98.3**	73.9 / 89.8 / 93.9

Two properties make selection at SD's operating point ($t_s = 8$ of $T = 64$) possible. First, partial latents are in distribution for DREAM's encoder, because Masking Warmup exposes the contrastive pair to a wide range of visibility throughout training. At 12.5% visibility, DREAM retrieves the correct image with 70.8 R@1, against 22.2 for an external CLIP (Table 13).

Second, early scores predict final quality. We generated 5 candidates for each of 1,000 CC12M prompts, recorded DREAM's alignment score at 12.5% decoded, let every candidate finish, and judged the final images with an external CLIP. The early score correlates with the final result (correlation 0.428 over all 5,000 trajectories): the weakest tenth by early score finishes at 25.20 CLIP Score and the strongest tenth at 31.70. Since SD chooses among candidates for the same prompt, we also measured within-prompt selection: picking by early score finds the best of 5 candidates 31.4% of the time (against 20% by chance), gaining +0.787 CLIP Score over random choice. SD therefore does not need a perfect ranking; it needs an early score that is informative about final quality and computable from sparse tokens, a combination that Masking Warmup creates and external rerankers lack.

E.4 Is the Gain a Self-Consistency Artifact?

Because SD selects candidates with DREAM's own contrastive encoder, which is trained on CC12M, part of its CC12M FID gain could in principle reflect a match between the selection distribution and the evaluation distribution. Such an artifact would help most where the selector and the evaluation share a distribution, and would not survive a change of training corpus or of evaluation captions. We observe the opposite pattern. The alignment gain is undiminished out of domain: SD improves CLIP Score by +0.98 on CC12M and +0.97 on zero-shot MS-COCO (Table 5), where the selector scores human-written captions from a corpus it never saw and is judged by an external CLIP it was never trained against. It also persists when the selector is trained on a different corpus (+0.21 CS on GPIC, Table 6). SD's effect on FID tracks caption coverage instead: positive in domain on a narrow corpus (4.56 → 4.25), negative out of domain on that corpus (10.02 → 10.38), and neutral once coverage is broad (10.24 → 10.22).

E.5 Compute Accounting for Semantically Aligned Decoding

Semantically Aligned Decoding operates under a fixed compute budget by trading off the number of candidates K against the step at which they are scored. Let T denote the total number of decoding steps ($T = 64$ unless noted otherwise, as described in Section 3.3), and let $t_s < T$ denote the step at which scoring is performed.

From step 0 to t_s, all K candidates are decoded in parallel; after scoring, only the winning candidate continues for the remaining $T - t_s$ steps. The total number of forward-encoder (NFE) passes is therefore

$$\text{NFE} = \underbrace{K \cdot t_s}_{\text{parallel candidates}} + \underbrace{(T - t_s)}_{\text{single-candidate decoding}} . \tag{4}$$

The fraction of the image decoded at scoring time is t_s/T. For the configuration analyzed in Section 5.3.3 (NFE = 128, $K = 9$, $T = 64$), this yields $t_s = 8$, meaning candidate selection occurs with only $8/64 = 12.5\%$ of the image decoded. For $K = 17$ at the same budget, $t_s = 4$ (6.25% decoded).

E.6 When Semantically Aligned Decoding Should and Should Not Help

At a matched budget of 128 NFE, SD reaches 4.25 FID against 4.50 for external reranking of $K = 2$ candidates, at higher throughput (Table 4). SD wins when the scorer must act on partial generations under a budget too small to complete all candidates, which Table 13 explains. It should not help when the budget allows completing every candidate, or when the text encoder has not seen partially visible inputs during training.

E.7 Generation Diversity

To verify that Semantically Aligned Decoding (SD) does not collapse generation toward preferred modes, we conducted a diversity analysis. We generated 20 candidates per prompt for 50 prompts and computed average pairwise cosine dissimilarity within each prompt group using an external CLIP ViT-B/32 encoder. Results are reported in Table 14.

Table 14: **Generation diversity.** Average pairwise cosine dissimilarity between candidates for the same prompt, measured with an external CLIP ViT-B/32 encoder. Higher is more diverse.

Condition	Avg. Intra-Prompt Diversity
DREAM, K=1 (no SD)	0.122
External CLIP reranker	0.117
DREAM (SD), K=9	0.119

SD and external CLIP reranking produce nearly identical diversity scores (0.119 vs. 0.117), both within 0.005 of the no-selection baseline, confirming that SD does not collapse generation toward

preferred modes any more than external reranking does. Diversity is preserved because the contrastive alignment space is broad: many visually distinct completions can be semantically consistent with a given prompt, so candidates initialized from independent stochastic seeds naturally remain diverse even after selection.

F Baselines and Component Isolation

F.1 REPA Baseline

Table 15: **REPA baseline ablations.** Teacher encoder and alignment layer. With CLIP-L, deeper alignment yields higher LP and layer 6 gives the best FID; DINOv2 (at its best-FID layer) gives higher LP but worse FID. The default REPA baseline is highlighted.

Teacher	CLIP-L	CLIP-L	CLIP-L	DINOv2
Alignment layer	4	6	8	8
LP ↑	58.0	62.5	66.1	**68.4**
FID ↓	4.71	**4.42**	4.76	4.67

Our REPA baseline adapts REPA [69] to masked image modeling by aligning the unmasked encoder tokens with the features of a frozen pretrained encoder computed on the full image (Appendix G).

Alignment layer. Following [69], we ablate the encoder layer at which representations are aligned with a pretrained CLIP-L encoder and select the layer with the lowest FID. Linear probing is always applied to the output of the vision encoder. Aligning at layer 6 yields the lowest FID (Table 15), and we use it for the REPA baseline throughout. Linear probing accuracy increases with alignment depth, while FID is best at the intermediate layer.

Teacher encoder. We also compare CLIP-L with DINOv2, the default teacher in REPA. For a fair comparison, we select the DINOv2 alignment layer with the same protocol, which gives layer 8. DINOv2 supervision yields higher linear probing accuracy but worse FID (Table 15): DINOv2 provides stronger category-level representations, but CLIP's text-aligned supervision is more effective for text-to-image generation. We therefore report the CLIP-supervised REPA baseline in the main paper.

F.2 Isolating Architectural Components

DREAM relies on several specialized components: the SD-VAE tokenizer, the MAR diffusion head, a CLIP-style text encoder, and a frozen T5-Large conditioning encoder. Every component except the contrastive branch is held fixed and identical across arms, so it is differenced out of our comparisons. The SD-VAE tokenizer and the diffusion head are shared by MAR, FLUID, REPA, and DREAM. The frozen T5-Large is shared by all generative arms and never touches the encoder, since text conditioning is decoder-only (Fig. 2), so it cannot affect any discriminative result. The two additions are each ablated: diffusion reconstruction by Table 1 (-1.7 LP without it), and the contrastive branch by attachment layer (66.6 $\rightarrow$ 72.5 LP, Table 9g), token set (Table 9f), and inference scorer (4.25 vs. 4.50 FID for external CLIP, Table 4). Table 19 confirms that DREAM-L's entire +11.6% parameter delta over FLUID-L is the CLIP text encoder, and our CLIP baseline on SD-VAE tokens matches the LP of pixel-based CLIP in StableRep [57]. Whether the conclusions hold with a different tokenizer or text encoder remains untested (Appendix A.1).

G Implementation Details

This section describes the architecture, training, inference, and evaluation protocols, and gives pseudocode for DREAM, REPA, and Semantically Aligned Decoding. The full implementation is available at https://github.com/chaoli-charlie/dream.

MAR [34] is a class-conditional encoder–decoder model in which the encoder takes class labels as input. FLUID [21] is a text-to-image model that uses a transformer architecture operating on continuous tokens, which are cross-attended with text tokens.

DREAM draws architectural inspiration from both MAR and FLUID. We adopt the encoder design from MAR but remove the class-conditioning pathway, ensuring that no textual information enters the visual encoder. On the decoder side, we use FLUID's architecture and introduce text conditioning only at this stage. This prevents shortcuts between visual features and text and ensures that all discriminative and dense-prediction capabilities learned by the encoder originate solely from visual inputs.

We align the encoder's visual representations with text using the CLIP contrastive loss [44]. Conceptually, DREAM removes text conditioning from the encoder and instead trains the encoder to predict the relevant textual semantics, reintroducing text conditioning only within the decoder.

G.1 Image Tokenizer and Detokenizer

We employ the `kl-f8-ft-EMA` autoencoder from Stable Diffusion [48], a widely used continuous tokenizer. The image is first encoded into a 32×32 latent grid with 4 channels per token. To maintain consistency with the discrete tokenizer, we group each 2×2 block of latent tokens into a single token, resulting in a final sequence of 256 tokens, each with 16 channels. The detokenizer reconstructs the image from the predicted continuous representations.

G.2 ViT Architecture

G.2.1 Vision Encoder

After the tokenizer, the latent sequence length becomes 256. During training, tokens are masked and dropped before being fed into the encoder. We prepend 64 buffer tokens to the unmasked sequence to ensure stability. We use the standard ViT architecture [18], a stack of Transformer blocks in which each block consists of a multi-head self-attention block and an MLP block. We use two learnable positional embeddings, one added to the input of the encoder and another added to the input of the decoder.

We use features from the encoder output for classification tasks, such as Linear Probing, Few-Shot Transfer Learning, and Fine-tuning. We apply average pooling to the encoder output for linear classification in Linear Probing and Fine-tuning, and we use the corresponding visual embeddings for semantic segmentation and depth estimation.

G.2.2 Text Encoders

We use two text encoders, each serving a distinct role during training. This separation enables direct comparability with prior work and ensures that our contrastive and generative objectives follow established conventions.

For contrastive alignment, captions are tokenized with the OpenAI CLIP tokenizer (77 tokens) and encoded using a CLIP text transformer, following the setup in [57]. The resulting features are projected into the vision encoder's latent space.

For generation, captions are tokenized with SentencePiece (128 tokens) and encoded using a frozen T5-Large model [45]. A lightweight 6-layer text aligner projects the T5 embeddings into the decoder's latent space, providing conditioning for the decoder. This text-conditioning module follows the design used in FLUID [21].

G.3 Model Sizes

We evaluate four model scales: **Base (B)**, **Large (L)**, **Huge (H)**, and **Giant (G)**. In all settings, the encoder and decoder share the same number of transformer blocks (e.g., a 32-block model allocates 16 blocks to the encoder and 16 to the decoder).

- **Base (B):** 24 blocks, hidden size 768
- **Large (L):** 32 blocks, hidden size 1024

- **Huge (H):** 40 blocks, hidden size 1280
- **Giant (G):** 48 blocks, hidden size 1664

The **Huge** configuration corresponds to a ∼1B-parameter encoder-decoder model. The **Giant** configuration (∼2.4B parameters in total) provides an encoder with approximately 1B parameters (921M), enabling us to study scaling behavior in the 1B-encoder regime.

G.4 Pre-Training

Table 16 summarizes the pre-training configurations of all models. Unless otherwise specified, all models are trained for 49 epochs on CC12M.

G.4.1 DREAM Training

DREAM uses a progressive Masking Warmup in which the masking ratio follows a shifting truncated Gaussian distribution. The minimum and maximum masking ratios are fixed at 0 and 1.0, respectively. The mean of the distribution increases linearly from 0 to 1.0 over the first 36 epochs and is then fixed at 1.0 for the remaining epochs. This schedule enables a smooth transition from fully visible inputs to the high-masking regime required for strong generative performance.

G.4.2 REPA Training

REPA [69] was originally proposed for diffusion transformers, where noisy intermediate encoder states are aligned with clean-image features from a pretrained vision encoder. In our adaptation for masked image modeling, we instead align the unmasked tokens of the vision encoder with the corresponding features computed from the full image using a pretrained CLIP-L encoder. This provides REPA-style feature supervision in a manner that is compatible with our masked-token training setup.

G.4.3 CLIP Training

Our CLIP baseline is trained without masking for the full 49 epochs.

Table 16: **Pre-training settings.** All models are trained on CC12M. The absolute learning rate is the base learning rate × batch size / 256 (8×10^{-4} at batch size 2048). DREAM uses $\sigma = 0.55$ for the main results (Section 5.3); the controlled study and ablations use $\sigma = 0.45$ (Appendix C). "–" denotes not applicable.

Config	CLIP	MAR / FLUID	REPA	DREAM
Optimizer	AdamW [38], $(\beta_1, \beta_2) = (0.9, 0.95)$			
Base learning rate	1×10^{-4}, constant after warmup			
Warmup / training epochs	12 / 49			
Batch size	2048			
Gradient clip	3.0			
Label dropout	0.1			
Augmentation	Random horizontal flip			
Weight decay	0.02	0.02	0.02	0.04
Masking ratio (min / max)	–	0.7 / 1.0	0.7 / 1.0	0.0 / 1.0
Masking ratio (std)	–	0.25	0.25	0.55
Masking warmup epochs	–	–	–	36
Max. masking for CLIP loss (ϕ)	–	–	–	75%
Min. masking for diffusion loss (γ)	–	–	–	50%
CLIP loss weight (λ)	–	–	–	0.005
REPA teacher / loss weight	–	–	CLIP-L / 0.5	–

G.4.4 FLUID and MAR Training

FLUID and MAR baselines are trained for 49 epochs following the MAR masking schedule [34]: masking ratios are sampled from a truncated Gaussian centered at 1.0 (100% masking) with a standard

deviation of 0.25, bounded to the range [0.7, 1.0]. This setup allows us to directly evaluate the vision encoder and assess the effect of applying text conditioning only in the decoder.

G.5 Inference

G.5.1 Generation

DREAM adopts the MAR [34] decoding procedure, which generates images by predicting a randomly selected set of latent tokens at each step. At inference time, we begin from a fully masked latent sequence and iteratively reduce the masking ratio following a cosine schedule. By default, we use 64 steps in this schedule. In each iteration, DREAM first predicts continuous latent values for all currently masked positions. It then applies token-wise temperature sampling by adding temperature-scaled noise to the predicted latent vectors. Finally, instead of using confidence-based or location-based ordering as in MAGE, DREAM follows MAR's fully randomized ordering: a subset of positions is sampled uniformly and re-masked to match the target masking ratio for the next iteration. This process progressively refines the latent representation until all positions are filled.

G.5.2 Classifier-Free Guidance

We adopt classifier-free guidance by randomly removing the text condition for 10% of training samples, replacing the prompt with a null token. This trains the model to produce both conditional and unconditional predictions using the same diffusion head.

During inference, we run the model twice: once with the null prompt and once with the given text. This produces the unconditional and conditional representations z_u and z_c. The guided noise prediction is computed as

$$\varepsilon = \varepsilon_\theta(x_t \mid t, z_u) \; + \; \omega \cdot \big(\varepsilon_\theta(x_t \mid t, z_c) - \varepsilon_\theta(x_t \mid t, z_u)\big),$$

where ω denotes the guidance strength. Since we use a fixed sampling temperature of 1, no additional temperature scaling is applied.

For sampling, we sweep over two CFG schedules: a constant scale and a linearly increasing scale, and report the best CLIP and FID scores.

G.6 Evaluation Protocols

Linear probing and fine-tuning follow the settings in Tables 17 and 18.

Table 17: **Linear probing settings.**

Config	Value
Optimizer	LARS [68]
Base LR (B/L/H/G)	0.1 / 0.05 / 0.05 / 0.02
Weight decay	0
Optimizer momentum	0.9
Batch size	4096
LR schedule	Cosine decay [37]
Warmup / training epochs	10 / 90
Augmentation	RandomResizedCrop

Table 18: **Fine-tuning settings.**

Config	Value
Optimizer	AdamW [38]
Optimizer momentum	$(\beta_1, \beta_2) = (0.9, 0.999)$
Base learning rate	2.5×10^{-4}
Weight decay	0.05
Layer-wise LR decay [1]	0.75
Batch size	1024
LR schedule	Cosine decay [37]
Warmup / training epochs	5 / 50
Label smoothing [52]	0.1
Augmentation	RandAug (9, 0.5) [14]
Mixup [72] / CutMix [71]	0.8 / 1.0
Random erase	0
Drop path [28]	0.2

G.6.1 Semantic Segmentation

For semantic segmentation, we follow the DINOv3 [51] setup. A linear layer is trained on top of the frozen patch features from the vision encoder and evaluated on ADE20K [74] (150 categories). We report mean IoU and Pixel Accuracy. The linear head is trained for 40k iterations with a learning rate of 1×10^{-4} using a cosine decay schedule with 1.5k warm-up steps.

G.6.2 Depth Estimation

For depth estimation, we follow the DINOv2 [42] protocol. A linear head is trained over frozen patch features and evaluated on NYU Depth v2 [50]. We report RMSE and ARel. Training uses 38.4k iterations with a learning rate of 3×10^{-4}, a cosine scheduler, and 12.8k warm-up steps.

G.7 Training Efficiency

Table 19: **Training efficiency** of Large models, measured on the same hardware (NVIDIA A100).

Model	Total Params	Train GFLOPs/sample	Training Time	Peak GPU Mem
FLUID-L	570M	772.8	~48 hours	43,827 MB
REPA-L	578M	825.2	~48 hours	44,872 MB (+2.4%)
DREAM-L	636M (+11.6%)	877.2 (+13.5%)	~50 hours (+4.2%)	45,688 MB (+4.2%)

Here, we compare the training efficiency between the T2I models. All models were trained on NVIDIA A100 GPUs. The results show that despite incorporating two text encoders, DREAM-L incurs only modest overhead compared to its baselines: a 11.6% increase in parameters and 13.5% increase in training compute over FLUID-L, while requiring only 4.2% more training time and 4.2% additional GPU memory. Additionally, we note that the increase in total parameters arises from the CLIP text encoder, whose size is fixed across model scales (B/L/H/G), so its relative parameter overhead diminishes as the backbone grows.

These numbers confirm that the additional architectural components of DREAM do not impose a significant practical burden during training.

G.8 Pseudocode

Algorithm 1 Training Loop for DREAM

Require: model $\mathcal{M}$, VAE, CLIP-style text encoder $\mathcal{E}_{\text{clip}}$, conditioning text encoder $\mathcal{E}_{\text{cond}}$, data loader $\mathcal{D}$, coefficient λ_{clip}, epoch index e, admission thresholds γ (diffusion) and ϕ (CLIP)

for each batch i, $(\mathbf{I}, \mathbf{T})$ in $\mathcal{D}$ **do**
  $\mathbf{Z} \leftarrow \text{VAE.Encode}(\mathbf{I})$
  $\mathbf{t}_{\text{clip}} \leftarrow \mathcal{E}_{\text{clip}}.\text{Encode}(\mathbf{T})$ ▷ text embeddings for CLIP loss
  $\mathbf{t}_{\text{cond}} \leftarrow \mathcal{E}_{\text{cond}}.\text{Encode}(\mathbf{T})$ ▷ text embeddings for conditioning
  $\mathbf{m} \leftarrow \textsc{MaskingScheduler}(\mathbf{Z},\ \hat{s} = e + \frac{i}{|\mathcal{D}|})$ ▷ mask depends on $\mathbf{Z}$ and training step
  $\mathbf{Z}_{\text{unmask}} \leftarrow \textsc{Unmask}(\mathbf{Z}, \mathbf{m})$ ▷ retain only unmasked tokens

  $(\mathbf{H}, \mathbf{R}) \leftarrow \textsc{EncoderForward}(\mathbf{Z}_{\text{unmask}})$ ▷ (1) forward encoder on unmasked tokens
  $\mathbf{z} \leftarrow \textsc{DecoderForward}(\mathbf{H}, \mathbf{m}, \mathbf{t}_{\text{cond}})$ ▷ (2) forward decoder conditioned on text

  $\mathcal{B}_{\text{diff}} \leftarrow \{b : \text{ratio}(\mathbf{m}_b) \geq \gamma\},\ \ \mathcal{B}_{\text{clip}} \leftarrow \{b : \text{ratio}(\mathbf{m}_b) \leq \phi\}$ ▷ filter, not clip
  $\mathcal{L}_{\text{mar}} \leftarrow \textsc{DiffusionLoss}(\mathbf{z}, \mathbf{Z}, \mathbf{m})$ over $\mathcal{B}_{\text{diff}}$ ▷ (3) MAR / diffusion loss
  $\mathbf{f}_{\text{img}} \leftarrow \textsc{MeanPool}(\mathbf{R})$ ▷ mean-pooled encoder image embedding
  $\mathcal{L}_{\text{clip}} \leftarrow \text{CLIPLoss}(\mathbf{f}_{\text{img}}, \mathbf{t}_{\text{clip}})$ over $\mathcal{B}_{\text{clip}}$ ▷ (4) image–text CLIP loss

  $\mathcal{L} \leftarrow \mathcal{L}_{\text{mar}} + \lambda_{\text{clip}} \cdot \mathcal{L}_{\text{clip}}$ ▷ (5) combined objective
  $\mathcal{L}.\textsc{backprop}()$ ▷ (6) backpropagation step
end for

Algorithm 2 Training Loop for REPA Baseline, Implemented for Masked Image Modeling

Require: model $\mathcal{M}$, VAE, conditioning text encoder $\mathcal{E}_{\text{cond}}$, REPA teacher encoder $\mathcal{E}_{\text{teacher}}$, data loader $\mathcal{D}$, coefficient λ_{repa}, epoch index e

for each batch $(\mathbf{I}, \mathbf{T})$ in $\mathcal{D}$ **do**
  $\mathbf{Z} \leftarrow \text{VAE.Encode}(\mathbf{I})$
  $\mathbf{t}_{\text{cond}} \leftarrow \mathcal{E}_{\text{cond}}.\text{Encode}(\mathbf{T})$ ▷ text embedding used solely for decoder conditioning
  $\mathbf{m} \leftarrow \textsc{MaskingScheduler}(\mathbf{Z},\ \hat{s} = e + \frac{i}{|\mathcal{D}|})$ ▷ mask depends on $\mathbf{Z}$ and training step
  $\mathbf{Z}_{\text{unmask}} \leftarrow \textsc{Unmask}(\mathbf{Z}, \mathbf{m})$ ▷ retain only unmasked tokens
  $(\mathbf{H}, \mathbf{R}) \leftarrow \textsc{EncoderForward}(\mathbf{Z}_{\text{unmask}})$ ▷ (1) encoder forward on unmasked tokens
  $\mathbf{z} \leftarrow \textsc{DecoderForward}(\mathbf{H}, \mathbf{m}, \mathbf{t}_{\text{cond}})$ ▷ (2) decoder reconstructs masked tokens
  $\mathcal{L}_{\text{mar}} \leftarrow \textsc{DiffusionLoss}(\mathbf{z}, \mathbf{Z}, \mathbf{m})$ ▷ (3) MAR / diffusion reconstruction loss
  $\mathbf{R}_{\text{teacher}} \leftarrow \mathcal{E}_{\text{teacher}}.\textsc{ForwardFeatures}(\mathbf{I})$ ▷ teacher encoder features
  $\mathcal{L}_{\text{repa}} \leftarrow \text{REPALoss}(\mathbf{R}, \mathbf{R}_{\text{teacher}})$ ▷ (4) REPA encoder alignment loss
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{mar}} + \lambda_{\text{repa}} \cdot \mathcal{L}_{\text{repa}}$ ▷ (5) combined MAR + REPA objective
  $\mathcal{L}.\textsc{backprop}()$ ▷ (6) backpropagation step
end for

Algorithm 3 Semantically Aligned Decoding

Require: Number of candidates K, scoring step t_s, total steps T, text $\mathcal{Y}$
Initialize candidate states $\{(\mathbf{x}^{(n)}, \mathbf{m}^{(n)})\}_{n=1}^{K}$
for step = 0 to $T-1$ **do**
  if step $< t_s$ **then** ▷ 1) Generate K candidates up to step t_s
    for $n = 1$ to K **do**
      $(\mathbf{x}^{(n)}, \mathbf{m}^{(n)}) \leftarrow \text{DecodeStep}(\mathbf{x}^{(n)}, \mathbf{m}^{(n)}, \mathcal{Y})$
    end for
  else if step $= t_s$ **then** ▷ 2) Score with text, retrieve best, then decode
    for $n = 1$ to K **do**
      $s_n \leftarrow \text{CLIPScore}(\mathbf{x}^{(n)}, \mathbf{m}^{(n)}, \mathcal{Y})$ ▷ encoder sees decoded tokens only
    end for
    $n^* \leftarrow \arg\max_n s_n$
    $(\mathbf{x}, \mathbf{m}) \leftarrow (\mathbf{x}^{(n^*)}, \mathbf{m}^{(n^*)})$
    $(\mathbf{x}, \mathbf{m}) \leftarrow \text{DecodeStep}(\mathbf{x}, \mathbf{m}, \mathcal{Y})$
  else ▷ 3) Continue decoding best candidate
    $(\mathbf{x}, \mathbf{m}) \leftarrow \text{DecodeStep}(\mathbf{x}, \mathbf{m}, \mathcal{Y})$
  end if
end for
return VAE.Decode($\mathbf{x}$)

H Assets and Licenses

This work uses several existing datasets, models, and software libraries which we list below.

Datasets.

- **Conceptual 12M (CC12M)** [9]. Used for pre-training. URL: https://github.com/google-research-datasets/conceptual-12m. License: provided by Google LLC "AS IS" and freely usable for any purpose with acknowledgement of Google as the data source. CC12M is a list of (image-URL, alt-text) pairs scraped from the web; the dataset distributors do not own the underlying images and we accessed images via the provided URLs subject to the original copyright of each web source.

- **ImageNet-1K (ILSVRC 2012)** [15]. Used for linear probing, fine-tuning, and as the source for the few-shot IN-1K split. URL: https://image-net.org/. License: ImageNet Terms of Access (non-commercial research and educational use only); ImageNet does not own the copyright of the underlying images.
- **ImageNet variants for robustness evaluation:** ImageNet-A [27] (MIT License, https://github.com/hendrycks/natural-adv-examples), ImageNet-R [26] (MIT License, https://github.com/hendrycks/imagenet-r), ImageNet-Sketch [58] (MIT License, https://github.com/HaohanWang/ImageNet-Sketch), ImageNet-Hard [53] (MIT License, https://github.com/taesiri/ImageNet-Hard), ObjectNet [2] (custom non-commercial research license, https://objectnet.dev/), and the ImageNetV2 splits MatchedFrequency, Threshold0.7, and TopImages [47] (MIT License, https://github.com/modestyachts/ImageNetV2).
- **MS-COCO** [36]. Used for zero-shot text-to-image evaluation (MS-COCO 30K). URL: https://cocodataset.org/. License: annotations licensed under Creative Commons Attribution 4.0 (CC BY 4.0); the images are not owned by the COCO Consortium and use of the images is subject to the original Flickr Terms of Use.
- **ADE20K** [74]. Used for semantic segmentation. URL: https://groups.csail.mit.edu/vision/datasets/ADE20K/. License: image annotations and software released under the BSD-3-Clause License (Copyright 2019 MIT, CSAIL); the underlying images are provided for non-commercial research and educational use, with MIT CSAIL not owning the copyright of the original images.
- **NYU Depth V2** [50]. Used for monocular depth estimation. URL: https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html. The original release does not specify a formal license; commonly redistributed preprocessed versions (e.g., FastDepth) are released under the MIT License.
- **Few-shot transfer datasets**: DTD [12] (research-only, https://www.robots.ox.ac.uk/~vgg/data/dtd/), Caltech-256 [23] (CC BY 4.0 via CaltechDATA), SUN397 [63] (research-only, https://vision.princeton.edu/projects/2010/SUN/), Food-101 [5] (research-only, ETH Zurich), PASCAL VOC 2007 [20] (Flickr Terms of Use; annotations free for research), STL-10 [13] (research-only), Oxford 102 Flowers [41] (research-only, University of Oxford), UC Merced Land Use [66] (public domain; images from USGS), EuroSAT [25] (MIT License), Country211 [55] (derived from YFCC100M; CC license metadata per image; Country211 split released by OpenAI under MIT License), RESISC45 [11] (research-only, https://gcheng-nwpu.github.io/), Stanford Dogs [29] (research-only, derived from ImageNet), MIT Indoor Scenes [43] (research-only, https://web.mit.edu/torralba/www/indoor.html).

Pretrained models.

- **Stable Diffusion VAE** (`kl-f8-ft-EMA`) [48]. Used as the continuous image tokenizer. URL: https://github.com/CompVis/stable-diffusion. License: CreativeML Open RAIL-M.
- **T5-Large** [45]. Used (frozen) as the decoder-side text encoder for generation. URL: https://huggingface.co/google-t5/t5-large. License: Apache License 2.0.
- **OpenAI CLIP** [44]. Used for: (i) the pretrained CLIP-L teacher in the REPA baseline, (ii) the CLIP ViT-B/32 used as an external reference encoder for the diversity analysis (Table 14), and (iii) the CLIP-style text encoder architecture and tokenizer for our contrastive branch. URL: https://github.com/openai/CLIP. License: MIT License.
- **DINOv2** [42]. Used as an alternative REPA teacher in the ablation in Section F.1; its linear-probe protocol is followed for NYU Depth V2 depth estimation. URL: https://github.com/facebookresearch/dinov2. License: Apache License 2.0.

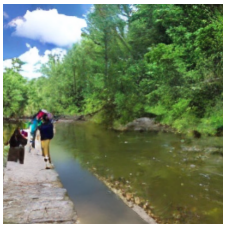
Code bases and software libraries.

- **MAR** [34]. The encoder architecture, MAR masking strategy, decoding procedure, and diffusion head implementation are adapted from the official MAR codebase. URL: https://github.com/LTH14/mar. License: MIT License.

- **REPA** [69]. We adapt the representation-alignment objective for the masked-image-modeling setting as a baseline. URL: https://github.com/sihyun-yu/REPA. License: MIT License.
- **SentencePiece** [32]. Used for tokenizing captions for the T5 text encoder. URL: https://github.com/google/sentencepiece. License: Apache License 2.0.
- **PyTorch** and standard Python scientific libraries (NumPy, SciPy, timm, etc.) under their respective open-source licenses (BSD-3-Clause, BSD, Apache 2.0, etc.).
- **DINOv3** [51]. The dense-prediction linear-probe protocol of DINOv3 is followed for ADE20K segmentation evaluation. URL: https://github.com/facebookresearch/dinov3. License: DINOv3 License (custom, non-Apache).

I Additional Generated Samples

Figures 8–10 show samples from three model scales for the same 20 CC12M captions.


Hiking along the Chipola River

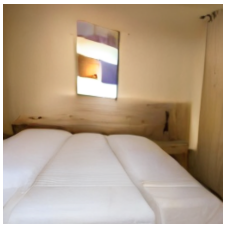

Skinny Broccoli and Mixed Vegetable Stir Fry - Skip takeout and make your own fast, easy, and healthy stir fry! Think of all the money and calories you'll save!!

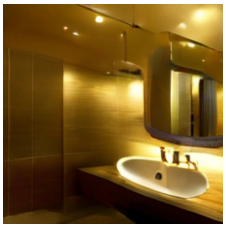

Home is Where the Art is IX <PERSON>


View of rooftop of Church of the Holy Sepulchre in Old City, Old City, UNESCO World Heritage Site, Jerusalem, Israel, Middle East

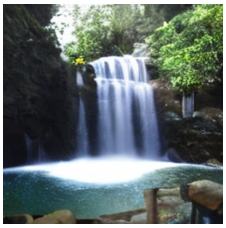

Shared Zone - road walking through some of Innes National Park


A soup bowl served with Egg drop soup topped with scallions and fried noodle

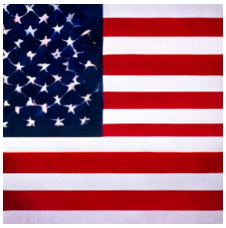

A bed or beds in a room at Chiaia 205


Unusual lighting can make the atmosphere in the bathroom too

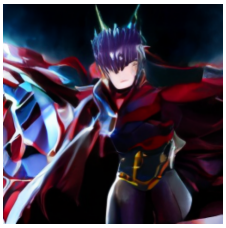

The Ghost (bike) Rider Comic Book Characters, Comic Books, Ghost Rider 2, Scary, Creepy, Cycling Art, Road Cycling, Bicycle Art, Bike Rider

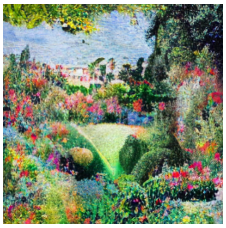

The Hunas Falls Hotel: Starting point of the waterfall


Tree Photograph - The Manzanita Tree by <PERSON>


HD Wallpaper American Flag with high-resolution 1920x1080 pixel. You can use this wallpaper 1920x1080


This Snow Covered Abandoned Mall Has Been Forgotten By Time

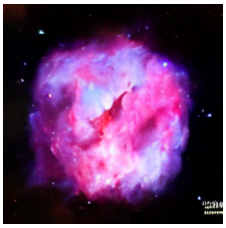

Wallpapers Anime, My Hero Academia, Blood, All Might Desktop Picture


The Art Print featuring the painting The Garden at Bougival by <PERSON>


The Porcelain Jars - Vigan, Ilocos Sur


The Famous church of the good shepherd photographed in the night with Milky way in the background


Fruit Still Life With Two Peaches And Blue And Green Grapes On The Branch And A Vine Leaf Artwork by <PERSON>


The Wizard Nebula: Sharpless 142

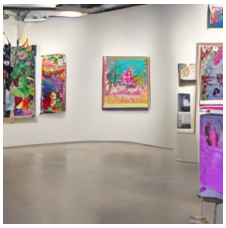
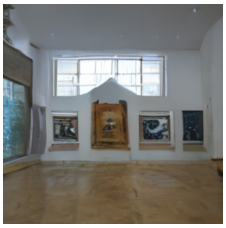

Local visual arts fans have a treat awaiting them at the Asheville Mall through April 28.

Figure 8: **Samples from DREAM-L (0.57B)** generated from CC12M captions with CFG = 5.0. The same 20 captions are used in Figs. 8–10 for comparison across scales. Captions are shown verbatim; CC12M replaces person names with the token <PERSON>.


Hiking along the Chipola River

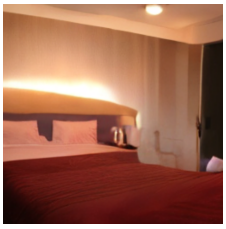

Skinny Broccoli and Mixed Vegetable Stir Fry - Skip takeout and make your own fast, easy, and healthy stir fry! Think of all the money and calories you'll save!!

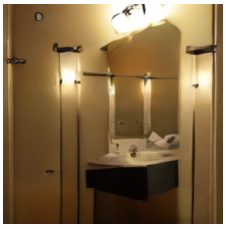

Home is Where the Art is IX <PERSON>

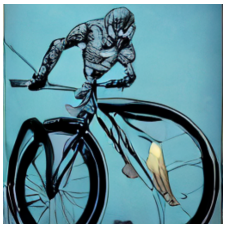

View of rooftop of Church of the Holy Sepulchre in Old City, Old City, UNESCO World Heritage Site, Jerusalem, Israel, Middle East


Shared Zone - road walking through some of Innes National Park


A soup bowl served with Egg drop soup topped with scallions and fried noodle


A bed or beds in a room at Chiaia 205


Unusual lighting can make the atmosphere in the bathroom too


The Ghost (bike) Rider Comic Book Characters, Comic Books, Ghost Rider 2, Scary, Creepy, Cycling Art, Road Cycling, Bicycle Art, Bike Rider

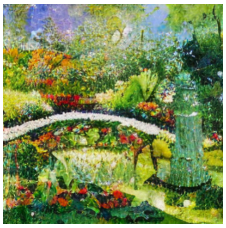

The Hunas Falls Hotel: Starting point of the waterfall


Tree Photograph - The Manzanita Tree by <PERSON>


HD Wallpaper American Flag with high-resolution 1920x1080 pixel. You can use this wallpaper 1920x1080


This Snow Covered Abandoned Mall Has Been Forgotten By Time


Wallpapers Anime, My Hero Academia, Blood, All Might Desktop Picture


The Art Print featuring the painting The Garden at Bougival by <PERSON>


The Porcelain Jars - Vigan, Ilocos Sur

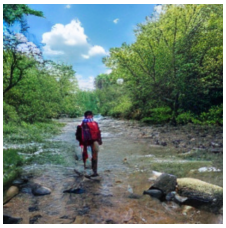

The Famous church of the good shepherd photographed in the night with Milky way in the background


Fruit Still Life With Two Peaches And Blue And Green Grapes On The Branch And A Vine Leaf Artwork by <PERSON>


The Wizard Nebula: Sharpless 142


Local visual arts fans have a treat awaiting them at the Asheville Mall through April 28.

Figure 9: **Samples from DREAM-H (1.1B)** generated from CC12M captions with CFG = 5.0. The same 20 captions are used in Figs. 8–10 for comparison across scales. Captions are shown verbatim; CC12M replaces person names with the token <PERSON>.


Hiking along the Chipola River

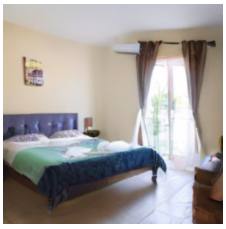

Skinny Broccoli and Mixed Vegetable Stir Fry - Skip takeout and make your own fast, easy, and healthy stir fry! Think of all the money and calories you'll save!!

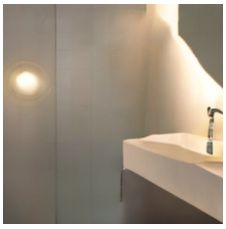

Home is Where the Art is IX <PERSON>


View of rooftop of Church of the Holy Sepulchre in Old City, Old City, UNESCO World Heritage Site, Jerusalem, Israel, Middle East


Shared Zone - road walking through some of Innes National Park


A soup bowl served with Egg drop soup topped with scallions and fried noodle


A bed or beds in a room at Chiaia 205


Unusual lighting can make the atmosphere in the bathroom too


The Ghost (bike) Rider Comic Book Characters, Comic Books, Ghost Rider 2, Scary, Creepy, Cycling Art, Road Cycling, Bicycle Art, Bike Rider

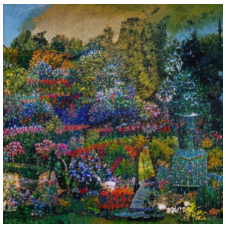

The Hunas Falls Hotel: Starting point of the waterfall


Tree Photograph - The Manzanita Tree by <PERSON>


HD Wallpaper American Flag with high-resolution 1920x1080 pixel. You can use this wallpaper 1920x1080


This Snow Covered Abandoned Mall Has Been Forgotten By Time


Wallpapers Anime, My Hero Academia, Blood, All Might Desktop Picture


The Art Print featuring the painting The Garden at Bougival by <PERSON>


The Porcelain Jars - Vigan, Ilocos Sur


The Famous church of the good shepherd photographed in the night with Milky way in the background


Fruit Still Life With Two Peaches And Blue And Green Grapes On The Branch And A Vine Leaf Artwork by <PERSON>


The Wizard Nebula: Sharpless 142


Local visual arts fans have a treat awaiting them at the Asheville Mall through April 28.

Figure 10: **Samples from DREAM-G (2.4B)** generated from CC12M captions with CFG = 5.0. The same 20 captions are used in Figs. 8–10 for comparison across scales. Captions are shown verbatim; CC12M replaces person names with the token <PERSON>.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's core contributions: identifying the masking-ratio conflict between the text-image contrastive and generative losses, unifying the two objectives with Masking Warmup and demonstrating their synergy, and Semantically Aligned Decoding, a zero-shot inference-time decoding strategy. Each claim is supported by the experiments in Section 5 and the appendices.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix A.1.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: A comprehensive model and training description are included in Section 4 and Appendix G, with training settings in Table 16.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: Code is available at https://github.com/chaoli-charlie/dream. All datasets used are public (Appendix H), and the paper additionally provides pseudocode for training all models in Appendix G.8.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All the training and testing details are provided in Section 4, Appendix G, and Tables 16–18.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: For the headline CC12M results, the 100M-pair GPIC results, and GenEval, we report the mean ± sample standard deviation (1-sigma) over 3 independent runs that vary initialization, data order, and generation noise (Tables 5 and 6 in Appendix B, and Table 12 in Appendix E.2). Ablation tables report single runs; Appendix B.1 discusses which ablation orderings exceed run-to-run noise.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides details of the compute resources in Appendix G.7.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research uses publicly available datasets, publicly available pretrained components, and does not involve human subjects or sensitive data collection.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Both positive and negative societal impacts are discussed in Appendix A.2.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We release only training and evaluation code; no pretrained model weights or new datasets are released, so the high-risk release scenarios covered by this question do not apply. Risks of text-to-image models in general are discussed in Appendix A.2.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The original papers that produced the assets used in the paper are cited. The license and terms of use are listed in Appendix H and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The only new asset is our code, released at `https://github.com/chaoli-charlie/dream` under the MIT License with documentation for data preparation, training, and evaluation. No new datasets or model checkpoints are released.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or any research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The research does not involve human subjects, so IRB approval is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not an important, original, or non-standard component of the method.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.