REPA-G: Test-Time Conditioning with Representation-Aligned Visual Features

Nicolas Sereyjol-Garros [1, 2] **Ellington Kirby** [1] **Victor Letzelter** [1, 3] **Victor Besnier** [1]

Nermin Samet [1]

[1] Valeo.ai, Paris, France
[2] LIGM, CNRS, Univ Gustave Eiffel, ENPC, Institut Polytechnique de Paris, France
[3] LTCI, Télécom Paris, Institut Polytechnique de Paris, France

Abstract

While representation alignment with self-supervised models has been shown to improve diffusion model training, its potential for enhancing inference-time conditioning remains largely unexplored. We introduce Representation-Aligned Guidance (REPA-G), a framework that leverages these aligned representations, with rich semantic properties, to enable test-time conditioning from features, in generation. By optimizing a similarity objective (the *potential*) at inference, we steer the denoising process toward a conditioned representation extracted from a pre-trained feature extractor. Our method provides versatile control at multiple levels of granularity, ranging from patch level matching via single patches to broad semantic guidance using global image feature tokens. We further extend this to multi-concept composition, allowing for the faithful combination of distinct concepts. REPA-G operates entirely at inference time with no additional training required, offering a flexible and precise alternative to often ambiguous text prompts or coarse class labels. Our approach achieves high-quality, diverse generations on ImageNet and COCO. Code is available at https://github.com/valeoai/REPA-G.

1 Introduction

In recent years, generative models have achieved impressive results in image synthesis [1, 2], reaching near photo-realistic quality. This progress has been driven in particular by diffusion models [3, 4] and flow matching [5, 6] approaches, which iteratively remove Gaussian noise from images during generation. Recently, a line of work has shown that, beyond the standard denoising objective, diffusion models can be trained to align their intermediate representations [7–11] using pre-trained self-supervised learning models such as DINOv2 [12]. This representation alignment has been shown to accelerate training and improve sample quality. However, it remains unclear how these aligned representations can be effectively leveraged at inference time.

Beyond improving sample quality, controllability has emerged as a central challenge in image synthesis. Current conditioning strategies typically rely on class labels [13] or textual descriptions [14]. Class-based conditioning provides only coarse control and is limited to predefined categories. Using text is more flexible but often imprecise and ambiguous (see the left panel of Figure 1).

In this work, we leverage representation alignment training in diffusion models for conditional image generation. Our method (REPA-G) enables test-time visual conditioning via inference-time optimization by exploiting the alignment between a diffusion model's internal representations and those of a vision encoder. By steering the diffusion process toward target feature tokens, we achieve

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

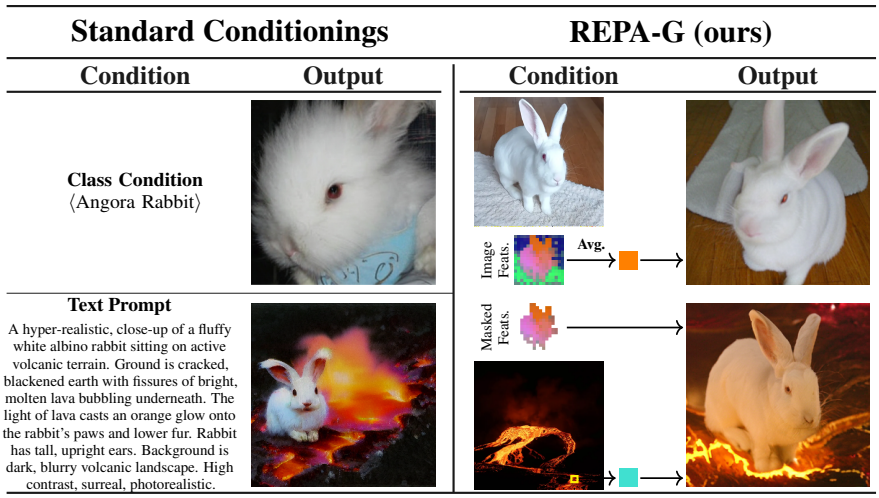


Figure 1: **Comparing class-label, text-prompt, and REPA-G conditioning.** All models are trained on ImageNet. *(Top)* We average extracted features (■) from a anchor image to generate a generic "rabbit" image. *(Bottom)* We combine a masked feature map with a specific "lava" patch (■) to synthesize a rabbit on a volcano. While text prompts require lengthy descriptions and often lack precision, our feature-based conditioning offers better compositional control and provides more precise generation.

control over generation without relying on a fixed set of class labels or potentially ambiguous textual prompts. This improves controllability while preserving flexibility and generality.

Specifically, we extract visual tokens from a real image using the same self-supervised learning network used during representation alignment training, for example DINOv2 [12]. We then guide generation using a single token or a set of tokens that capture the concept at varying levels of granularity . The model can be conditioned with multiple tokens extracted from a mask to preserve shape or pose. Finally, conditioning with the average of all image tokens provides a broad concept signal such as *rabbit*, *car* (see the top-right of Figure 1).

We achieve this by computing the gradient of a *potential* that quantifies the alignment between representations of generated sample and conditioning data points, corresponding to semantic *concepts* tokens. In addition to predicted scores and the velocities, the sampling procedure is guided by this gradient. We then generalize the idea to multiple potentials by aligning with several concept tokens simultaneously. Our proposed method pulls relevant features toward the specified concepts enabling faithful generation, and makes it possible to generate flexible compositions (Figure 1, bottom right).

In summary, we introduce the first test-time visual conditioning framework for diffusion models trained with representation alignment. Our contributions are as follows:

We introduce Representation-Aligned Guidance (REPA-G). REPA-G leverages representation alignment during sampling to align internal diffusion features with targets from a vision encoder. Our method operates entirely at inference time without requiring additional parameters or finetuning.

We analyze the properties of the representation space for feature conditioning. We justify that the embedding space provided by self-supervised feature alignment is well suited for this task.

We conduct extensive experiments, showing control over concrete and abstract visual concepts. Our framework handles objects and background semantics, enabling both spatial and concept-level guidance as well as concept composition.

2 Related works

Representation alignment for generation. Image synthesis is dominated by diffusion and flow matching models [4, 5]. These methods enable photorealistic image generation at the cost of a slow generation speed and a long training time [15]. Recently, REPA [16] argues that the training speed

of latent diffusion models is largely constrained by how quickly they learn a meaningful internal image representation [7]. To mitigate this, they guide the learning by aligning intermediate model projections with feature maps from a pre-trained encoder, which accelerates training and improves generation quality. iREPA [17] shows that spatial structure in representations correlates more strongly with generation quality than global information. They use a convolutional projection instead of a MLP and a spatial normalization layer. VA-VAE [18] applies a latent alignment loss solely in the VAE latent space prior to diffusion training. Finally, REPA-E [8] enables end-to-end training by backpropagating the REPA loss through both the diffusion model and the VAE encoder. This jointly shapes their internal representations, yielding better alignment and higher-quality generations. These representation-alignment methodologies have since been extended to various modalities beyond static images, including video [10], LiDAR point clouds [11], and audio synthesis [19].

Conditional generation in diffusion models. Beyond sample quality, recent research has increasingly emphasized controllability as a central aspect of image synthesis, often considered as important as visual fidelity or diversity [20, 21]. Early adaptations of diffusion models to image generation were largely unconditional [4]. Subsequent works introduced conditioning through auxiliary classifiers [22] or classifier-free guidance [23]. Nowadays, diffusion models are commonly conditioned during training on various modalities, including class labels [22], textual descriptions [14, 24], or reference images for image editing [25]. These mechanisms typically require task-specific training and architectural modifications that remain fixed at inference. An alternative line of work explores post-training strategies [26–28], which enable conditioning on structural signals like depth or edge maps. While effective, these approaches introduce additional parameters and require extra post-training steps, increasing both model complexity and computational cost, making them not directly comparable and orthogonal to our approach.

Our method enables test-time control in trained flow models, without additional training or parameters, provided that representation alignment holds. This is orthogonal to image inpainting methods [29, 30], which typically require explicit training on masked data and text prompts, often using Stable Diffusion models [14] pre-trained on large-scale, scene-centric datasets. In contrast, REPA-G targets general guidance in generation, allowing conditioning on arbitrary features rather than text.

To the best of our knowledge, we are the first to explore such test-time conditioning with visual features in this context. TFG [31] provides a unified framework for various training-free methods, but it does not explore visual feature guidance.

[32] study the interpretability of intermediate features in an unconditional diffusion model, rather than proposing a test-time conditioning method.

3 Preliminaries

Let p_0 be a (clean) data distribution on $\mathcal{X}$, which may correspond to pixel space in image generation. We describe hereafter the training of a flow matching model to be able to produce new samples from $p_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to p_0 while being "representation-aligned" with a pretrained backbone ϕ. For more details on the training and sampling of flow models see App. A.

The flow model v_θ can be expressed as $v_\theta = g_\theta \circ f_\theta$ where $f_\theta : \mathcal{X} \times [0,1] \to \mathcal{Z}$, projects input into a representation $f_\theta(x_t, t)$ at a given layer, and $g_\theta : \mathcal{Z} \to \mathcal{X}$ process this representation to construct the prediction. REPA [16] suggests optimizing the flow matching denoising loss $\mathcal{L}^{\text{diff}}(\theta)$ along with an alignment loss using an additional projection layer $h_\theta : \mathcal{Z} \to \mathcal{Z}'$:

$$\mathcal{L}^{\text{align}}(\theta) = -\mathbb{E}_{\substack{t\sim\mathcal{U}(0,1) \\ (x_0,x_1)\sim p_0\times p_1}} \Big[\mathscr{V}\big((h_\theta \circ f_\theta)(x_t, t),\, \phi(x_0)\big)\Big]. \tag{1}$$

in a compound loss $\mathcal{L}^{\text{diff}}(\theta) + \beta\mathcal{L}^{\text{align}}(\theta)$, with $\beta > 0$.

In (1), $\mathscr{V} : \mathcal{Z}' \times \mathcal{Z}' \to \mathbb{R}$ is a *potential* that measures similarity between the features, predicted by f_θ and projected with h_θ, with those of a pretrained (frozen) backbone $\phi : \mathcal{X} \to \mathcal{Z}'$ evaluated on the *clean* image. When $\mathcal{Z}' = \mathbb{R}^{N\times d}$, N is the number of features and d is the features dimension, $\mathscr{V}$ is typically expressed as an average patch-wise similarity $\mathscr{V}(h_t, h^\star) = \frac{1}{N}\sum_{n=1}^{N}\langle [h_t]_n, [h^\star]_n\rangle$ where $\langle\cdot,\cdot\rangle$ denotes the euclidean dot product on $\mathbb{R}^d$ and $[h]_n$ is the n-th row of $h \in \mathbb{R}^{N\times d}$. Importantly, $[h_t]_n$ and $[h^\star]_n$ are set to have unit norm.



Figure 2: **Impact of representation alignment on feature space.** We perform k-means clustering ($k = 1{,}000$) on four feature spaces across ImageNet (see App. C.1). For a reference image (with red frame), we visualize others from its assigned cluster. Without alignment, SiT fails to form semantic groupings, making its latent space unsuitable for conditioning. In contrast, the aligned model successfully replicates the teacher's semantic structure both before and after the projection layer, resulting in semantically consistent clusters.

4 Guiding generation at inference

We perform inference by leveraging flow models trained with representation alignment, enabling generation conditioned on features $\phi(x_c)$ from a reference image x_c. This approach is analogous to classifier guidance [3, 22], but extends the paradigm to continuous features rather than discrete classes.

We formulate the time dependent *posterior* distribution with respect to the condition $h^\star$ as $p_t(h^\star|x;\lambda) \propto \exp\left(\lambda \mathscr{V}\left((h_\theta \circ f_\theta)(x,t), h^\star\right)\right)$ with $\lambda > 0$ controlling the strength of the conditioning. To generate samples x conditioned on $\phi(x_c)$ in accordance with classifier guidance literature [15, 22], we introduce a guidance term that modifies the score function $\nabla_x \log p_t(x_t)$ used during sampling as:

$$\nabla_x \log p_t(x_t) + \lambda \nabla_x \mathscr{V}\left((h_\theta \circ f_\theta)(x_t,t), \phi(x_c)\right) \ , \tag{2}$$

The guidance term allows to steer the samples toward regions of low energy while the unconditional score transport them toward the image manifold. Under assumptions provided in App. B.1, sampling with this modified score allow to recover the tilted distribution $p_0(x|h^\star;\lambda)$. It results in an effective way to condition the generation as shown in the 2D toy example in App. B.2. We also show in Section 7 that it allows to generate high quality images with high semantic alignment with the condition. Details of the algorithm are provided in App. B.3.

5 Properties of representation space

We study how representation alignment [16] shapes the internal feature space of diffusion transformers and how this impacts feature conditioning. To this end, we use SiT models [6] trained on ImageNet [33]. In this setting, the generation is done in the latent space of an autoencoder, but the visual backbone ϕ takes images as input.

We argue that effective conditioning requires: (i) a semantically meaningful embedding space, where nearby embeddings correspond to similar concepts; and (ii) a smooth mapping from conditioning features to conditional distributions, so similar embeddings yield similar conditional densities.

Representation alignment enables semantic features in diffusion transformers. Self-supervised vision models learn rich representations from large-scale unlabeled data by enforcing invariance to augmentations while preserving semantic content. Among them, DINOv2 provides particularly strong semantic features and is widely used for representation alignment in diffusion models [7–9, 16].

In contrast, the diffusion denoising objective alone rarely produces semantic features. Figure 2 highlights this difference: clustering ImageNet with DINOv2 features yields coherent, concept-level groups, whereas clustering SiT's internal features produces clusters non-semantic clusters. Training SiT with a representation alignment loss restores semantic information in its internal feature space.

Representation alignment ensures a well-conditioned mapping from features to image densities. Following [32], we evaluate the Euclidean embedding property to ensure the conditioning space structure transfers to the conditional densities. This requires the distance d^2 between the conditional

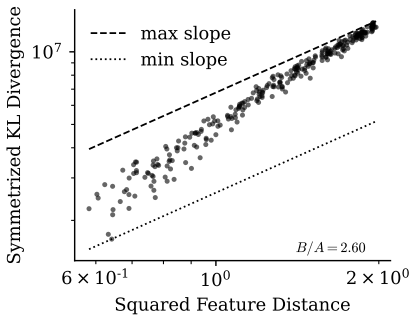


Figure 3: **Correlation between embedding and density distances.** The narrow range of the B/A ratio indicates a well-conditioned space where distances in any direction behave consistently.



Figure 4: **Semantic interpolation in the feature space.** Samples are generated by bilinearly interpolating the global conditioning features between four anchor images. The smooth transitions show the semantic coherence and stability of the mapping between the embedding space and conditional densities.

densities $p_1 = p(\cdot|\phi(x_1))$ and $p_2 = p(\cdot|\phi(x_2))$ to scale with the squared distance between $\phi(x_1)$, and $\phi(x_2)$. Specifically, there must exist $0 < A \leq B$ and B/A not too large such that for all x_1, x_2:

$$A\|\phi(x_1) - \phi(x_2)\|^2 \leq d^2(p_1, p_2) \leq B\|\phi(x_1) - \phi(x_2)\|^2 \ . \tag{3}$$

We define the density dissimilarity as:

$$d^2(p_1, p_2) = \frac{\lambda}{N} \sum_{n=1}^{N} \langle [\mathbb{E}_{p_1}[\phi(x)] - \mathbb{E}_{p_2}[\phi(x)]]_n, [\phi_1 - \phi_2]_n \rangle \ , \tag{4}$$

with more details in App. C.2. Figure 3 shows a strong relation between embedding and density distances, with a tight ratio $B/A = 2.60$, confirming the stability of the conditional space.

We further validate the semantic coherence of the embedding space and the smoothness of the density mapping in Figure 4 by interpolating between conditioning features. The smooth transitions confirm the meaningful structure of the embedding space and its stable mapping to conditional densities.

6 Design choices for the Potential $\mathscr{V}$

We have presented a guidance method for test-time visual conditioning within the representatively aligned class of diffusion models. While Section 3 outlined the required properties of the feature extractor ϕ, the model is specifically trained to align with the potential:

$$\mathscr{V}(h, h^\star) = \frac{1}{N} \sum_{n=1}^{N} \langle [h]_n, [h^\star]_n \rangle \ , \tag{5}$$

where $h, h^\star \in \mathbb{R}^{N \times d}$. In this section, we discuss how (5) can be adapted during inference to achieve specific characteristics in the generated images.

6.1 Guidance via Independent Patch Alignment (IPA)

Eq. (5) can be generalized to allow for spatial flexibility:

$$\mathscr{V}_P(h, h^\star) = \sum_{n=1}^{N} \sum_{m=1}^{N} P_{n,m} \langle [h]_n, [h^\star]_m \rangle \ , \tag{6}$$

where $P \in \mathbb{R}^{N \times N}$ is a weight matrix such that $P_{n,m}$ defines the interaction strength between predicted patch n and conditioning patch m, normalized such that $\sum_n \sum_m P_{n,m} = 1$. By manipulating P, we achieve fine-grained control over the generation process. Because P is determined independently of the feature alignments, we refer to this class of conditioning as *Independent Patch Alignment* (IPA). We detail three specific configurations of P below.

Alignment with full feature map. Setting $P = \frac{1}{N} I$ recovers the original alignment objective (5). This configuration enforces a dense, spatial constraint, resulting in a stochastic reconstruction of the conditioning image.

Alignment with feature mask. When only specific regions of the conditioning image are relevant, we define a binary mask $m \in \{0,1\}^N$ and its corresponding index set $\mathscr{S} = \{n \mid m_n = 1\}$. The weights are defined as $P_{n,m} = \frac{1}{|\mathscr{S}|}\mathbf{1}[m = n]\mathbf{1}[n \in \mathscr{S}]$. This *masked-conditioning* potential preserves information within the masked region while allowing for variation in the surrounding areas.

Alignment with an average concept. To steer generation toward the global semantic content of a reference image without enforcing pixel-wise spatial fidelity, we utilize an *average-concept* potential. Here, we set $P_{n,m} = \frac{1}{N^2 \|\bar{h}\| \|\bar{h^\star}\|}$, which is equivalent to aligning the spatial average of the predicted features with the spatial average of the conditioning features, $\bar{h}^\star$. This preserves the general semantic category while removing all spatial constraints.

Alignment with a single concept. Alternatively, we consider *single-concept* alignment, where the full generated feature map is compared against a single target patch i. This corresponds to $P_{n,m} = \frac{1}{N}\mathbf{1}[m = i]$, focusing the guidance signal on a local feature from the conditioning image.

6.2 Guidance via Selective Patch Alignment (SPA)

While IPA uses constant weighting, *Selective Patch Alignment* dynamically weights predicted patches based on their similarity to a target concept $h_i^\star$, as follows:

$$\mathscr{V}^{\mathrm{SPA}}(h, h^\star) = T \log \left[\sum_{n=1}^{N} \exp\left(\frac{\langle [h]_n, [h^\star]_i \rangle}{T} \right) \right] , \tag{7}$$

where the temperature T modulates selection sparsity. As $T \to 0$, the potential recovers a hard-maximum over patches; as $T \to \infty$, it converges to the uniform average-concept baseline. This soft-maximum mechanism allows the model to adaptively localize concepts, bypassing the rigid spatial constraints of the conditioning image. More intuition and justifications are given in App. D.1.

7 Experiments

Datasets. We evaluate on ImageNet [33], which contains 1.2M images across 1,000 classes, and on COCO [34], which includes 123K images. Both datasets are preprocessed to a resolution of 256×256.

Implementation details. We evaluate the performance of our proposed test-time visual feature guidance mechanisms by analyzing their generated outputs. For our experiments, we use a pretrained flow matching models with representation alignment on the ImageNet. We first evaluate the conditional generation capabilities of REPA-G across varying granularities of feature types. We then explore the ability of our method to compose features from multiple images. We follow [6] by using a Stochastic Differential Equation integrated via the Euler-Maruyama solver with 250 steps.

7.1 Test-time guidance with single source

In this series of experiments, we use a single "anchor image" whose features guide the generation process. We generate images using guidance mechanisms derived from Independent Patch Alignment (IPA), including: (i) Full Feature Map, utilizing all patch features; (ii) Masked Feature Map, utilizing a subset of patch features; and (iii) Average Feature Map, utilizing the global average of all patch features. To generate the masked feature maps, we extract DINOv2 [12] patch features, compute the first principal component, and apply a threshold set to 0.5 to isolate the foreground [35].

We evaluate three flow matching models: a standard SiT-XL [6] (trained without representation alignment), REPA [7] and REPA-E [8], both trained with representation alignment. In Table 1 and Table 2, we remove class conditioning to focus solely on visual conditioning, while App. E.3 reports results with both class and visual conditioning. For visual conditioning, we extract internal features ($\mathrm{F}_{\mathrm{SiT}}$), at $t = 0$, from the same backbone layer across all models, specifically the layer before the projection layer used in the REPA variants, i.e. $f_\theta(x_t, t)$. For the REPA-based models, we also

Table 1: **Distribution-level comparison of REPA-G generations on ImageNet [33].** We compare the standard SiT backbone [6] (no representation alignment) against REPA [7] and REPA-E [8] variants. For each method block, the first row denotes results for unconditional generation, serving as a baseline, and $\mathbf{F_{SiT}}$ and $\mathbf{F_{DINO}}$ for REPA-G.

		Full Feature Map					Masked Feature Map					Average Feature Map				
Model	**Cond.**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
SiT	-	**45.02**	**9.02**	**22.33**	**0.50**	**0.63**	**45.02**	**9.02**	22.33	**0.50**	**0.63**	**45.02**	**9.02**	**22.33**	**0.50**	**0.63**
	F_{SiT}	71.35	72.08	14.73	0.33	0.54	45.85	19.23	**29.36**	0.40	0.62	65.99	22.52	16.69	0.36	0.52
REPA	-	26.58	6.85	41.31	0.56	**0.70**	26.58	6.85	41.31	0.56	**0.70**	26.58	6.85	41.31	0.56	**0.70**
	$\mathbf{F_{DINO}}$	7.23	8.86	199.69	0.65	0.69	14.17	10.59	135.08	0.58	0.64	29.06	16.79	83.85	0.46	0.69
	$\mathbf{F_{SiT}}$	**2.09**	**6.17**	**260.97**	**0.74**	**0.70**	**2.67**	**4.86**	**222.71**	**0.74**	0.69	**6.26**	**6.14**	**159.14**	**0.68**	**0.70**
REPA-E	-	15.07	4.46	55.14	0.65	**0.69**	15.07	4.46	55.14	0.65	**0.69**	15.07	**4.46**	55.14	0.65	**0.69**
	$\mathbf{F_{DINO}}$	**1.45**	**4.07**	**264.69**	**0.76**	**0.69**	2.30	4.84	212.25	0.75	0.67	3.24	5.20	188.92	0.73	0.67
	$\mathbf{F_{SiT}}$	2.15	6.64	264.48	0.75	0.68	**1.79**	**4.13**	**237.37**	**0.76**	0.68	**2.50**	4.81	**201.13**	**0.74**	**0.69**

Table 2: **Instance-level comparison of REPA-G generations on ImageNet [33].** Setup is the same as in Table 1. Alignment is measured with DINOv2 [12], JEPA [36], and CLIP [37] feature spaces, supplemented by PSNR for pixel-level fidelity. We highlight ours conditioning methods with $\mathbf{F_{SiT}}$ and $\mathbf{F_{DINO}}$.

		Full Feature Map				Masked Feature Map				Average Feature Map			
Model	**Cond.**	**DINOv2**	**JEPA**	**CLIP**	**PSNR**	**DINOv2**	**JEPA**	**CLIP**	**PSNR**	**DINOv2**	**JEPA**	**CLIP**	**PSNR**
SiT	F_{SiT}	0.27	0.36	0.43	15.35	0.44	0.48	0.46	17.98	0.12	0.35	0.83	7.74
REPA	$\mathbf{F_{DINO}}$	0.75	0.61	0.58	11.02	0.77	0.60	0.57	10.63	0.76	0.69	0.92	8.00
	$\mathbf{F_{SiT}}$	**0.85**	**0.78**	**0.71**	**20.46**	**0.87**	**0.79**	**0.71**	**20.72**	**0.84**	**0.84**	**0.95**	**10.82**
REPA-E	$\mathbf{F_{DINO}}$	0.83	0.69	0.64	15.23	0.85	0.68	0.63	14.36	**0.91**	0.83	**0.96**	10.97
	$\mathbf{F_{SiT}}$	**0.86**	**0.76**	**0.71**	**18.68**	**0.88**	**0.78**	**0.71**	**19.07**	**0.91**	**0.88**	**0.96**	**11.96**

evaluated features extracted after the projection layer (i.e. $h_\theta(f_\theta(x_t, t))$) but observed no significant difference in performance; further details are provided in App. E.2. Additionally, for the aligned models, we extend our evaluation to include DINOv2 features (F_{DINO}) as an external conditioning.

Distribution-level evaluation. We first evaluate the generated outputs at the distribution level using standard metrics, FID [38], sFID [39], IS [40], Precision, and Recall [41]. Distribution level evaluation is non-trivial, as a clear target conditional density is unavailable. To address this, we approximate the unconditional target distribution by aggregating the conditional densities across anchor features. We consider three feature granularities: (i) full feature maps, (ii) masked feature maps, and (iii) average features. These conditioning signals are extracted from 50,000 anchor images sampled uniformly across all classes from the ImageNet training set. Using these anchor features, we generate 50,000 images with REPA-G and compare them against the whole dataset.

We report results on ImageNet (Table 1) and present a zero-shot evaluation on COCO (Table 3) using the same models. The first rows show that SiT does not respond well on REPA-G due to an ambiguous feature space, as the FID increases with IPA. In contrast, REPA and REPA-E benefit from feature alignment, allowing them to better react to the conditioning across all granularity levels.

Instance-level evaluation. To verify that the generated samples follow the given conditions, we evaluate quality at the instance level using feature extractors DINOv2 [12], JEPA [36], and CLIP [37]. We compute the average alignment score (i.e., cosine similarity) between the features of each anchor image and the corresponding generated image across three granularities. For the average feature map, alignment is calculated between global average features; for the full feature map, it is computed as the mean of all patch-wise similarities; and for the masked setup, it is the mean of patch similarities within the masked region. Furthermore, we report PSNR to measure the low-level pixel fidelity.

The results in Table 2 indicate that achieving high similarity to the anchor images requires a well-structured representation space. The vanilla SiT model fails to follow the anchor images, whereas REPA and REPA-E nearly perfectly reconstruct the condition across all levels of granularity. Figure 5 (left) presents qualitative results of IPA, demonstrating that our method effectively conditions generation across each setting (additional examples provided in App. E.11).

Information density of visual features and text. Visual features are expected to provide more information than long and detailled caption. To confirm this intuition, we compare REPA-G against the text-to-image (T2I) model CAD-I [42] as a baseline in Table 4. Unlike most T2I models trained

Table 3: **Distribution-level zero-shot evaluation on the COCO [34] dataset.** The setup is the same as in Table 1. Within each block, the first row reports unconditional generation results as a baseline, while $\mathbf{F}_{\text{SiT}}$ and $\mathbf{F}_{\text{DINO}}$ denote our method.

		Full Feature Map					Masked Feature Map					Average Feature Map				
Model	**Cond.**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
SiT	-	**45.13**	**30.15**	**22.33**	**0.42**	**0.54**	**45.13**	**30.15**	**22.33**	**0.42**	**0.54**	**45.13**	**30.15**	**22.33**	**0.42**	**0.54**
	$\mathbf{F}_{\text{SiT}}$	77.26	108.79	9.86	0.25	0.37	51.88	42.36	15.76	0.27	0.52	66.09	46.88	16.67	0.28	0.45
REPA	-	37.85	29.11	**41.31**	0.45	0.58	37.85	29.11	**41.31**	0.45	0.58	37.85	29.11	**41.31**	0.45	0.58
	$\mathbf{F}_{\text{DINO}}$	12.96	29.05	31.11	0.55	0.57	20.51	31.04	26.15	0.45	0.53	32.04	39.05	20.59	0.35	0.53
	$\mathbf{F}_{\text{SiT}}$	**6.18**	**25.35**	33.08	**0.65**	**0.62**	**6.63**	**24.37**	32.05	**0.65**	**0.61**	**9.46**	**25.89**	29.43	**0.60**	**0.60**
REPA-E	-	36.23	29.98	**55.13**	0.51	0.59	36.23	29.98	**55.13**	0.51	0.59	36.23	29.98	**55.13**	0.51	0.59
	$\mathbf{F}_{\text{DINO}}$	**4.61**	**23.27**	35.97	**0.67**	**0.64**	5.65	23.35	34.54	0.64	0.63	6.07	24.09	32.76	0.63	**0.63**
	$\mathbf{F}_{\text{SiT}}$	5.70	25.55	34.29	0.66	0.62	**4.92**	**23.03**	34.78	**0.66**	**0.63**	**5.45**	**23.98**	33.81	**0.64**	**0.63**



Figure 5: **Qualitative comparison on ImageNet [33] using REPA-E [8] with DINOv2 features. (Left)** We show single-source conditioning across three levels of granularity: full, masked, and averaged feature maps given *anchor image* and *input mask*. **(Right)** We illustrate multi-source composition. Within each group, the anchor object and target background are shown at a small scale alongside the enlarged generation. We use full features from the anchor and a single feature patch sampled from the target's background.

on massive, scene-centric datasets, CAD-I is specifically trained on ImageNet. We use long captions extracted by [42] from the COCO validation set as conditioning signals for CAD-I to generate the described scenes. For our method, we utilize both full and average feature maps extracted from the internal SiT representations before the projection layer trained with REPA-E to generate the corresponding images. The performance gap between text-to-image and visual feature conditioning shows that visual features provide as expected a denser, more informative signal than text captions.

Resolutions scalability. Although 256×256 resolution is the standard benchmark in generative modeling research, we additionally fine-tune our 256×256 model to generate 512×512 outputs for 40k steps in order to evaluate scalability to higher resolutions. The results in App. E.7 confirm that our method scales effectively to higher resolutions, reducing FID from 33 to 5.

Generalization across semantic vision encoders. To assess whether REPA-G depends on a specific semantic vision encoder, we evaluate it on three SiT-XL/2 models trained with REPA and aligned with DINOv2, CLIP, and Perception Encoder [43] (PE), respectively. As shown in Table 6, REPA-G consistently improves generation quality across all three representation spaces. Notably, these models are trained for only 400k steps, 10× fewer than our main models. This indicates that effective conditioning does not require fully converged representation alignment; once the internal features acquire sufficient semantic organization, they already provide a useful guidance signal.

Generalization beyond REPA-style models. Our guidance mechanism is not inherently restricted to models trained with a REPA-style objective. Rather, it requires internal representations that form a semantically organized and sufficiently noise-robust feature space, such that feature similarity provides a meaningful guidance signal. To support this broader applicability, we evaluate our method on Self-Flow [44], which learns semantic internal representations without relying on the standard REPA objective or an additional pretrained self-supervised backbone. Self-Flow employs a teacher–

Table 4: **Comparison with text-to-image (T2I) generation.** We compare our method against CAD-I [42], an ImageNet-trained T2I baseline, using COCO validation captions as conditioning signals. Our method utilizes full and average feature maps extracted from REPA-E [8].

Model	Cond.	FID ↓	sFID ↓	IS ↑	Prec. ↑	Rec. ↑
CAD-I	Text	37.98	29.87	25.60	0.46	0.37
REPA-E	Avg. Feat.	3.58	11.10	32.69	0.75	0.78
REPA-E	Full Feat.	**2.11**	**8.35**	**36.01**	**0.98**	**0.96**

Table 5: **Compositional generation evaluation on ImageNet [33].** We assess visual quality with PickScore and concept adherence using CLIP, and compare REPA-G variants using IPA and SPA against standard baselines.

Cond.	Pick Score	CLIP Score	Target Sim.	Anchor Sim.	Combined Sim.
Uncond.	0.175	0.100	0.098	0.113	0.105
Class Cond.	**0.198**	0.220	0.110	0.243	0.177
Interp.	0.188	0.190	0.156	0.183	0.170
IPA→IPA	0.196	0.232	0.372	**0.452**	0.412
IPA→SPA	0.196	**0.238**	**0.510**	0.422	**0.466**

student framework with asynchronous noise timesteps, where the student predicts representations produced by the teacher from a less corrupted input. Our guidance transfers directly to these learned representations, and, as shown in Table 7, full-map guidance substantially improves generation quality. These results indicate that the proposed guidance is not tied to a particular representation-alignment objective, but can be applied more broadly when the generative model provides a semantically structured internal feature space. We provide qualitative results in Figure 11.

Table 6: **Generalization across semantic alignment backbones.** We present the FID and DINOv2 Alignment scores for REPA models aligned with CLIP, PE, and DINOv2, evaluating unconditional generation and different feature-conditioning granularities (Full, Average, Masked).

Cond.	FID ↓			DINOv2 Align. ↑		
	CLIP	PE	DINOv2	CLIP	PE	DINOv2
-	40.0	44.6	36.2	-	-	-
Full	8.3	22.2	4.8	0.5	0.4	0.7
Average	18.7	19.1	9.8	0.6	0.6	0.7
Masked	12.0	18.1	20.6	0.6	0.4	0.7

Table 7: **Generalization beyond REPA-style models.** We apply REPA-G on Self-Flow [44] with full feature map conditioning and show that it can extend to model trained with non REPA-style objective provided that the learned internal representation is semantic and noise-robust.

Cond.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	DINOv2 Align.↑
REPA-E						
-	15.07	4.46	55.14	0.65	0.69	-
Full Feat.	2.15	6.64	264.48	0.75	0.68	0.86
Self-Flow						
-	35.51	5.79	27.62	0.55	0.63	-
Full Feat.	5.49	4.71	161.02	0.71	0.68	0.72

Comparison to training-free baseline. To the best of our knowledge, TFG [31] is the only fair baseline that, similar to REPA-G, does not require additional parameters or training steps. We adapt the official implementation of TFG to support feature-based guidance, using DINOv2 [12] as the target predictor. However, in our setup, TFG does not produce images of sufficient visual quality, with FID scores around 80. We provide the full experimental setup and results in App. E.5.

Comparison with adapter-based conditioning. IP-Adapter-style methods provide a relevant reference-based generation baseline, but differ substantially from REPA-G in both their assumptions and training requirements. Despite requiring additional training, IP-Adapter [27] yields an FID that is 6 points higher, 25% lower DINOv2 alignment on ImageNet, and lower generation throughput than REPA-G. Further details are provided in App. E.6.

7.2 Test-time guidance with multiple sources

We evaluate concept-blending by composing anchor images (source objects) with target features (backgrounds). We apply two simultaneous conditioning: first, an object-level IPA using the anchor's DINO features, and a target background condition implemented via IPA or SPA. In this multi-source setting, in every denoising step, we sum up both potentials and compute once the gradient of the resulting joint potential that incorporates both the anchor and the target. The resulting images should preserve the anchor object while adopting the target background's characteristics. We evaluate on a 50-class ImageNet subset representing diverse objects. For each class, 100 random samples are combined with 8 hand-selected target background features, yielding 40,000 images per method.

As novel image compositions are steering away from the real distribution, FID is inapplicable. We therefore evaluate image quality via Pick Score [45] and concept adherence via CLIP Score [46], using the prompt: *An image of a/an [Anchor Class] with a [Target Class] background.* Additionally,

we introduce patch-wise similarity metrics: each generated patch is assigned to the anchor or target based on maximum cosine similarity. We then compute Anchor, Target, and Combined similarities to quantify feature adherence to each source.

We compare our method against three baselines: (i) unconditional, (ii) class-conditional (using the anchor class), and (iii) interpolated conditional generation, which blends class embeddings via adapted Classifier-Free Guidance. Our compositional conditioning is evaluated using two variants: IPA followed by IPA (IPA→IPA) and IPA followed by SPA (IPA→SPA). As shown in Table 5, REPA-G significantly outperforms baselines in prompt alignment and image quality. While class-conditional baselines are limited by an inability to encode specific target backgrounds (e.g., a tiger with a water background), REPA-G leverages DINO features to achieve superior anchor and target alignment. Notably, the SPA variant is most effective at blending sources, yielding the highest CLIP Score (see App. E.9). Figure 5 (right) highlights that REPA-G enables multi-source composition.

Hyperparameters. We run an hyparameter sweep for single and multi-source conditioning. Similar to classifier guidance [22], the parameter λ controls the strength of the conditioning. For single-source conditioning, we find that increasing λ in (2) improves FID up to a saturation point, after which performance plateaus, while preserving both precision and recall thanks to the anchor. Results are shown in App. E.1.

For multi-source conditioning, we found a low optimal value for temperature $T = 0.1$ indicating that SPA is indeed beneficial as a large temperature would recover the IPA→IPA variant. While in practice the user may have to tune the λ of each energy to balance conditioning contributions, we provide a default setting used for the experiment with both guidance scale set to $\lambda = 10{,}000$.

Runtime. REPA-G predicts both the features and the velocity field in a single forward pass, with feature extraction occurring early in the network at the 8th transformer block. This significantly reduces the cost of the backward pass making REPA-G only 26% slower than unconditional generation and faster than other guidance mechanisms. We provide a detailed runtime analysis in App. E.8.

Reconstruction–generation trade-off. Full-map conditioning with large λ approaches stochastic reconstruction, while weaker guidance or average-feature conditioning permits increasingly diverse generations. Reducing λ or using average-feature conditioning relaxes spatial constraints and increases variation across samples from the same anchor, yielding a controllable reconstruction-generation trade-off. We provide the full analysis, quantitative and qualitative results in App. E.4.

8 Conclusion

We introduced REPA-G, an inference-time framework that leverages aligned self-supervised representations to condition diffusion models. By guiding the sampling process, REPA-G enables semantic control without the need for retraining. Our work show that feature conditioning, while impracticable with a vanilla model, becomes possible with a well-aligned feature space and this without being the initial motivation for such a training. While REPA-style methods are primarily introduced for training, we show that they can also provide inference-time guidance at no additional cost.

Limitations. For multi-source composition, complex blending scenarios may require the user to tune the λ hyperparameter, which introduces a slight overhead. Besides, more strongly conflicting conditions remain challenging: incompatible identities may blend, spatial or viewpoint constraints can induce competing gradients, one condition may dominate, pose control requires spatially corresponding features, and far out-of-distribution combinations may produce artifacts (see App. E.10 for failure cases). Addressing these limitations and integrating our guidance with orthogonal conditioning methods represent promising directions for future research.

Impact statement. This paper presents an image conditioning method which unlocks the full potential of image generation models, but these capabilities may be misused to produce harmful content and can amplify biases present in the training data.

Acknowledgments and Disclosure of Funding

We acknowledge EuroHPC Joint Undertaking for awarding the project ID EHPC-REG-2024R02-234 access to Karolina, Czech Republic. We thank the reviewers for their insightful comments.

References

[1] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[2] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *NeurIPS*, 2020.

[5] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

[6] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024.

[7] Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, Kai Wang, and Yang You. REPA Works Until It Doesn't: Early-Stopped, Holistic Alignment Supercharges Diffusion Training. *NeurIPS*, 2025.

[8] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for End-to-End Tuning with Latent Diffusion Transformers. *ICCV*, 2025.

[9] Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-REPA: Aligning Diffusion U-Nets to ViTs. *NeurIPS*, 2025.

[10] Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. *ICLR*, 2026.

[11] Nicolas Sereyjol-Garros, Ellington Kirby, Victor Besnier, and Nermin Samet. R3DPA: Leveraging 3d representation alignment and rgb pretrained priors for LiDAR scene generation. *ICRA*, 2026.

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

[13] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022.

[15] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[16] 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. *ICLR*, 2025.

[17] Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What Matters for Representation Alignment: Global Information or Spatial Structure? *ICLR*, 2025.

[18] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. Generation: Taming Optimization Dilemma in Latent Diffusion Models. In *CVPR*, 2025.

[19] Tri Ton, Ji Woo Hong, and Chang D Yoo. Taro: Timestep-adaptive representation alignment with onset-aware conditioning for synchronized video-to-audio synthesis. In *International Conference on Computer Vision (ICCV)*, 2025.

[20] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023.

[21] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *ICLR*, 2023.

[22] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[24] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022.

[25] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[26] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[27] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arxiv:2308.06721*, 2023.

[28] Nick Stracke, Stefan Andreas Baumann, Joshua Susskind, Miguel Angel Bautista, and Björn Ommer. Ctrloralter: conditional loradapter for efficient 0-shot control and altering of t2i models. In *ECCV*, 2024.

[29] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *CVPR*, 2023.

[30] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, 2023.

[31] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified Training-Free Guidance for Diffusion Models. *NeurIPS*, 2024.

[32] Zahra Kadkhodaie, Stéphane Mallat, and Eero Simoncelli. Elucidating the representation of images within an unconditional diffusion model denoiser. *arXiv preprint arXiv:2506.01912*, 2025.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[34] 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 *ECCV*, 2014.

[35] Yunqiu Lv, Jing Zhang, Nick Barnes, and Yuchao Dai. Weakly-supervised contrastive learning for unsupervised object discovery. *Trans. Img. Proc.*, 2024.

[36] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *CVPR*, 2023.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[38] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[39] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In *ICML*, 2021.

[40] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training generative adversarial networks. In *NeurIPS*, 2016.

[41] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *NeurIPS*, 2019.

[42] Lucas Degeorge, Arijit Ghosh, Nicolas Dufour, David Picard, and Vicky Kalogeiton. How far can we go with imagenet for text-to-image generation? *arXiv preprint arXiv:2502.21318*, 2025.

[43] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. *NeurIPS*, 2025.

[44] Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *ICML*, 2026.

[45] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *NeurIPS*, 2023.

[46] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[47] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[48] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *NeurIPS*, 2014.

[49] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *NeurIPS*, 2020.

[50] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2022.

[51] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 2011.

[52] Kieran Didi, Francisco Vargas, Simon Mathis, Vincent Dutordoir, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. A framework for conditional diffusion modelling with applications in motif scaffolding for protein design. In *NeurIPS 2023 Workshop*, 2023.

[53] L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales*, volume 2. Cambridge university press, 2000.

[54] Luc Devroye. Chapter 4 nonuniform random variate generation. In Shane G. Henderson and Barry L. Nelson, editors, *Simulation*, Handbooks in Operations Research and Management Science. Elsevier, 2006.

[55] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *IEEE Transactions on Big Data*, 2025.

[56] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[57] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.

[58] Adriaan Daniël Fokker. Die mittlere energie rotierender elektrischer dipole im strahlungsfeld. *Annalen der Physik*, 1914.

[59] Max Planck. Ueber einen satz der statistichen dynamik und eine erweiterung in der quantumtheorie sitz. ber. *Preussischen Akad. Wiss*, 1917.

[60] Diederik P Kingma. Adam: A method for stochastic optimization. *ICLR*, 2015.

[61] Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? A fine-grained analysis in a simple setting. *NeurIPS*, 2024.

[62] Yasaman Haghighi, Bastien van Delft, Mariam Hassan, and Alexandre Alahi. Layersync: Self-aligning intermediate layers. *ICLR*, 2026.

[63] Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. *NeurIPS*, 2025.

[64] Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *NeurIPS*, 2026.

This appendix supplements the main paper with additional details regarding;

- Complementary Details on *Preliminaries* (App. A)
- Complementary Details on *Guiding Generation at Inference* (App. B)
  - Sampling from the tilted distribution (App. B.1)
  - Experiments on toy data to compare with the tilted distribution (App. B.2)
  - Our algorithm for test-time conditioning (App. B.3)
- Complementary Details on *Properties of Representation Space*
  - Impact of representation alignment on ImageNet Clusters (App. C.1)
  - More technical details on Euclidean Embedding Property (App. C.2)
- Complementary Details on Design of Potential $\mathscr{V}$ (App. D)
- More Analysis and Experimental Results (App. E)
  - More on hyperparameters (App. E.1)
  - More results on unconditional generation (App. E.2)
  - Results on conditional generation (App. E.3)
  - Analysis of the balance between stochastic reconstruction and generation diversity (App. E.4)
  - Comparison to adapted training free guidance baseline with visual feature (App. E.5)
  - Comparison with adapter-based conditioning (App. E.6)
  - Results on high resolution generation (App. E.7)
  - Detailed runtime analysis (App. E.8)
  - More technical details on multiple-Source compositional generation (App. E.9).
  - Failure cases for compositional generation (App. E.10).
  - More qualitative results (App. E.11).
- Licences of assets and codebase used (App. F)

A Complementary details on *Preliminaries* (Section 3)

A.1 Flow matching training and sampling

Let $p_1 = \mathcal{N}(0, I)$ be a standard Gaussian prior on $\mathcal{X}$. We define an interpolation process by sampling independently $x_0 \sim p_0$ and $x_1 \sim p_1$ and set $x_t = \alpha_t x_0 + \sigma_t x_1$, with typically $\alpha_t = 1 - t$ and $\sigma_t = t$. For such a process, there exists a *probability-flow* ordinary differential equation (PF-ODE) $\dot{x} = v^\star(x, t)$ with a velocity field $v^\star(x, t)$ such that the probability flow p_t induced by the PF-ODE at time t is the time-marginal density of x_t. Conditional flow models train a velocity field $v_\theta : (x, t) \in \mathcal{X} \times [0, 1] \to v_\theta(x, t) \in \mathcal{X}$ with

$$\mathcal{L}^{\text{diff}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),(x_0,x_1)\sim p_0\times p_1}\left[\|v_\theta\left(x_t, t\right) - \dot{x}_t\|_2^2\right] . \tag{8}$$

A necessary condition for $\mathcal{L}^{\text{diff}}$ to reach a global optimum at $\theta = \theta^\star$ is that, for each x and t, we have:

$$v_\theta(x, t) = v^\star\left(x, t\right) = \mathbb{E}_{(x_0,x_1)\sim p_0\times p_1}[\dot{x}_t \mid x_t = x] . \tag{9}$$

Using Tweedie's formula [51], we can relate $v^\star$ to the *score* $\nabla_x \log p_t(x)$ of the marginal distribution p_t of x_t. In the particular case of $(\alpha_t, \sigma_t) = (1 - t, t)$, we have:

$$v^\star\left(x, t\right) = -\frac{1}{1-t}x - \frac{t}{1-t}\nabla_x \log p_t(x) . \tag{10}$$

To sample from p_0, one can draw $x_1 \sim p_1$ and integrate the PF-ODE backward from $t = 1$ to $t = 0$ using, e.g., Euler's method. In practice, a Stochastic Sampler (e.g., Euler-Maruyama) can be used to improve sample quality during inference as in [16], integrating a reverse-time stochastic differential equation (SDE) of the form

$$\mathrm{d}x_t = (v^\star(x_t, t) - t\nabla_x \log p_t(x_t))\,\mathrm{d}t + \sqrt{2t}\mathrm{d}\bar{W}_t , \tag{11}$$

which shares the same marginals p_t as the PF-ODE. The term in parentheses corresponds to the *drift*, while the stochastic term $\sqrt{2t}\,\mathrm{d}\bar{W}_t$ is the *diffusion*, where $\bar{W}_t = W_{1-t}$ denotes a (time-reversed) standard Brownian motion. In (11), $v^\star$ is replaced by v_θ and the estimated score $\nabla_x \log p_t(x)$ is deduced from (10), where t is clipped to $\varepsilon > 0$ to avoid instabilities.

B Complementary details on *Guiding generation at inference* (Section 4)

B.1 Sampling the tilted distribution

General classifier guidance does not exactly target the tilted distribution [61], defined as:

$$\tilde{p}_0(x; x_c) \propto p_0(x) e^{\lambda \mathscr{V}(\phi(x), \phi(x_c))} \ . \tag{12}$$

Below, we present two assumptions under which modifying the score function via the gradient of the potential $\mathscr{V}$, as done in (2), would theoretically allow sampling from this tilted distribution (12). While our method operates within the classifier guidance framework and does not strictly require exact sampling from this density, we provide these assumptions as supplementary grounding for the score modification (2). This complements the standard justification derived from Bayes' rule [15].

B.1.1 Assumptions

Mirroring (9), we can show the following proposition.

Proposition B.1. *A necessary condition for* (1) *to reach a global optimum when $\mathscr{V}$ is an average patch-wise similarity with unit length vectors is that for each $x \in \mathcal{X}$ and $t \in [0, 1]$:*

$$(h_\theta \circ f_\theta)(x, t) = \mathbb{E}_{(x_0, x_1) \sim p_0 \times p_1} \left[\phi(x_0) \mid x_t = x \right] \ . \tag{13}$$

We use the notation established in Section 3. Below, we provide the derivation for Proposition B.1, assuming $N = 1$ without loss of generality.

Proof of proposition B.1. The potential $\mathscr{V}$ can be expressed in terms of the L_2 distance as follows:

$$\begin{aligned} \mathscr{V}\big((h_\theta \circ f_\theta)(x_t, t),\, \phi(x_0)\big) &= \langle (h_\theta \circ f_\theta)(x_t, t), \phi(x_0) \rangle \\ &= 1 - \frac{1}{2} \| (h_\theta \circ f_\theta)(x_t, t) - \phi(x_0) \|_2^2 \, , \end{aligned}$$

where the second equality holds because the vectors in the scalar product are constrained to unit length. Consequently, minimizing the alignment loss $\mathcal{L}^{\text{align}}$ defined in (1) is equivalent to minimizing the following objective over θ:

$$\mathbb{E}_{\substack{t \sim \mathcal{U}(0,1) \\ (x_0, x_1) \sim p_0 \times p_1}} \Big[\| (h_\theta \circ f_\theta)(x_t, t) - \phi(x_0) \|_2^2 \Big].$$

From the properties of conditional expectation in minimum mean square error (MMSE) estimation, this quantity is minimized when, for every $x \in \mathcal{X}$ and $t \in [0, 1]$:

$$(h_\theta \circ f_\theta)(x, t) = \mathbb{E}_{x_0, x_1 \sim p_0 \times p_1} [\phi(x_0) \mid (1 - t)x_0 + t x_1 = x] \ .$$

□

Assumption B.2 (Global Optimality)**.** We assume the model trained with $\mathcal{L}^{\text{diff}}(\theta) + \beta \mathcal{L}^{\text{align}}(\theta)$ has reached a global optimum, such that the conditions (9) and (13) are satisfied.

Assumption B.3 (Vanishing Gap)**.** Let $Z_t(x; x_c) \triangleq \mathbb{E}_{(x_0, x_1) \sim p_0 \times p_1} \left[e^{\lambda \mathscr{V}(\phi(x_0), \phi(x_c))} \mid x_t = x \right]$ denotes the expected energy at state x_t. By Jensen's inequality:

$$\log Z_t(x; x_c) \geq \lambda \mathbb{E}_{(x_0, x_1)} \left[\mathscr{V}\left(\phi(x_0), \phi(x_c)\right) \mid x_t = x \right] \ . \tag{14}$$

We assume this bound is tight (i.e., the Jensen Gap vanishes).

B.1.2 Guidance of the process

Let us consider the following forward time SDE

$$\mathrm{d}x_t = \left(v^\star(x_t, t) + t \nabla_x \log p_t(x_t) \right) \mathrm{d}t + \sqrt{2t} \mathrm{d}W_t \ . \tag{15}$$

Starting from the Fokker-Plank equation [58, 59], we have:

$$\begin{aligned}\frac{\partial p_t}{\partial t} &= -\nabla \cdot (p_t(v_t^\star + t\nabla \log p_t)) + t\Delta p_t \\ &= -\nabla \cdot (p_t v_t^\star) - \nabla \cdot (tp_t \nabla \log p_t)) + t\Delta p_t \\ &= -\nabla \cdot (p_t v_t^\star) - \nabla \cdot (t\nabla p_t)) + t\Delta p_t \\ &= -\nabla \cdot (p_t v_t^\star) - t\Delta p_t + t\Delta p_t \\ &= -\nabla \cdot (p_t v_t^\star)\ .\end{aligned}$$

We see that the forward SDE (15) shares the same marginals as the ODE $\dot{x}_t = v^\star(x_t, t)$, which we use during the forward.

We utilize the following Lemma, established by [52, 53], to connect modified SDEs to tilted distributions.

Lemma B.4 (Adapted from [52])**.** *Given a backward SDE of the form* $\mathrm{d}x_t = (v^\star(x_t,t) - t\nabla_x \log p_t(x_t))\mathrm{d}t + \sqrt{2t}\ \mathrm{d}\bar{W}_t$ *with* $x_1 \sim p_1$*, the reverse-time SDE:*

$$\mathrm{d}x_t = (v^\star(x_t,t) - t\nabla_x \log p_t(x_t) - 2t\nabla_x \log Z_t(x_t; x_c))\,\mathrm{d}t + \sqrt{2t}\mathrm{d}\bar{W}_t\ , \tag{16}$$

satisfies $\mathrm{Law}(x_0) = \tilde{p}_0(\cdot; x_c)$*, where* $\tilde{p}_0(x; x_c) \propto p_0(x)e^{\lambda \mathscr{V}(\phi(x),\phi(x_c))}$.

Proposition B.5. *Assume* $\mathscr{V}(h, h^\star) = \langle h, h^\star\rangle$*. Under Assumptions B.2 and B.3, for a flow model* v_θ *trained with representation alignment, the SDE:*

$$\mathrm{d}x_t = (v^\star(x_t,t) - t\nabla_x \log p_t(x_t) - 2\lambda t\nabla \mathscr{V}(x_t, \phi(x_c)))\,\mathrm{d}t + \sqrt{2t}\,\mathrm{d}\bar{W}_t\ , \tag{17}$$

with $x_1 \sim p_1$*, produces samples distributed according to the tilted distribution* $\tilde{p}_0(\cdot; x_c)$.

Proof of lemma B.4. Consider the forward SDE (15), where the marginals p_t are induced by the interpolation process $x_t = \alpha_t x_0 + \sigma_t x_1$, with $x_0 \sim p_0$ and $x_1 \sim p_1$. We define the tilting term $p(\phi(x_c) \mid x_0) = e^{\lambda \mathscr{V}(\phi(x_0),\phi(x_c))}$ and the corresponding time-dependent potential $Z_t(x; x_c) \triangleq \mathbb{E}_{x_0 \sim p(x_0|x_t)}\left[e^{\lambda \mathscr{V}(\phi(x_0),\phi(x_c))} \mid x_t = x\right]$. Applying Proposition 2.3. from [52], the reverse-time SDE is given by:

$$\begin{aligned}\mathrm{d}x_t &= (v^\star(x_t,t) + t\nabla_x \log p_t(x_t) - 2t(\nabla_x \log p_t(x_t) + \nabla_x \log Z_t(x; x_c)))\,\mathrm{d}t + \sqrt{2t}\mathrm{d}\bar{W}_t \\ &= (v^\star(x_t,t) - t(\nabla_x \log p_t(x_t) + 2\nabla_x \log Z_t(x; x_c)))\,\mathrm{d}t + \sqrt{2t}\mathrm{d}\bar{W}_t\end{aligned}$$

Initialised at $x_1 \sim p_1$, this SDE yields samples $x_0 \sim \tilde{p}_0(x; x_c) \propto p_0(x)e^{\lambda \mathscr{V}(\phi(x),\phi(x_c))}$, effectively sampling from the tilted distribution. □

Proof of proposition B.5. By Assumption B.2, the optimality condition (13) holds, which implies that the network recovers the conditional expectation: $f_\theta(x, t) = \mathbb{E}_{(x_0,x_1)\sim p_0 \times p_1}[\phi(x_0) \mid x_t = x]$. Consequently:

$$\mathscr{V}(f_\theta(x,t), \phi(x_c)) = \mathbb{E}_{(x_0,x_1)\sim p_0\times p_1}[\langle \phi(x_0), \phi(x_c)\rangle \mid x_t = x]\ . \tag{18}$$

Under Assumption B.3, the gradient of this expectation is equivalent to $\nabla_x \log Z_t(x; x_c)$. Substituting this result into the formulation of the reverse SDE from Lemma B.4 completes the proof. □

B.2 Toy example

To visualize the capacity of REPA-G for conditional generation, we consider a simple 2D distribution $p_0(x) = \frac{1}{2}\mathbf{1}[x \in C_1 \cup C_2]$, where $C_1 = \{x = (x_1, x_2); -1 \le x_1 \le 0\,, 0 \le x_2 \le 1\}$ and $C_2 = \{x = (x_1, x_2); 0 \le x_1 \le 1\,, -1 \le x_2 \le 0\}$. To learn a velocity field mapping p_0 to the standard normal distribution $p_1 = \mathcal{N}(0, I)$, we train a 5-layer MLP with ReLU activation functions following the REPA procedure [16]. We used Adam [60] optimizer with learning rate of 10^{-3}, a batch size of 512, and 300 epochs, with a dataset size of 100,000 points. The model utilizes 512-channel intermediate layers, taking a 2D coordinate and time as input to output a 2D velocity vector. We extract representations from the third layer and map them to the alignment target space using a 2-layer MLP projection head.

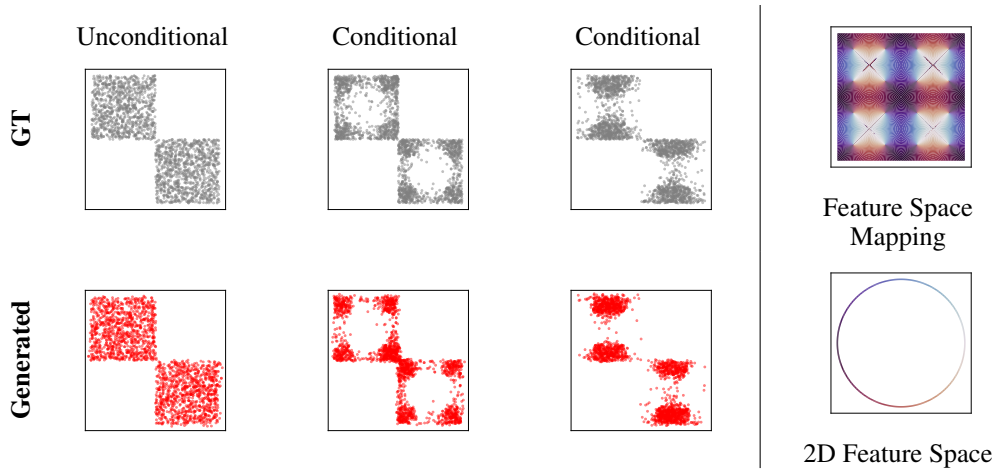


Figure 6: **Representation alignment with diffusion in 2D space.** Ground-truth samples are shown in grey, with corresponding samples generated via diffusion in red. The conditional results in the second and third columns are guided by target feature vectors $[-1, 0]$ and $[0, -1]$, respectively. The last column shows the feature space mapping, mapping each data points to its corresponding 2D feature.

To mimic the ℓ_2 normalization used in REPA, we design a 2D target feature space constrained to the unit circle. Specifically, the target embedding $\phi(x)$ is defined as:

$$\phi(x) = [\frac{t}{\sqrt{t^2 + (h-w)^2}}, \frac{h-w}{\sqrt{t^2 + (h-w)^2}}], \tag{19}$$

where t, h, and w represent the normalized distances from the cell center, the nearest vertical edge, and the nearest horizontal edge, respectively. The model is trained for 300 epochs using the composite loss $\mathcal{L}^{\text{diff}} + \beta\mathcal{L}^{\text{align}}$ with $\beta = 0.5$.

Figure 6 illustrates 2,000 generated samples (red) alongside the ground-truth distribution (grey). For conditional density sampling, we employ a rejection algorithm [54] to generate reference points from the tilted distribution. Diffusion sampling is performed via an Euler-Maruyama solver with 250 steps and an optional guidance scale of $\lambda = 2$. The conditional results in the third and fourth columns are induced by target feature vectors $[-1, 0]$ and $[0, -1]$, respectively.

B.3 Algorithm

Algorithm 1 describes the test-time conditioning process using a potential function $\mathscr{V}$ that is parametrized by the conditioning features and the guidance scale λ. Optionally, it can be a combination of joint potential, i.e, $\mathscr{V} = \lambda^1\mathscr{V}^1 + \lambda^2\mathscr{V}^2$.

Algorithm 1 REPA-G

Input: Initial noise $x_1 \sim \mathcal{N}(0, I)$, conditioning features $h^\star$, guidance scale λ, potential $\mathscr{V}$, and step size Δt.
for $t = 1$ **down to** 0 **do**
  Compute velocity $v_\theta(x_t, t)$ and internal representation $h_t = (h_\theta \circ f_\theta)(x_t, t)$.
  Compute the score $\nabla_{x_t} \log p_t(x_t)$ from the velocity (10).
  Compute guidance gradient $\nabla_{x_t}\mathscr{V}(h_t, h^\star)$.
  Sample $\epsilon \sim \mathcal{N}(0, I)$.
  Update x_t using the Euler-Maruyama discretization of the reverse-time SDE (17):
    $x_{t-\Delta t} = x_t - (v_\theta(x_t, t) - t\nabla_{x_t} \log p_t(x_t) - 2\lambda t\nabla_{x_t}\mathscr{V}(h_t, h^\star))\,\Delta t + \sqrt{2t\Delta t}\,\epsilon$.
end for
Return x_0.



Figure 7: **Impact of representation alignment on SiT's feature space.** We compare clusters across ImageNet [33] using four feature spaces. For each space, we extract features for the entire ImageNet dataset and perform k-means clustering into 1,000 discrete clusters. Then, we identify the specific cluster associated with a given reference image (indicated by the red frame) and randomly sample other images from that same cluster. Without alignment, the standard SiT model fails to form semantic clusters; features that are close in latent space represent unrelated concepts, which makes these internal representations unsuitable for conditioning. The aligned model successfully mimics the feature space of the visual teacher backbone both before and after the projection layer. This results in semantically consistent clusters. SiT (Align) refers to REPA [17] features extracted before the projection layer, whereas SiT (Align + Proj) denotes features extracted after the projection layer.

C Complementary details on *Properties of representation space* (Section 5)

C.1 Clustering of ImageNet

We perform k-means clustering across ImageNet [33] using $k = 1000$ clusters across four distinct feature spaces via the FAISS library [55]. We use the normalized global average of each image's spatial feature map. Features are extracted from: (i) Dinov2 [12], (ii) SiT-XL/2 with representation alignment [16] at the 8th layer (both before and after projection), and (iii) the baseline SiT-XL/2 trained without alignment [6] at the same layer. Figure 7 provide two additional cluster comparisons for each feature space.

C.2 Euclidean embedding property

Following [32], we investigate the Euclidean embedding property. As established in Section 3, there must exist $0 < A \leq B$ and where the ratio B/A is reasonably small such that for all x_1, x_2:

$$A\|\phi(x_1) - \phi(x_2)\|^2 \leq d^2(p_1, p_2) \leq B\|\phi(x_1) - \phi(x_2)\|^2.$$

We now justify the choice of the metric d^2. Let ϕ_1 and ϕ_2 be two embeddings that induce two tilted conditional densities p_1 and p_2 (12), respectively.

In this experiment, we utilize the full feature map to adhere strictly to our theoretical framework. For this setup, we define d^2 as the symmetric KL divergence:

$$d^2(p_1, p_2) := \mathrm{KL}(p_1 \| p_2) + \mathrm{KL}(p_2 \| p_1) \tag{20}$$

$$= \mathbb{E}_{p_1}\left[\log \frac{p_1}{p_2}\right] + \mathbb{E}_{p_2}\left[\log \frac{p_2}{p_1}\right] \tag{21}$$

$$= \mathbb{E}_{p_1}\left[\lambda\big(\mathscr{V}(\phi(x), \phi_1) - \mathscr{V}(\phi(x), \phi_2)\big)\right] + \mathbb{E}_{p_2}\left[\lambda\big(\mathscr{V}(\phi(x), \phi_2) - \mathscr{V}(\phi(x), \phi_1)\big)\right] \tag{22}$$

$$= \frac{\lambda}{N} \sum_{n=1}^{N} \Big(\mathbb{E}_{p_1}[\langle [\phi(x)]_n, [\phi_1 - \phi_2]_n \rangle] + \mathbb{E}_{p_2}[\langle [\phi(x)]_n, [\phi_2 - \phi_1]_n \rangle] \Big) \tag{23}$$

$$= \frac{\lambda}{N} \sum_{n=1}^{N} \left\langle \big[\mathbb{E}_{p_1}[\phi(x)] - \mathbb{E}_{p_2}[\phi(x)]\big]_n, [\phi_1 - \phi_2]_n \right\rangle . \tag{24}$$

D Complementary details on design of *Potential* $\mathscr{V}$ (Section 6)

D.1 Selective Patch Alignment (SPA)

We define the Selective Patch Alignment (SPA) potential as:

$$\mathscr{V}^{\mathrm{SPA}}(h, h^\star) = T \log \left[\sum_{n=1}^{N} \exp\left(\frac{\langle [h]_n, [h^\star]_i \rangle}{T} \right) \right] . \tag{25}$$

As discussed in Section 6.2, the temperature parameter T modulates the sparsity of this selection. In the limit $T \to 0$, the potential is dominated by the patch most similar to $h_i^\star$, effectively performing a "hard" selection. Conversely, as $T \to \infty$, the patches are weighted equally, recovering the global averaging behavior of IPA. Consequently, SPA can be viewed as a generalization of IPA, where T serves as a tunable parameter to optimize spatial correspondence. This relationship is further elucidated by the gradient:

$$\nabla_{h_k} \mathscr{V}^{\mathrm{SPA}}(h, h^\star) = N\, \sigma\left[\frac{1}{T} \big(\langle h_n, h_i^\star \rangle \big)_{n \in [1,N]} \right]_k \nabla_{h_k} \mathscr{V}^{\mathrm{IPA}}(h, h^\star), \tag{26}$$

where $\sigma(\cdot)$ denotes the softmax function. Here, the gradient of SPA is simply a reweighted version of the IPA gradient, where the weights are determined by the relative similarity of each patch to the target feature.

E More experimental results (Section 7)

E.1 Hyperparameter selection

Table 8: **Guidance scale (λ) robustness.** FID and DINOv2 Alignment scores for single-source conditioning under various values of the guidance scale λ.

λ	Full				Masked				Average			
	FID (↓)	Prec. (↑)	Rec. (↑)	Align. (↑)	FID (↓)	Prec. (↑)	Rec. (↑)	Align. (↑)	FID (↓)	Prec. (↑)	Rec. (↑)	Align. (↑)
5000	3.3	0.72	0.67	0.6	3.7	0.72	0.66	0.7	3.2	0.76	0.64	0.9
10000	2.8	0.75	0.67	0.7	2.8	0.74	0.66	0.8	2.8	0.76	0.65	0.9
20000	1.6	0.76	0.69	0.8	2.4	0.75	0.66	0.8	2.7	0.75	0.67	0.9
50000	1.4	0.76	0.69	0.8	2.3	0.75	0.67	0.9	3.3	0.73	0.68	0.9
100000	1.5	0.75	0.70	0.9	2.6	0.74	0.68	0.9	4.5	0.71	0.69	0.9

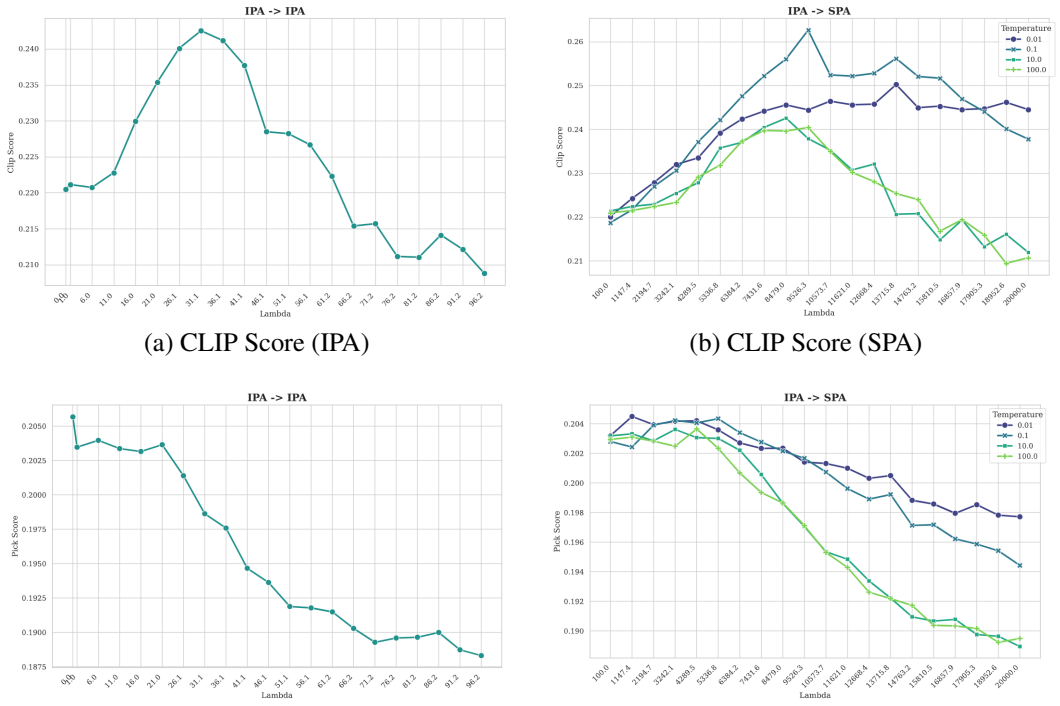
Robustness in single-source guidance We present a quantitative evaluation of the impact of the guidance scale (λ) on generation quality and feature alignment for single-source conditioning on ImageNet in Table 8. Numbers are computed using the same $50{,}000$ image anchors evenly distributed across the class of ImageNet [33]. Here we report FID, Precision and Recall alongside alignment scores across DINOv2 feature spaces. For full feature map conditioning, we observe that increasing λ consistently improves both FID and alignment scores before plateauing. We keep λ fixed at $50{,}000$ for

all other single source conditioning settings for simplicity. It is also the best performing value for the masked conditioning setting, while a lower value of $\lambda = 20{,}000$ seems to be slightly more effective for the average conditioning setting. Importantly, the performance of single source conditioning remains robust across a wide range of λ values, showing that this parameter does not introduce an overhead of careful tuning.

Hyperparameters for multi-source guidance. Hyperparameters λ and temperature T are set to $10{,}000$ and 0.1, respectively, based on a grid search over a diverse 5-class ImageNet subset (i.e., *goldfish, shark, golden retriever, bald eagle, tiger*). For each class, 10 random samples are combined with 8 hand-selected target background features.

For multi-source composition (IPA $\rightarrow$ SPA), Figure 8 shows tuning follows a smooth concave trend, making the effective range easy to identify. The best CLIP score, 0.263 is reached around $\lambda \approx 9,500$ at temperature 0.1. Performance is especially robust at lower temperatures (0.01 to 0.1), where CLIP scores remain high across a broad range of λ.

The sensitivity plots provide a tendency of the relationship between λ, temperature, and output quality. While different compositions might require some tuning to balance the contributions of the conditions, adjusting conditioning strength is common when guiding diffusion models.



(a) CLIP Score (IPA)

(b) CLIP Score (SPA)

(c) Peak (IPA)

(d) Peak (SPA)

Figure 8: Sensitivity plots illustrating the relationship between λ, temperature, and output quality.

E.2 More results on unconditional generation

As detailed in Section 7, we also evaluated features extracted after the projection layer for REPA-based models, but observed no significant performance difference. These results, provided in Table 9, complement those in Table 1.

We further complement Table 2 by providing alignment scores within the MAE [56] and MoCov3 [57] feature spaces. As shown in Table 10, these scores remain consistent with the trends observed across other feature representations. Similarly, we report alignment scores for the COCO dataset [34] in Table 11.

Table 9: **Distribution-level comparison of unconditional REPA-G generations on ImageNet [33].** We compare the standard SiT backbone [6] (trained without representation alignment) against REPA [7] and REPA-E [8] variants. All models are trained on ImageNet. For each method block, the first row denotes metrics for unconditional generation, serving as a baseline for the REPA-G results. $\mathbf{F}^*_{\mathbf{SiT}}$ denotes features extracted after the projection layer in REPA-based models. Gray results are from Table 1 in the main text for ease of comparison.

		Full Feature Map					Masked Feature Map					Average Feature Map				
Model	**Cond.**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
SiT	-	**45.02**	**9.02**	**22.33**	**0.50**	**0.63**	**45.02**	**9.02**	**22.33**	**0.50**	**0.63**	**45.02**	**9.02**	**22.33**	**0.50**	**0.63**
	F_{SiT}	71.35	72.08	14.73	0.33	0.54	51.88	42.36	15.76	0.27	0.52	65.99	22.52	16.69	0.36	0.52
REPA	-	26.58	6.85	41.31	0.56	**0.70**	26.58	6.85	41.31	0.56	**0.70**	26.58	6.85	41.31	0.56	**0.70**
	F_{DINO}	7.23	8.86	199.69	0.65	0.69	14.17	10.59	135.08	0.58	0.64	29.06	16.79	83.85	0.46	0.69
	F_{SiT}	**2.09**	**6.17**	**260.97**	**0.74**	**0.70**	**2.67**	**4.86**	**222.71**	**0.74**	0.69	**6.26**	**6.14**	**159.14**	**0.68**	**0.70**
	F^*_{SiT}	6.65	7.59	186.60	0.66	0.70	12.29	9.72	135.77	0.61	0.65	22.10	13.86	92.34	0.54	0.66
REPA-E	-	15.07	4.46	55.14	0.65	**0.69**	15.07	4.46	55.14	0.65	**0.69**	15.07	**4.46**	55.14	0.65	**0.69**
	F_{DINO}	**1.45**	**4.07**	**264.69**	**0.76**	**0.69**	2.30	4.84	212.25	0.75	0.67	3.24	5.20	188.92	0.73	0.67
	F_{SiT}	2.15	6.64	264.48	0.75	0.68	**1.79**	**4.13**	**237.37**	**0.76**	0.68	**2.50**	4.81	**201.13**	**0.74**	**0.69**
	F^*_{SiT}	**1.42**	4.10	257.23	**0.76**	**0.70**	2.01	4.73	216.19	0.76	0.67	2.65	5.05	191.79	**0.76**	0.65

Table 10: **Instance-level comparison of unconditional REPA-G generations on ImageNet [33].** Setup is the same as in Table 1. Alignment is measured with DINOv2 [12], JEPA [36], CLIP [37], MAE [56] and MoCov3 [57] feature spaces, supplemented by PSNR for pixel-level fidelity. $\mathbf{F}^*_{\mathbf{SiT}}$ denotes features extracted after the projection layer in REPA-based models. Gray results are from Table 1 in the main text for ease of comparison.

		Full Feature Map						Masked Feature Map						Average Feature Map					
Model	**Feat.**	**DINOv2**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**	**DINOv2**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**	**DINOv2**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**
SiT	F_{SiT}	**0.27**	**0.36**	**0.43**	**0.93**	**0.78**	15.35	**0.44**	**0.48**	**0.46**	**0.95**	**0.82**	17.98	**0.12**	**0.35**	**0.83**	**0.99**	**0.95**	7.74
REPA	F_{DINO}	0.75	0.61	0.58	0.94	0.86	11.02	0.77	0.60	0.57	0.94	0.84	10.63	0.76	0.69	0.92	0.99	0.98	8.00
	F_{SiT}	**0.85**	**0.78**	**0.71**	**0.98**	**0.95**	20.46	**0.87**	**0.79**	**0.71**	**0.98**	**0.94**	20.72	**0.84**	**0.84**	0.95	**0.99**	**0.990**	10.82
	F^*_{SiT}	0.72	0.63	0.59	0.94	0.87	11.32	0.74	0.63	0.58	0.94	0.86	11.02	0.77	0.72	0.92	0.99	0.98	8.01
REPA-E	F_{DINO}	0.83	0.69	0.64	0.96	0.90	15.23	0.85	0.68	0.63	0.95	0.88	14.36	**0.91**	0.83	0.96	0.99	0.99	10.97
	F_{SiT}	**0.86**	**0.76**	**0.71**	**0.97**	**0.94**	18.68	**0.89**	**0.78**	**0.71**	**0.98**	**0.94**	19.07	0.91	**0.88**	**0.96**	**0.99**	**0.99**	11.96
	F^*_{SiT}	0.81	0.70	0.65	0.96	0.91	15.91	0.83	0.71	0.65	0.96	0.90	15.13	0.90	0.86	0.96	0.99	0.99	11.12

E.3 Results on conditional generation

Analogous to our unconditional experiments, we evaluate conditional generation by providing the class index alongside the noisy latent as input. Results for this setup are reported in Table 12 for distribution-level metrics and Table 13 for instance-level alignment.

E.4 Analysis of the balance between stochastic reconstruction and generation diversity

With full-feature-map conditioning and sufficiently large λ, REPA-G approaches stochastic reconstruction. This setting provides a useful sanity check for the guidance mechanism: recovering the spatial and semantic structure of an anchor image from Gaussian noise is not guaranteed, and both vanilla SiT and the adapted TFG baseline (App. E.5) fail under the same feature-conditioning setup. Successful reconstruction therefore indicates that representation alignment produces an internal feature space capable of meaningfully steering the denoising trajectory.

Stochastic reconstruction is also useful for precise image manipulation. When combined with masks or additional conditions, it enables selected content to be preserved while allowing other regions to change. More generally, reconstruction and semantic generation form a continuum controlled by both

Table 11: **Zero-shot evaluation on COCO [34] dataset using instance-level metrics.** Alignment is measured with DINOv2 [12], JEPA [36], CLIP [37], MAE [56] and MoCov3 [57] feature spaces, supplemented by PSNR for pixel-level fidelity. $\mathbf{F}^*_{\mathbf{SiT}}$ denotes features extracted after the projection layer in REPA-based models.

		Full Feature Map						Masked Feature Map						Average Feature Map					
Model	**Cond.**	**DINOv2**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**	**DINOv2**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**	**DINOv2**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**
SiT	F_{SiT}	**0.25**	**0.35**	**0.4**	**0.93**	**0.76**	**15.18**	**0.4**	**0.45**	**0.44**	**0.95**	**0.82**	**18.56**	**0.11**	**0.38**	**0.81**	**0.99**	**0.95**	**7.66**
REPA	F_{DINO}	0.74	0.60	0.56	0.94	0.85	10.72	0.75	0.56	0.55	0.93	0.84	10.85	0.74	0.68	0.90	0.99	0.98	7.72
	F_{SiT}	**0.84**	**0.78**	**0.7**	**0.98**	**0.95**	**20.61**	**0.84**	**0.76**	**0.69**	**0.98**	**0.94**	**21.38**	**0.8**	**0.82**	**0.92**	**0.99**	**0.99**	**10.47**
	F^*_{SiT}	0.70	0.63	0.56	0.94	0.87	11.01	0.71	0.59	0.56	0.94	0.86	11.28	0.73	0.70	0.90	0.99	0.98	7.73
REPA-E	F_{DINO}	0.82	0.68	0.63	0.95	0.89	14.78	0.84	0.64	0.62	0.95	0.88	14.72	**0.9**	0.82	0.94	1.00	0.99	10.66
	F_{SiT}	**0.84**	**0.76**	**0.7**	**0.97**	**0.94**	**18.58**	**0.86**	**0.75**	**0.69**	**0.98**	**0.94**	**19.67**	0.89	**0.86**	**0.95**	**0.99**	**0.99**	**11.63**
	F^*_{SiT}	0.79	0.70	0.63	0.95	0.91	15.57	0.81	0.67	0.63	0.95	0.90	15.62	0.88	0.84	0.94	1.00	0.99	10.89

Table 12: **Distribution-level comparison of class-conditional REPA-G generations on ImageNet [33].** We compare the standard SiT backbone [6] (trained without representation alignment) against REPA [7] and REPA-E [8] variants. All models are trained on ImageNet. For each method block, the first row denotes metrics for unconditional generation, serving as a baseline for the REPA-G results. F^*_{SiT}denotes features extracted after the projection layer in REPA-based models.

		Full Feature Map					Masked Feature Map					Average Feature Map				
Model	**Cond.**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
SiT	-	**10.17**	**8.53**	**124.93**	**0.67**	**0.67**	**10.17**	**8.53**	**124.93**	**0.67**	**0.67**	**10.17**	**8.53**	**124.93**	**0.67**	**0.67**
	F_{SiT}	71.05	49.21	17.42	0.29	0.58	62.65	13.98	19.82	0.34	0.65	130.14	53.16	2.08	0.50	0.04
REPA	-	5.92	**5.92**	157.10	0.70	**0.68**	5.92	5.92	157.10	0.70	**0.68**	**5.92**	**5.92**	**157.10**	**0.70**	**0.68**
	F_{DINO}	6.24	8.43	197.64	0.69	0.64	13.68	10.21	127.72	0.61	0.60	27.94	15.32	84.04	0.49	0.66
	F_{SiT}	**2.14**	6.71	**254.99**	**0.77**	0.65	**3.34**	**5.68**	**199.55**	**0.77**	0.63	11.24	8.59	122.54	0.63	0.67
	F^*_{SiT}	5.60	7.14	186.45	0.71	0.64	11.77	9.41	131.34	0.64	0.60	21.98	13.61	91.11	0.56	0.62
REPA-E	-	1.71	4.19	220.60	0.77	0.66	**1.71**	**4.19**	220.60	0.77	**0.66**	**1.71**	**4.19**	**220.60**	**0.77**	0.66
	F_{DINO}	1.53	4.10	266.85	0.78	0.66	3.46	5.03	188.62	0.74	0.65	8.07	7.41	160.08	0.66	**0.69**
	F_{SiT}	1.97	6.67	**269.26**	0.78	0.66	1.98	4.24	**223.53**	**0.77**	0.65	3.23	5.39	192.17	0.72	0.67
	F^*_{SiT}	**1.53**	**4.04**	257.50	**0.78**	**0.66**	2.85	4.76	193.55	0.75	0.65	6.12	7.37	165.95	0.70	0.66

the guidance strength λ and the granularity of the conditioning features. Reducing λ weakens the attraction toward the anchor and increases stochastic variation, while average-feature conditioning removes patch-wise spatial correspondence and retains primarily global semantic information, thereby producing more diverse samples. This behavior is analogous to the fidelity–diversity relationship commonly observed with classifier and classifier-free guidance.

Our qualitative results illustrate this continuum. In the multi-source examples in Figure 5, full-feature conditioning preserves the semantic identity of the reference while still allowing substantial variations in properties such as color and shape. Similarly, in Figure 1, combining rabbit and lava features produces interactions, such as orange illumination reflected on the rabbit's fur, that are not directly copied from either source but instead emerge from the generative prior.

We further quantify the ability of REPA-G to generate diverse samples from the same condition. For each of the 1,000 ImageNet classes, we randomly select one anchor image and generate 10 samples using different random seeds. In Figure 9, we report the mean pairwise LPIPS distance, pairwise DINOv2 patch cosine distance, and DINOv2-based Vendi score, where larger values indicate greater sample diversity. Across all metrics, two consistent trends emerge: diversity decreases as λ increases, while average-feature conditioning remains substantially more diverse than full-map conditioning, particularly at large guidance scales, due to the removal of explicit patch-wise spatial constraints.

We repeat the same analysis using 1,000 randomly selected COCO anchors as a held-out dataset, which is out of distribution for the ImageNet-trained generator. The same trends are observed on COCO, with no reduction in diversity relative to ImageNet. Together with the quantitative COCO evaluation and the in-the-wild references in Figure 1, these results indicate that REPA-G is not simply retrieving or memorizing ImageNet training examples. Additional qualitative results with a lower guidance scale of $\lambda = 2,000$ are shown in Figure 10.

E.5 Training free guidance with visual feature

TFG [31] presents a unified approach to bring together every training free guidance for diffusion models in the same framework. It can use as guidance signal the output of a classifier or any general

Table 13: **Distribution-level comparison of class-conditional REPA-G generations on ImageNet [33].** We compare the standard SiT backbone [6] (trained without representation alignment) against REPA [7] and REPA-E [8] variants. All models are trained on ImageNet. For each method block, the first row denotes metrics for unconditional generation, serving as a baseline for the REPA-G results. $\mathbf{F^*_{SiT}}$ denotes features extracted after the projection layer in REPA-based models.

		Full Feature Map						Masked Feature Map						Average Feature Map					
Model	**Cond.**	**DINO**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**	**DINO**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**	**DINO**	**JEPA**	**CLIP**	**MAE**	**MoCo**	**PSNR**
SiT	F_{SiT}	**0.3**	**0.39**	**0.43**	**0.94**	**0.78**	**14.79**	**0.4**	**0.44**	**0.44**	**0.95**	**0.81**	**17.66**	**0.13**	**0.37**	**0.83**	**0.99**	**0.95**	**8.64**
REPA	F_{DINO}	0.74	0.59	0.57	0.94	0.85	10.47	0.76	0.58	0.56	0.93	0.83	10.30	0.75	0.68	0.92	**0.99**	0.98	8.05
	F_{SiT}	**0.84**	**0.77**	**0.7**	**0.98**	**0.94**	**19.6**	**0.86**	**0.77**	**0.7**	**0.98**	**0.94**	**20**	**0.79**	**0.8**	**0.93**	**0.99**	**0.99**	**10.36**
	F^*_{SiT}	0.71	0.61	0.58	0.94	0.86	10.71	0.73	0.61	0.57	0.94	0.85	10.60	0.76	0.70	0.92	**0.99**	0.98	8.02
REPA-E	F_{DINO}	0.82	0.68	0.63	0.95	0.89	14.79	0.84	0.66	0.62	0.95	0.87	13.84	0.88	0.79	0.95	**0.99**	0.99	10.34
	F_{SiT}	**0.85**	**0.76**	**0.7**	**0.97**	**0.94**	**18.57**	**0.87**	**0.77**	**0.7**	**0.97**	**0.93**	**18.81**	**0.89**	**0.85**	**0.96**	**0.99**	**0.99**	**11.58**
	F^*_{SiT}	0.80	0.69	0.64	0.96	0.90	15.33	0.82	0.68	0.63	0.95	0.89	14.41	0.88	0.80	0.95	**0.99**	0.99	10.45

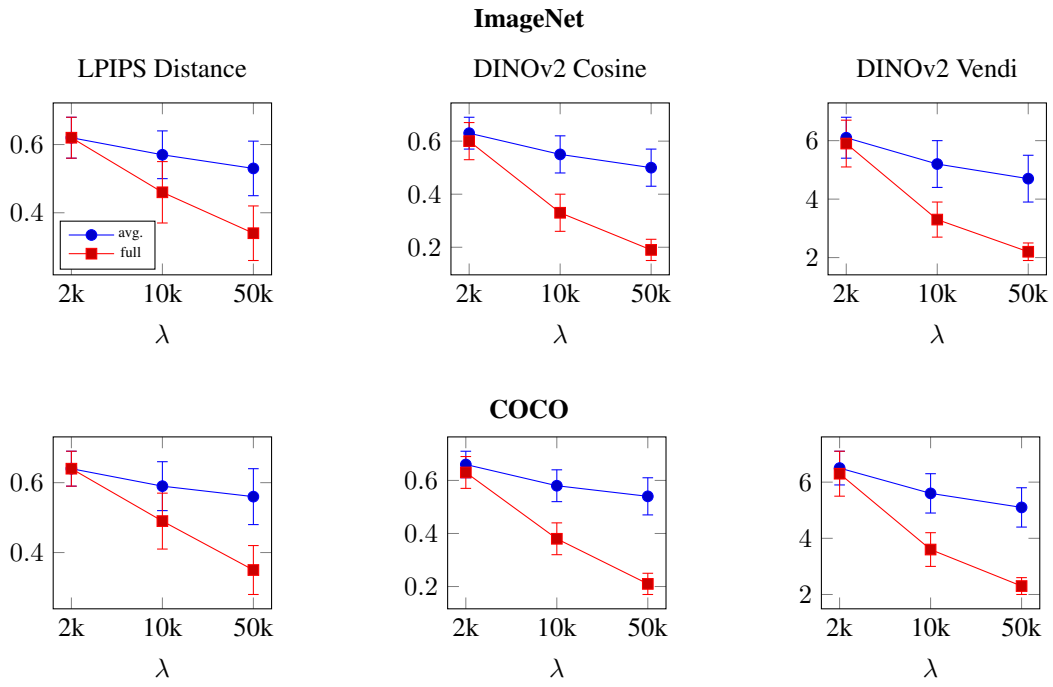


Figure 9: **Effect of guidance strength on generation diversity.** Impact of λ on LPIPS distance, DINOv2 cosine distance, and DINOv2-based Vendi score for samples generated from ImageNet and COCO anchors.



Figure 10: **Qualitative results of generation diversity.** ImageNet examples are shown in the top row and COCO examples in the bottom row. For each example, the anchor image is shown on the left, followed by three samples generated with *average feature* conditioning and three with *full feature map* conditioning. We use a small guidance scale of $\lambda = 2{,}000$.

energy function. Critically, contrarily to classifier guidance where the model is trained to classify noisy input, the guidance signal is extracted here from clean data. We introduce a new setting not tested in their work to see if TFG can be used with visual features as guidance signal. We use as target predictor the same energy function as in our method based on alignment of features patches. In REPA-G, the predicted features are extracted from the noisy latent with the aligned diffusion transformer. In contrast, they have to predict at each step an approximation of the clean latent sample and decode it to image space. Then, the it is possible to extract features from it with a vision encoder, here DINOv2, and compare them with target features. It introduces much more computational overhead, as the backpropagation is done through the vision encoder, the decoder and the entire diffusion network versus only the first 8 transformer blocks in our method.

We use their style transfer hyperparameters, as this is the closest task to ours. We set $N_{iter} = 1$ and $\rho = 0$ for faster sampling, and $\mu = 1000$ to provide sufficiently strong conditioning. We use the REPA-E model for generation.

We show in Figure 11 that TFG fails in the simple setting of stochastic reconstruction of an image, i.e. guiding with a full feature map. While the condition is recognizable in the generated image, the quality is very poor due to visual artifacts. This behaviour is confirmed by the quantitative evaluation in Table 14, where we report a very high FID and low alignment scores, except for DINOv2 alignment which is directly optimized. Lowering $\mu = 1000$ doesn't allow to generate images with better quality that keeps the condition recognizable.

We emphasize that a target predictor trained directly on noisy latents, such as ours, naturally outperforms a predictor operating on clean data, which relies on heavier approximations to guide generation. Indeed, such trained predictors are presented as an upper bound for TFG's performance in their original work. A key strength of REPA-G is that its target predictor emerges naturally from the diffusion model's alignment procedure, originally motivated solely by convergence speed and sample quality. Although never explicitly intended to serve as a guidance signal at inference, we demonstrate that it fulfills this role exceptionally well without requiring any additional training.

Table 14: **TFG performance.** Evaluation of TFG with different guidance granularities.

Method	Features	FID	sFID	IS	Prec.	Rec.	DINOv2	JEPA	CLIP	PSNR
REPA-G	Full	1.4	4.1	264.7	0.76	0.69	0.83	0.69	0.64	15.2
	Masked	2.3	4.8	212.2	0.75	0.67	0.85	0.68	0.63	14.4
	Average	3.2	5.2	188.9	0.73	0.67	0.91	0.83	0.96	11.0
TFG	Full	74.6	53.6	47.4	0.26	0.40	0.91	0.43	0.54	10.0
	Masked	85.8	47.2	36.0	0.24	0.38	0.94	0.45	0.55	10.0
	Average	81.1	37.0	35.8	0.23	0.41	0.99	0.51	0.91	8.5

E.6 Comparison with adapter-based conditioning

Adapter-based methods learn additional image-prompt projection and cross-attention parameters from paired training data. In contrast, once a representation-aligned diffusion model is available, REPA-G requires no additional learned parameters, architectural modifications, or task-specific training. Instead, it directly exploits the model's aligned feature space at inference time and supports full-map, masked, global-average, single-patch, and multi-source conditioning within the same formulation. Unlike methods explicitly trained for structural conditioning, however, REPA-G does not provide direct control from inputs such as edge maps, depth maps, or poses.

To better position REPA-G relative to learned adapter-based conditioning, we train an IP-Adapter-style module on ImageNet for 100k steps on top of REPA-E, using DINOv2 full-map features. The diffusion backbone is kept frozen, while the input projection, normalization, and image cross-attention parameters are optimized. During sampling, we apply classifier-free guidance to the image condition with a scale of 6. Qualitative results on ImageNet (see Figure 11) show that the adapter follows the reference image, but preserves its spatial layout less precisely than REPA-G, consistent with the lower patch-wise DINOv2 alignment reported in Table 15. Its distribution-level metrics also improve over the unconditional baseline. Runtime results are reported in Table 17; due to classifier-free guidance and approximately 150M additional parameters, IP-Adapter also achieves noticeably lower throughput than REPA-G.

Table 15: **Comparison with trained adapter-based conditioning.** We compare full-feature-map conditional generation using IP-Adapter and REPA-G on ImageNet and COCO, with REPA-E as the shared base model.

	ImageNet						COCO					
Condition	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	DINOv2↑	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	DINOv2↑
-	15.07	4.46	55.14	0.65	**0.69**	-	36.23	29.98	**55.13**	0.51	0.59	-
IP-Adapter	7.52	10.65	145.36	0.71	0.63	0.62	9.90	32.23	30.45	0.62	0.56	0.61
REPA-G	**1.45**	**4.07**	**264.69**	**0.76**	**0.69**	**0.83**	**4.61**	**23.27**	35.97	**0.67**	**0.64**	**0.82**

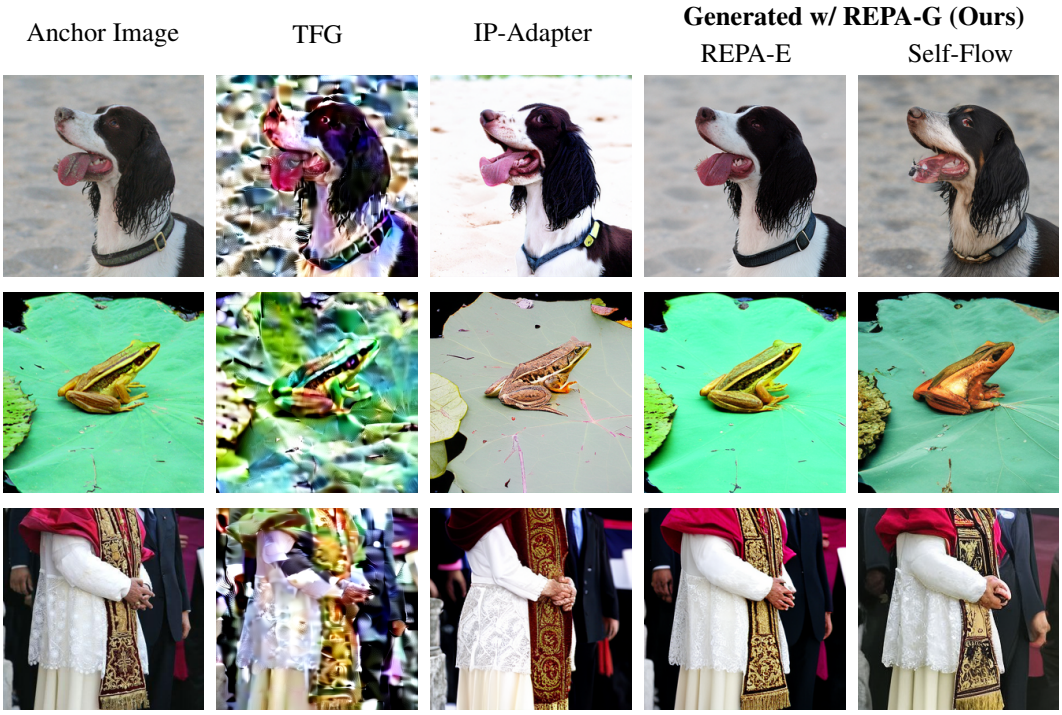


Figure 11: **Visual comparison with the training-free baseline TFG [31] and the trained IP-Adapter [27].** While the images generated with TFG respect the conditioning to the anchors, perceptual artifacts appear and degrade image quality. On the other hand, IP-adapter manages to generate images faithful to the anchor but requires additional training and parameters.
REPA-G on the Self-Flow model [44] also generates realistic images.

E.7 Resolution scalability

The supervision is done at the patch level, which is independent of the resolution. Therefore, our method should be able to scale to higher resolutions without any architectural modification. To validate this claim, we fine-tuned the 256×256 REPA model to output at 512×512 for 40k steps and evaluated the performance of REPA-G. The results below in Table 16 confirm that our method successfully scales, maintaining superior image quality than unconditional sampling.



Figure 12: **Qualitative results for resolution scalability.** Using the fine-tuned 512 $\times$ 512 model, REPA-G successfully stochastically reconstruct the conditioning image at a 512 $\times$ 512 resolution. The **top row** shows the anchor images, and the **bottom row** shows the corresponding results for REPA-G (512x512).

Table 16: **Resolution scalability.** We evaluate our method's ability to scale to higher resolutions (512×512) and compare the performance across unconditional generation with different granularity settings for our feature conditioning.

	FID	sFID	IS	Prec.	Rec.	DINOv2 Align.
Uncond	33.1	10.1	39.1	0.61	0.6	-
Full	5.4	6.0	178.8	0.75	0.67	0.61
Masked	6.9	5.9	149.1	0.73	0.65	0.68
Average	8.9	5.6	139.9	0.73	0.66	0.85

E.8 Runtime analysis

We measure throughput on a single H100 GPU batch size 32, averaging across $n = 20$ batches after a 5-batch warmup. The 95% confidence intervals (computed from the standard deviation over the 20 batches) are also reported alongside the means to confirm the statistical stability of our measurements. As reported in Table 17, backpropagating gradients through the first 8-transformer blocks naturally increases latency; however, the resulting inference speeds remain well within an acceptable range.

We provide a detailed analysis of the computational overhead using an H100 GPU at 256 and 512 resolution with batch size 32. While REPA-G introduces a 26% throughput reduction compared to unconditional generation due to the backward pass, the resulting inference speeds remains well within an acceptable range. For reference only, we report the throughput of classifier-free guidance (CFG), a guidance broadly used in diffusion models despite a throughput reduction of 48%. Closer to our method, TFG requires backpropagating through the entire network, resulting in a significant throughput reduction. In contrast, our method only utilizes 2GB of additional GPU memory with batch size 32 (+25%) to store activations for the backward pass. This is a manageable trade-off for the level of semantic control achieved.

Our multi-source composition adds no extra throughput overhead. Indeed, all guidance signals are computed from the same set of feature maps extracted during the forward pass.

We observe similar behaviour of our method at 512 resolution. The number of patches increases with the resolution, leading to a higher absolute computational cost for the backward pass.

Table 17: **Inference throughput.** Feature conditioning requires backpropagating gradients through the initial blocks of the transformer. While this results in a slight reduction in throughput, the computational overhead remains minimal. The baseline is the standard class-conditioned REPA-E model [8], whereas our approach incorporates feature guidance into this framework.

Image Resolution	Method	Throughput (img/s)	Peak GPU memory (GB)
256	Uncond.	0.50	8
	Ours (Single source)	0.37	10
	Ours (Multi source)	0.37	10
	TFG	0.21	52
	CFG	0.26	8
	IP-Adapter	0.19	9
512	Uncond.	0.12	23
	Ours	0.08	29

E.9 Complementary details on multiple-source compositional generation

More details on classes. In Table 18, we provide the list of 50 ImageNet [33] classes and 8 hand-selected target backgrounds used to evaluate multi-source compositional generation.

Patch metrics for multi-concept. As detailed in Section 7, we evaluate the quality of our compositions using patch-wise metrics. For each patch n in the generated feature map h_{gen}, we calculate the cosine similarity against both the corresponding anchor patch $[h_{\text{ref}}]_n$ and the broadcasted target

Table 18: A diverse set of 50 ImageNet [33] classes and 8 hand-selected target backgrounds used to evaluate multi-source compositional generation.

Super-category	Fine-grained Class Names
Animals	Golden retriever, Tabby cat, Zebra, Bullfrog, Tarantula, Goldfish, Monarch butterfly, Crane, Bald eagle, Tiger
Movables	Bicycle, Motor scooter, Mountain bike, Baby stroller, Lawnmower, Shopping cart, Tricycle, Car wheel, Fire engine, Sports car
Food Flora	Strawberry, Banana, Granny Smith, Plate, Pizza, Cheeseburger, Agaric, Acorn, Burrito, Bee
Tools Tech	Acoustic guitar, Broom, Hammer, Screwdriver, Soccer ball, Tennis racket, Crash helmet, Wall clock, Laptop, Ping-pong ball
Household	Teapot, Coffeepot, Toaster, Vase, Cowboy hat, Running shoe, Backpack, Umbrella, Teddy bear, Balloon
Our hand-selected target backgrounds	
Background	Grass, Dirt, Snow, Ice, Carpet, Water, Brick, Wood

feature $[\tilde{h}_{\text{target}}]_n$:

$$s_{\text{anchor},n} = \cos([h_{\text{gen}}]_n, [h_{\text{ref}}]_n), \quad s_{\text{target},n} = \cos([h_{\text{gen}}]_n, [\tilde{h}_{\text{target}}]_n) \tag{27}$$

Next, patches are assigned to the partition (Ω_{anchor} or Ω_{target}) where they exhibit the highest similarity.

$$n \in \begin{cases} \Omega_{\text{anchor}} & \text{if } s_{\text{anchor},n} > s_{\text{target},n} \\ \Omega_{\text{target}} & \text{otherwise} \end{cases} \tag{28}$$

The resulting scores, $\mathcal{S}_{\text{anchor}}$ and $\mathcal{S}_{\text{target}}$, are the average alignments within their respective partitions:

$$\mathcal{S}_{\text{anchor}} = \frac{1}{|\Omega_{\text{anchor}}|} \sum_{n \in \Omega_{\text{anchor}}} s_{\text{anchor},n} \tag{29}$$

$$\mathcal{S}_{\text{target}} = \frac{1}{|\Omega_{\text{target}}|} \sum_{n \in \Omega_{\text{target}}} s_{\text{target},n} \tag{30}$$

Finally, we compute a combined score by taking the arithmetic mean of $\mathcal{S}_{\text{anchor}}$ and $\mathcal{S}_{\text{target}}$:

$$\mathcal{S}_{\text{combined}} = \frac{1}{2} \left(\mathcal{S}_{\text{target}} + \mathcal{S}_{\text{anchor}} \right) \tag{31}$$

These metrics capture the model's precision in structural preservation alongside its responsiveness to the target potential $\mathscr{V}$.

E.10 Failure Cases of Multiple-Source Compositional Generation

We exhibit in Figure 13 some failure cases of multiple-source compositional generation. Using two images with inconsitent view points as condition can lead to unrealistic blending as the resulting image will be subject to both view points. Similarly, inconsistent aspect ratios can lead to unrealistic blending. More generally, the model can fail to find a good compromise between the two conditions.

E.11 More Qualitative Results

Figure 14 and 15 provides additional qualitative results on ImageNet [33] and COCO [34].



Figure 13: **Failure cases of multiple-source compositional generation.** Top row: Inconsistent aspect ratio. Middle row: Failed blending. Bottom row: Inconsistent view point.

F Licenses

In this work, we utilize several open-source datasets and codebases for training, evaluation, and baseline comparisons. We strictly adhere to their respective licenses and terms of use and confirm that our use of these assets aligns with their intended purpose of academic research.

- **ImageNet** [33]: Available under non-commercial use.
- **COCO** [34]: Annotations are licensed under the Creative Commons Attribution 4.0 License (CC-BY 4.0), and images are subject to their respective Flickr terms of use.
- **SiT** [6], **REPA** [7], **REPA-E** [8] and **CAD-I** [42]: We utilize their released codebases and pre-trained models, which follow their provided open-source licenses (MIT License).
- **DINOv2** [12] and **Perception Encoder** [43]: Licensed under the Apache 2.0 License.
- **CLIP** [37]: Licensed under the MIT License.
- **JEPA** [36], **MoCo v3** [57] and **MAE** [56]: Pre-trained models and code are distributed under the CC-BY-NC 4.0 license.

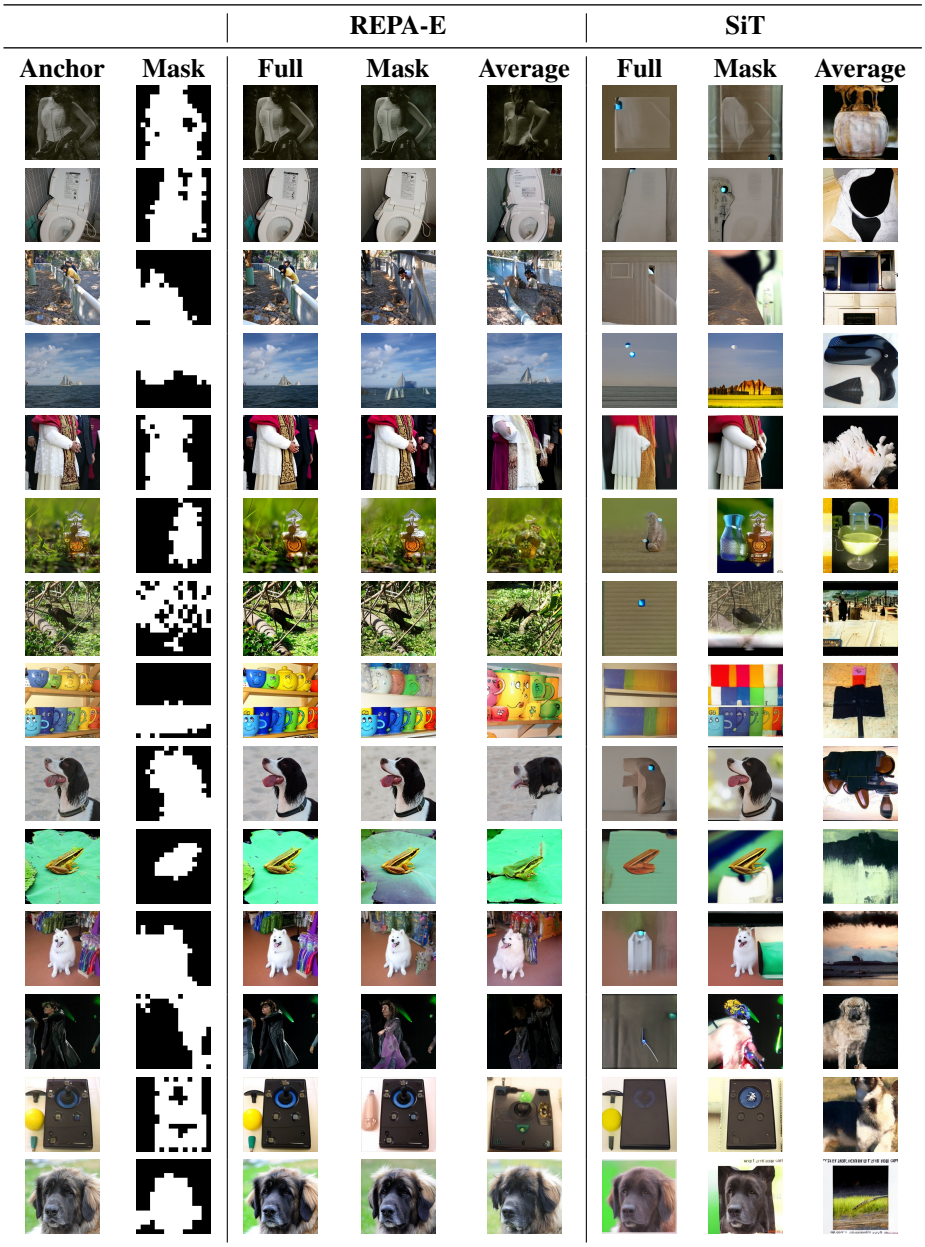


Figure 14: **Qualitative results on ImageNet dataset [33].** We present randomly selected samples from various conditioning modes. The first row shows the ground-truth (anchor) images, followed by their unsupervised foreground masks. Subsequent rows show generations using the full feature map, the masked feature map (using the corresponding mask), and the averaged feature map. All visualizations are from the following configuration: an unconditioned model guided by features from the diffusion model before the projection layer. The following block displays the corresponding results for standart SiT [6] model trained without representation alignment and REPA-E [8]. We observe that SiT models trained without representation alignment struggle to produce coherent results under feature guidance.



Figure 15: **Qualitative results on COCO dataset [34].** We present randomly selected samples from various conditioning modes. The first row shows the ground-truth (anchor) images, followed by their unsupervised foreground masks. Subsequent rows show generations using the full feature map, the masked feature map (using the corresponding mask), and the averaged feature map. All visualizations are from the following configuration: an unconditioned model guided by features from the diffusion model before the projection layer. The following block displays the corresponding results for standart SiT [6] model trained without representation alignment and REPA-E [8]. We observe that SiT models trained without representation alignment struggle to produce coherent results under feature guidance.

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 claims made in the abstract and introduction accurately summarize our technical contributions and empirical findings backed by experiments in Section 7 and Appendix.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: We discuss the limitations of our proposed approach, including failure cases related to varying scene viewpoints, in the Conclusion section (Section 8) and Appendix E.10.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: All assumptions and proofs for theoretical claims regarding the tilted distribution and score modifications are fully detailed in Appendix B.

Guidelines:

- The answer [N/A] 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: All datasets, network architectures, and setup details required to reproduce the main experiments are provided in Section 7 and Appendix E.

   Guidelines:

   - The answer [N/A] 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: We provide the implementation for our toy example in the supplementary material. The full codebase will be released publicly upon acceptance of the paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

Answer: [Yes]

Justification: Implementation details, hyperparameters, and experimental settings are provided in Section 7 and Appendix E.

Guidelines:

- The answer [N/A] 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: [No]

Justification: We do not report error bars for the generative metrics (FID, sFID, IS) as it is computationally expensive to run multiple evaluations (generating 50,000 images per run), which is standard practice in generative modeling literature.

Guidelines:

- The answer [N/A] 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: We detail the required compute resources and provide a comprehensive runtime and memory overhead analysis in Appendix E.8.

   Guidelines:

   - The answer [N/A] 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 conducted in this paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: we provide an impact statement at the beginning of Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release new datasets or models.

    Guidelines:

    - The answer [N/A] 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: We properly cite all datasets (ImageNet, COCO) and pre-trained models (DINOv2, SiT, etc.) used in our experiments, which are standard community assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: We do not release new datasets or large assets. The source code for the toy experiment is provided in the supplementary material.

    Guidelines:

    - The answer [N/A] 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: This research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] 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: This research does not involve human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We did not use LLMs as an important, original, or non-standard component of the core methods in this research.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.