

SODA: Bottleneck Diffusion Models for Representation Learning

Drew A. Hudson*

Daniel Zoran

Mateusz Malinowski

Andrew K. Lampinen

Andrew Jaegle

James L. McClelland

Loic Matthey

Felix Hill

Alexander Lerchner

Google DeepMind

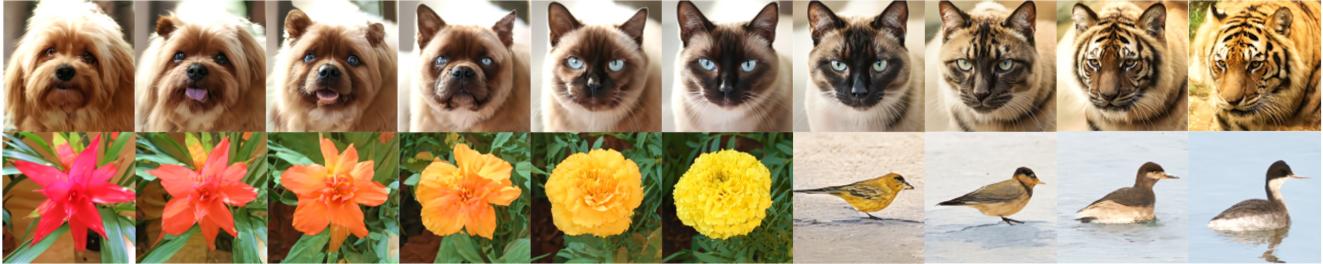


Figure 1. **Latent Interpolations.** SODA learns to encode images into compact latent representations. By traversing its latent space, we can interpolate between images, morphing from one image category to another and smoothly transitioning between semantic attributes.

Abstract

We introduce SODA, a self-supervised diffusion model, designed for representation learning. The model incorporates an image encoder, which distills a source view into a compact representation, that, in turn, guides the generation of related novel views. We show that by imposing a tight bottleneck between the encoder and a denoising decoder, and leveraging novel view synthesis as a self-supervised objective, we can turn diffusion models into strong representation learners, capable of capturing visual semantics in an unsupervised manner. To the best of our knowledge, SODA is the first diffusion model to succeed at ImageNet linear-probe classification, and, at the same time, it accomplishes reconstruction, editing and synthesis tasks across a wide range of datasets. Further investigation reveals the disentangled nature of its emergent latent space, that serves as an effective interface to control and manipulate the produced images. All in all, we aim to shed light on the exciting and promising potential of diffusion models, not only for image generation, but also for learning rich and robust representations. See our website at soda-diffusion.github.io.

1. Introduction

What I cannot create, I do not understand.

— Richard P. Feynman

Synthesis, the ability to create, is considered among the highest manifestations of learning [1, 2]. As opposed to passive analysis of a text or an image, conceiving them out of thin air involves profound understanding of the underly-

*Main contributor. The paper presents the findings of Drew’s internship project at Google DeepMind, hosted by Felix Hill and Alexander Lerchner. Contact: dorarad@google.com; lerchner@google.com.

ing factors and intricate generative processes that give rise to the final product [3]. Indeed, learning to write in a new language is often more challenging than reading it. Figuring out the solution to a math problem is fundamentally harder than verifying it [4]. And just as the chef learns more about the culinary arts than the diner to prepare a tasty meal, and the novelist knows more about narrative structures than the reader to tell a good story, the artist better grasps perspective and composition to craft a breathtaking masterpiece.

Analogously, in AI, the recent years have witnessed remarkable progress at the generative domain, with large-scale diffusion modeling proving to be a powerful and flexible technique that can create vivid imagery of astonishing realism and incredible detail. And yet, while the vast majority of research harnesses these models for the straightforward goal of synthesis or editing alone [5–17], only little attention has been given to their representational capacity [18–20], leaving this promising direction rather unexplored. Surely, models that can weave from scratch such rich depictions of high fidelity, likely learn much along the way about the underlying properties, processes, and components that make up the resulting pictures. How then can we leverage this untapped potential of diffusion models for the purpose of representation learning, and extract the knowledge they acquire for the benefit of downstream tasks?

Motivated to achieve this aim, we present SODA, a self-supervised diffusion model, designed for both perception and synthesis. It couples an image encoder with the classic diffusion decoder [5], both trained in tandem for novel view generation [21] – a task we choose to employ here, not only for its own sake, but as a self-supervised objective. The encoder transforms an input view into a concise latent representation, which then guides the denoising of an output view, by modulating the decoder’s activations.

This setup introduces a desirable information bottleneck

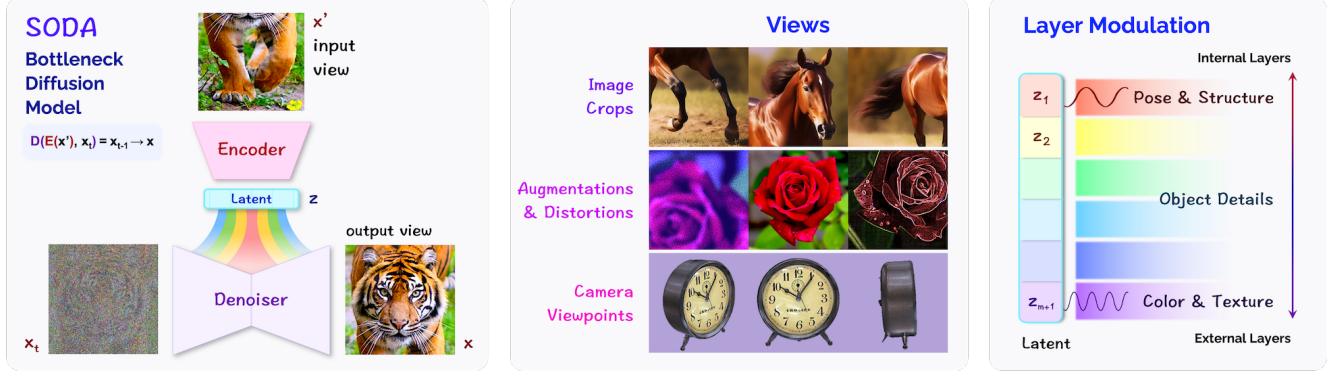


Figure 2. (**Left**) **Model Overview.** SODA consists of two modules: an *encoder* and a *denoising decoder*. The encoder transforms a source view into a *compact latent representation*, which guides the denoising of a target view, by having a dedicated latent subvector modulating each of the decoder’s layers (**Right, Layer Modulation**). The model is useful both for representation learning, by applying the latents to downstream perception tasks, as well as for synthesis, by iteratively generating novel views. (**Center**) **Views.** We consider views as any set of images that relate visually or semantically: They can be crops, augmentations, or images captured from different camera perspectives.

between the encoder and the decoder [22], that in contrast to the typical diffusion framework, equips our model with an explicit and interpretable visual latent space. As our experiments confirm, its advantages are twofold: it both encourages the emergence of disentangled and informative **representations** that capture image key properties and semantics, which thus can be applied to downstream tasks, and further provides effective means to control and manipulate the produced outputs, for the gain of image editing and **synthesis**. We further devise and integrate multiple new ideas into the network architecture and training procedure: layer modulation, modified classifier-free guidance, and an inverted noise schedule, so to maximize its representation skills.

We demonstrate our model’s strengths and versatility by evaluating it along a series of classification, reconstruction and synthesis tasks, spanning an extensive collection of datasets that covers both the simulated and real-world kinds. SODA possesses strong representation skills, attaining high performance in linear-probing experiments over the ImageNet dataset among others. Moreover, it excels at the task of few-shot novel view generation, and can flexibly synthesize images either conditionally or unconditionally, as indicated by metrics of fidelity, consistency and diversity. Finally, we inspect the model’s emergent latent space and discover its disentangled nature, which offers controllability over the semantic traits of the images it produces, as validated both qualitatively and quantitatively.

Overall, SODA integrates together three research ideas that we seek to establish and promote: First, diffusion models are not only adept at image generation, but are also capable of learning strong representations. Second, novel view synthesis can serve as a powerful self-supervised objective for model pre-training. And third, the compactness of the latent space, which could be reached by constricting the bottleneck between the encoder and the denoiser, plays a pivotal role in enhancing the latent representations’ quality,

informativeness and interpretability.

2. Related Work

Diffusion. The advent of diffusion models has lately marked a breakthrough in the field of visual synthesis. Originally inspired by theories of thermodynamics [23], it approaches generative modeling by following a reversible and iterative denoising process, the forward direction of which slowly erodes the structure within the data distribution, while the backward direction is gradually restoring it. Since its early inception back in 2015 [24], tremendous strides have been made in the quality and diversity of the created outputs, thanks to innovations of the framework’s training and sampling techniques [25–28]. Consequently, diffusion models have been widely adopted for numerous tasks and modalities [29–33], synthesizing images, videos, audio and text [34–38], and even advancing planning [39] and drug discovery [40], effectively becoming one of the leading paradigms for generative modeling nowadays.

But while most literature highlights its generative feats, only a handful of works have studied diffusion modeling’s representational capacity, mainly repurposing pre-trained text-to-image models for classification [41], segmentation [42], or multimodal reasoning [43]. The reliance on such models makes it unclear whether the downstream capabilities arise from the diffusion approach itself, or are actually attributable to the exceptionally large scales, long training and voluminous captioned data, which, essentially, provides rich and textual semantic supervision. To address this shortcoming, we focus here instead on the fully-unsupervised regime, and train our model from scratch on standardized benchmarks, seeking to assess the value and potential of diffusion-based representations derived from images alone.

Visual Encoding. Closer to our work is DRL [20], that extends early research on denoising auto-encoders [44–47], and conditions a denoiser on an encoded clean version of its

own target. It is mainly explored from a theoretical perspective, along with preliminary results on MNIST and CIFAR-10. DiffAE [48] follows up, integrating style modulation into the encoder [49–51], while InfoDiffusion [52] regularizes it with mutual-information loss. Our approach builds upon this line of research, but instead of auto encoding the same image, we generate novel views. Notably, we discover that this, in turn, remarkably enhances the model’s representation skills, as evidenced by substantial gains in downstream performance. We further couple this idea with multiple technical innovations, pertaining both architecture and optimization, geared to realize the representational capabilities of diffusion models to their fullest. And in contrast to prior works, we provide an extensive empirical study of diffusion-based representation learning, encompassing a broad suite of datasets over multiple different tasks.

Hybrid Models. A couple of partially related works are unCLIP [8] and Latent Diffusion Models [10], both of which utilize a frozen pre-trained encoder (CLIP and VQ-GAN respectively) to cast images onto a compressed latent space over which a diffusion model can operate. Consequently, we note that the latent representations used in both these approaches are in fact not derived by diffusion itself, but rather through either contrastive or adversarial pre-training. As such, they differ fundamentally from our study, which aims to explore the effectiveness of diffusion-based pre-training as a means for representation learning.

Downstream Tasks. For each of the tasks we explore – classification, disentanglement, reconstruction, and novel view synthesis – we compare SODA to the leading prior works. These include models such as SimCLR, DINO, and MAE for linear-probe classification [53–58], NeRF-based approaches for novel view generation [59, 60], and classic variational models for the task of disentanglement [61–63]. Whereas these techniques are designed for particular objectives or depend on domain-specific assumptions, SODA exhibits a greater degree of versatility, as it tackles representational and generative tasks alike.

3. Approach

SODA is a self-supervised diffusion model that learns a bidirectional mapping between images and latents. It consists of an image encoder $\mathcal{E}(\mathbf{x}') = \mathbf{z}$ that casts an *input view* \mathbf{x}' into a low-dimensional latent \mathbf{z} , which is then used to guide the synthesis of a novel *output view* \mathbf{x} , that relates to the input \mathbf{x}' (Figure 2). Concretely, \mathbf{x} is produced through a diffusion process that is conditioned on the encoding \mathbf{z} via *feature modulation* [49]. This design equips SODA with an explicit and compact latent space, which not only offers ample control over the generative process, but can also be leveraged for downstream perception tasks (Section 4)¹.

¹Our model is named after the soda drink. Indeed, the fizzing in soda bottles is an everyday example of the diffusion phenomena.

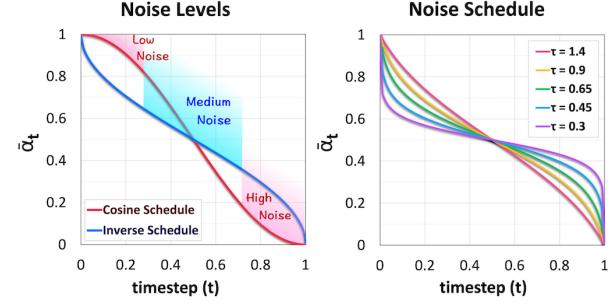


Figure 3. **Inverted Noise Schedule** used for SODA’s training. It prioritizes medium noise levels, which aid representation learning.

We first present an overview of the model (**Section 3.1**, **Figure 2**), followed by an in-depth discussion of each of its core components: the encoder’s architectural design (**Section 3.2**), the mechanisms involved in the synthesis of novel views (**Section 3.3**), and the optimization techniques we develop to cultivate strong and meaningful representations (**Section 3.4**).

3.1. Model Overview

As a denoising diffusion model [5, 25], SODA is formally defined by a pair of forward and backward Markov chains, iteratively transforming a sample \mathbf{x}_T from the normal distribution into the target one (\mathbf{x}_0) and vice versa. Each forward step t erodes \mathbf{x}_t by adding low Gaussian noise ϵ_t according to a fixed variance schedule α_t . Meanwhile, the respective backward step performs image denoising, and aims to estimate ϵ_t in order to recover \mathbf{x}_{t-1} from its successor \mathbf{x}_t . It is carried out by a decoder \mathcal{D} , implemented as a convolutional UNet [64] (with $2m + 1$ activation layers h_i).

To tackle the denoising challenge, we assist the decoder by conditioning it on a latent vector \mathbf{z} , which guides its operation by modulating the activations \mathbf{h} at each of its layers through adaptive group normalization that controls their scale and bias: $\mathbf{z}_s\text{GroupNorm}(\mathbf{h}) + \mathbf{z}_b$ where $(\mathbf{z}_s, \mathbf{z}_b)$ are linear projections of \mathbf{z} , applied evenly across the activation grid. See Appendix B for the architectural details of the decoder, as well as closed-form equations of the diffusion forward and backward steps. The latent \mathbf{z} is created by the newly introduced image encoder \mathcal{E} , discussed next.

3.2. The Encoder

The latent representation \mathbf{z} is at the model’s core, serving as the communication channel between the encoder \mathcal{E} and the denoising decoder \mathcal{D} , while guiding the latter through the diffusion process. It is derived by a ResNet encoder $\mathcal{E}(\mathbf{x}') = \mathbf{z}$, representing a clean source view \mathbf{x}' that semantically or visually relates to the target view \mathbf{x} (Section 3.3).

The driving motivation behind this idea stems from the denoising task’s inherent *under-determination*: Seeking to fulfill it to the best of its ability, the decoder will leverage



Figure 4. **Latent Controllability (Unsupervised).** We explore SODA’s latent space and discover directions that correspond to semantic attributes like face size and structure, lighting and viewpoint, maturity, expressions, fur length and color, texture, brightness and tone. SODA is trained fully unsupervised over images only. It successfully disentangles and encodes meaningful aspects into its learned representations.

any pertinent knowledge or useful clue that could inform it of the missing content to fill in. This in turn incentivizes the encoder to distill into \mathbf{z} the most striking and prominent commonalities between the source and target views. Since we further constrict the encoding \mathbf{z} into a **low-dimensional space**, and use it to guide the decoder via **global feature modulation**, it yields a fruitful combination that encourages \mathbf{z} to specifically capture the image’s *high-level semantics*, while delegating the reconstruction of *localized and high-frequency details* to the denoiser itself.

From that perspective, learning a latent \mathbf{z} that supports image **denoising**, rather than pure reconstruction from scratch, as in most auto-encoders [65–67], liberates our encoder from the need to compress *all* information about the image into the representation, and let it instead focus on the image’s most distinctive and descriptive qualities.

3.2.1 Layer Modulation & Masking

To enhance the model’s latent space disentanglement, we introduce two intertwined mechanisms of modulation and masking. In **layer modulation**, we partition the latent vector \mathbf{z} into $m + 1$ sections – half the number of layers in the decoder \mathcal{D} . We use each \mathbf{z}_i to modulate the respective pair or layers (h_i, h_{2m-i}) , thereby promoting specialization among the latent sub-vectors. Just like light rays refracted through a prism, they are encouraged to capture visual traits at different levels of granularity, from coarser to finer, so to guide the decoder’s operation through the layers.

To improve the localization and reduce the correlations among the m sub-vectors, we present **layer masking** – a layer-wise generalization of classifier-free guidance [27]. During training, we zero out a random subset of $\mathbf{z}_{1:m}$, effectively performing layer-wise guidance dropout, that mitigates the decoder’s reliance on sub-vector dependencies, allowing them to decouple and specialize independently. At

sampling, we extrapolate the model’s output in the conditional direction: $\epsilon_\theta(\mathbf{x}_t|\mathbf{z}) - \epsilon_\theta(\mathbf{x}_t|0)$. This endows SODA with finer control over the generative process, and opens the door for image editing and style mixing [49], as we can selectively condition the decoder on some levels of granularity, like structural or positional aspects, while giving it free rein to unconditionally vary other ones, such as lighting, texture, or color palette (see supplementary figures).

3.3. Novel View Generation

We loosely consider *views* to be any set of images that hold some relation among each other, such as visual or semantic (Figure 2): they can be various augmentations or distortions of an original image, as is commonly explored in the contrastive learning literature [58], they can show a 3D object from different poses and perspectives [21], or they can simply share the same semantic category with one another.

Cardinality. We permit the trivial singular case where all views are identical, which then turns the model into an auto-encoder. Conversely, we can extend the conditioning and create a novel view based on a *set of k input views*, instead of just a single one. We map each input \mathbf{x}^i to its latent with a shared encoder \mathcal{E} , and aggregate the resulting latents $\mathbf{z}^{1:k}$ into a single vector \mathbf{z} , either by taking their mean, or by processing them through a shallow transformer.

Perspective. The model can incorporate richer forms of conditional information, such as the camera perspective associated with each view: Specifically, for experiments over 3D datasets like ShapeNet (Section 4.2.2), we concatenate a grid of ray positions and directions $\mathbf{r} = (\mathbf{o}, \mathbf{d})$, embedded with sinusoidal positional encoding [68], to the linearly-mapped RGB channels of the source and denoised views, \mathbf{x}' and \mathbf{x}_t . This allows us to conditionally generate novel views that match the requested pose and orientation. See supplementary for illustrations and implementation details.

Guidance. We extend classifier-free guidance for novel view synthesis, and instead of masking just the latent z , we randomly and independently mask either the latent or the pose information r . As our empirical findings suggest, this idea not only hones the model’s generative skills, but further enables conditioning on partial information, allowing SODA to either unconditionally conceive *novel objects* at a requested pose, or alternatively, generate arbitrary novel views of given objects at the absence of source or target positional information, based on an image only (Appendix G).

Cross Attention. The technique of layer modulation offers the encoding z with *global control* over each of the decoder’s layers, by evenly setting their scales and biases across the grid. We further study alternative mechanisms, and explore the integration of cross attention, so to support *spatial modulation*. Instead of layer-wise modulation (which we use by default), we partition the latent z into n sub-vectors, and perform cross attention between the sub-vectors set and the decoders’ activations, akin to the word-based attention common in text-to-image generation. We find that cross attention aids the model at 3D novel view synthesis, while layer modulation performs better for image editing, reconstruction, and representation learning.

3.4. Training & Sampling

Noise Schedule. We train the model with the standard MSE objective [5], but introduce a new noise schedule to better fit the representation learning task: Indeed, diffusion models commonly set the variance of the additive noise term ϵ to follow either a cosine [69], sigmoid [70] or linear [5] decay schedules, prioritizing noise levels that are close to the margins, either of the high or low ends (Figure 3). Those schedules have been found useful for image synthesis.

However, from a representation learning perspective, denoising images with overly high or low noise levels fails to provide effective training signal for the model to learn from: Too little noise does not present the denoiser with a challenging enough task, thereby diminishing the encoder’s necessity. Meanwhile, too heavy noise puts excess pressure on the latent z to fully capture every pixel-level detail of the image, turning denoising into mere reconstruction. We thus incorporate a new inverted noise schedule, that promotes medium noise levels in lieu of the extremes, which proves highly conducive to representation quality (Section 4.1).

Additional Settings. Two modifications we find beneficial for representation learning are: (1) adding low Gaussian noise to the *encoder’s input images*; (2) optionally setting the encoder to have a higher learning rate than the decoder, so to positively impact their learning dynamics by allowing the encoder to adapt faster as it guides the decoder in the denoising task. While the model is robust to the selection of the learning-rate ratio, tuning it could improve downstream results. Once trained, we use DDPM [5] for sampling.

Table 1. **Linear-Probe Classification on ImageNet**, evaluating discriminative and generative approaches of comparable model sizes. *Top1 (Crop+Flip)* presents scores with light augmentations only, revealing the greater robustness of generative approaches in general and SODA in particular to data augmentations. (*) denotes cropping only, and (†) denotes no augmentation. (*) indicates that the decoder is not used for the downstream linear probing.

Method	Arch.	#θ	Top1	Top5	Top1 Crop+Flip
Discriminative Approaches					
Supervised [71]	RN50×2	94	79.9	95.0	-
SimCLR [58]	RN50×2	94	74.2	92.0	46.7
BYOL [72]	RN50×2	94	77.4	93.6	63.8
SwAV [57]	RN50×2	94	73.5	-	54.2
DINO [56]	ViT-B/16	86	74.9	-	61.1
SwAV + multi-crop	RN50×2	94	77.3	-	58.7
DINO + multi-crop	ViT-B/16	86	78.2	-	65.3
Generative Approaches					
Vanilla EncDec	RN50×2	118	8.5	17.6	10.2
Vanilla AutoEnc	RN50×2	118	14.3	28.9	-
iGPT [55]	GPT-2	76	41.9	-	41.9*
iGPT-L [55]	GPT-2	1386	65.2	-	65.2*
BEiT [54]	ViT-B/16	86	56.7	-	-
BigBiGAN [73]	RV50×4	86	61.3	81.9	61.3†
MAE [53]	ViT-B/16	86	68.0	-	68.0
Diffusion-based Approaches (Generative)					
Palette [30]	UNet	118	11.4	22.3	8.7
Unconditional Diffusion	UNet	118	24.5	44.4	28.3
SODA w/o bottleneck	RN50×2	94+35*	34.2	52.9	29.7
SODA w/o modulation	RN50×2	94+32*	56.7	78.5	51.1
SODA w/o novel views	RN50×2	94+36*	55.1	75.5	48.2
SODA w/o noise sched.	RN50×2	94+36*	62.0	82.9	56.8
SODA (ours)	RN50×2	94+36*	72.2	90.5	69.1

Table 2. **Image Reconstruction on ImageNet**, comparing discrete [74, 75] and continuous [8] approaches. The suffix of discrete methods refers to their codebook cardinality.

Method	Latent Dim	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
DALL-E2	1024	9.0	0.11	16.53	0.66
DALL-E,8K	512×32×32	22.8	0.73	32.01	1.95
VQGAN,1K	256×16×16	19.4	0.50	7.94	1.98
VQGAN,16K	256×16×16	19.9	0.51	4.98	1.83
VQGAN,8K	256×32×32	22.2	0.65	1.49	1.17
SODA (ours)	2048	23.6	0.93	2.77	0.19

4. Experiments

We evaluate SODA through a suite of quantitative and qualitative experiments, demonstrating its strong representation skills and generative capabilities over 12 different datasets grouped into 4 tasks: We begin with **linear-probe classification** (Section 4.1), showing the model’s utility for downstream perception. We proceed to **image reconstruction** and few-shot **novel view synthesis** (Section 4.2), illustrating its ability to envision 3D objects from new unseen perspectives. We then explore the model’s **disentanglement & controllability** (Section 4.3), as substantiated by comparative analysis and latent-space interpolations.

In the supplementary and our website website (soda-diffusion.github.io), we provide additional samples, visualizations and animations, and provide further details about our evaluation procedures, discussing the **datasets, metrics, baselines, and implementation details** (Appendices C-F). We conclude with **ablation studies** (Appendix G), that em-

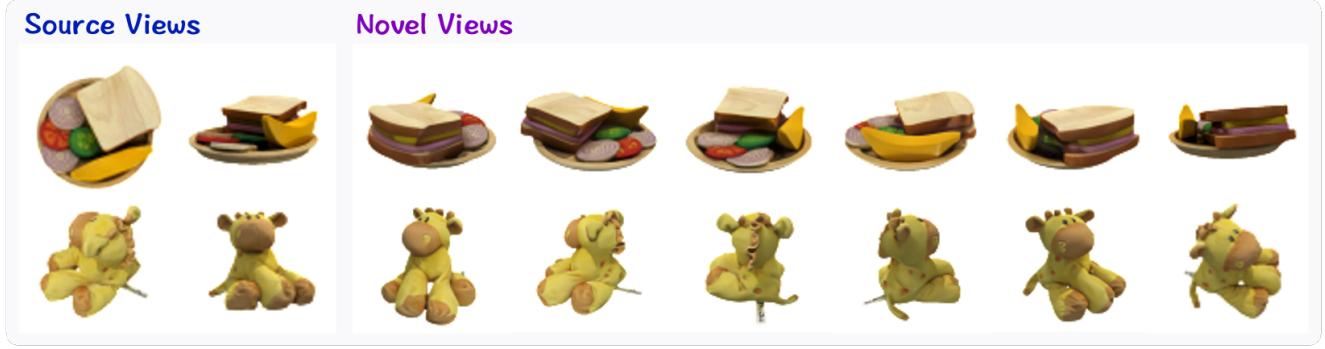


Figure 5. **Novel View Synthesis**, where given two source views of GSO objects, SODA generates their images from new perspectives.

pirically validate the contribution of the model’s components and design choices. Taken altogether, the evaluation offers solid evidence for the efficacy, robustness and versatility of our approach.

4.1. Linear-Probe Classification

We assess the quality of SODA’s learned representations through linear-probe analysis [72, 76–78], over ImageNet and CelebA, which complement one another: the former calls for fine-grained clustering into 1000 possible *categories*, while the latter involves rich identification of diverse semantic *attributes*. We train our model in a self-supervised fashion: using RandAugment [79] for ImageNet, and Gaussian data augmentation for CelebA. We then fit a linear classifier over the latent vectors z that predicts the category or attributes, and measure the resulting performance. We note that training diffusion models for representation learning is computationally efficient, since iterative sampling is necessary for generative purposes only.

As shown in Table 1, SODA reaches **72.24%** accuracy (**top1**) on the ImageNet1K linear-probe classification task, outshining competing generative approaches such as MAE, BEIT and iGPT [53–55], and significantly reducing the gap with discriminative and contrastive approaches like DINO, SwAV, and SimCLR [56–58]. Meanwhile, for CelebA, our model attains the strongest results (72.7% F1) compared to competing approaches (Supp Table 6), eclipsing even the language-supervised CLIP embeddings (71.1% F1).

SODA proves remarkably **robust to the choice of data augmentation**, as it performs strongly regardless of the selected strategy, seeing only a minor decrease of 3.1% when switching the heavier RandAugment for a lighter **crop+flip** augmentation. This stands in stark contrast to the high sensitivity of contrastive methods to data augmentations, with e.g. BYOL and SimCLR suffering from major drops of 13.6% and 27.5% respectively when light augmentation is applied (crop+flip), and other approaches relying on particular schemes such as MultiCrop [57] among others.

The newly introduced **image encoder** plays an instrumental role in the model’s downstream performance, en-

abling a 3x boost over features obtained from an *unconditional diffusion* model [19]², which for ImageNet scores 24.49% only. A significant improvement further arises from the use of **novel view synthesis as a self-supervised representation learning objective**: Indeed, maintaining a distinction between the model’s source and target views yields a 17.12% increase for ImageNet, compared to when they match (i.e. auto-encoding), even though the same data augmentation is applied in both cases (Figure 6)³.

Other contributors include the **compact bottleneck** and **feature modulation**, which respectively raise accuracy by 38.04% and 15.51% for ImageNet, and 12.74% and 11.25% F1 for CelebA (Supp Table 11). Designing a new inverted **noise schedule** that favors medium noise levels in lieu of the extremes likewise strengthen the model’s representation capacity, eliciting a 10.25% increase over ImageNet.

4.2. Visual Synthesis

Next, we analyze the model’s generative skills, evaluating its ability to faithfully **reconstruct** an image x from its latent encoding z (Section 4.2.1)⁴, and **generate novel views** of 3D objects from requested camera perspectives, given one or more conditional source views (Section 4.2.2). We evaluate the targets and predictions’ similarity along multiple dimensions: pixel-wise (PSNR [81]), structural (SSIM [82]), perceptual (FID [83]) which accounts for sharpness and realism, and semantic (LPIPS [84]) (Appendix E.2).

4.2.1 Image Reconstruction

As indicated by Table 2, SODA produces excellent reconstructions, surpassing competing approaches like VQGAN [75], StyleGAN2 [85], DALL-E [74] and unCLIP (DALL-E2) [8], especially in terms of structural and semantic similarity (SSIM & LPIPS). Visually speaking, our samples

²The baseline obtains image encodings by pooling the features of the best-performing denoiser layer (a middle one) over a lightly-noised image.

³For auto-encoding, we sample a new augmentation at every training step, but use it both as the source and as the target.

⁴To evaluate our model at the task of reconstruction, we keep the source and the target views the same during training $x' = x_0$.

Table 3. **Novel View Synthesis**, from a single source view, on real and synthetic objects, comparing Geometry-free and -aware approaches. *PSNR*, *SSIM*, and *LPIPS* respectively express pixel-wise, structural and semantic similarity, while *FID* captures sharpness and realism. For these experiments, we integrate cross attention with feature modulation (Section 3.3). (★) denotes short sampling, with 20 steps only.

Method	GSO				ShapeNet				NMR			
	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
Geometry-Aware Approaches												
PixelNeRF [60]	24.93	0.919	48.72	0.086	26.58	0.940	25.34	0.073	28.19	0.932	34.54	0.082
NeRF-VAE [59]	22.20	0.882	74.83	0.113	24.60	0.915	45.79	0.101	25.56	0.893	71.67	0.134
Geometry-Free Approaches												
Vanilla Auto-Encoder	24.46	0.941	75.96	0.129	24.71	0.943	42.23	0.096	24.22	0.948	51.42	0.116
SRT [80]	21.97	0.877	38.64	0.110	26.31	0.934	17.96	0.073	25.69	0.898	7.90	0.090
Diffusion-based Approaches (Geometry-Free)												
Palette [30]	13.42	0.672	8.49	0.199	14.44	0.582	6.93	0.177	14.10	0.609	6.03	0.212
DALL-E2 [8]	15.68	0.793	8.32	0.147	18.75	0.823	6.54	0.101	20.32	0.899	3.61	0.087
SODA w/o bottleneck	20.97	0.926	4.39	0.071	24.31	0.949	2.83	0.051	23.31	0.948	0.75	0.053
SODA w/o modulation	21.02	0.929	4.10	0.069	25.02	0.944	2.97	0.048	25.34	0.949	0.77	0.051
SODA (shorter sampling)*	24.38	0.930	2.35	0.065	26.71	0.946	1.31	0.046	27.13	0.936	1.10	0.063
SODA (ours)	24.97	0.945	1.51	0.054	27.42	0.947	0.74	0.039	28.71	0.952	0.81	0.048

are sharper and crispier than DALL-E’s, while being more accurate than StyleGAN2 and unCLIP inversions, perhaps due to their lack of a trainable encoder (see supplementary examples). The results are significant given the order-of-magnitude lower dimensionality of the latents z from which we restore the images: 2K for SODA versus 65-524K for VQGAN and DALL-E⁵, illustrating here an advantage of continuous representations over discrete codebooks.

4.2.2 Novel View Generation

For the task of few-shot novel view synthesis, we focus on the 3D regime and look into 3 datasets that span both synthetic object renderings and real-world scans of household items (Google Scanned Objects [86], custom ShapeNet [87], and NMR [88]). We compare our model to geometry-free and -aware approaches such as PixelNeRF [60], Scene Representation Transformer (SRT) [80], and Palette [30]. We condition the models on 1-9 source views, and test them on held-out validation objects that do not appear in training.

SODA consistently beats the competing approaches across the 3 datasets and for different numbers of source views, as indicated by FID, SSIM and LPIPS (Table 3 and Supplementary Figure 7). It reaches the largest gains along LPIPS and FID, producing significantly sharper images that better match the source views both structurally and semantically. We observe that settings of 1-3 source views benefit the most from our model, where for the single-source case, it improves FID scores by an order-of-magnitude and often almost halves the LPIPS scores. For the GSO dataset, as we increase the source views number, we score a little lower on the pixel-wise PSNR than PixelNeRF, perhaps due to the probabilistic nature of our approach. Yet, in terms of computational efficiency, contrary to the slow and heavy rendering of geometry-aware methods, SODA maintains strong performance with as little as 20 sampling steps.

Figure 5 and the supplementary animations feature objects synthesized from various perspectives, showcasing

⁵The overall latent dimension is $16 \times 16 \times 256 - 32 \times 32 \times 512$.

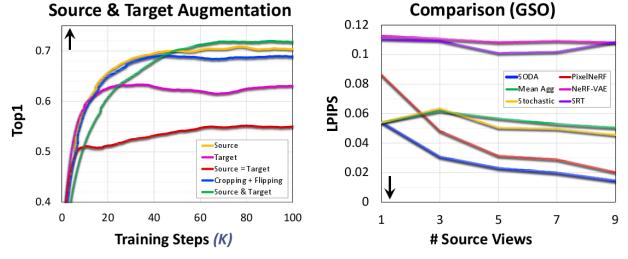


Figure 6. (**Left**): Contribution of the Novel View Synthesis objective to downstream classification on ImageNet. (**Right**): Generative quality comparison with baseline approaches and aggregation methods as we increase the number of source views.

the viewpoint consistency SODA achieves. For multiple sources, we find that our proposed transformer-based view aggregation (Section 3.3) surpasses the stochastic conditioning technique of 3DiM [89] (Figure 6). Our approach further outperforms the denoiser-only Palette diffusion model, which fits translational tasks that closely follow the source layout, like colorization or super-resolution, but struggles at structural transformations, corroborating the need for our dedicated image encoder.

4.3. Disentanglement & Controllability

The concept of disentanglement has been a recurring theme in representation learning research over the years [90–94]. While formal definitions may vary [95–99], a common aim lies in the discovery of abstract and meaningful latent representations that linearly align with the natural axes of variation. Disentanglement could enhance the encodings’ interpretability, and, in the context of generative modeling, support greater controllability. In the following, we inspect the model’s latent space, and analyze it quantitatively and qualitatively along the dimensions of disentanglement, controllability and informativeness.

4.3.1 Qualitative Evaluation

Latent Interpolations. We begin by visualizing latent interpolations for our model, linearly traversing the latent space from one vector z_1 to another z_2 (Figure 1 and the supplementary figures & animations). We observe smooth variations over traits of texture and structure. Notable in particular are image categories that seamlessly morph from one to another (e.g. from a tiger to a cat to a dog to a wolf), while for CelebA, we see gradual transformations ranging from broad shifts of pose and orientation to finer transitions of hair, facial features and expressions.

Attribute Manipulation. We go beyond interpolations and identify meaningful latent directions that correspond to individual axes of variation. To infer them, we explore two techniques: **supervised**, by normalizing the linear-probe’s weight matrix, and **unsupervised**, through PCA decomposition [100] (details at Appendix E.3). Figure 4 and the supplementary figures show perturbations along the discovered directions, which influence various semantic properties: from age and gender in CelebA, to tone, clarity and lighting conditions in LSUN, to object’s dimensions, thickness and material in ShapeNet and GSO. Indeed, we see that these manipulations are disentangled: one attribute is altered while the others are mostly kept intact, demonstrating the well-behaved nature of SODA’s emergent latent space and the quality and strength of its learned representations. Indeed, we emphasize that both SODA’s training and the following PCA-based discovery of interpretable latent directions are fully-unsupervised, derived from images only.

Layer Modulation. We investigate the effect of layer modulation and masking (Section 3.2.1), blocking the encoder’s guidance from select decoder’s layers and inspecting the impact on the produced outputs. As illustrated in the supplementary, it allows for selective modification of input images at different levels of granularity, so to preserve certain factors while unconditionally regenerating other ones. We can thus resample a new color palette while retaining shape and structure, or change the background while preserving the subject’s identity. We find that layer masking improves the model’s robustness to these forms of partial conditioning. Meanwhile, the role played by the initial Gaussian **noise map** x_T is closely linked to the chosen augmentation scheme: it controls fine stochastic subtleties like fur or freckles when SODA is trained to reconstruct the source image, and could conversely shape the underlying layout when heavier data augmentations are applied.

4.3.2 Quantitative Evaluation

To quantitatively bolster the findings above, we analyze our approach with DCI [94], which measures representations along Disentanglement, Completeness and Informativeness by assessing the degree of 1-to-1 correspondence between

Table 4. **Disentanglement Analysis (DCI)** for various datasets. SODA achieves improvements of 27.2-58.3% in Disentanglement, 5.0-23.8% in Completeness, and comparable Informativeness to variational approaches. We report here the 3 metrics’ average. StyleGAN2 achieves 68.71% (W) and 69.07% (W+) for CelebA. See Supplementary Table 7 for the full comparison.

Method	CelebA	CUB	MPI3D	3DShapes	SmallNORB
AnnealedVAE [106]	53.09	46.94	30.95	72.11	33.14
FactorVAE [63]	54.76	47.62	43.99	91.26	50.04
DIP-VAE (I) [107]	58.63	46.39	55.94	94.11	47.16
DIP-VAE (II) [107]	57.02	47.22	44.40	94.07	51.20
β -VAE [67]	56.96	47.39	47.77	94.45	50.40
β -TCVAE [108]	56.61	48.59	50.33	88.51	50.56
SODA w/o layer mod.	65.61	54.75	70.71	80.09	59.10
SODA (ours)	74.67	56.98	73.41	94.08	64.78

latent and ground-truth factors of variation (Appendix E.3). We evaluate models over multiple semantically-annotated datasets, ranging from the diagnostic SmallNORB and MPI3D to the realistic CUB and CelebA [101–105].

As Table 4 and supplementary Tables 6 and 7 show, SODA outshines both variational and adversarial approaches, improving Disentanglement by 27.2-58.3% and Completeness by 5.0-23.8% across 4 different datasets, with the sole exception of the synthetic 3DShapes, for which both SODA and most variational methods attain excellent scores. For Informativeness, results are mostly comparable, with SODA taking the lead for some datasets, while StyleGAN or DIP-VAE [107] improving scores for others. Our experiments further validate the contribution of layer modulation and masking, respectively yielding 3.2-13.5% and 2.5-8.9% increases in latent-space Disentanglement, and 3.8% and 1.7% mean increase in Completeness. Visually, SODA’s samples are significantly sharper than the variational ones, and it achieves remarkable boosts in realism (FID) and semantic similarity (LPIPS).

5. Conclusion

We introduced SODA, a self-supervised diffusion model, designed for both perception and synthesis. It re-purposes the task of novel view generation as a training objective for representation learning. By conditioning a denoiser on an image encoder, and imposing an information bottleneck between the two, SODA learns strong semantic representations that enable downstream classification, as well as reconstruction, editing and synthesis. While we focused on single-object images, as in LSUN, ShapeNet, or ImageNet, we believe that exploring the applicability of our approach to dynamic compositional scenes is a promising direction for future research. We hope our work will help bridging the gap between novel view synthesis and self-supervised learning, two flourishing topics that are often pursued independently, and bring us one step closer to unlocking the potential of generative models in general and diffusion models in particular to advance the representational frontier.

References

- [1] David R Krathwohl. A revision of bloom’s taxonomy: An overview. *Theory into practice*, 41(4):212–218, 2002. 1
- [2] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 1
- [3] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006. 1
- [4] Christos H Papadimitriou. Computational complexity. In *Encyclopedia of computer science*, pages 260–265. 2003. 1
- [5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3, 5, 14
- [6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 14
- [7] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 14, 21
- [8] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 3, 5, 6, 7, 16, 21
- [9] 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. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.
- [10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3
- [11] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.
- [12] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.
- [13] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.
- [14] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022.
- [15] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.
- [16] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023.
- [17] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 1
- [18] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. *arXiv preprint arXiv:2303.16203*, 2023. 1
- [19] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. *arXiv preprint arXiv:2303.09769*, 2023. 6, 20
- [20] Korbinian Abstreiter, Sarthak Mittal, Stefan Bauer, Bernhard Schölkopf, and Arash Mehrjou. Diffusion-based representation learning. *arXiv preprint arXiv:2105.14257*, 2021. 1, 2
- [21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 4
- [22] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 2
- [23] Sybren Ruurds De Groot and Peter Mazur. *Non-equilibrium thermodynamics*. Courier Corporation, 2013. 2
- [24] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2
- [25] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 3
- [26] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.
- [27] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4, 14
- [28] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 2
- [29] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022. 2

- [30] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. [5](#), [7](#), [21](#), [23](#)
- [31] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022.
- [32] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. *arXiv e-prints*, pages arXiv–2303, 2023.
- [33] Allan Jabri, Sjoerd van Steenkiste, Emiel Hoogeboom, Mehdi SM Sajjadi, and Thomas Kipf. DORsal: Diffusion for object-centric representations of scenes et al. *arXiv e-prints*, pages arXiv–2306, 2023. [2](#)
- [34] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. [2](#)
- [35] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [36] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.
- [37] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.
- [38] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022. [2](#)
- [39] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. [2](#)
- [40] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022. [2](#)
- [41] Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero-shot classifiers. *arXiv preprint arXiv:2303.15233*, 2023. [2](#)
- [42] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. *arXiv preprint arXiv:2303.02153*, 2023. [2](#)
- [43] Benno Krojer, Elinor Poole-Dayan, Vikram Voleti, Christopher Pal, and Siva Reddy. Are diffusion models vision-and-language reasoners? *arXiv preprint arXiv:2305.16397*, 2023. [2](#)
- [44] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. [2](#)
- [45] Krzysztof J Geras and Charles Sutton. Scheduled denoising autoencoders. *arXiv preprint arXiv:1406.3269*, 2014.
- [46] B Chandra and Rajesh Kumar Sharma. Adaptive noise schedule for denoising autoencoder. In *Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3–6, 2014. Proceedings, Part I* 21, pages 535–542. Springer, 2014.
- [47] Qianjun Zhang and Lei Zhang. Convolutional adaptive denoising autoencoders for hierarchical feature extraction. *Frontiers of Computer Science*, 12:1140–1148, 2018. [2](#)
- [48] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsu, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022. [3](#), [16](#), [22](#)
- [49] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. [3](#), [4](#), [14](#), [21](#)
- [50] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. [14](#)
- [51] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. [3](#), [23](#)
- [52] Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. Infodiffusion: Representation learning using information maximizing diffusion models. *arXiv preprint arXiv:2306.08757*, 2023. [3](#)
- [53] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. [3](#), [5](#), [6](#), [16](#), [17](#), [21](#)
- [54] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2021. [5](#), [21](#)
- [55] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. [5](#), [6](#), [21](#)
- [56] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. [5](#), [6](#), [17](#), [21](#)
- [57] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. [5](#), [6](#), [21](#)

- [58] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3, 4, 5, 6, 16, 17, 21
- [59] Adam R Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Sona Mokrá, and Danilo Jimenez Rezende. Nerf-vae: A geometry aware 3d scene generative model. In *International Conference on Machine Learning*, pages 5742–5752. PMLR, 2021. 3, 7, 15, 22
- [60] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 3, 7, 15, 22, 24
- [61] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019. 3, 19, 22
- [62] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016.
- [63] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 3, 8, 16, 18, 22
- [64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3
- [65] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 4
- [66] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 22
- [67] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016. 4, 8, 16, 18, 22
- [68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4, 14, 15, 24
- [69] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 5, 14
- [70] Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022. 5
- [71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 14
- [72] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 5, 6, 21
- [73] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019. 5
- [74] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 5, 6, 21
- [75] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 5, 6, 16, 21
- [76] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. 6
- [77] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [78] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 6
- [79] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 6, 17, 23
- [80] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6229–6238, 2022. 7, 15, 22, 24
- [81] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 6, 19
- [82] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6, 19

- [83] 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. *Advances in neural information processing systems*, 30, 2017. 6, 19
- [84] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6, 19
- [85] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 6, 16
- [86] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 7, 16, 21
- [87] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7, 16, 21
- [88] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 7, 16, 21
- [89] Daniel Watson, William Chan, Ricardo Martin Bralla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2022. 7, 15, 22
- [90] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 7
- [91] James J DiCarlo and David D Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007.
- [92] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992.
- [93] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.
- [94] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International conference on learning representations*, 2018. 7, 8, 19
- [95] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992. 7
- [96] Karl Ridgeway. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299*, 2016.
- [97] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.
- [98] Alessandro Achille and Stefano Soatto. On the emergence of invariance and disentangling in deep representations. *arXiv preprint arXiv:1706.01350*, 125:126–127, 2017.
- [99] Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018. 7
- [100] Erik Häkkinen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems*, 33:9841–9850, 2020. 8, 20
- [101] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104. IEEE, 2004. 8, 16, 21
- [102] Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *Advances in Neural Information Processing Systems*, 32, 2019. 16, 21
- [103] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 16
- [104] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 16, 21
- [105] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August, 15(2018):11*, 2018. 8, 16, 21
- [106] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv e-prints*, pages arXiv–1804, 2018. 8, 16, 18, 22
- [107] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018. 8, 16, 18, 22
- [108] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018. 8, 16, 18, 22
- [109] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv e-prints*, pages arXiv–1412, 2014. 14
- [110] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the

- impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 14
- [111] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 14
- [112] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *International Conference on Learning Representations*, 2016. 14, 23
- [113] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 14
- [114] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 14
- [115] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. 2018. 14
- [116] Hongchang Gao, Jian Pei, and Heng Huang. Progan: Network embedding via proximity generative adversarial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1308–1316, 2019. 15
- [117] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 16, 21
- [118] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 16, 21
- [119] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 16, 21
- [120] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 16, 21
- [121] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 16
- [122] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3761, 2022. 16
- [123] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 17
- [124] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 17
- [125] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015. 19
- [126] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 19
- [127] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987. 20
- [128] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 21
- [129] Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv e-prints*, pages arXiv–1901, 2019. 21, 23
- [130] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 21
- [131] Vincent Sitzmann, Semon Rezhikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34:19313–19325, 2021. 22

Supplementary Material

A. Overview

In the following, we discuss additional analysis of our approach, and provide further description of the model structure, implementation details, and evaluation procedures. **Appendix B** offers an overview of diffusion models' preliminaries and equations. In **Appendix C**, we then specify the chosen hyperparameters, training techniques, and sampling methods. **Appendices D–F** respectively review the datasets, metrics, and baselines we consider in this study. Finally, in **Appendix G**, we present ablation and variation studies that assess the contribution of each of our design choices, complementing the principal ones explored in the main paper.

We plan very soon to add to the supplementary and our website (soda-diffusion.github.io) a variety of animations and visualizations of outputs generated by the model over different datasets, spanning image reconstructions, viewpoint traversals, latent interpolations, unsupervised attribute discovery and manipulation, demonstration of style and content (or structure) separation, qualitative impact of layer masking and variation of the initial noise map for different training data augmentation schemes, and samples conditioned on partial information.

B. Model Overview & Diffusion Preliminaries

As a denoising diffusion model [5], SODA is formally defined by a pair of forward and backward Markov chains that represent a T -steps transformation from a normal distribution $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ into the learned data distribution $\mathbf{x}_0 \sim p_\theta(\mathbf{x})$ and vice versa, where $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$. Each forward step t erodes \mathbf{x}_t by adding a small Gaussian noise according to a fixed variance schedule α_t , sampling:

$$\mathbf{x}_t \sim \mathcal{N}(\sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t) \mathbf{I})$$

Meanwhile, each reverse step t performs image denoising, and aims to estimate ϵ_t in order to recover $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{z}, \mathbf{c})$ where the latent representation $\mathbf{z} \in \mathbb{R}^D$ serves as a guidance source for denoising the image, and is produced by the encoder through $\mathbf{z} = \mathcal{E}(\mathbf{x}', \mathbf{c}')$, the image \mathbf{x}' is a related clean input view given to the encoder, and \mathbf{c}, \mathbf{c}' denote optional conditions for the encoder and decoder respectively (e.g. source and target camera perspectives of a 3D object). We note that this formulation contrasts with unconditional diffusion models, which rely on (\mathbf{x}_t, t) only. The reverse step is realized by a denoising decoder \mathcal{D} that predicts $\epsilon = \epsilon_\theta(\mathbf{x}_t, t, \mathbf{z}, \mathbf{c})$. Thanks to the reparametrization trick [5], we can then sample the following:

$$\mathbf{x}_{t-1} \sim \mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\cdot) \right), \sigma_t^2 \mathbf{I}\right)$$

where $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ is the product of the variances up to step t , and σ_t^2 is either a fixed or learned variance term. To train the model, we can readily obtain \mathbf{x}_t with the closed-form computation (where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$):

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_{t-1} + (1 - \bar{\alpha}_t) \epsilon$$

and couple it with the simplified re-weighted MSE training objective (where ϵ_θ is estimated by the model):

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, \epsilon, t} [\|\epsilon - \epsilon_\theta\|_2^2]$$

In terms of the architecture, our model consists of an image encoder \mathcal{E} (ResNet or ViT), and a denoising decoder \mathcal{D} that follows the classic structural design of prior literature [5, 6], featuring a UNet implemented as a stack of residual, convolutional, and either downsampling or upsampling layers (in the encoding and decoding modules of the UNet respectively), that are further linked by symmetric skip connections. The decoder \mathcal{D} notably integrates Adaptive Group Normalization layers [6, 49, 50] throughout, allowing \mathbf{z} and t to modulate the decoder's activations of each layer \mathbf{h} , by scaling and shifting them channel-wise:

$$\text{AdaGN}(\mathbf{h}, \mathbf{z}, t) = \mathbf{z}_s (\mathbf{t}_s \text{GroupNorm}(\mathbf{h}) + \mathbf{t}_b) + \mathbf{z}_b$$

where $(\mathbf{t}_s, \mathbf{t}_b)$ and $(\mathbf{z}_s, \mathbf{z}_b)$ are both obtained by linear projections, the former of a sinusoidal timestep embedding of t [68], and the latter of the latent representation \mathbf{z} created by the image encoder \mathcal{E} .

C. Implementation Details

Architecture. See Table 12 for our chosen hyperparameters. In terms of the training objective, optimization scheme and empirical configuration, we adopt most of the common settings of recent works [5–7], and specifically use the Adam optimizer [109], gradient accumulation, and exponential moving average for the model's weights; for the ResNet encoder [71]: variant v2 ResNet [110], Xavier initialization [111], ReLU non-linearity, dropPath [112], and mean pooling; and for the UNet decoder: truncated normal initialization (JAX default), GeLU non-linearity [113], $\sqrt{2}$ rescaling of residual connections, BigGAN re-sampling order [114], and self attention in the decoder's low-resolution layers (8–32).

Training. For each dataset, we train the model until convergence, as measured by lack of improvement over a set number of training steps along a validation metric of choice (either downstream accuracy or SSIM). For sampling, we use discrete-time DDPM [5], classifier-free guidance [27] and 1000 diffusion timesteps, practically strided into 75–250 steps [69]. We implement SODA in JAX [115], and run our experiments either on NVIDIA Tesla V100s or TPUs (v2).

Table 5. **Novel View Synthesis (FID)**, comparing different approaches and aggregation methods as we vary the number of source views. *Stochastic Conditioning* guides each sampling step with a randomly-chosen source view. *Mean Aggregation* conditions on multiple source views by averaging their latents, while *Transformer Aggregation* instead uses a shallow transformer to aggregate the view representations.

# Source Views	GSO				ShapeNet					
	1	3	5	7	9	1	3	5	7	9
NeRF-VAE [59]	74.835	79.984	76.965	81.334	80.926	45.791	42.165	36.592	35.441	34.134
SRT [80]	38.642	70.665	40.728	51.936	74.705	17.956	16.336	13.717	27.719	28.026
PixelNeRF [60]	48.721	20.659	7.934	5.906	3.622	25.341	9.679	4.591	3.607	2.557
SODA with Mean Aggregation	1.508	2.290	2.060	2.183	1.754	0.736	0.696	0.667	0.679	0.742
SODA with Stochastic Conditioning [89]	1.508	2.117	1.360	1.177	1.228	0.736	0.706	0.686	0.679	0.697
SODA (Transformer Aggregation)	1.508	0.962	0.797	0.711	0.653	0.736	0.491	0.458	0.378	0.319

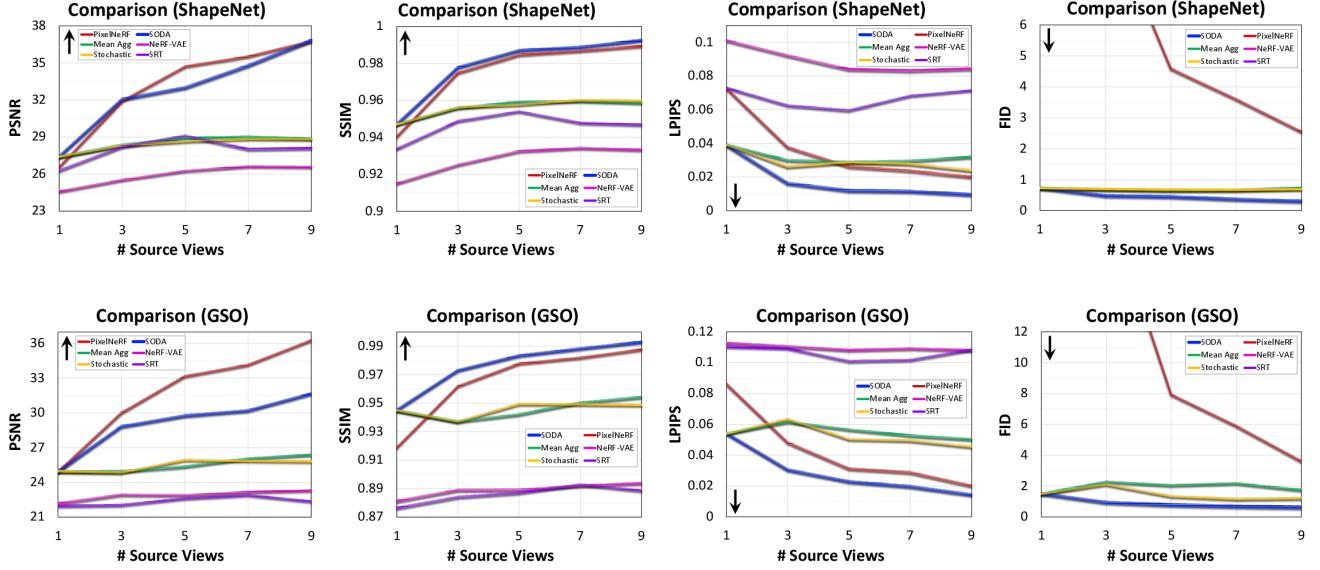


Figure 7. **Number of source views’ impact on models performance**, along PSNR, SSIM, LPIPS and FID. We see that as we increase the number of views, SODA achieves a bit higher SSIM score than PixelNeRF, comparable or lower PSNR, better LPIPS score, and much better FID. Other approaches lag behind PixelNeRF and SODA. In terms of view aggregation, we see that stochastic conditioning performs similarly to averaging the view representations, and that our transformer-based aggregation performs robustly better than these alternatives.

Positional Encoding. We employ sinusoidal positional encoding [68] to represent both timesteps and, in the case of pose-conditional view synthesis, spatial coordinates, either xy grids for the 2D case or camera rays’ origins and directions for 3D, normalized to a range of $[-1, 1]$. In contrast to the original encoding scheme used to represent discrete word positions, we further scale the arguments of sin and cos by a factor of $2\pi s$ (with s being a hyperparameter), so to increase the distinction among the positional encodings (Figure 10).

Pose Conditioning. Throughout the paper, we experiment with several different flavors of the novel view synthesis task: either generating a view conditionally, matching a 3D pose or 2D coordinates, or alternatively, in a pose-unconditional fashion: where given a source view, the model is asked to generate arbitrary novel views at perspectives of its choice). For the conditional case, we represent each perspective by a $H \times W$ 2D grid – of $(x_0, y_0) \times (x_1, y_1)$

in the 2D case, and ray positions and directions in the 3D case – embedded by sinusoidal positional encoding and concatenated to linearly-mapped RGB channels of the corresponding view, after the first layer of the encoder and the denoiser respectively. In Appendix G, we compare different ways to represent the rays, such as through normalization, by casting them on a plane or a sphere, or by summing up their positions and directions.

Learning Rates. For the ImageNet dataset, we maintain a different learning rate between the encoder and the denoiser, at a ratio of $\frac{lr_E}{lr_D} > 1$. We practically implement it by following the idea of learning rate equalization [116], scaling down the initialized weights of the encoder by a factor of k (by scaling down the standard deviation of the initialization distribution), and then having the network itself scale them back up by k , effectively scaling the encoder’s gradients by k . While the model is robust to the selection of the learning rate ratio, we find that a ratio of 2 yields

Table 6. **Performance Comparison on CelebA** of classification, reconstruction, and disentanglement, considering variational, adversarial and diffusion-based approaches. *Disen.* stands for Disentanglement, *Comp.* for Completeness, and *Info.* for Informativeness.

Method	Latent Dim	F1 \uparrow	Disen. \uparrow	Comp. \uparrow	Info. \uparrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow
Variational Approaches									
Vanilla Auto-Encoder	2048	66.35	38.94	29.82	84.52	21.61	0.906	65.50	0.327
AnnealedVAE [106]	2048	68.94	42.99	30.75	85.53	15.94	0.686	145.28	0.433
FactorVAE [63]	2048	71.26	46.34	29.87	88.06	20.17	0.890	87.19	0.331
DIP-VAE (I) [107]	2048	71.94	49.09	37.58	89.23	19.83	0.884	81.02	0.316
DIP-VAE (II) [107]	2048	71.85	48.56	33.37	89.12	19.95	0.887	82.41	0.307
β -VAE [67]	2048	71.98	49.04	32.46	89.37	19.94	0.886	78.65	0.314
β -TCVAE [108]	2048	71.82	49.11	31.63	89.10	20.18	0.892	80.33	0.315
Adversarial Approaches									
VQGAN [75]	256 \times 16 \times 16	-	-	-	-	23.28*	0.773*	-	0.311*
StyleGAN2-W [85]	512	-	53.26	56.15	96.71	16.76*	0.662*	-	0.394*
StyleGAN2-W+ [85]	512 \times 14	-	52.71	60.03	94.46	21.42*	0.813*	-	0.345*
Diffusion-based Approaches									
Unconditional Diffusion	-	63.67	41.33	30.72	84.94	-	-	25.33	-
DiffAE [48]	512	68.70	64.39	39.25	84.61	15.28*	0.681*	-	0.392*
DALL-E2 (with CLIP) [8]	1024	71.08	51.60	37.82	87.87	9.34	0.311	21.91	0.484
SODA (ResNet50 \times 2)	2048	72.65	79.93	53.62	90.44	18.78	0.859	9.54	0.273

optimal downstream results (Appendix G).

D. Datasets, Preprocessing & Augmentations

D.1. Datasets Overview

Throughout this work, we evaluate models over various datasets grouped into multiple tasks, as summarized by Table 10 and through the textual description below:

Representation Learning & Reconstruction:

Each image in the following datasets is associated with a category label (or for CelebA, with multiple attribute annotations).

- (1) **Imagenet1K** [117]: includes diverse images of objects among 1,000 categories of e.g. animals, instruments, furniture and food items.
- (2) **CelebA-HQ** [105]: features face images, annotated with 40 binary semantic properties like age, gender, or hair color; used also for quantitative disentanglement analysis.
- (3) **LSUN** [118]: partitioned into multiple categories of objects (like cars, cats and horses) and scenes (e.g. bedrooms and churches); See Table 10 for full list.
- (4) **Animal Faces-HQ (AFHQ)** [119]: covers various breeds of cats, dogs and wildlife.
- (5) **Oxford Flowers 102** [120]: features diverse flowers from the United Kingdom.

Novel View Synthesis:

Each image in the following datasets is associated with the camera perspective it was captured from, expressed as a grid of ray positions and directions $r = (\mathbf{o}, \mathbf{d})$.

- (6) **NMR** [88]: consists of ShapeNet [87] objects' renderings at 24 fixed views, evenly spaced around a surrounding ring with constant radius and altitude; images of 64 \times 64 resolution. We use the SoftRas data split [121].

(7) **ShapeNet**: our custom ShapeNet renderings dataset, featuring 120 views randomly sampled from an upper hemisphere, with random azimuth ϕ , altitude θ , and radius $r \in [r_{\min}, r_{\max}]$; 256 \times 256 resolution; created by the Blender-based Kubric library [122].

(8) **Google Scanned Objects (GSO)** [86]: includes scans of real-world household items, which we render with Blender following the same protocol described above.

Disentanglement (Quantitative):

Each image in the following datasets is associated with discrete semantic attribute annotations.

- (9) **SmallNORB** [101]: contains toy images belonging to 5 categories like animals and vehicles, captured from various camera perspectives and lighting conditions.
- (10) **3DShapes** [103]: includes images of a centered object among varied combinations of shape, color, size and orientation (4 shapes, 8 scales, 15 orientations, and 10 possible colors for the object, wall and floor).
- (11) **MPI3D** [102]: includes 4 splits of either synthetic or real objects, held by a robotic arm, with different discrete attributes (4-6 shapes, 4-6 colors, 2 sizes, 3 background colors, and 3 \times 40 \times 40 camera perspectives).
- (12) **Caltech-UCSD Birds (CUB-200-2011)** [104]: contains images of various bird species, annotated with 312 binary semantic properties.

D.2. Data Preprocessing

Resolution. We resize all images for training and evaluation to a source resolution of 256 \times 256, inputted into the encoder \mathcal{E} , and target resolution 128 \times 128, produced by the decoder \mathcal{D} , with the exception of CUB and ImageNet: for the former, we center-crop and pad each image based on its associated bird's bounding box; for the latter, we first resize the target images to 256 \times 256, and then center-crop them to 224 \times 224, matching prior literature [53, 58].

We keep the model's output resolution as 128 since ac-

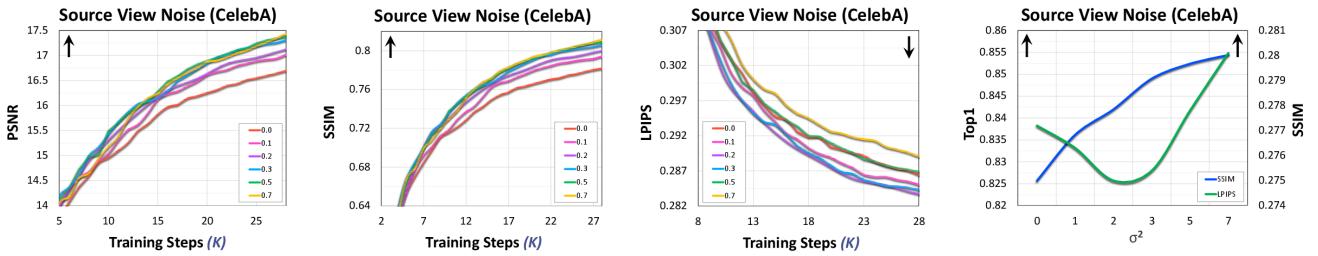


Figure 8. **Impact of source view noise** on the model’s generative and representational capabilities.

cording to diffusion models’ practices, higher-resolution images are commonly produced through cascading [123], where a core module first generates images of resolution 64 or 128, and these are subsequently post-processed by an independent super-resolution module, rather than being created as high-resolution directly. Indeed, this technique has been shown to improve the overall sample quality, and could readily fit with our approach as well.

Normalization. We normalize the input images fed into the encoder \mathcal{E} based on ImageNet mean and variance statistics [124], while linearly scaling the target images of the denoising decoder \mathcal{D} to the range $[-1, 1]$, following the standard procedures.

Data Splits. For each dataset, we either use the default splits, or if not provided, split them into 80% training, 10% validation and 10% testing. Data is shuffled at training time. We note that for all the multi-view datasets: NMR, ShapeNet, GSO, and smallNorb, we intentionally keep all the views of each object exclusively grouped within one of the splits, and consequently, all the objects used for evaluation are not included in the training set.

D.3. Data Augmentation

We study several augmentation schemes, applied for different tasks and datasets: by default, we use random resize cropping, horizontal flipping and optionally RandAugment [79] data augmentation on both the source and target views \mathbf{x}' and \mathbf{x} (encoded and denoised respectively). Specifically, at every training step, we randomly augment each view, at the rates specified in Table 12. To train the subsequent downstream classifier, we perform cropping and flipping only, and finally, at evaluation time, perform only center-cropping, following the standard linear probing protocols of prior self-supervision learning works [53, 56]. When training the diffusion model, we also find it conducive to add low Gaussian noise to the encoded source view, similarly to the noise added to the denoised target view.

Meanwhile, for multi-view 3D datasets such as NMR, GSO and ShapeNet, we do not apply data augmentations, and instead, randomly sample one view as the source and another as the target, further supplied by their respective

camera perspectives (Appendix C). In this case, we allow for conditioning on multiple source views, and conduct experiments over $k \in [1 - 9]$ sources. Lastly, to illustrate the ability of SODA to learn useful representations even without relying on data augmentation, we perform ablations on datasets used as is, forgoing augmentations of any kind.

E. Evaluation & Metrics

We explore SODA for multiple types of tasks and purposes: downstream linear-probe classification and disentanglement analysis for assessing the quality of the learned representations, as well as image reconstruction and novel view synthesis for evaluating the model’s generative capabilities. These skills are measured both through qualitative inspection of the latent space, with visualizations that demonstrate its impact on the model’s outputs (including in particular latent interpolations and unsupervised attribute discovery), as well as through an assortment of metrics that quantify each of the capabilities as discussed below.

E.1. Linear Probing

In Section 4.1, we analyze the model’s learned latent representations by measuring their predictive performance on a downstream classification task. Following the common evaluation protocol [53, 56, 58], we first train our model on a collection of images, and then fit a linear classifier that considers the latent encodings produced by the model and use them to predict each respective image’s category or semantic attributes. The classifier is either trained on the frozen representations \mathbf{z} subsequently to the training of the diffusion model, or alternatively, trained with it in tandem by blocking the gradient flow between the two networks – we find that both approaches achieve similar results.

When training the classifier, we refrain from applying weight decay, and adhere to either light augmentation of cropping and flipping for ImageNet or no augmentation in other cases. The latents \mathbf{z} are normalized before being fed to the classifier, concretely, by processing them with an unparameterized batch normalization [53], which only tracks mean and variance statistics and lacks the follow-up affine

Table 7. **Disentanglement Analysis**, comparing SODA to variational approaches on various datasets. Our model achieves improvements of 27.2-58.3% in Disentanglement, 5.0-23.8% in Completeness, and comparable Informativeness. Its reconstructions are often sharper and more accurate. Metrics: *Disen.* stands for Disentanglement, *Comp.* for Completeness, and *Info.* for Informativeness. *PSNR*, *SSIM*, and *LPIPS* respectively express pixel-wise, structural and perceptual/semantic similarity, while *FID* captures sharpness and fidelity.

Method	Disen. \uparrow	Comp. \uparrow	Info. \uparrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow
MPI3D (Toy)							
AnnealedVAE [106]	18.43	21.44	52.47	29.49	0.852	118.18	0.169
FactorVAE [63]	22.41	27.06	60.79	34.60	0.959	31.64	0.075
DIP-VAE (I) [107]	51.12	41.33	72.05	37.34	0.977	22.33	0.057
DIP-VAE (II) [107]	24.88	27.37	64.83	36.68	0.976	25.52	0.062
β -VAE [67]	25.34	28.57	64.65	36.86	0.977	24.61	0.061
β -TCVAE [108]	32.21	40.10	64.03	35.17	0.965	30.81	0.072
SODA w/o layer mod.	83.44	55.65	85.38	50.21	0.998	2.64	0.015
SODA (ours)	87.38	54.79	84.78	50.72	0.999	1.49	0.014
MPI3D (realistic)							
AnnealedVAE [106]	18.76	19.93	51.66	32.54	0.968	55.95	0.201
FactorVAE [63]	30.80	36.63	60.82	34.51	0.980	29.76	0.181
DIP-VAE (I) [107]	45.61	42.37	66.59	36.23	0.986	24.90	0.169
DIP-VAE (II) [107]	29.63	35.85	64.85	36.33	0.986	25.14	0.170
β -VAE [67]	25.14	26.15	61.48	36.54	0.987	24.46	0.168
β -TCVAE [108]	35.81	41.07	66.83	36.71	0.988	24.41	0.167
SODA w/o layer mod.	71.73	51.07	76.53	39.40	0.995	3.42	0.109
SODA (ours)	85.19	56.26	77.78	40.65	0.996	3.84	0.069
MPI3D (real)							
AnnealedVAE [106]	17.62	17.41	53.60	31.97	0.963	40.33	0.083
FactorVAE [63]	33.99	41.48	59.74	34.19	0.978	33.70	0.062
DIP-VAE (I) [107]	53.01	40.35	72.05	36.68	0.988	18.31	0.037
DIP-VAE (II) [107]	33.54	38.72	65.57	36.37	0.987	20.36	0.041
β -VAE [67]	48.56	48.20	68.19	36.64	0.988	19.35	0.038
β -TCVAE [108]	48.60	44.68	67.03	36.47	0.987	19.74	0.040
SODA w/o layer mod.	75.47	51.30	77.32	41.03	0.998	1.57	0.007
SODA (ours)	81.19	50.92	77.69	42.51	0.997	0.61	0.006
MPI3D (complex)							
AnnealedVAE [106]	21.08	21.27	57.73	31.37	0.952	39.71	0.077
FactorVAE [63]	37.50	48.55	68.12	32.93	0.966	36.27	0.067
DIP-VAE (I) [107]	58.49	51.67	76.64	35.14	0.980	23.50	0.042
DIP-VAE (II) [107]	35.00	40.21	72.30	35.03	0.979	27.44	0.045
β -VAE [67]	51.82	50.70	74.47	35.58	0.982	22.78	0.039
β -TCVAE [108]	42.99	46.39	74.26	35.42	0.981	24.90	0.042
SODA w/o layer mod.	86.55	56.60	77.43	44.23	0.998	0.47	0.007
SODA (ours)	89.76	56.98	78.22	43.18	0.997	0.44	0.006
3DShapes							
AnnealedVAE [106]	58.14	66.97	91.23	30.77	0.994	54.83	0.063
FactorVAE [63]	87.62	87.52	98.64	30.37	0.994	48.89	0.058
DIP-VAE (I) [107]	99.75	82.59	99.97	34.23	0.997	31.04	0.031
DIP-VAE (II) [107]	99.22	83.01	99.99	33.87	0.997	32.41	0.033
β -VAE [67]	99.87	83.51	99.96	33.93	0.997	29.52	0.032
β -TCVAE [108]	90.92	74.78	99.81	34.62	0.998	29.03	0.031
SODA w/o layer mod.	92.18	78.21	99.27	51.75	0.9997	0.36	0.0003
SODA (ours)	98.60	84.76	98.88	52.54	0.9999	0.32	0.0002
SmallNORB							
AnnealedVAE [106]	21.50	17.62	60.30	28.40	0.900	164.94	0.271
FactorVAE [63]	37.01	44.58	68.52	27.67	0.885	148.40	0.267
DIP-VAE (I) [107]	34.13	35.51	71.85	28.92	0.907	114.48	0.226
DIP-VAE (II) [107]	37.94	45.76	69.89	29.15	0.909	122.10	0.232
β -VAE [67]	37.79	42.40	71.01	29.25	0.912	120.31	0.229
β -TCVAE [108]	37.43	43.51	70.75	29.20	0.909	120.28	0.230
SODA w/o layer mod.	63.12	45.82	68.36	16.06	0.756	47.90	0.253
SODA (ours)	72.60	51.56	70.19	15.47	0.734	44.81	0.235
CUB							
AnnealedVAE [106]	37.53	11.59	91.71	14.73	0.244	275.35	0.716
FactorVAE [63]	39.14	11.68	92.04	16.26	0.517	244.70	0.637
DIP-VAE (I) [107]	36.44	10.61	92.12	15.05	0.421	233.65	0.642
DIP-VAE (II) [107]	37.43	12.19	92.05	14.97	0.409	227.14	0.647
β -VAE [67]	38.22	11.88	92.06	15.10	0.428	234.52	0.639
β -TCVAE [108]	40.43	13.39	91.95	15.26	0.389	237.96	0.652
SODA w/o layer mod.	62.05	14.35	87.86	12.96	0.423	20.30	0.503
SODA (ours)	65.40	17.43	88.10	13.04	0.344	17.21	0.492

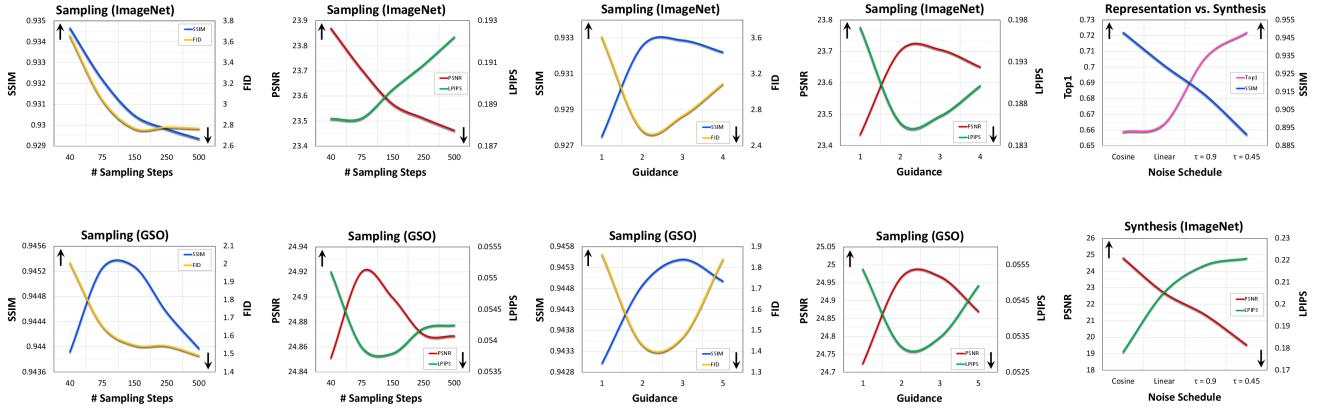


Figure 9. Impact of classifier-free guidance and number of sampling steps and noise schedule on the model’s performance.

transformation. After normalizing the latents, we use 0.1 dropout for regularization, and for ImageNet, apply label smoothing of 0.1. Since the annotated datasets we explore all have discrete labels, we use softmax cross entropy to train the classifier, and report its performance along metrics such as F1 for binary attributes, and top1 accuracy for other ones.

E.2. Image synthesis

In Section 4.2.1, we analyze the capacity of SODA to both reconstruct an input view and generate novel views. Given a target image x , we assess the quality of a synthesized output \hat{x} through multiple complementary metrics that range from visual to semantic similarity:

- (1) **Peak Signal-to-Noise Ratio (PSNR)** \uparrow (*measured in dB*) [81]: is directly derived from the mean MSE between x and \hat{x} , and it thereby measures **pixel-wise** similarity. It may rate a blurry estimation as highly consistent with the target, as long as they match well with each other *on average*.
- (2) **Structural Similarity Index Measure (SSIM)** \uparrow (*ranges between $[-1, 1]$*) [82]: compares images along three perceptual factors: luminance, contrast and **structure**, and is thus better correlated with the Human Visual System (HVS).
- (3) **Learned Perceptual Image Patch Similarity (LPIPS)** \downarrow (*often normalized to be in $[0, 1]$*) [84]: computes the distance between the target and synthesized images in the feature space of a supervised pre-trained network, such as VGG [125], and therefore serves as an indicator for **semantic** similarity.
- (4) **Fréchet inception distance (FID) score** \downarrow (*is ≥ 0*) [83]: quantifies **realism and sharpness** of the generated images by comparing their distribution to that of the target ground-truth images. It concretely achieves

it by considering the mean and variance of each, in a latent feature space, e.g. of the Inception model [126]. When assessing unconditionally-generated images, the FID score further expresses their diversity, but in the case of conditional synthesis, either as reconstructions or with pose conditioning, it mainly reflects their fidelity, sharpness and lack of distortions (also known as R-FID in this context).

For fair comparison, we compute these metrics over all approaches using the same metrics’ implementations, and specifically, casting the images to the range of $[0, 1]$ for PSNR, $[-1, 1]$ for FID and LPIPS, and using a uniform kernel to calculate SSIM scores.

E.3. Disentanglement

In Section 4.3, we examine the latent space of our model and assess its degree of disentanglement and controllability through quantitative and qualitative evaluation methods:

DCI metrics (Disentanglement, Completeness & Informativeness) \uparrow (*at a range of $[0, 100\%]$*) [94]: measures the 1:1 alignment between the latent representation z and the natural (ground-truth) factors of variation c . **Disentanglement** reflects the extant to which each latent variable \hat{z}_j (the j ’th axis of the vector z) corresponds to a unique natural factor c_j . **Completeness** inversely measures the extant to which each natural factor c_j is captured by a single latent variable \hat{z}_j . Finally, **Informativeness** indicates the predictability of the natural factors c from the latent encoding z . These metrics are derived from the normalized importance matrix of a learned classifier and its performance, where the classifier is based on either gradient boosting or Lasso (we use the former). Our implementation of these metrics closely follows Locatello *et al.* [61].

Latent Interpolation: we randomly pick two images x_1, x_2 from each dataset, encode them to obtain z_1, z_2 , and then decode back the latents along a linear segment that con-

Table 8. **Ablations on ShapeNet**, varying the **camera viewpoint encoding schemes**, including **coordinate system**: Polar or Cartesian, rays’ representation **method**: by origin and direction $[o, d]$, or with a weighted sum $o + s_d \cdot d$, and **conditioning** mode: through 2D grid concatenation or vector-based modulation (either of *absolute* or *relative* camera perspective).

Method	Coordinates	PSNR \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow
Concat	Cartesian	27.21	0.945	0.82	0.040
Concat	Polar	27.10	0.940	0.93	0.041
Sphere	Cartesian	27.12	0.946	0.75	0.040
Normalized	Cartesian	27.00	0.941	0.78	0.041
Plane	Cartesian	27.03	0.943	0.78	0.041
Sphere	Polar	27.42	0.947	0.74	0.039

Conditioning	Coordinates	PSNR \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow
Absolute	Polar	23.81	0.870	2.65	0.059
Grid+Relative	Polar	26.81	0.941	0.87	0.040
Grid+Absolute	Polar	27.27	0.946	0.80	0.040
2D Grid	Polar	27.42	0.947	0.74	0.039

Table 9. **Ablations on ImageNet** for feature modulation (*mod.*), evaluated through reconstruction and classification.

Ablation	PSNR \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow	Top1 \uparrow
w/o scale mod.	21.33	0.910	5.33	0.224	70.31
w/o layer mod.	17.61	0.800	9.99	0.377	68.10
sum mod.	13.78	0.541	15.24	0.483	61.46
concat mod.	13.57	0.522	16.68	0.494	59.87
SODA (default)	23.63	0.931	2.77	0.191	72.24

nects between the endpoints: $z_1 + (z_2 - z_1) \cdot t$ for $t \in [0, 1]$, which results in a visualization of the latent traversal.

Principal Component Analysis (PCA) [100, 127]: we encode a sample set of N images (1,000-10,000), and perform PCA decomposition over the obtained latents $z_{i=1}^N$, which yields the latent directions s_j of the greatest variation. We then traverse the latent space along the discovered directions: $z + s_j t$ for $t \in [-\sqrt{\lambda_j}, \sqrt{\lambda_j}]$ where λ_j is the respective eigenvalue and $\sqrt{\lambda_j}$ is the standard deviation along direction s_j . Doing so allows us to visualize the impact of these latent directions on the model’s generated images, and indeed, we find they strongly correlate with semantically-meaningful manipulations.

Thanks to layer modulation (Section 3.2.1), we can further perform PCA over chosen sub-vectors of z that are responsible for guiding decoder’s layers of interest. This enables the discovery of latent directions that control particular levels of granularity, from low-frequency structural aspects to high-frequency factors like texture and color, enhancing the model’s overall controllability.

Classifier-based Attribute Manipulation: For datasets with binary attributes annotations, such as CelebA and CUB, we can produce similar visualizations to the ones described above by examining the weight matrix’s rows of the linear probes we train for the downstream classification experiments (Section 4.1). Indeed, these probes are trained to capture the latent directions that correspond to the presence or absence of the semantic attribute annotations that accompany the datasets we study. The key difference between the

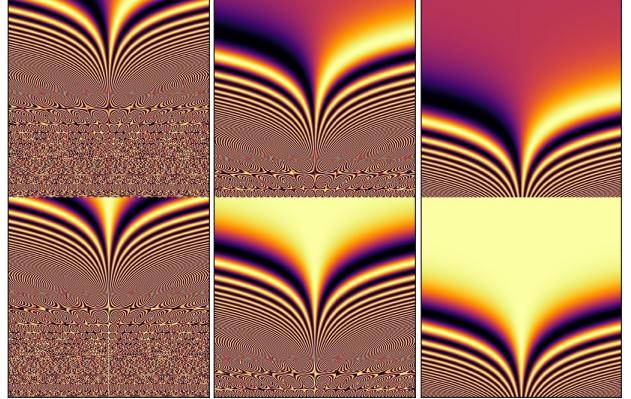


Figure 10. **Positional Encoding Scales.** We visualize the positional encodings in the range of $[-1, 1]$, with each embedding visualized vertically within each plot. When the positional encoding scale is set to a too low value (**right column**), the encodings are less distinctive and their capacity is under-utilized (as many of the features (rows) get the same value for all positions). Meanwhile, a too high value (**left column**) damages the positional encodings empirical performance. We thus make sure to select a medium scale (**middle column**) for optimal performance.

PCA-based approach and this technique is that the former is unsupervised while the latter is not.

F. Baselines

For each of the tasks we explore, we compare our model to the respective leading approaches, as well as to additional ablated baselines that we design. Here, we list and review all the baseline methods we compare to.

F.1. Task-Agnostic Baselines

First, we implement multiple baselines and ablated models within our diffusion codebase, and report their performance across the range of tasks:

- (1) **EncDec**: a vanilla encoder-decoder $\hat{x} = \mathcal{D}(\mathcal{E}(x'))$, sharing the same encoder and decoder architectures as SODA (for \mathcal{D} , the decoding module of the UNet, but being trained to generate the target image x from scratch, with no denoising). We train this model both with and without $(x = x')$ data augmentation.
- (2) **Uncond**: an unconditional diffusion model \mathcal{D} (see also DDAE [19]). To obtain an encoding z for an image x , we compute $\mathcal{D}(\tilde{x})$ over a lightly-noised version of x and pool the activations from the middle layer of the UNet denoiser (The layer index and noise degree are hyperparameters chosen for highest performance).
- (3) **Palette**: a diffusion model that instead of having a dedicated encoder \mathcal{E} , concatenates the source image x' to the denoised image x_t directly, and inputs both of them to a UNet denoiser $\mathcal{D}(x_t, x')$ (also known as

Table 10. **Datasets Configuration & Statistics.** (*) LSUN sizes per partition: Bedrooms (3.03M), Church Outdoors (126K), Bird (2.31M), Car (5.52M), Cat (1.66M), Dog (5.05M), Horse (2.00M).

Dataset	Size	Resolution (Raw)	# Categories / Attributes	Augmentation / Views	Source View Noise Scale	ResNet Size	Learning Rate	Batch Size	Guidance
Imagenet1K [117]	1.28M	Varied	1000	RandAugment	0.10	50×2	4×10^{-4}	4096	2
CelebA-HQ [105]	30K	1000	2×40	Gaussian Noise	0.22	50×2	4×10^{-4}	4096	2
LSUN [118]	(*)	Varied	-	Gaussian Noise	0.22	50×2	4×10^{-4}	4096	2
AFHQ [119]	15K	512	3	Gaussian Noise	0.22	50×2	4×10^{-4}	4096	2
NMR [88]	1.05M	64	13	24×43.8K	0.00	50	2×10^{-4}	2048	2
ShapeNet [87]	6.28M	256	55	120×52K	0.00	50	2×10^{-4}	2048	2
GSO [86]	120K	256	17	120×1K	0.00	50	2×10^{-4}	2048	2.5
SmallNORB [101]	24.3K	96	18,5,9,6	Gaussian Noise	0.05	18	1×10^{-4}	1024	3
3D-Shapes [104]	480K	64	4,10,10,10,15,8	Gaussian Noise	0.05	18	1×10^{-4}	1024	3
MPI3D (Toy) [102]	1.03M	64	6,6,2,3,3,40,40	Gaussian Noise	0.05	18	1×10^{-4}	1024	3
MPI3D (Realistic) [102]	1.03M	64	6,6,2,3,3,40,40	Gaussian Noise	0.05	18	1×10^{-4}	1024	3
MPI3D (Real) [102]	1.03M	64	6,6,2,3,3,40,40	Gaussian Noise	0.05	18	1×10^{-4}	1024	3
MPI3D (Complex) [102]	461K	64	4,4,2,3,3,40,40	Gaussian Noise	0.05	18	1×10^{-4}	1024	3
Caltech-UCSD Birds (CUB) [104]	11.8K	Varied	2×312	Gaussian Noise	0.05	18	1×10^{-4}	1024	3
Oxford Flowers [120]	8.2K	Varied	102	Gaussian Noise	0.05	18	1×10^{-4}	1024	3

Image-to-Image diffusion model [30]).

- (4) **unCLIP (Dall-E2)** [8]: a diffusion model that relies on a frozen pretrained CLIP [128] as the encoder \mathcal{E} . We emphasize that we do not refer here to the already trained Dall-E2 model, but rather to its architecture, and so we train its denoiser (in a comparable size to our model) from scratch along with the frozen pretrained CLIP encoder, for each dataset of interest.
- (5) **w/o bottleneck**: an ablation of SODA with no bottleneck, which rather encodes the input image x' into a 2D feature grid $z^{w \times h}$, with no global pooling, and conditions the denoising on it through cross-attention (similarly to text-to-image diffusion models [7]).
- (6) **w/o modulation**: an ablation of SODA that broadcasts and concatenates the latent z to linearly-mapped RGB channels of the denoised image x_t , instead of applying modulation through adaptive group normalization (also called a spatial broadcast decoder [129]).

F.2. Linear-Probe Classification

For downstream classification, we compare our model to a diverse array of leading self-supervised learning approaches: **generative** methods like MAE [53], BEIT [54] and iGPT [55] split each image into a grid of tokens or patches, mask some patches and predict them back from the unmasked ones, oftentimes using a transformer backbone.

Meanwhile, **discriminative** approaches leverage contrastive learning (as in SimCLR [58]), clustering techniques (as in SwAV [57]), and distillation (as in DINO [56] and BYOL [72]) to derive visual representations. At the core of these methods is a strong reliance on rich data augmentations, which are essentially the driving force that allows the to perform unsupervised clustering.

Consequently, contrastive learning approaches operate well at tasks that involve identification of an image’s category, as is the case for ImageNet, but may struggle to capture finer traits that are altered by the augmentations. The

semantic properties they may or may not encode into the learned representations heavily depend on the particularities of the data augmentation scheme they employ, since they are basically encouraged to form a latent space that is invariant to the augmentation applied, instead casting different augmentations into similar representations.

Contrary to these two kinds of approaches, both of which are unsuitable for high-quality image generation, SODA stands out being able to both encode input images into meaningful latents, and also **synthesize** back crisp output images, conditionally and unconditionally. It learns **compact** and disentangled representations, which contrast with the large, potentially discrete, 2D grids learned by alternative approaches, and as demonstrated in Section 4.1, is **robust to the chosen data augmentation** scheme, operating well even in its absence.

Our comparison to the approaches discussed in this subsection relies on the performance reports in their respective publications over the ImageNet1K dataset, with the exception of the crop+flip accuracy for SwAV and DINO’s for which we retrain the models.

F.3. Image Reconstruction

We examine the performance of varied models for the task of image reconstruction: Dall-E [74] and VQGAN [75] employ a **discrete variational** auto-encoder [130], which casts input images into 2D token grids, based on a trainable codebook. These approaches then couple the auto-encoder with a prior-distribution model, to enable unconditional image synthesis. However, for our purposes (image reconstruction), we consider the auto-encoder module only.

The **adversarial** StyleGAN model [49] can also be used for image reconstruction, by applying optimization-based inversion techniques to infer back latents from images. Given an image x , they leverage gradient descent to reverse engineer the latent z that gives rise to an output \hat{x} that is as close as possible to the image x while still staying

Table 11. **Ablations on CelebA**, evaluated through classification, reconstruction, and disentanglement. The no-bottleneck ablation (*) has skip connections between the model’s input and output, making the reconstruction task trivial, but simultaneously damaging the learned representations’ quality. *Disen.* stands for Disentanglement, *Comp.* for Completeness, and *Info.* for Informativeness.

Ablation	F1 \uparrow	Disen. \uparrow	Comp. \uparrow	Info. \uparrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow
w/o bottleneck	58.89	36.61	26.12	83.15	27.29*	0.985*	6.61*	0.099*
w/o modulation	60.38	64.51	40.85	84.11	14.53	0.734	20.58	0.332
w/o layer modulation	70.90	67.25	42.53	87.06	16.79	0.821	10.35	0.288
w/o layer mask	70.36	68.41	43.41	87.45	16.98	0.820	11.39	0.285
w/o scale modulation	69.65	67.41	44.84	87.63	16.53	0.816	10.49	0.290
sum modulation	68.31	61.12	40.53	85.88	16.71	0.816	17.95	0.290
concat modulation	67.58	61.34	40.67	86.02	16.67	0.816	17.35	0.289
SODA (ResNet50, default)	71.63	73.90	48.81	87.64	18.24	0.842	10.09	0.275

on the model’s learned manifold. While these techniques tend to produce samples that share semantic properties with the source images, they oftentimes fail to reconstruct them faithfully. Finally, we compare our model to the **diffusion-based DiffAE** [48], which, in contrast to our study, focuses on auto-encoding only, and can be viewed as a predecessor of our approach, as discussed in Section 2.

We assess the reconstruction capabilities of the approaches described in this subsection by evaluating a sample set of images produced by their associated public pre-trained checkpointed models.

F.4. Novel View Synthesis

For novel view synthesis of 3D objects, we compare SODA to a collection of **geometry-free and -aware** approaches designed for few-shot settings: **PixelNeRF** [60] learns to translate a small number of source views into a neural radiance field, and then use volumetric rendering techniques to generate new ones. **NeRF-VAE** [59] extends this idea by leveraging amortized variational inference to learn probabilistic neural scene representations. In contrast to these specialized methods, designed specifically for 3D environments, SODA proves considerably more versatile, successfully addressing a broader spectrum of tasks and datasets.

As an alternative to differentiable rendering, geometry-free approaches often use attention mechanisms to directly transform source views into targets: **Scene Representation Transformer (SRT)** [80] parametrizes scenes with the computationally lighter and faster Light-Field formulation [131], and synthesize output views from new perspectives by directly attending to the input views’ encodings. The diffusion-based 3DiM [89] goes further and makes extensive use of cross-attention throughout all of its network’s layers so to directly map sources to targets. In contrast to these approaches, we intentionally introduce a bottleneck into our model that induces a meaningful and compact latent space. This, in turn, offers much tighter control over the model’s generative process, opening the door for both semantic manipulation of given scenes, as well as unconditional synthesis of new ones – two new capabilities that are out of these prior works’ reach.

We evaluate the methods described in this subsection ei-

ther using the authors’ official implementations (for NeRF-VAE), or with our own re-implementations (for PixelNeRF and SRT), matching the originally reported performance.

F.5. Disentanglement

In terms of disentanglement, we analyze SODA over a suite of semantically-annotated datasets, and compare it with a series of **variational** approaches [66], which are traditionally known for encouraging the formation of disentangled representations: **β -VAE** [67], re-weights the KL regularization term to constrain the latents’ capacity; **AnnealedVAE** [106] slowly relaxes the encoder-decoder bottleneck so to foster gradual learning; **FactorVAE** [63] and **β -TCVAE** [108] encourage factorization of the latent distribution by reducing the correlations among the axes; **DIP-VAE** [107] (variants I and II) penalizes the mismatch between the prior and the posterior, so to similarly encourage factorization within the latter. We evaluate these methods using the official *disentanglement-lib* TensorFlow repository [61], while modifying the backbone encoder and decoder architectures to match the ones used in SODA, for better comparability.

G. Ablation Studies

To gain better insight into the relative contributions our design decisions make, we conduct thorough ablation and variation studies for each of the model’s components, inspecting the (1) **feature modulation** used to propagate information between the encoder and the denoiser, (2) **data augmentation** strategies for the source and target views, encoding and conditioning schemes of (3) **positional information** for our 3D multi-view experiments, (4) **sampling configurations** of the denoising process and its classifier-free guidance, and finally, (5) the encoder and denoiser’s respective **sizes, dimensions and learning rates**.

This study joins ablations presented through the main paper (Sections 4.1 and 4.3.2) that attest to the strengths and benefits of the model’s core aspects and key innovations, like bottleneck compactness (Section 3.2), layer modulation (Section 3.2.1), redesigned noise schedule (Section 3.4), and incorporation of novel view synthesis as a self-supervised training objective (Section 3.3).

Table 12. **Hyperparameters** of our model, including the encoder, denoiser, linear probe, and view aggregation transformer (for 3D experiments), as well as optimization, sampling and augmentation settings. (*) Depends on the dataset. (†) Applied for ImageNet only. (*) We use a base value of 64 for ImageNet pre-training for downstream classification, and 128 otherwise.

Hyperparameter	Value	Hyperparameter	Value
Optimization		Denoiser	
Learning Rate*	$(1\text{-}4) \times 10^{-4}$	Attention Heads Number	4
Batch Size*	1024-4096	Hidden Layer Multiplier	4
Learing Rate Schedule†	Cosine Decay	Architecture	UNet
Learing Rate After Decay†	$0.25 \times LR$	Resolution	128
Learning Rate Decay Steps†	1.2×10^5	Base Channels*	64-128
Weight Decay	0.05	Channels multipliers	1,1,2,3,4
EMA Decay Rate	0.9999	Residual blocks per resolution	2
Warmup Steps	5000	Selt-Attention resolution	8,16,32
Gradient Clipping Norm	0.5	Attention Head Dimension	64
Optimizer	Adam	Normalization Type	GroupNorm [51]
β_1	0.9	Dropout Rate	0.1
β_2	0.95	Sampling	
Model		Classifier-Free Guidance*	2-3
Latent Dimension*	128-2048	Diffusion Training Steps	1000
Bottleneck Dropout	0.1	Sampling Strided Steps*	75-250
Classifier-Free Guidance Masking Rate	0.12	Linear Probe	
Layer Masking Rate	0.15	Weight Initialization Scale	0.02
Positional Encoding Dimension	512	Bias Initialization Scale†	-10.0
Positional Encoding Scale (Figure 10)	0.0001	Dropout Rate	0.1
Encoder		Augmentation Rate (RandomResizedCrop)	0.65
Architecture	ResNet	Label Smoothing†	0.1
Size*	18, 50, 50×2	Data Augmentation	
Version	v2	Augmentation Rate (Cropping+Flipping)	0.95
Resolution	256 (224†)	Distortion Rate (RandAugment [79])	0.65
LR multiplier	2	Distortion Layers Number	2
DropPath [112]	0.1	Distortion Magnitude	9
View Aggregation Transformer		Distortion Magnitude STD	0.5
Depth	2	Source View Noise Scale*	0-0.22

G.1. Feature Modulation

We explore multiple modulation variants and examine how they fare in terms of generative skills and downstream performance (Tables 7, 9 and 11). As the results suggest, **modulation-based conditioning** proves considerably more effective than alternative mechanism such as input concatenation $[x_t, x']$ (Palette [30]) or spatial broadcasting (*w/o modulation*) [129], with respective deltas of 15.5% and 12.3% at classification over ImageNet (top1) and CelebA (F1), and 0.32 (out of 1.0) mean SSIM improvement at novel view synthesis. **Layer modulation** proves beneficial too, enhancing disentanglement scores, with up to 13.9% improvement, and generative capabilities, with 0.13 increase in SSIM and halving of LPIPS for ImageNet reconstructions.

We further assess ways to integrate the guidance of the timestep t and the latent z , and as an alternative to our **two-stage guidance** approach, where the denoiser’s activations h are modulated first by t and subsequently by z , we map them instead to a single pair (w_s, w_b) either through summation or concatenation (*sum/concat mod.*) which is then used to modulate the activations: AdaGN(h, t, z) = w_s GroupNorm(h) + w_b . However, our two-step strategy proves stronger than this variant. Likewise, **scaling the features multiplicatively** with z_s , as opposed to adding a bias term z_b only, leads to small improvements across different datasets.

G.2. Data Augmentation

We examine the impact of data augmentations on the model’s performance, and analyze variations of the augmentation method itself as well as the inputs it is applied to (Figure 6). At training, our model receives two inputs: a clean view x' processed by the encoder \mathcal{E} , and a noisy view x_t denoised by the decoder $\mathcal{D}(x_t|x')$, aiming to recover $x = x_0$. With the exception of native multi-view datasets (e.g. ShapeNet), we create the source and target views x' and x_t by applying random data augmentations at each training step on the original image x (from the dataset).

We test the impact of applying augmentations either just to the source view, just to the target view, to both, or to none of them. We observe that **augmenting the source** is more critical than the target in terms of its influence on downstream classification performance, and that the model still achieves 55.1 when the source and the target views remain equal. Moreover, we find it valuable to add **low Gaussian noise** to the source views read by the encoder, yielding 1.3% improvement in ImageNet classification accuracy and improving LPIPS scores relatively by 33%.

G.3. Pose Conditioning

We compare different encoding schemes of the camera perspectives for the 3D novel view synthesis task (Table 8). Given a camera pose p , we can use a closed-form calculation to derive a 2D grid of rays $r = (o, d)$ of dimension

Table 13. **Ablations on ShapeNet**, varying the **classifier-free guidance** settings: either masking the latent z that encodes the source image view, masking the pose information r of the source and target views, or independently masking both.

Masking	PSNR \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow
ShapeNet				
Latent	27.42	0.947	0.74	0.039
Pose	27.16	0.940	0.96	0.041
Latnet + Pose	27.11	0.938	0.95	0.041
GSO				
Latent	24.12	0.937	2.22	0.065
Pose	24.25	0.939	2.48	0.062
Latnet + Pose	24.97	0.945	1.51	0.054

$H \times W \times 6$ with origins \mathbf{o} and directions \mathbf{d} . We can then represent each ray through concatenation: $[\mathbf{o}, \mathbf{d}]$ (*concat*), as commonly done in prior works [60, 80], or instead, express them with a parametric sum: $\mathbf{o} + s_d \cdot \mathbf{d}$, where s_d is a scaling factor that can be chosen in different ways: either normalizing \mathbf{d} to a length of 1, casting it onto the image plane, or, as we propose, on a sphere that centers at the object, i.e. the origin (*Normalized, Plane and Sphere*). We can further describe the rays either using Polar or Cartesian coordinates, embedded with sinusoidal positional encoding [68] as explained above (Appendix C). We compare these alternatives, and find that **casting the rays on a sphere** performs most effectively, and that for this case, **Polar coordinates** outperform the Cartesian ones.

We further experiment with representing the camera pose as a single vector \mathbf{p} that captures its position and direction in Polar coordinates, either considering $\mathbf{p}_t - \mathbf{p}_s$, the relative camera transformation from the source to the target, or concatenating the two absolute viewpoints $[\mathbf{p}_s, \mathbf{p}_t]$. We then encode the information with sinusoidal positional encoding [68]), and use the resulting vector to guide the denoiser’s operation through feature modulation, similarly to the latent z . However, the ablations show that integrating the camera perspective by concatenating a 2D grid of rays surpasses both the modulation-based pose conditioning as well as a hybrid alternative that simultaneously uses both techniques.

Finally, we experiment with different masking techniques as part of the classifier-free guidance (Section 3.3), either randomly masking the latent representation z that encodes the source image view, masking the pose information (namely, the rays 2D grid r), or independently masking both. We interestingly note that the ideal masking vary for different datasets: while masking of both the latent and pose improves performance for the real-world Google Scanned Objects, it reduces the performance for ShapeNet. Qualitatively, masking both the pose r and the latent z enhances the model’s generative flexibility, allowing it to synthesize either novel objects at requested camera perspectives, or arbitrary novel views even at the absence of source or target’s pose information (supplementary figures will be added very soon).

G.4. Sampling Configuration

We vary the guidance strength g and timesteps striding l (i.e. number of timesteps used at sampling) and analyze their impact on the generated images’ quality along different metrics (Figure 9). The model is robust to variation in both settings, with optimal values commonly achieved at $g = 2$ and $l = 150$ (considering different metrics and datasets). Classifier-free guidance consistently yields higher-quality images than unguided sampling, while too strong guidance (like ≥ 5) results in a slight reduction in scores and potential visual artifacts.

As per the number of sampling steps, while PSNR and LPIPS scores tend to remain constant, we interestingly observe an inverse correlation between the process length’s influence on FID vs. SSIM, the former reflecting sharpness and fidelity while the latter capturing similarity to the target: Sampling images over more steps tends to improve their realism, but may simultaneously induce subtle variations, as the samples begin to slightly move away from the mean estimated target. As aforementioned, we find that $l = 150$ offers a favorable balance between these two qualities.