Return of Unconditional Generation: A Self-supervised Representation Generation Method

Tianhong Li **Dina Katabi** **Kaiming He**
MIT CSAIL

Abstract

Unconditional generation—the problem of modeling data distribution without relying on human-annotated labels—is a long-standing and fundamental challenge in generative models, creating a potential of learning from large-scale unlabeled data. In the literature, the generation quality of an unconditional method has been much worse than that of its conditional counterpart. This gap can be attributed to the lack of semantic information provided by labels. In this work, we show that one can close this gap by generating semantic representations in the representation space produced by a self-supervised encoder. These representations can be used to condition the image generator. This framework, called Representation-Conditioned Generation (RCG), provides an effective solution to the unconditional generation problem without using labels. Through comprehensive experiments, we observe that RCG significantly improves unconditional generation quality: *e.g.*, it achieves a new state-of-the-art FID of 2.15 on ImageNet 256×256, largely reducing the previous best of 5.91 by a relative 64%. Our unconditional results are situated in the same tier as the leading class-conditional ones. We hope these encouraging observations will attract the community's attention to the fundamental problem of unconditional generation. Code is available at https://github.com/LTH14/rcg.

1 Introduction

Generative models have been long developed as *unsupervised* learning methods in the history, *e.g.*, in the seminal works including GAN [27], VAE [39], and diffusion models [57]. These fundamental methods focus on learning the probabilistic distributions of data, without relying on the availability of human annotations. This problem, often categorized as *unconditional generation* in today's community, is in pursuit of utilizing the vast abundance of unannotated data to learn complex distributions.

However, unconditional image generation has been largely stagnant in comparison with its conditional counterpart. Recent research [18, 54, 10, 11, 24, 50] has demonstrated compelling image generation quality when conditioned on class labels or text descriptions provided by humans, but its quality degrades *significantly* without these conditions. Closing the gap between unconditional and conditional generation is a challenging and scientifically valuable problem: it is a necessary step towards unleashing the power of large-scale unannotated data, which is a common goal in today's deep learning community.

We hypothesize that such a performance gap is because human-annotated conditioning introduces rich semantic information to simplify the generative process. In this work, we largely close this gap by taking inspiration from a closely related area—unsupervised/self-supervised learning.[1] We

[1] In this paper, the term of "unsupervised learning" implies "not using human supervision". Thus, we view self-supervised learning as a form of unsupervised learning. The distinction between these two terminologies is beyond the scope of this work.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3985

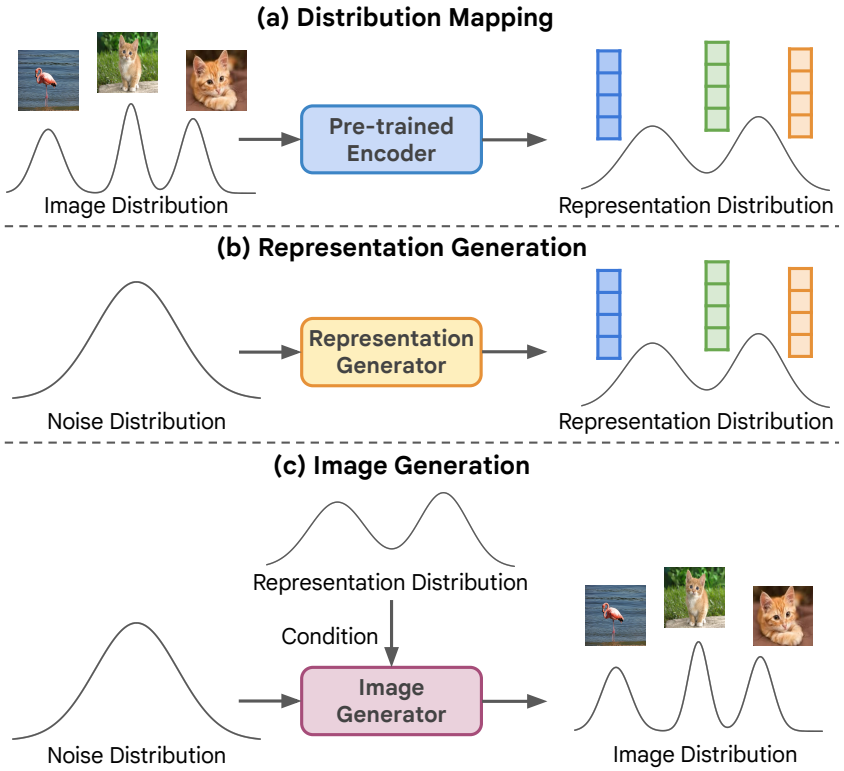

Figure 1: **The Representation-Conditioned Generation (RCG) framework** for unconditional generation. RCG consists of three parts: (a) it uses a pre-trained self-supervised encoder to map the image distribution to a representation distribution; (b) it learns a representation generator that samples from a noise distribution and generates a representation subject to the representation distribution; (c) it learns an image generator (*e.g.*, which can be ADM [18], DiT [50], or MAGE [41]) that maps a noise distribution to the image distribution conditioned on the representation distribution.

observe that the *representations* produced by a strong self-supervised encoder (*e.g.*, [30, 12, 8, 14]) can also capture a lot of semantic attributes without human supervision, as reflected by their transfer learning performance in the literature. These self-supervised representations can serve as a form of conditioning without violating the unsupervised nature of unconditional generation, creating an opportunity to get rid of the heavy reliance on human-annotated labels.

Based on this observation, we propose to first unconditionally generate a self-supervised representation and then condition on this representation to generate the images. As a preprocessing step (Figure 1a), we use a pre-trained self-supervised encoder (*e.g.*, MoCo [14]) to map the image distribution into the corresponding representation distribution. In this representation space, we train a representation generator without any conditioning (Figure 1b). As this space is low-dimensional and compact [65], learning the representation distribution is favorably feasible with unconditional generation. In practice, we implement it as a very lightweight diffusion model. Given this representation space, we train a second generator that is conditioned on these representations and produces images (Figure 1c). This image generator can conceptually be any image generation model. The overall framework, called *Representation-Conditioned Generation* (RCG), provides a new paradigm for unconditional generation.[2]

RCG is conceptually simple, flexible, yet highly effective for unconditional generation. RCG greatly improves unconditional generation quality regardless of the specific choice of the image generator (Figure 2), reducing FID by 71%, 76%, 82%, and 51% for LDM-8, ADM, DiT-XL/2, and MAGE-L, respectively. This indicates that RCG largely reduces the reliance of current generative models on

[2]The term "unconditional generation" implies "not conditioned on human labels". As such, RCG is an unconditional generation solution.

https://doi.org/10.52202/079017-3985

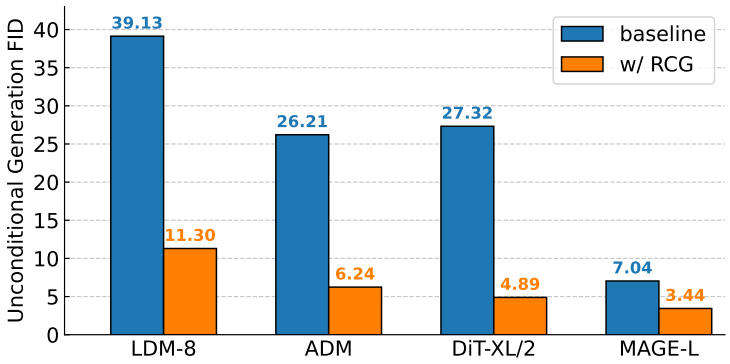

Figure 2: **Unconditional Image Generation can be largely improved by our RCG framework**. Regardless of the specific form of the image generator (LDM [54], ADM [18], DiT [50], or MAGE [41]), RCG massively improves the unconditional generation quality. Generation quality is measured by FID on ImageNet with a 256×256 resolution. All comparisons between models without and with RCG are conducted under controlled conditions to ensure fairness. The technical details and more metrics are in Section 4.1.

manual labels. On the challenging ImageNet 256×256 benchmark, RCG achieves an unprecedented 2.15 FID for unconditional generation. This performance not only largely outperforms previous unconditional methods, but more surprisingly, can catch up with the strong leading methods that are *conditional* on class labels. We hope our method and encouraging results will rekindle the community's interest in the fundamental problem of unconditional generation.

2 Related Work

Generative Models. Generative models aim at accurately modeling data distribution to generate new data point that resembles the original data. One stream of generative models is built on top of generative adversarial networks (GANs) [27, 69, 37, 70, 7]. Another stream is based on a two-stage scheme [63, 53, 10, 67, 40, 41, 11]: first tokenize the image into a latent space and then apply maximum likelihood estimation and sampling in the token space. Diffusion models [33, 59, 18, 54, 52] have also achieved superior results on image synthesis.

The design of a generative model is mostly orthogonal to how it is conditioned. However, literature has shown that unconditional generation often significantly lags behind conditional generation under the same design[18, 41, 10], especially on complex data distributions.

Unconditional Generation. Unconditional generation is the fundamental problem in the realm of generative models. It aims to model the data distribution without relying on human annotations, highlighted by seminal works of GAN [27], VAE [39], and diffusion models [57]. It has demonstrated impressive performance in modeling simple image distributions such as scenes or human faces [23, 10, 18, 54], and has also been successful in applications beyond images where human annotation is challenging or impossible, such as novel molecular design [66, 28, 26], medical image synthesis [71, 16, 47], and audio generation [48, 42, 25]. However, recent research in this domain has been limited, largely due to the notable gap between conditional and unconditional generation capabilities of recent generative models on complex data distributions [46, 18, 19, 41, 3, 61].

Prior efforts to narrow this gap mainly group images into clusters in the representation space and use the cluster indices as underlying class labels to provide conditioning [46, 43, 3, 35]. However, these methods assume that the dataset is clusterable, and the optimal number of clusters is close to the number of classes. Additionally, these methods fall short of generating diverse representations—they are unable to produce different representations within the same cluster or underlying class.

Representations for Image Generation. Prior works have explored the effectiveness of exploiting representations for image generation. DALL-E 2 [52], a *text-conditional* image generation model, first converts text prompts into image embeddings, and then uses these embeddings as the conditions

https://doi.org/10.52202/079017-3985

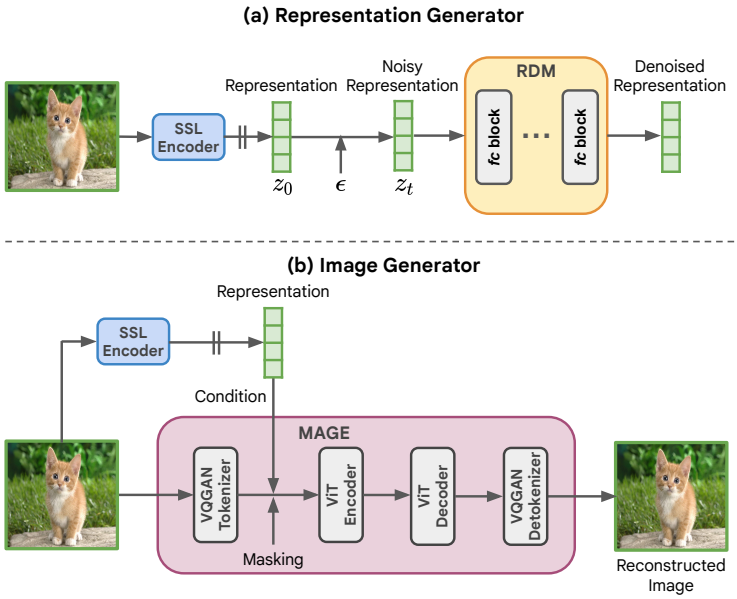

Figure 3: **RCG's training framework.** The pre-trained self-supervised image encoder extracts representations from images and is fixed during training. To train the representation generator, we add standard Gaussian noise to the representations and ask the network to denoise them. To train the MAGE image generator, we add random masking to the tokenized image and ask the network to reconstruct the missing tokens conditioned on the representation extracted from the same image.

to generate images. In contrast, RCG for the first time demonstrates the possibility of directly generating image representations *from scratch*, a necessary step to enable conditioning on self-supervised representations in unconditional image generation. Another work, DiffAE [51], trains an image encoder in an end-to-end manner with a diffusion model as decoder, aiming to learn a meaningful and decodable image representation. However, its semantic representation ability is still limited (e.g., compared to self-supervised models like MoCo [14], DINO [8]), which largely hinders its performance in unconditional generation. Another relevant line of work is retrieval-augmented generative models [5, 4, 9], where images are generated based on representations extracted from existing images. Such semi-parametric methods heavily rely on ground-truth images to provide representations during generation, a requirement that is impractical in many generative applications.

3 Method

Directly modeling a complex high-dimensional image distribution is a challenging task. RCG decomposes it into two much simpler sub-tasks: first modeling the distribution of a compact low-dimensional representation, and then modeling the image distribution conditioned on this representation distribution. Figure 1 illustrates the idea. Next, we describe RCG and its extensions in detail.

3.1 The RCG Framework

RCG comprises three key components: a pre-trained self-supervised image encoder, a representation generator, and an image generator. Each component's design is elaborated below:

Distribution Mapping. RCG employs an off-the-shelf image encoder to convert the image distribution to a representation distribution. This image encoder has been pre-trained using self-supervised contrastive learning methods, such as MoCo v3 [14], on ImageNet. This approach regularizes the representations on a hyper-sphere while achieving state-of-the-art performance in representation learning. The resulting distribution is characterized by two essential properties: it is simple enough to be modeled effectively by an *unconditional* representation generator, and it is rich in high-level semantic content, which is crucial for guiding image generation. These attributes are vital for the effectiveness of the following two components.

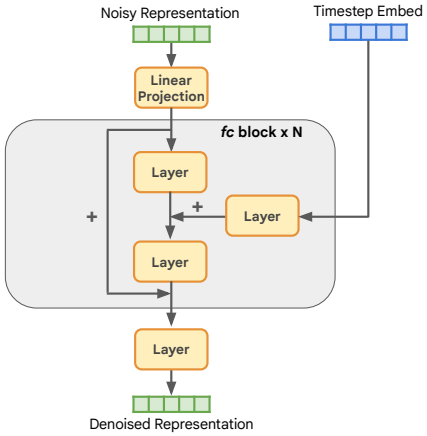

Figure 4: **Representation generator's backbone architecture.** Each "Layer" consists of a LayerNorm layer [1], a SiLU layer [22], and a linear layer. The backbone consists of an input layer that projects the representation to hidden dimension C, followed by N fully-connected (fc) blocks, and an output layer that projects the hidden latent back to the original representation dimension. The diffusion timestep is embedded and added to every fc block.

Representation Generation. In this stage, we want to generate abstract, unstructured representations without conditions. To address this issue, we develop a diffusion model for unconditional representation generation, which we call a representation diffusion model (RDM). RDM employs a fully-connected network with multiple fully-connected residual blocks as its backbone (Figure 4). Each block consists of an input layer, a timestep embedding projection layer, and an output layer, where each layer consists of a LayerNorm [1], a SiLU [22], and a linear layer. Such an architecture is simply controlled by two parameters: the number of blocks, N, and the hidden dimension, C.

RDM follows DDIM [58] for training and inference. As shown in Figure 3a, during training, image representation z_0 is mixed with standard Gaussian noise variable ϵ: $z_t = \sqrt{\alpha_t} z_0 + \sqrt{1-\alpha_t}\epsilon$. The RDM backbone is then trained to denoise z_t back to z_0. During inference, RDM generates representations from Gaussian noise following the DDIM sampling process [58]. Since RDM operates on highly compacted representations, it brings marginal computation overheads for both training and generation (Appendix B.1), while providing substantial semantic information for the image generator, introduced next.

Image Generation. The image generator in RCG crafts images conditioned on self-supervised representations. Conceptually, such an image generator can be any modern conditional image generative model by substituting its original conditioning (e.g., class label or text) with representations. In Figure 3b, we take MAGE [41], a parallel decoding generative model as an example. The image generator is trained to reconstruct the original image from a masked version of the image, conditioned on the representation of the same image. During inference, the image generator generates images from a fully masked image, conditioned on the representation generated by the representation generator.

We experiment with four representative generative models: ADM [18], LDM [54], and DiT [50] are diffusion-based frameworks, and MAGE [41] is a parallel decoding framework. Our experiments show that all four generative models achieve much better performance when conditioned on high-level self-supervised representations (Table 1).

3.2 Extensions

Our RCG framework can easily be extended to support guidance even in the absence of labels, and to support conditional generation when desired. We introduce these extensions as follows.

Enabling Guidance in Unconditional Generation. In class-conditional generation, the presence of labels allows not only for class conditioning but can also provides additional "guidance" in the generative process. This mechanism is often implemented through classifier-free guidance in class-conditional generation methods [32, 54, 11, 50]. In RCG, the representation-conditioning behavior enables us to also benefit from such guidance, even in the absence of labels.

Specifically, RCG follows [32, 11] to incorporate guidance into its MAGE generator. During training, the MAGE generator is trained with a 10% probability of not being conditioned on image representations, analogous to [32] which has a 10% probability of not being conditioned on labels. For each inference step, the MAGE generator produces a representation-conditioned logit, l_c, and

https://doi.org/10.52202/079017-3985

Table 1: **RCG significantly improves the unconditional generation performance of current generative models**, evaluated on ImageNet 256×256. All numbers are reported under the unconditional generation setting.

Unconditional generation		FID↓	IS↑
LDM-8 [54]	baseline	39.13	22.8
	w/ RCG	**11.30** (−27.83)	**101.9** (+79.1)
ADM [18]	baseline	26.21	39.7
	w/ RCG	**6.24** (−19.97)	**136.9** (+97.2)
DiT-XL/2 [50]	baseline	27.32	35.9
	w/ RCG	**4.89** (−22.43)	**143.2** (+107.3)
MAGE-B [41]	baseline	8.67	94.8
	w/ RCG	**3.98** (−4.69)	**177.8** (+83.0)
MAGE-L [41]	baseline	7.04	123.5
	w/ RCG	**3.44** (−3.60)	**186.9** (+63.4)

an unconditional logit, l_u, for each masked token. The final logits, l_g, are calculated by adjusting l_c away from l_u by the guidance scale, τ: $l_g = l_c + \tau(l_c - l_u)$. The MAGE generator then uses l_g to sample the remaining masked tokens. Additional implementation details of RCG's guidance are provided in Appendix A.

Simple Extension to Class-conditional Generation. RCG seamlessly enables conditional image generation by training a task-specific conditional RDM. Specifically, a class embedding is integrated into each fully-connected block of the RDM, in addition to the timestep embedding. This enables the generation of class-specific representations. The image generator then crafts the image conditioned on the generated representation. As shown in Table 3 and Appendix C, this simple modification allows users to specify the class of the generated image while keeping RCG's superior generative performance, all without the need to retrain the image generator.

4 Experiments

We evaluate RCG on the ImageNet 256×256 dataset [17], which is a common benchmark for image generation and is especially challenging for unconditional generation. Unless otherwise specified, we do not use ImageNet labels in any of the experiments. We generate 50K images and report the Frechet Inception Distance (FID) [31] and Inception Score (IS) [55] as the standard metrics for assessing the fidelity and diversity of the generated images. Evaluations of precision and recall are included in Appendix B.1. Unless otherwise specified, we follow the evaluation suite provided by ADM [18]. **All ablations and results on other datasets are included in Appendix B.1.**

4.1 Observations

We extensively evaluate the performance of RCG with various image generators and compare it to the results of state-of-the-art unconditional and conditional image generation methods. Several intriguing properties are observed.

RCG significantly improves the unconditional generation performance of current generative models. In Table 1, we evaluate the proposed RCG framework using different image generators. The results demonstrate that conditioning on generated representations substantially improves the performance of all image generators in unconditional generation. Specifically, it reduces the FID for unconditional LDM-8, ADM, DiT-XL/2, MAGE-B, and MAGE-L by 71%, 76%, 82%, 54%, and 51%, respectively. We further show that such improvement is also universal across different datasets, as demonstrated by the results on CIFAR-10 and iNaturalist in Appendix B.1. These findings confirm that RCG markedly boosts the performance of current generative models in unconditional generation, significantly reducing their reliance on human-annotated labels.

Moreover, such outstanding performance can be achieved with lower training cost compared to current generative models. In Figure 5, we compare the training cost and unconditional generation FIDs

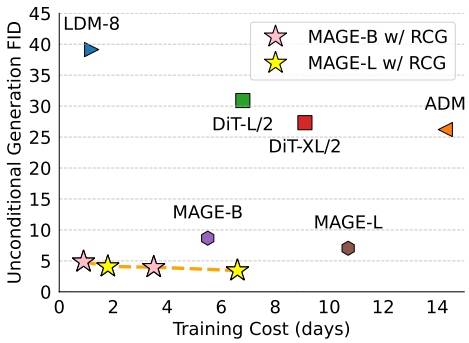

Figure 5: **RCG achieves outstanding unconditional generation performance with less training cost.** All numbers are reported under the unconditional generation setting. The training cost is measured using a cluster of 64 V100 GPUs. Given that the MoCo v3 ViT encoder is pre-trained and not needed for generation, its training cost is excluded. Detailed computational cost is reported in Appendix B.1.

Table 2: **RCG largely improves the state-of-the-art in unconditional image generation** on ImageNet 256×256. All numbers are reported under the unconditional generation setting. Following common practice, we report the number of parameters used during generation. † denotes semi-parametric methods which require ground-truth ImageNet images during generation.

Unconditional generation	#params	FID↓	IS↑
BigGAN [19]	∼70M	38.61	24.7
ADM [18]	554M	26.21	39.7
MaskGIT [10]	227M	20.72	42.1
RCDM† [5]	-	19.0	51.9
IC-GAN† [9]	∼75M	15.6	59.0
ADDP [61]	176M	8.9	95.3
MAGE-B [41]	176M	8.67	94.8
MAGE-L [41]	439M	7.04	123.5
RDM-IN† [4]	400M	5.91	158.8
RCG (MAGE-B)	239M	3.98	177.8
RCG (MAGE-L)	502M	3.44	186.9
RCG-G (MAGE-B)	239M	3.19	212.6
RCG-G (MAGE-L)	502M	**2.15**	**253.4**

of RCG and current generative models. RCG achieves a significantly lower FID with less training cost than current generative models. Specifically, MAGE-B with RCG achieves an unconditional generation FID of 4.87 in less than a day when trained on 64 V100 GPUs. This demonstrates that decomposing the complex tasks of unconditional generation into much simpler sub-tasks can significantly simplify the data modeling process.

RCG largely improves the state-of-the-art in unconditional image generation. In Table 2, we compare MAGE with RCG and previous state-of-the-art methods in unconditional image generation. As shown in Figure 8 and Table 2, RCG can generate images with both high fidelity and diversity, achieving an FID of 3.44 and an Inception Score of 186.9. These results are further enhanced with the guided version of RCG (RCG-G), which reaches an FID of 2.15 and an Inception Score of 253.4, significantly surpassing previous methods of unconditional image generation.

RCG's unconditional generation performance rivals leading methods in class-conditional image generation. In Table 3, we perform a system-level comparison between the *unconditional* RCG and state-of-the-art *class-conditional* image generation methods. MAGE-L with RCG is comparable to leading class-conditional methods, with and without guidance. These results demonstrate that RCG, for the first time, improves the performance of unconditional image generation on complex data distributions to the same level as that of state-of-the-art class-conditional generation methods, effectively bridging the historical gap between class-conditional and unconditional generation.

In Table 4, we further conduct an apple-to-apple comparison between the class-conditional versions of LDM-8, ADM, and DiT-XL/2 and their unconditional counterparts using RCG. Surprisingly, with RCG, these generative models consistently outperform their class-conditional versions by a noticeable margin. This demonstrates that the rich semantic information from the unconditionally generated representations can guide the generative process even more effectively than class labels.

https://doi.org/10.52202/079017-3985

Table 3: **System-level comparison: RCG's unconditional generation performance rivals leading methods in class-conditional image generation** on ImageNet 256×256. Following common practice, we report the number of parameters used during generation. Class-conditional results are marked in gray.

Methods	#params	w/o Guidance FID↓	w/o Guidance IS↑	w/ Guidance FID↓	w/ Guidance IS↑
Class-conditional					
ADM [18]	554M	10.94	101.0	4.59	186.7
LDM-4 [54]	400M	10.56	103.5	3.60	247.7
U-ViT-H/2-G [2]	501M	-	-	2.29	263.9
DiT-XL/2 [50]	675M	9.62	121.5	2.27	278.2
DiffiT [29]	-	-	-	1.73	276.5
BigGAN-deep [6]	160M	6.95	198.2	-	-
MaskGIT [10]	227M	6.18	182.1	-	-
MDTv2-XL/2 [24]	676M	5.06	155.6	**1.58**	314.7
CDM [34]	-	4.88	158.7	-	-
MAGVIT-v2 [68]	307M	3.65	200.5	1.78	**319.4**
RIN [36]	410M	3.42	182.0	-	-
VDM++ [38]	2B	**2.40**	**225.3**	2.12	267.7
RCG, conditional (MAGE-L)	512M	2.99	215.5	2.25	300.7
Unconditional					
RCG (MAGE-L)	502M	3.44	186.9	2.15	253.4

Table 4: **Apple-to-apple comparison: RCG's unconditional generation outperforms the class-conditional counterparts of current generative models**, evaluated on ImageNet 256×256. MAGE does not report its class-conditional generation performance. Class-conditional results are marked in gray.

Methods		FID↓	IS↑
LDM-8 [54]	w/ class labels	17.41	72.9
	w/ RCG	**11.30**	**101.9**
ADM [18]	w/ class labels	10.94	101.0
	w/ RCG	**6.24**	**136.9**
DiT-XL/2 [50]	w/ class labels	9.62	121.5
	w/ RCG	**4.89**	**143.2**

As shown in Table 3 and Appendix C, RCG also supports class-conditional generation with a simple extension. Our representation diffusion model can easily adapt to class-conditional representation generation, thereby enabling RCG to also adeptly perform class-conditional image generation. This result demonstrates the effectiveness of RCG in leveraging its superior unconditional generation performance to benefit downstream conditional generation tasks.

Importantly, such an adaptation does not require retraining the representation-conditioned image generator. For any new conditioning, only the lightweight representation generator needs to be retrained. This potentially enables pre-training of the self-supervised encoder and image generator on large-scale unlabeled datasets, and task-specific training of conditional representation generator on a small-scale labeled dataset. We believe that this property, similar to self-supervised learning, allows RCG to both benefit from large unlabeled datasets and adapt to various downstream generative tasks with minimal overheads. We leave the exploration on this direction to future work.

4.2 Qualitative Insights

In this section, we showcase the visualization results of RCG, providing insights into its superior generative capabilities. Figure 8 illustrates RCG's unconditional image generation results on ImageNet 256×256. The model is capable of generating both diverse and high-quality images without relying on human annotations. The high-level semantic diversity in RCG's generation stems from

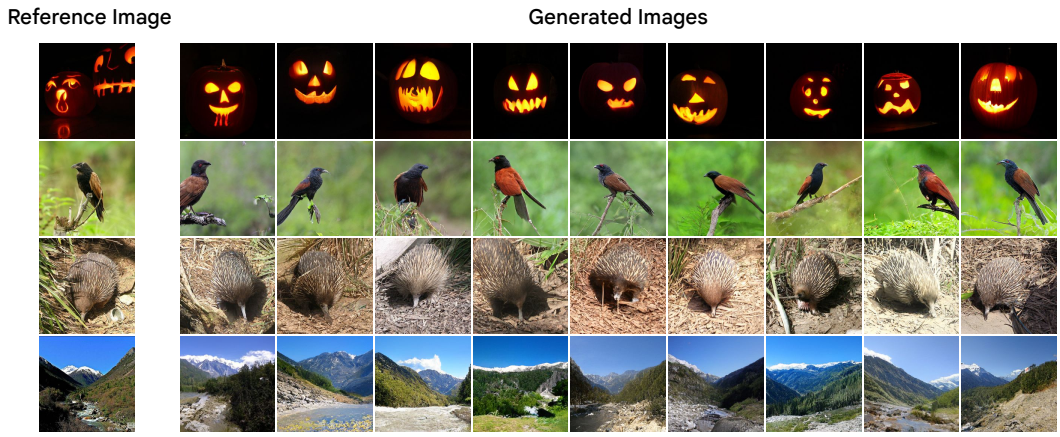

Figure 6: RCG can generate images with diverse appearances but similar semantics from the same representation. We extract representations from reference images and, for each representation, generate a variety of images from different random seeds.

Figure 7: RCG's results conditioned on interpolated representations from two images. The semantics of the generated images gradually transfer between the two images.

its representation generator, which models the distribution of representations and samples them with varied semantics. By conditioning on these representations, the complex data distribution is broken down into simpler, representation-conditioned sub-distributions. This decomposition significantly simplifies the task for the image generator, leading to the production of high-quality images.

Besides high-quality generation, the image generator can also introduce significant low-level diversity in the generative process. Figure 6 illustrates RCG's ability to generate diverse images that semantically align with each other, given the same representation from the reference image. The images generated by RCG can capture the semantic essence of the reference images while differing in specific details. This result highlights RCG's capability to leverage semantic information in representations to guide the generative process, without compromising the diversity that is important in unconditional image generation.

Figure 7 further showcases RCG's semantic interpolation ability, demonstrating that the representation space is semantically smooth. By leveraging RCG's dependency on representations, we can semantically transition between two images by linearly interpolating their respective representations. The interpolated images remain realistic across varying interpolation rates, and their semantic contents smoothly transition from one image to another. For example, interpolating between an image of "Tibetan mastiff" and an image of "wool" could generate a novel image featuring a dog wearing a woolen sweater. This also highlights RCG's potential in manipulating image semantics within a low-dimensional representation space, offering new possibilities to control image generation.

https://doi.org/10.52202/079017-3985

5 Discussion

Computer vision has entered a new era where learning from extensive, unlabeled datasets is becoming increasingly common. Despite this trend, the training of image generation models still mostly relies on labeled datasets, which could be attributed to the large performance gap between conditional and unconditional image generation. Our paper addresses this issue by exploring *Representation-Conditioned Generation*, which we propose as a nexus between conditional and unconditional image generation. We demonstrate that the long-standing performance gap can be effectively bridged by generating images conditioned on self-supervised representations and leveraging a representation generator to model and sample from this representation space. We believe this approach has the potential to liberate image generation from the constraints of human annotations, enabling it to fully harness the vast amounts of unlabeled data and even generalize to modalities that are beyond the scope of human annotation capabilities.

Acknowledgements. This work was supported by the GIST MIT Research Collaboration grant funded by GIST. Tianhong Li was also supported by the Mathworks Fellowship. We thank Huiwen Chang, Saining Xie, Zhuang Liu, Xinlei Chen, and Mike Rabbat for their discussion and feedback. We also thank Xinlei Chen for his support on MoCo v3.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[3] Fan Bao, Chongxuan Li, Jiacheng Sun, and Jun Zhu. Why are conditional generative models better than unconditional ones? *arXiv preprint arXiv:2212.00362*, 2022.

[4] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems*, 35:15309–15324, 2022.

[5] Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Int. Conf. on Learning Representations (ICLR)*, 2019.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Int. Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

[9] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34:27517–27529, 2021.

[10] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

[11] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *icml*, pp. 1597–1607. PMLR, 2020.

[14] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Int. Conference on Computer Vision (ICCV)*, pp. 9640–9649, 2021.

[15] Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024.

[16] Pedro Costa, Adrian Galdran, Maria Ines Meyer, Meindert Niemeijer, Michael Abràmoff, Ana Maria Mendonça, and Aurélio Campilho. End-to-end adversarial retinal image synthesis. *IEEE transactions on medical imaging*, 37(3):781–791, 2017.

[17] 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*, pp. 248–255. Ieee, 2009.

[18] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[19] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. on Learning Representations (ICLR)*, 2021.

[21] DC Dowson and BV666017 Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

[22] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[23] 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*, pp. 12873–12883, 2021.

[24] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023.

[25] Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It's raw! audio generation with state-space models. In *International Conference on Machine Learning*, pp. 7616–7633. PMLR, 2022.

[26] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[27] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. 2014.

[28] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

[29] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. *arXiv preprint arXiv:2312.02139*, 2023.

[30] 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*, pp. 9729–9738, 2020.

[31] 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.

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

[33] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

[34] 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.

 https://doi.org/10.52202/079017-3985

[35] Vincent Tao Hu, David W. Zhang, Yuki M. Asano, Gertjan J. Burghouts, and Cees G. M. Snoek. Self-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18413–18422, June 2023.

[36] Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

[37] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[38] Diederik P Kingma and Ruiqi Gao. Understanding the diffusion objective as a weighted integral of elbos. *arXiv preprint arXiv:2303.00848*, 2023.

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

[40] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

[42] Jen-Yu Liu, Yu-Hua Chen, Yin-Cheng Yeh, and Yi-Hsuan Yang. Unconditional audio generation with generative adversarial networks and cycle regularization. *arXiv preprint arXiv:2005.08526*, 2020.

[43] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14286–14295, 2020.

[44] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[45] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[46] Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *International conference on machine learning*, pp. 4183–4192. PMLR, 2019.

[47] Ali Madani, Mehdi Moradi, Alexandros Karargyris, and Tanveer Syeda-Mahmood. Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. In *2018 IEEE 15th International symposium on biomedical imaging (ISBI 2018)*, pp. 1038–1042. IEEE, 2018.

[48] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*, 2016.

[49] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

[50] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

[51] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–10629, 2022.

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

[53] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

[55] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[56] Dave Salvator. Nvidia developer blog. https://developer.nvidia.com/blog/getting-immediate-speedups-with-a100-tf32, 2020.

[57] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

[58] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[59] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Int. Conf. on Learning Representations (ICLR)*, 2021.

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

[61] Changyao Tian, Chenxin Tao, Jifeng Dai, Hao Li, Ziheng Li, Lewei Lu, Xiaogang Wang, Hongsheng Li, Gao Huang, and Xizhou Zhu. Addp: Learning general representations for image recognition and generation with alternating denoising diffusion process. *arXiv preprint arXiv:2306.05423*, 2023.

[62] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

[63] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[64] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

[65] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

[66] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[67] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[68] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[69] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[70] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Int. Conference on Machine Learning (ICML)*, pp. 7354–7363, 2019.

[71] Tianyang Zhang, Huazhu Fu, Yitian Zhao, Jun Cheng, Mengjie Guo, Zaiwang Gu, Bing Yang, Yuting Xiao, Shenghua Gao, and Jiang Liu. Skrgan: Sketching-rendering unconditional generative adversarial networks for medical image synthesis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22*, pp. 777–785. Springer, 2019.

[72] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

 https://doi.org/10.52202/079017-3985

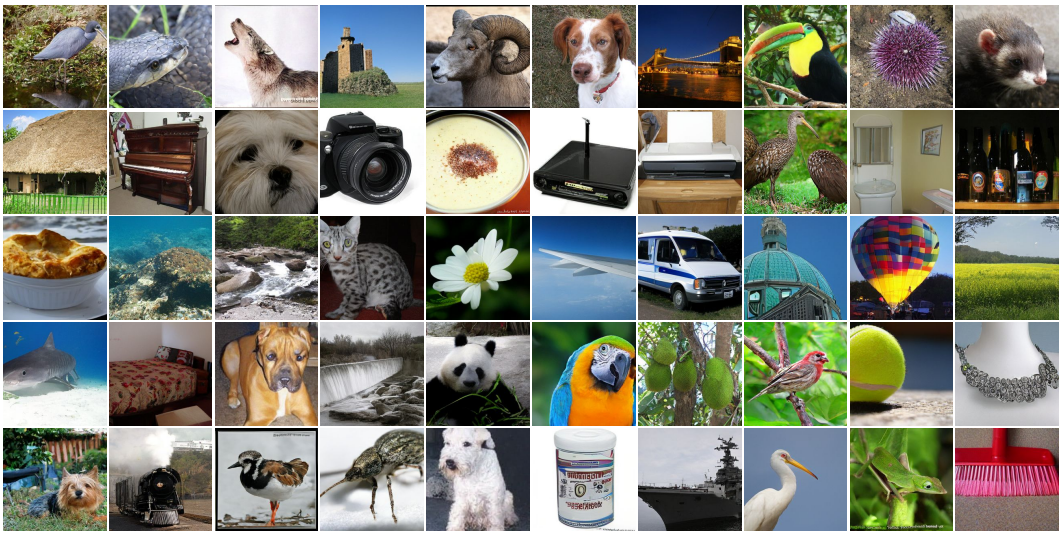

Figure 8: **Unconditional generation results** of RCG on ImageNet 256×256. RCG can generate realistic images with diverse semantics without human annotations.

A Implementation Details

In this section, we describe the implementation details of RCG, including hyper-parameters, model architecture, and training paradigm. We also include a copy of our code in the supplementary material. **All codes and pre-trained model weights will be made publicly available.**

Image Encoder. We use Vision Transformers (ViTs) [20] pre-trained with MoCo v3 [14] as the default image encoder. We evaluate three ViT variants (ViT-S, ViT-B, and ViT-L) in the main paper, each trained on ImageNet for 300 epochs. We utilize the image representation after the MLP projection head, favoring its adjustable dimensionality. An output dimension of 256 has proven the most effective. The representation of each image is normalized by its own mean and variance. Detailed training recipes of our pre-trained image encoder can be found in [14].

Representation Diffusion Model (RDM). Our RDM architecture employs a backbone of multiple fully connected blocks. We use 12 blocks and maintain a consistent hidden dimension of 1536 across the network. The timestep t is discretized into 1000 values, each embedded into a 256-dimensional vector. For class-conditional RDM, we embed each class label into a 512-dimensional vector. Both timestep and class label embeddings are projected to 1536 dimensions using different linear projection layers in each block. Detailed hyper-parameters for RDM's training and generation can be found in Table 5.

Image Generator. We experiment with ADM [18], LDM [54], DiT [50], and MAGE [41] as the image generator. For representation-conditioned ADM, LDM and DiT, we substitute the original class embedding conditioning with the image representation. We follow ADM's original training recipe to train representation-conditioned ADM for 400 epochs. We follow LDM-8's original training recipe, with modifications including a batch size of 256, a learning rate of 6.4e-5, and a training duration of 40 epochs. We follow the DiT training scheme in [15], which trains DiT-XL for 400 epochs with batch size 2048 and a linear learning rate warmup for 100 epochs. The β_2 of the AdamW optimizer is set to 0.95. For representation-conditioned MAGE, we replace the default "fake" class token embedding $[\texttt{C}_0]$ with the image representation for conditioning.

During the training of RCG's image generator, the image is resized so that the smaller side is of length 256, and then randomly flipped and cropped to 256×256. The input to the SSL encoder is further resized to 224×224 to be compatible with its positional embedding size. Our implementation of guidance follows Muse [11], incorporating a linear guidance scale scheduling. Detailed hyper-parameters for our representation-conditioned MAGE are provided in Table 6.

https://doi.org/10.52202/079017-3985

Table 5: **RDM implementation details.**

config	value
#blocks	12
hidden dimension	1536
#params	63M
optimizer	AdamW [45]
learning rate	5.12e-4
weight decay	0.01
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	512
learning rate schedule	constant
training epochs	200
augmentation	Resize(256) RandCrop(256) RandomFlip (0.5)
diffusion steps	1000
noise schedule	linear
DDIM steps	250
η	1.0

Table 6: **Repsentation-conditioned MAGE implementation details.**

config	value
optimizer	AdamW [45]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
batch size	4096
learning rate schedule	cosine decay [44]
warmup epochs	10
training epochs	800
gradient clip	3.0
label smoothing [60]	0.1
dropout	0.1
augmentation	Resize(256) RandCrop(256) RandomFlip (0.5)
masking ratio min	0.5
masking ratio max	1.0
masking ratio mode	0.75
masking ratio std	0.25
rep. drop rate	0.1
parallel-decoding temperature	6.0 (B) 11.0 (L)
parallel-decoding steps	20
guidance scale (τ)	1.0 (B) 6.0 (L)
guidance scale schedule	linear [11]

B Additional Quantitative Results

B.1 Ablations

This section provides a comprehensive ablation study of the three core components of RCG. Our default setup uses MoCo v3 ViT-B as the pre-trained image encoder, an RDM with a 12-block, 1536-hidden-dimension backbone trained for 100 epochs, and a MAGE-B image generator trained for 200 epochs. Unless otherwise specified, all other properties and modules are set to the default settings during each component's individual ablation. The FID in this section is evaluated against the ImageNet validation set.

Distribution Mapping. Table 7 ablates the image encoder. Table 7a compares image encoders trained via various self-supervised learning methods (MoCo v3, DINO, and iBOT), highlighting

 https://doi.org/10.52202/079017-3985

Table 7: **Distribution mapping ablation experiments.** The default encoder is MoCo v3 ViT-B with 256 projection dimension. Default settings are marked in gray.

Method	FID	IS
No condition	14.23	57.7
MoCo v3 [14]	**5.07**	142.5
DINO [8]	7.53	160.8
iBOT [72]	8.05	148.7
DeiT [62] (supervised)	5.51	**211.7**

(a) **Pre-training.** RCG achieves good performance with encoders pre-trained with different contrastive learning and supervised learning methods.

Model	params	lin.	FID	IS
ViT-S	22M	73.2	5.77	120.8
ViT-B	86M	76.7	5.07	142.5
ViT-L	304M	77.6	**5.06**	**148.2**

(b) **Model size.** RCG scales up with larger pre-trained encoders with better linear probing accuracy.

Projection Dim	FID	IS
32	9.14	81.0
64	6.09	119.2
128	5.19	**143.3**
256	**5.07**	142.5
768	6.10	112.7

(c) **Projection dimension.** The dimensionality of the image representation is important in RCG's performance.

Table 8: **Representation generation ablation experiments.** The default RDM backbone is of 12 blocks and 1536 hidden dimensions, trained for 100 epochs, and takes 250 sampling steps during generation. The representation Frechet Distance (rep FD) is evaluated between 50K generated representations and representations extracted from the ImageNet training set by MoCo v3 ViT-B. Default settings are marked in gray.

#Blocks	FID	IS	rep FD
3	7.53	113.5	0.71
6	5.40	132.9	0.53
12	**5.07**	**142.5**	**0.48**
18	5.20	141.9	0.50
24	5.13	141.5	0.49

(a) **Model depth.** A deeper RDM can improve generation performance.

Hidden Dim	FID	IS	rep FD
256	12.99	67.3	5.98
512	9.07	99.8	1.19
1024	5.35	132.0	0.56
1536	**5.07**	142.5	**0.48**
2048	5.09	**142.8**	**0.48**

(b) **Model width.** A wider RDM can improve generation performance.

Epochs	FID	IS	rep FD
10	5.94	124.4	0.87
50	5.21	138.3	0.54
100	5.07	142.5	0.48
200	5.07	**145.1**	**0.47**
300	**5.05**	144.3	**0.47**

(c) **Training epochs.** Training RDM longer improves generation performance.

#Steps	FID	IS	rep FD
20	5.80	120.3	0.87
50	5.28	133.0	0.55
100	5.15	138.1	**0.48**
250	**5.07**	142.5	**0.48**
500	**5.07**	**142.9**	0.49

(d) **Diffusion steps.** More sampling steps can improve generation performance.

Table 9: **Image generation ablation experiments.** The default image generator is MAGE-B trained for 200 epochs. Table 9c evaluates different τ using MAGE-L with RCG trained for 800 epochs and the FID is evaluated following ADM suite. Default settings are marked in gray.

Conditioning	FID	IS
No condition	14.23	57.7
Cluster label	6.60	121.9
Class label	5.83	**147.3**
Generated rep.	**5.07**	142.5
Oracle rep.	4.37	149.0

(a) **Conditioning.** Conditioning on generated representations improves over all baselines in FID.

Epochs	FID	IS
100	6.03	127.7
200	5.07	142.5
400	4.48	158.8
800	**4.15**	**172.0**

(b) **Training epochs.** Longer training can improve generation performance.

τ	0.0	1.0	3.0	5.0	6.0	7.0
FID	3.44	2.59	2.29	2.31	**2.15**	2.31
IS	186.9	228.5	251.3	252.7	**253.4**	252.6

(c) **Classifier-free guidance scale.** $\tau = 6$ achieves the best FID and IS for RCG-L.

their substantial improvements over the unconditional baseline. Additionally, an encoder trained with DeiT [62] in a supervised manner also exhibits impressive performance, indicating RCG's adaptability to both supervised and self-supervised pre-training approaches.

We also notice that using representations from MoCo v3 achieves better FID than using representations from DINO/iBOT. This is likely because only MoCo v3 uses an InfoNCE loss. Literature has shown that optimizing InfoNCE loss can maximize uniformity and preserve maximal information in the representation. The more information in the representation, the more guidance it can provide for the image generator, leading to better and more diverse generation. To demonstrate this, we compute the uniformity loss on representations [65]. Lower uniformity loss indicates higher uniformity and more information in the representation. The uniformity loss of representations from MoCo v3, DINO, and iBOT is -3.94, -3.60, and -3.55, respectively, which aligns well with their generation performance.

Table 7b assesses the impact of model size on the pre-trained encoder. Larger models with better linear probing accuracy consistently enhance generation performance, although a smaller ViT-S model still achieves decent results.

We further analyze the effect of image representation dimensionality, using MoCo v3 ViT-B models trained with different output dimensions from their projection head. Table 7c shows that neither excessively low nor high-dimensional representations are ideal – too low dimensions lose vital image information, while too high dimensions pose challenges for the representation generator.

Representation Generation. Table 8 ablates the representation diffusion model and its effectiveness in modeling representation distribution. The RDM's depth and width are controlled by the number of fc blocks and hidden dimensions. Table 8a and Table 8b ablate these parameters, indicating an

Table 10: **CIFAR-10 and iNaturalist results.** RCG consistently improves unconditional image generation performance on different datasets.

Dataset	Methods		FID
CIFAR-10	Improved DDPM [49]	baseline	3.29
		w/ RCG	**2.62**
		w/ class labels	2.89
iNaturalist 2021	MAGE-B	baseline	8.64
		w/ RCG	**4.49**
		w/ class labels	4.55

optimal balance at 12 blocks and 1536 hidden dimensions. Further, Table 8c and Table 8d suggest that RDM's performance saturates at 200 training epochs and 250 diffusion steps.

Besides evaluating FID and IS on generated images, we also assess the Frechet Distance (FD) [21] between the generated representations and the ground-truth representations. A smaller FD indicates that the distribution of generated representations more closely resembles the ground-truth distribution. Since the MoCo v3 encoder is trained on the ImageNet training set, the representation distribution in the training set can be slightly different from that in the validation set. To establish a better reference point, we compute the FD between 50K randomly sampled representations from the training set and the representations from the entire training set, which should serve as the lower bound of the FD for our representation generator. The result is an FD of 0.38, demonstrating that our representation generator (with an FD of 0.48) can accurately model the representation distribution.

We also evaluate the representation generator against the validation set, resulting in an FD of 2.73. As a reference point, the FD between 50K randomly sampled representations from the training set and the validation set is 2.47, which is also close to the FD of our representation generator.

Image Generation. Table 9 ablates RCG's image generator. Table 9a experiments with MAGE-B under different conditioning. MAGE-B with RCG significantly surpasses the unconditional and clustering-based baselines, and further outperforms the class-conditional baseline in FID. This shows that representations could provide rich semantic information to guide the generative process. It is also quite close to the "upper bound" which is conditioned on oracle representations from ImageNet *real* images, demonstrating the effectiveness of the representation generator in producing realistic representations.

We also ablate the training epochs of the image generator and the guidance scale τ, as shown in Table 9b and Table 9c. Training MAGE longer keeps improving the generation performance, and $\tau = 6$ achieves the best FID and IS.

B.2 Other Datasets

In this section, we evaluate RCG on datasets other than ImageNet to validate its consistent effectiveness across different datasets. We select CIFAR-10 and iNaturalist 2021 [64]. CIFAR-10 represents a relatively simple and low-dimensional image distribution, and iNaturalist 2021 represents a more complex image distribution, with 10,000 classes and 2.7 million images. For CIFAR-10, we employ SimCLR [13] trained on CIFAR-10 as the image encoder and Improved DDPM [49] as the image generator. The FID is evaluated between 50,000 generated images and the CIFAR-10 training set. For iNaturalist, we employ MoCo v3 ViT-B trained on ImageNet as the image encoder and MAGE-B as the image generator. The FID is evaluated between 100,000 generated images and the iNaturalist validation set, which also consists of 100,000 images.

As shown in Table 10, RCG consistently enhances unconditional image generation performance on both CIFAR-10 and iNaturalist 2021, demonstrating its universal effectiveness across various datasets. Notably, the improvement on complex data distributions such as ImageNet and iNaturalist is more significant than on simpler data distributions such as CIFAR-10. This is because RCG decomposes a complex data distribution into two relatively simpler distributions: the representation distribution and the data distribution conditioned on the representation distribution. Such decomposition is particularly effective on complex data distributions, such as natural images, paving the way for generative models to model unlabeled complex data distributions.

https://doi.org/10.52202/079017-3985

Table 11: **Computational cost.** RCG achieves a much smaller FID with similar or less computational cost as baseline methods. The number of parameters, training cost, and the number of training epochs of the representation generator and the image generator are reported separately.

Unconditional Generation	#Params (M)	Training Cost (days)	Epochs	Throughput (samples/s)	FID
LDM-8 [54]	395	1.2	150	0.9	39.13
ADM [18]	554	14.3	400	0.05	26.21
DiT-L [50]	458	6.8	400	0.3	30.9
DiT-XL [50]	675	9.1	400	0.2	27.32
MAGE-B [41]	176	5.5	1600	3.9	8.67
MAGE-L [41]	439	10.7	1600	2.4	7.04
RCG (MAGE-B)	63+176	0.1+0.8	100+200	3.6	4.87
RCG (MAGE-B)	63+176	0.2+3.3	200+800	3.6	3.98
RCG (MAGE-L)	63+439	0.3+1.5	100+200	2.2	4.09
RCG (MAGE-L)	63+439	0.6+6.0	200+800	2.2	3.44

Table 12: RCG's unconditional generation FID, IS, precision and recall on ImageNet 256×256, evaluated following ADM's suite [18].

Methods	FID↓	IS↑	Prec.↑	Rec.↑
RCG (MAGE-B)	3.98	177.8	0.84	0.47
RCG (MAGE-L)	3.44	186.9	0.82	0.52
RCG-G (MAGE-B)	3.19	212.6	0.83	0.48
RCG-G (MAGE-L)	2.15	253.4	0.81	0.53

B.3 Computational Cost

In Table 11, we present a detailed evaluation of RCG's computational cost, including the number of parameters, training costs, and generation throughput. The training cost of all image generators is measured using a cluster of 64 V100 GPUs. The training cost of RDM is measured using 1 V100 GPU, divided by 64. The generation throughput is measured on a single V100 GPU. As LDM and ADM measure their generation throughput on a single NVIDIA A100 [54], we convert it to V100 throughput by assuming a ×2.2 speedup of A100 vs V100 [56].

As shown in the Table 11, RCG requires significantly lower training costs to achieve great performance. For instance, it achieves an FID of 4.87 in less than one day of training. Moreover, the training and inference costs of the representation generator are marginal compared to those of the image generator. This efficiency potentially enables for lightweight adaptation to various downstream generative tasks by training only the representation generator on small-scale labeled datasets.

B.4 Precision and Recall

In Table 12, we report the unconditional generation precision and recall of RCG, evaluated on ImageNet 256×256 following the ADM suite [18]. Larger models as well as incorporating guidance (RCG-G) both improve recall while slightly decreases precision.

C Additional Qualitative Results

We include more qualitative results, including class-unconditional image generation (Figure 9), class-conditional image generation (Figure 10 and Figure 11), and the comparison between generation results with or without guidance (Figure 13). All these results demonstrate RCG's superior performance in image generation. We also include some failure cases in Figure 12.

D Limitations and Negative Impact

Limitations. Like any other generative models, RCG can also produce unrealistic or low-quality results (see Appendix C for some examples).

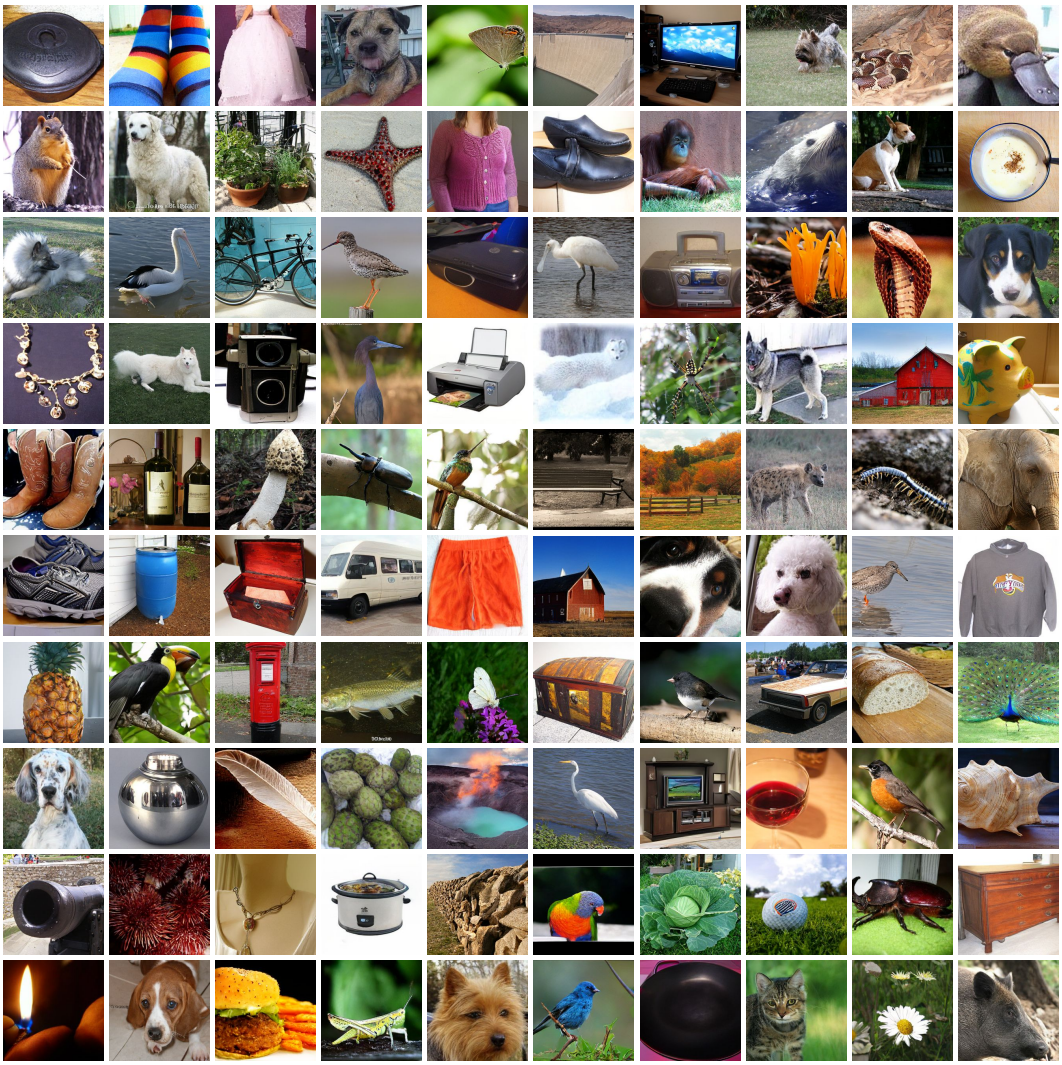

Figure 9: More RCG class-unconditional image generation results on ImageNet 256×256.

Societal Impact. Despite the rapid advancements in generative models, they also carry potential negative societal impacts. For instance, such models can amplify existing biases present in internet data. RCG, being a generative model, is not immune to these issues. However, it is important to note that RCG operates within an unconditional generation framework, which does not depend on human-provided labels. This characteristic might possess the potential to mitigate the influence of human biases, offering a more neutral approach to data generation compared to traditional conditional models.

https://doi.org/10.52202/079017-3985

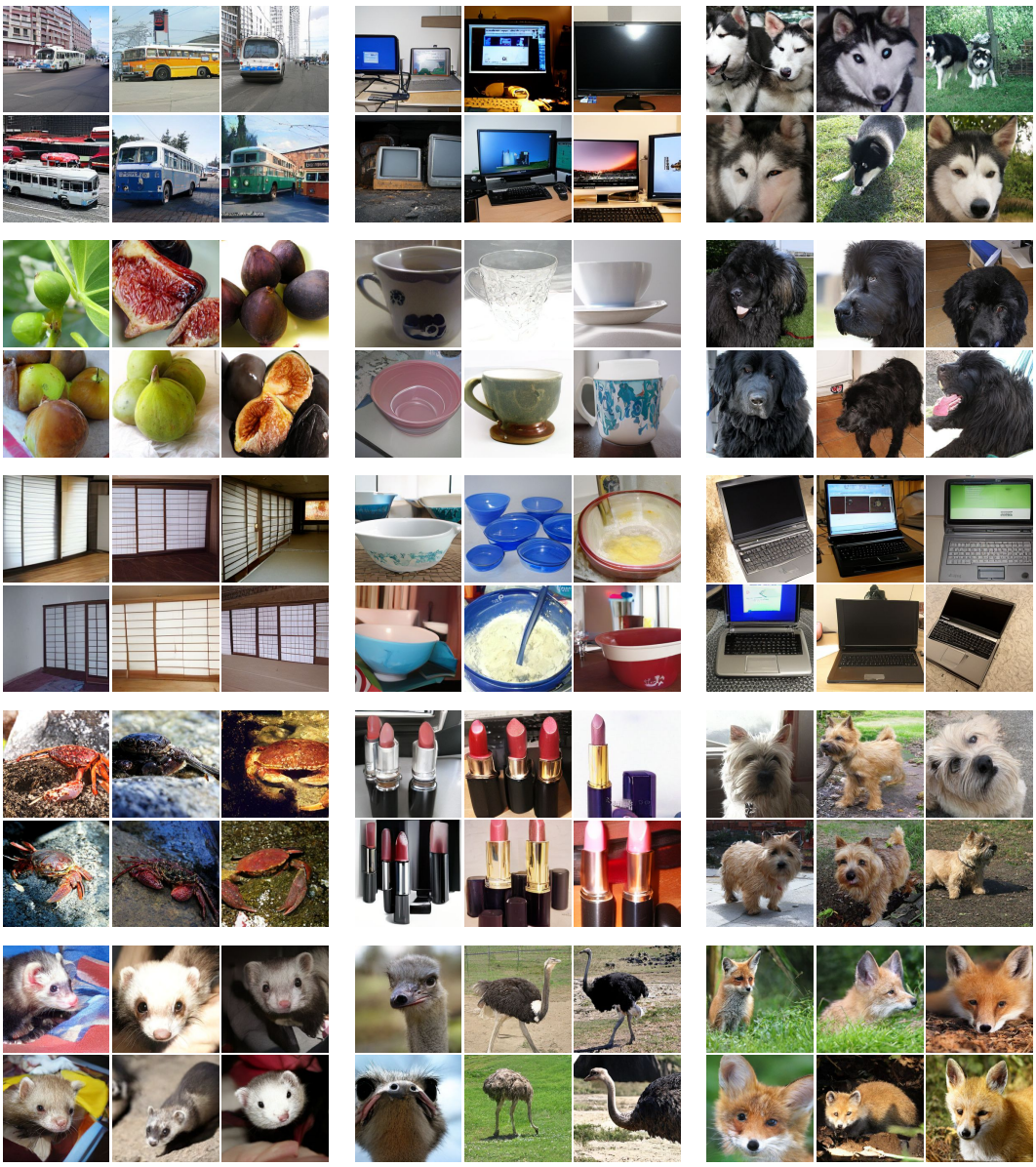

Figure 10: RCG class-conditional image generation results on ImageNet 256×256. Classes are 874: trolleybus, 664: monitor, 249: malamute; 952: fig, 968: cup, 256: Newfoundland; 789: shoji, 659: mixing bowl, 681: notebook; 119: rock crab, 629: lipstick, 192: cairn; 359: ferret, 9: ostrich, 277: red fox.

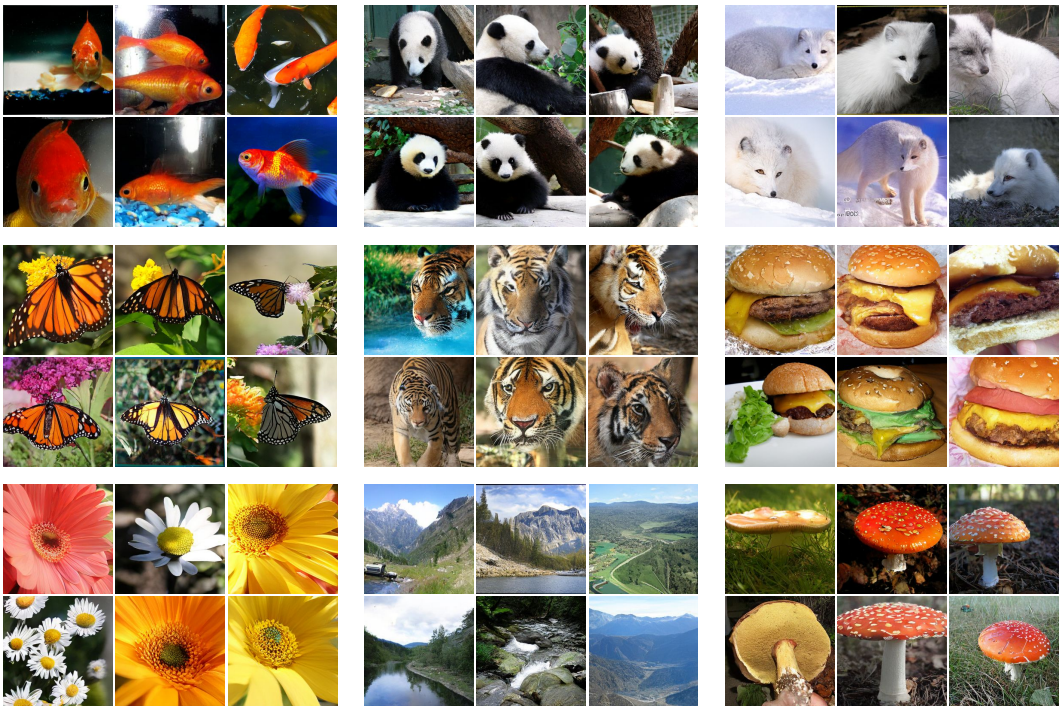

Figure 11: RCG class-conditional image generation results on ImageNet 256×256. Classes are 1: goldfish, 388: panda, 279: Arctic fox; 323: monarch butterfly, 292: tiger, 933: cheeseburger; 985: daisy, 979: valley, 992: agaric

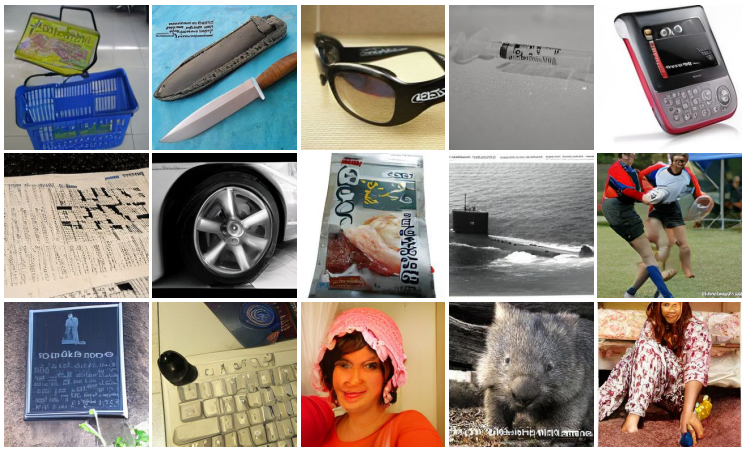

Figure 12: Similar to other generative models on ImageNet, RCG also could face difficulty in generating texts, regular shapes (such as keyboard and wheel), and realistic human.

https://doi.org/10.52202/079017-3985

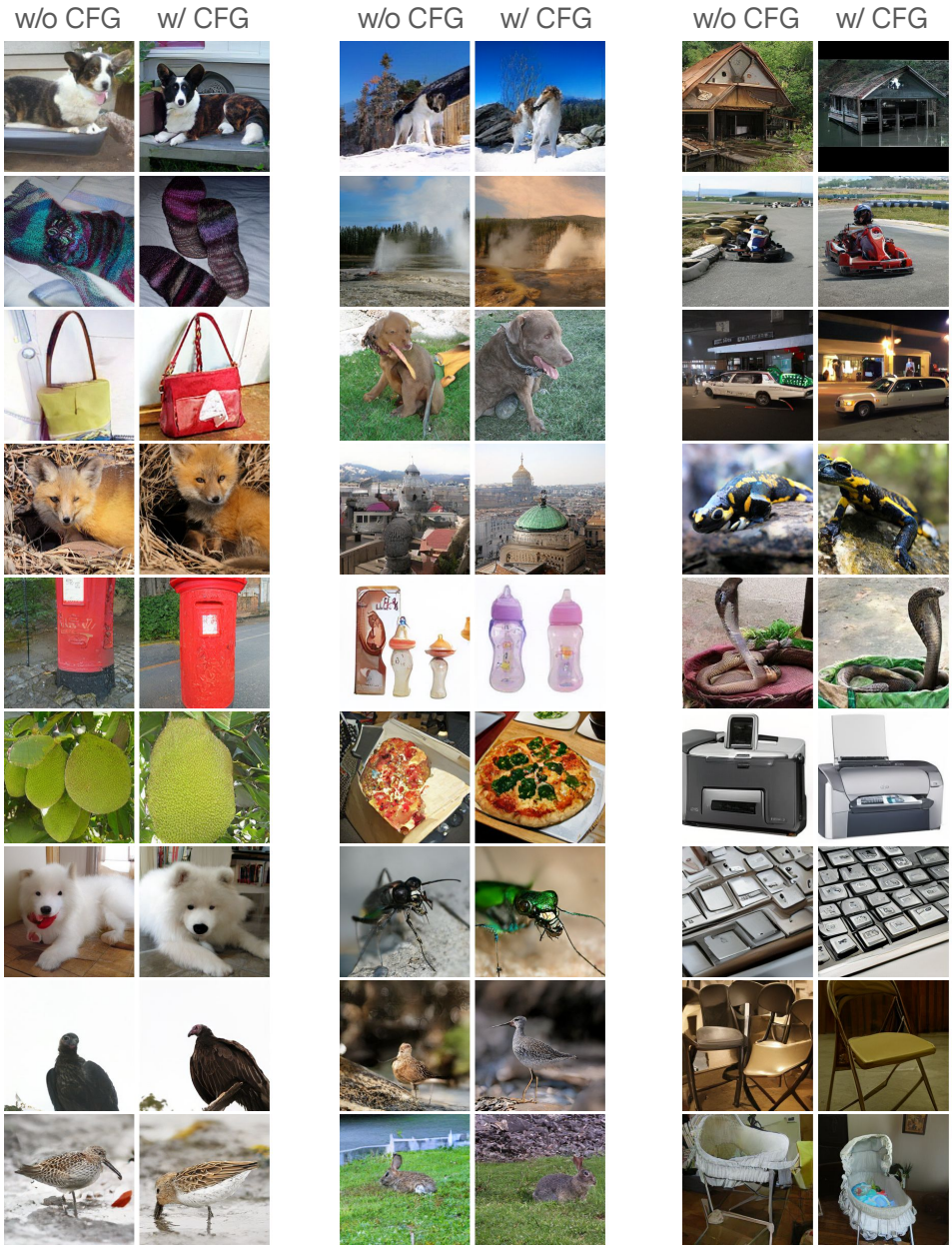

Figure 13: Class-unconditional image generation results on ImageNet 256×256, with or without guidance. RCG achieves strong generation performance even without guidance. Incorporating guidance further improves the generation quality.

https://doi.org/10.52202/079017-3985

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: This paper presents an unconditional image generation method that rivals the performance of the state-of-the-art class-conditional generation methods.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Appendix D.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

https://doi.org/10.52202/079017-3985

Justification: This paper does not include theoretical contribution.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: See Appendix A.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Code is available at https://github.com/LTH14/rcg.

Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: See Appendix A.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: Following common practice in the generative modeling literature, we do not report error bars in this paper because of the heavy computation overheads.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-3985

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: See subsection B.3.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: see Appendix D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [Yes]

    Justification: We will require the users to adhere to usage guidelines for our released models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly cite the original assets in the paper.

    Guidelines:

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

https://doi.org/10.52202/079017-3985

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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