One-Step Diffusion Distillation via Deep Equilibrium Models

Zhengyang Geng [*1] **Ashwini Pokle** [*1] **J. Zico Kolter** [1 2]

Abstract

Diffusion models excel at producing high-quality samples but naively require hundreds of iterations, prompting multiple attempts to distill the generation process into a faster network. However, many existing approaches suffer from a variety of challenges: the process for distillation training can be complex, often requiring multiple training stages, and the resulting models perform poorly when utilized in single-step generative applications. In this paper, we introduce a simple yet effective means of distilling diffusion models *directly* from initial noise to the resulting image. Of particular importance to our approach is to leverage a new Deep Equilibrium (DEQ) model as the distilled architecture: the Generative Equilibrium Transformer (GET). Our method enables fully offline training with just noise/image pairs from the diffusion model while achieving superior performance compared to existing one-step methods on comparable training budgets. We demonstrate that the DEQ architecture is crucial to this capability, as GET matches a $5\times$ larger ViT in terms of FID scores while striking a critical balance of computational cost and image quality. Code, checkpoints, and datasets will be released.

1. Introduction

Diffusion models (Sohl-Dickstein et al., 2015; Song and Ermon, 2019; Ho et al., 2020; Song et al., 2021b) have emerged as a powerful class of generative models due to their remarkable performance on a wide range of generative tasks (Rombach et al., 2022; Ramesh et al., 2022; Saharia et al., 2022; Meng et al., 2021; Nichol et al., 2022; Kong et al., 2020; Ho et al., 2022). These models are trained with a denoising objective derived from score matching (Hyvärinen and Dayan, 2005; Song and Ermon, 2019), variational inference (Sohl-Dickstein et al., 2015; Ho et al., 2020), or optimal transport (Liu et al., 2023; Lipman et al., 2023), enabling them to generate clean data samples by progressive denoising the initial Gaussian noise. Despite the promising results, one major drawback of diffusion models is their slow generative process, which often necessitates hundreds to thousands of model evaluations (Ho et al., 2020; Song et al., 2021b).

In an effort to speed up the sampling of diffusion models, researchers have proposed distillation methods (Salimans and Ho, 2022; Meng et al., 2022; Zheng et al., 2022; Song et al., 2023; Berthelot et al., 2023) aimed at distilling the long sampling chain into a more efficient few-step or single-step process. However, these methods often require carefully designed distillation targets, multiple training passes, and maintenance of dual model copies which increases memory and compute requirements.

In this work, our objective is to streamline the distillation of diffusion models while retaining the perceptual quality of the generated images. To this end, we introduce a simple and effective technique that distills a multi-step diffusion process into a single-step generative model, using solely noise/image pairs. At the heart of our technique is the Generative Equilibrium Transformer (GET), a novel Deep Equilibrium (DEQ) model (Bai et al., 2019). GET can be interpreted as an infinite depth network using weight-tied transformer layers, which also allows for the adaptive layer evaluations in the forward pass, striking a balance between inference speed and sample quality.

Our direct approach for distillation, via noise/image pairs generated by a diffusion model, can be applied to both ViT (Dosovitskiy et al., 2021; Peebles and Xie, 2022) and GET. Yet, in our experiments, we show that GET, in particular, is capable of achieving substantially better quality than ViTs using smaller models. Indeed, GET delivers perceptual image quality on par with or superior to other complex distillation techniques, such as progressive distillation (Salimans and Ho, 2022; Meng et al., 2022), in the context of both class-conditional and unconditional image generation.

To summarize, we make the following key contributions:

- We propose Generative Equilibrium Transformer (GET), a deep equilibrium vision transformer that is well-suited for *single-step* generative models.

[*]Equal contribution [1]Carnegie Mellon University [2]Bosch Center for AI. Correspondence to: Zhengyang Geng <zgeng2@cs.cmu.edu>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

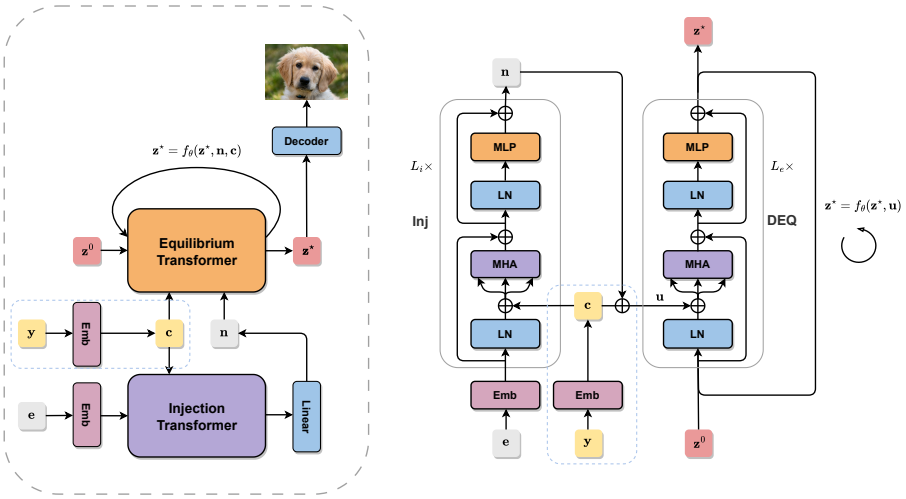

Figure 1. **Generative Equilibrium Transformer (GET).** (*Left*) GET consists of two major components: Injection transformer and Equilibrium transformer. The Injection transformer transforms noise embeddings into an input injection for the Equilibrium transformer. The Equilibrium transformer is the equilibrium layer that takes in noise input injection and an optional class embedding and solves for the fixed point. (*Right*) Details of transformer blocks in the Injection transformer (**Inj**) and Equilibrium transformer (**DEQ**), respectively. Blue dotted boxes denote optional class label inputs.

- We streamline diffusion distillation by training GET directly on noise/image pairs sampled from diffusion models, which turns out to be a simple yet effective strategy for producing one-step generative models in both class-conditional and unconditional cases.

2. Preliminaries

In this section, we will briefly introduce Deep Equilibrium (DEQ) Models. We provide a detailed background on diffusion models, distillation, and other fast sampling techniques for diffusion models in Appendix A.

DEQs (Bai et al., 2019) are neural networks of infinite depth, which solve for fixed points in the forward pass,

$$\lim_{k\to\infty} f_\theta(\mathbf{z}^k; \mathbf{x}) = f_\theta(\mathbf{z}^\star; \mathbf{x}) = \mathbf{z}^\star \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{d_x}$ is the input injection, $\mathbf{z}^\star \in \mathbb{R}^{d_z}$ is the output, and f_θ is the equilibrium layer parametrized by θ.

DEQs can utilize implicit differentiation to differentiate through the fixed point analytically. The Jacobian of $\mathbf{z}^\star$ with respect to the model weights θ is given by

$$\frac{\partial \mathbf{z}^\star}{\partial \theta} = \left(I - \frac{\partial f_\theta}{\partial \mathbf{z}^\star}\right)^{-1} \frac{\partial f_\theta(\mathbf{z}^\star; \mathbf{x})}{\partial \theta} \tag{2}$$

3. Generative Equilibrium Transformer

We introduce the Generative Equilibrium Transformer (GET), a Deep Equilibrium vision transformer designed for learning one-step generative models from diffusion models.

$$\mathbf{h}, \mathbf{c} = \mathrm{Emb}(\mathbf{e}),\ \mathrm{Emb}(\mathbf{y});\ \text{if } \mathbf{y} \notin \emptyset \tag{3}$$

$$\mathbf{n} = \mathrm{InjectionT}(\mathbf{h}, \mathbf{c}) \tag{4}$$

$$\mathbf{z}^\star = \mathrm{EquilibriumT}(\mathbf{z}^\star, \mathbf{n}, \mathbf{c}) \tag{5}$$

$$\tilde{\mathbf{x}} = \mathrm{Decoder}(\mathbf{z}^\star) \tag{6}$$

GET. Generative Equilibrium Transformer (GET) directly maps Gaussian noises $\mathbf{e}$ and optional class labels $\mathbf{y}$ to images $\tilde{\mathbf{x}}$. The major components of GET include the injection transformer (InjectionT, Eq. (4)) and the equilibrium transformer (EquilibriumT, Eq. (5)). The InjectionT transforms tokenized noise embedding $\mathbf{h}$ to an intermediate representation $\mathbf{n}$ that serves as the input injection for the equilibrium transformer. The EquilibriumT, which is the equilibrium layer, solves for the fixed point $\mathbf{z}^\star$ by taking in the noise injection $\mathbf{n}$ and an optional class embedding $\mathbf{c}$. Finally, this fixed point $\mathbf{z}^\star$ is decoded and rearranged to generate an image sample $\tilde{\mathbf{x}}$ (Eq. (6)). Figure 1 provides an overview of the GET architecture. Note that because we are directly distilling the entire generative process, there is no need for a time embedding t as is common in standard diffusion models.

Noise Embedding. GET first converts an input noise $\mathbf{e} \in \mathbb{R}^{H\times W\times C}$ into a sequence of 2D patches $\mathbf{p} \in \mathbb{R}^{N\times(P^2\cdot C)}$, where C is the number of channels, P is patch size, H and W denotes height and width of the original image, and $N = HW/P^2$ is the resulting number of patches. Let D denote the width of the network. We follow ViT to use a linear layer to project the N patches to D dimensional embedding. We add standard sinusoidal position encoding (Vaswani et al., 2017) to produce the noise embedding $\mathbf{h}$.

InjectionT & EquilibriumT. Both InjectionT and EquilibriumT are composed of a sequence of Transformer blocks. InjectionT is called only once to produce the noise injection $\mathbf{n}$, while EquilibriumT defines the function f_θ of the implicit layer $\mathbf{z}^\star = f_\theta(\mathbf{z}^\star, \mathbf{n}, \mathbf{c})$ that is called multiple times—creating a weight-tied computational graph—until convergence. A linear layer is added at the end of InjectionT to compute the noise injection $\mathbf{n}_l \in \mathbb{R}^{N\times 3D}$, $l \in [L_e]$, for each of the L_e GET blocks in EquilibriumT.

Transformer Block. GET utilizes a near-identical block design for the noise injection (InjectionT) and the equilibrium layer (EquilibriumT), differing only at the injection interface. Specifically, the transformer block is built upon the standard Pre-LN transformer block (Xiong et al., 2020; Dosovitskiy et al., 2021; Peebles and Xie, 2022):

$$\begin{aligned}\mathbf{z} &= \mathbf{z} + \text{Attention}\left(\text{LN}\left(\mathbf{z}\right), \mathbf{u}\right)\\ \mathbf{z} &= \mathbf{z} + \text{FFN}\left(\text{LN}\left(\mathbf{z}\right)\right)\end{aligned}$$

Here, $\mathbf{z} \in \mathbb{R}^{N\times D}$ represents the latent token, $\mathbf{u} \in \mathbb{R}^{N\times 3D}$ is the input injection, LN, FFN, and Attention stand for Layer Normalization (Ba et al., 2016), a 2-layer Feed-Forward Network with a hidden dimension of size $D \times E$, and an attention (Vaswani et al., 2017) layer with an injection interface, respectively.

Injection Interface. For blocks in the injection transformer, $\mathbf{u}$ is equal to the class embedding token $\mathbf{c} \in \mathbb{R}^{1\times 3D}$ for conditional image generation, i.e., $\mathbf{u} = \mathbf{c}$ for conditional models, and $\mathbf{u} = \mathbf{0}$ otherwise. In contrast, for blocks in the equilibrium transformer, $\mathbf{u}$ is the broadcast sum of noise injection $\mathbf{n} \in \mathbb{R}^{N\times 3D}$ and class embedding token $\mathbf{c} \in \mathbb{R}^{1\times 3D}$, i.e., $\mathbf{u} = \mathbf{n} + \mathbf{c}$ for conditional models and $\mathbf{u} = \mathbf{n}$ otherwise.

We modify the standard transformer attention layer to incorporate an additive injection interface before the query $\mathbf{q} \in \mathbb{R}^{N\times D}$, key $\mathbf{k} \in \mathbb{R}^{N\times D}$, and value $\mathbf{v} \in \mathbb{R}^{N\times D}$,

$$\begin{aligned}\mathbf{q}, \mathbf{k}, \mathbf{v} &= \mathbf{z}\mathbf{W}_i + \mathbf{u}\\ \mathbf{z} &= \text{MHA}\left(\mathbf{q}, \mathbf{k}, \mathbf{v}\right)\\ \mathbf{z} &= \mathbf{z}\mathbf{W}_o\end{aligned}$$

where $\mathbf{W}_i \in \mathbb{R}^{D\times 3D}$, $\mathbf{W}_o \in \mathbb{R}^{D\times D}$. The injection interface enables interactions between the latent tokens and the input injection in the multi-head dot-product attention (MHA) operation,

$$\begin{aligned}\mathbf{q}\mathbf{k}^\top &= (\mathbf{z}\mathbf{W}_q + \mathbf{u}_q)(\mathbf{z}\mathbf{W}_k + \mathbf{u}_k)^\top\\ &= \mathbf{z}\mathbf{W}_q\mathbf{W}_k^\top\mathbf{z}^\top + \mathbf{z}\mathbf{W}_q\mathbf{u}_k^\top + \mathbf{u}_q\mathbf{W}_k^\top\mathbf{z}^\top + \mathbf{u}_q^\top\mathbf{u}_k,\end{aligned}$$

where $\mathbf{W}_q, \mathbf{W}_k \in \mathbb{R}^{D\times D}$ are slices from $\mathbf{W}_i$, and $\mathbf{u}_q, \mathbf{u}_k \in \mathbb{R}^{N\times D}$ are slices from $\mathbf{u}$. This scheme adds no more computational cost compared to the standard MHA operation, yet it achieves a similar effect as cross-attention and offers good stability during training.

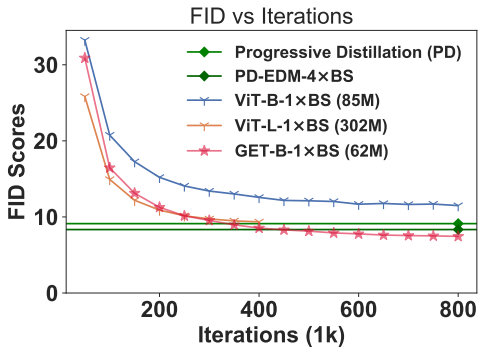

Figure 2. **Data Efficiency and Parameter Efficiency of GET**: GET outperforms PD and a 5× larger ViT in fewer iterations, yielding better FID scores.

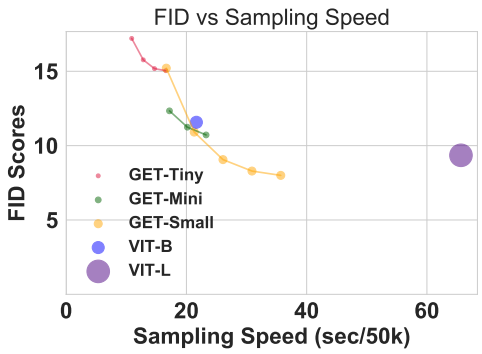

Figure 3. **Sampling speed of GET**: GET can sample faster than large ViTs, while achieving better FID scores. The size of each individual circle is proportional to the model size. For GETs, we vary the number of iterations in the Equilibrium transformer.

Image Decoder. The output of the GET-DEQ is first normalized with Layer Normalization (Ba et al., 2016). The normalized output is then passed through another linear layer to generate patches $\bar{\mathbf{p}} \in \mathbb{R}^{N\times D}$. The resulting patches $\bar{\mathbf{p}}$ are rearranged back to the resolution of the input noise $\mathbf{e}$ to produce the image sample $\tilde{\mathbf{x}} \in \mathbb{R}^{H\times W\times C}$.

3.1. Experiment Results

4. Experiments

We evaluate the effectiveness of our proposed Generative Equilibrium Transformer (GET) in the offline distillation of diffusion models on single-step class-conditional and unconditional image generation. Here, we use "single-step" to refer to the use of a single model evaluation while generating samples. We train and evaluate ViTs and GETs of varying scales on these tasks. GETs exhibit substantial data and parameter efficiency in offline distillation compared to the strong ViT baseline. Note that owing to the computational resources required to fully evaluate models, we report all our results on CIFAR-10 (Krizhevsky, 2009); extensions

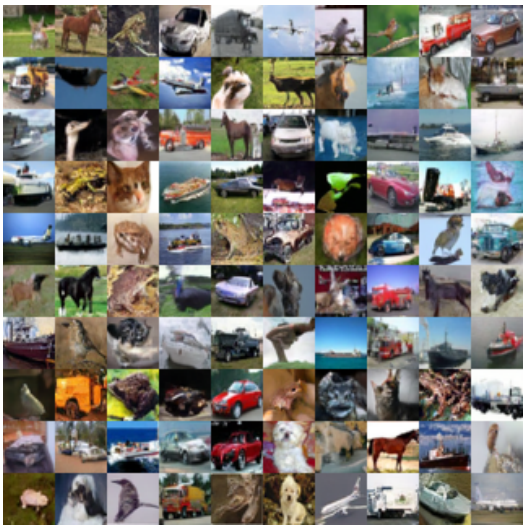

Figure 4. CIFAR-10 image samples from unconditional GET.

Figure 5. CIFAR-10 image samples from class-conditional GET. Each row corresponds to a class in CIFAR-10.

to the ImageNet-scale (Deng et al., 2009) are possible, but would require substantially larger GPU resources.

4.1. Experiment setup

In this section, we describe our offline distillation procedure and summarize our evaluation metrics. For a detailed description of our data collection process, network configs, and training specifics, please refer to the Appendix C.

Offline Distillation. We distill a pretrained EDM (Karras et al., 2022) into ViTs and GETs by training on a dataset $\mathcal{D}$ with noise/image pairs sampled from the teacher diffusion model using a reconstruction loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{e},\mathbf{x}\sim\mathcal{D}}\|\mathbf{x} - G_\theta(\mathbf{e})\|_1$$

Table 1. Performance on unconditional CIFAR-10

Method	NFE ↓	FID ↓	IS ↑
Diffusion Models			
DDPM (Ho et al., 2020)	1000	3.17	9.46
DDIM (Song et al., 2021a)	1000	4.04	-
Score SDE (Song et al., 2021b)	2000	2.2	9.89
DDIM (Song et al., 2021a)	10	13.36	-
LSGM (Vahdat et al., 2021)	147	2.10	
FastDPM (Kong and Ping, 2021)	10	9.9	
DPM-solver (Lu et al., 2022)	10	4.7	-
DEIS (Zhang and Chen, 2023)	10	4.17	-
EDM (Karras et al., 2022)	35	2.04	9.84
Continuous Flows			
2-ReFlow(+Distill) (Liu et al., 2023)	1	12.21 (4.85)	8.08 (9.01)
3-ReFlow(+Distill) (Liu et al., 2023)	1	8.15 (5.21)	8.47 (8.79)
Flow Matching (Diffusion) (Lipman et al., 2023)	183	8.06	-
Flow Matching (OT) (Lipman et al., 2023)	142	6.35	-
PFGM (Xu et al., 2022)	110	2.35	9.68
GANs			
StyleGAN2 (Karras et al., 2020b)	1	8.32	9.18
StyleGAN-XL (Sauer et al., 2022)	1	1.85	-
Diffusion Distillation			
KD (Luhman and Luhman, 2021)	1	9.36	8.36
PD (Salimans and Ho, 2022)	1	9.12	-
DFNO (Zheng et al., 2022)	1	4.12	-
TRACT-EDM (Berthelot et al., 2023)	1	4.17	-
TRACT (Berthelot et al., 2023)	1	5.02	-
PD-EDM (Salimans and Ho, 2022; Song et al., 2023)	1	8.34	8.69
CD (Song et al., 2023)	1	3.55	9.48
Consistency Models			
CT (Song et al., 2023)	1	8.70	8.49
CT (Song et al., 2023)	2	5.83	8.85
Ours			
GET-Base	1	6.91	9.16

where $\mathbf{x}$ is the desired ground truth image, $G_\theta(\cdot)$ is unconditional ViT/GET with parameters θ, and $\mathbf{e}$ is the initial Gaussian noise. To train a class-conditional GET, we also use class labels $\mathbf{y}$ in addition to noise/image pairs:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{e},\mathbf{y},\mathbf{x}\sim\mathcal{D}}\|\mathbf{x} - G^c_\theta(\mathbf{e},\mathbf{y})\|_1$$

where $G^c_\theta(\cdot)$ is class-conditional ViT/GET with parameters θ. As is the standard practice, we also maintain an exponential moving average (EMA) of weights of the model, which in turn is used at inference time for sampling.

Table 2. Generative performance on class-conditional CIFAR-10. w indicates the level of classifier guidance.

Method	NFE ↓	FID ↓	IS ↑
GANs			
BigGAN (Brock et al., 2018)	1	14.73	9.22
StyleGAN2-ADA (Karras et al., 2020a)	1	2.42	10.14
Diffusion Models			
DDIM (Meng et al., 2022)	2048	2.73	9.66
EDM (Karras et al., 2022)	35	1.79	-
NCSN++-G (Chao et al., 2022)	2000	2.25	-
EDM-G++ (Kim et al., 2022)	35	1.64	-
Diffusion Distillation			
Guided Distillation ($w = 0$) (Meng et al., 2022)	1	8.34	8.63
Guided Distillation ($w = 0.3$) (Meng et al., 2022)	1	7.34	8.90
Guided Distillation ($w = 1$) (Meng et al., 2022)	1	8.62	9.21
Guided Distillation ($w = 2$) (Meng et al., 2022)	1	13.23	9.23
Ours			
GET-Base	1	6.25	9.40

Table 3. Comparison of relevant training and hyperparameter settings for common distillation techniques. GET requires neither multiple training phases nor any trajectory information. We only count the number of models involved in the forward pass and exclude EMA in #Models. † indicates offline distillation techniques. ▲For CD, we count the VGG network used in the perceptual loss (Zhang et al., 2018).

Model	FID ↓	IS ↑	BS	Training Phases	#Models	Trajectory	Teacher
KD (Luhman and Luhman, 2021)†	9.36	-	4×	1	1	✗	DDIM
PD (Salimans and Ho, 2022)	9.12	-	1×	$\log_2(T)$	2	✓	DDIM
DFNO (Zheng et al., 2022)†	4.12	-	2×	1	1	✓	DDIM
TRACT (Berthelot et al., 2023)	14.40	-	2×	1	1	✓	DDIM
TRACT (Berthelot et al., 2023)	4.17	-	2×	2	1	✓	EDM
PD-EDM (Salimans and Ho, 2022; Song et al., 2023)	8.34	8.69	4×	$\log_2(T)$	2	✓	EDM
CD▲ (Song et al., 2023)	3.55	9.48	4×	1	3	✓	EDM
Ours†	7.42	9.16	1×	1	1	✗	EDM
Ours†	6.91	9.16	2×	1	1	✗	EDM
Guided Distillation (Meng et al., 2022)	7.34	8.90	4×	$\log_2(T)+1$	3	✓	DDIM
Ours†	6.25	9.40	2×	1	1	✗	EDM

Efficiency. Models trained with offline distillation require high data efficiency to make optimal use of limited training data sampled from pretrained diffusion models. In Figure 2, we observe that even with a fixed and limited offline data budget of 1M samples, GET achieves parity with online distilled EDM (Karras et al., 2022; Salimans and Ho, 2022; Song et al., 2023) while using only half the number of training iterations. For comparison, PD, TRACT, and CM use a much larger data budget of 96M, 256M, and 409.6M samples, respectively. Moreover, GET is able to match the FID score of a 5× large ViT, suggesting substantial parameter efficiency.

Sampling Speed. Figure 3 illustrates the sampling speed of both ViT and GET. A smaller GET (37.2M) can achieve faster sampling than a larger ViT (302.6M) while achieving lower FID scores.

One-Step Image Generation. We provide results for unconditional and class-conditional image generation on CIFAR-10 in Table 1 and Table 2, respectively. GET outperforms a much more complex distillation procedure—PD with classifier-free guidance—in class-conditional image generation. GET also outperforms PD and KD in terms of FID score for unconditional image generation. This effectiveness is intriguing, given that our approach for offline distillation is relatively simpler when compared to other state-of-the-art distillation techniques. In Table 3, we have outlined key differences in distillation techniques.

Qualitative results. We visualize uncurated CIFAR-10 (Krizhevsky, 2009) samples generated by GET in Figure 4 and Figure 5. GET can learn rich semantics and world knowledge from the dataset. For instance, GET has learned the symmetric layout of dog faces solely using reconstruction loss in the pixel space, as shown in Figure 5.

Scaling Model Size. We conduct extensive experiments to understand the trends of sample quality as we scale the model size of GET. Table 8 provides a summary of our findings on single-step unconditional image generation. We find that even small GET models with 10-20M parameters can generate images with sample quality on par with NAS-derived AutoGAN (Gong et al., 2019).

5. Conclusion and Limitations

We propose a simple yet effective approach to distill diffusion models into generative models capable of sampling with just a single model evaluation. Our method involves training a Generative Equilibrium Transformer (GET) architecture directly on noise/image pairs generated from a pretrained diffusion model, eliminating the need for trajectory information and temporal embedding. As our method for offline distillation relies on deterministic samplers to ensure a unique mapping between initial noise $\mathbf{e}$ and image $\mathbf{x}$, it cannot be directly applied to stochastic samplers which do not satisfy this requirement. However, this limitation also applies to many other distillation techniques, as they cannot maintain their fidelity under stochastic trajectories (Luhman and Luhman, 2021; Salimans and Ho, 2022; Berthelot et al., 2023). Overall, we find that GET demonstrates superior performance over more complex online distillation techniques such as progressive distillation (Salimans and Ho, 2022; Meng et al., 2022) in both class-conditional and unconditional settings. In addition, a small GET can generate higher quality images than a 5× larger ViT, sampling faster while using less training memory and fewer compute FLOPs, demonstrating its effectiveness.

6. Acknowledgements

Zhengyang Geng and Ashwini Pokle are supported by grants from the Bosch Center for Artificial Intelligence.

References

Cem Anil, Ashwini Pokle, Kaiqu Liang, Johannes Treutlein, Yuhuai Wu, Shaojie Bai, J Zico Kolter, and Roger B Grosse. Path independent equilibrium models can better exploit test-time computation. *Advances in Neural Information Processing Systems*, 35:7796–7809, 2022. 12

Uri M Ascher and Linda R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam, 1998. 11, 12

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3, 12

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 12

Shaojie Bai, Vladlen Koltun, and J Zico Kolter. Multiscale deep equilibrium models. *Advances in Neural Information Processing Systems*, 33:5238–5250, 2020. 12

Shaojie Bai, Vladlen Koltun, and J. Zico Kolter. Stabilizing Equilibrium Models by Jacobian Regularization. In *International Conference on Machine Learning (ICML)*, 2021. 12

Shaojie Bai, Zhengyang Geng, Yash Savani, and J Zico Kolter. Deep equilibrium optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 620–630, 2022. 12, 13

Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. *arXiv preprint arXiv:2206.07309*, 2022a. 11

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022b. 11

David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation, 2023. 1, 4, 5, 11

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 4, 12

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 11, 13

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 11

Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, Yi-Chen Lo, Chia-Che Chang, Yu-Lun Liu, Yu-Lin Chang, Chia-Ping Chen, and Chun-Yi Lee. Denoising likelihood score matching for conditional score-based data generation. *arXiv preprint arXiv:2203.14206*, 2022. 4

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021. 11

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 13

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. 4

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 11

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Neural Information Processing Systems (NeurIPS)*, 2021. 11, 12

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021. 11

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. GENIE: Higher-Order Denoising Diffusion Solvers. In *Advances in Neural Information Processing Systems*, 2022. 11

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 3, 11, 12, 13

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. 11

Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 5, 14

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 12

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. 13

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*, 2020. 1, 4, 11, 12

Jonathan Ho, Tim Salimans, Alexey A Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Neural Information Processing Systems (NeurIPS)*, 2022. 1

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 13, 14

Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International Conference on Machine Learning (ICML)*, 2021. 11

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 1

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021. 11

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. 12

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020a. 4

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, 2020b. 4

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Neural Information Processing Systems (NeurIPS)*, 2022. 4, 5, 11, 12, 13

Dongjun Kim, Yeongmin Kim, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv preprint arXiv:2211.17091*, 2022. 4

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Neural Information Processing Systems (NeurIPS)*, 2021. 11

Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021. 4, 11

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. 1

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 3, 5, 12

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019. 11

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 11

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 11

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 4

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022. 11

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 4

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 11

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 11

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 13

Cheng Lu, Jianfei Chen, Chongxuan Li, Qiuhao Wang, and Jun Zhu. Implicit normalizing flows. *arXiv preprint arXiv:2103.09527*, 2021. 12

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022. 4, 11

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021. 4, 5, 11

Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 1

Chenlin Meng, Ruiqi Gao, Diederik P Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022. 1, 4, 5, 11

Paul Micaelli, Arash Vahdat, Hongxu Yin, Jan Kautz, and Pavlo Molchanov. Recurrence without recurrence: Stable video landmark detection with deep equilibrium models. *arXiv preprint arXiv:2304.00600*, 2023. 12

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning (ICML)*, 2021. 11

Alexander Quinn 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. In *International Conference on Machine Learning (ICML)*, 2022. 1

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 13

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2020. 11

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-performance Deep Learning Library. In *Neural Information Processing Systems (NeurIPS)*, 2019. 12, 14

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 1, 3, 11, 12, 13, 14

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018. 12

Ashwini Pokle, Zhengyang Geng, and J Zico Kolter. Deep equilibrium approaches to diffusion models. In *Neural Information Processing Systems (NeurIPS)*, 2022. 12

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 11

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 11

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 11

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. 1, 11

Max Revay, Ruigang Wang, and Ian R Manchester. Lipschitz bounded equilibrium networks. *arXiv preprint arXiv:2010.01732*, 2020. 12

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J Liu, Sharan Narang, Wei Li, and Yanqi Zhou. Exploring the limits of transfer learning with a unified text-to-text transformer. 2019. 11

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. 1, 12

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. *Neural Information Processing Systems (NeurIPS)*, 2022. 1

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 1, 4, 5, 11, 13

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. 13

Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*, 2021. 11

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 4

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015. 1, 11

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021a. 4, 11, 12

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Neural Information Processing Systems (NeurIPS)*, 2019. 1, 11

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021b. 1, 4, 11

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 1, 4, 5, 11

Russell Tsuchida and Cheng Soon Ong. Deep equilibrium models as estimators for continuous latent variables. In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1671. PMLR, 2023. 12

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021. 4

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Neural Information Processing Systems (NeurIPS)*, 2017. 2, 3, 11, 12, 13

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective works best for zero-shot generalization? In *International Conference on Machine Learning*, pages 22964–22984. PMLR, 2022. 11

Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021. 11

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 11

Ezra Winston and J Zico Kolter. Monotone operator equilibrium networks. *Advances in neural information processing systems*, 33:10718–10728, 2020. 12

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021. 11

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning (ICML)*, 2020. 3, 12

Yilun Xu, Ziming Liu, Max Tegmark, and Tommi S. Jaakkola. Poisson flow generative models. In *Neural Information Processing Systems (NeurIPS)*, 2022. 4

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. 4, 11

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. 5

Sai Zhang, Liangjia Zhu, and Yi Gao. An efficient deep equilibrium model for medical image segmentation. *Computers in Biology and Medicine*, 148:105831, 2022. 12

Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. *arXiv preprint arXiv:2211.13449*, 2022. 1, 4, 5, 11

Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. In *The Eleventh International Conference on Learning Representations*, 2023. 11

A. Background and Related Work

Diffusion Models. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021a; Kingma et al., 2021; Dhariwal and Nichol, 2021) or score-based generative models (Song and Ermon, 2019; Song et al., 2021b) progressively perturb images with an increasing amount of Gaussian noise and then reverse this process through sequential denoising to generate images. Specifically, consider a dataset of i.i.d. samples p_{data}, then the diffusion process $\{\mathbf{x}(t)\}_{t=0}^{T}$ for $t \in [0, T]$ is given by an Itô SDE (Song et al., 2021b):

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \tag{7}$$

where $\mathbf{w}$ is the standard Wiener process, $\mathbf{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is the drifting coefficient, $g(\cdot) : \mathbb{R} \to \mathbb{R}$ is the diffusion coefficient, and $\mathbf{x}(0) \sim p_{\text{data}}$ and $\mathbf{x}(T) \sim \mathcal{N}(0, I)$. All diffusion processes have a corresponding deterministic process known as the probability flow ODE (PF-ODE) (Song et al., 2021b) whose trajectories share the same marginal probability densities as the SDE. This ODE can be written as:

$$\mathrm{d}\mathbf{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}, \sigma(t))\mathrm{d}t \tag{8}$$

where $\sigma(t)$ is the noise schedule of diffusion process, and $\nabla_{\mathbf{x}} \log p(\mathbf{x}, \sigma(t))$ represents the score function. Karras et al. (2022) show that the optimal choice of $\sigma(t)$ in Eq. (8) is $\sigma(t) = t$. Thus, the PF-ODE can be simplified to $\mathrm{d}\mathbf{x}/\mathrm{d}t = -t\nabla_{\mathbf{x}} \log p(\mathbf{x}, \sigma(t)) = (\mathbf{x} - D_\theta(\mathbf{x}; t))/t$, where $D_\theta(\cdot, t)$ is a denoiser function parametrized with a neural network that minimizes the expected L_2 denoising error for samples drawn from p_{data}. Samples can be efficiently generated from this ODE through numerical methods like Euler's method, Runge-Kutta method, and Heun's second-order solver (Ascher and Petzold, 1998).

Distillation techniques for diffusion models. Knowledge distillation (KD) (Luhman and Luhman, 2021) proposed to distill a multi-step DDIM (Song et al., 2021a) sampler into a single-step sampler by training the student model on 1M synthetic image samples. There are several key differences between KD and our work: GET does not employ any temporal embedding and predicts images instead of noise. Further, GET is built upon ViT, unlike the UNet in KD. Additionally, we demonstrate the effectiveness of our approach on both unconditional and class-conditional image generation.

Progressive distillation (PD) (Salimans and Ho, 2022) proposes a strategy for online distillation to distill a T-step teacher DDIM (Song et al., 2021a) diffusion model into a new $T/2$ step student DDIM model, repeating this process until one-step models are achieved. Transitive closure time-distillation (TRACT) (Berthelot et al., 2023) generalizes PD to distill $N > 2$ steps together at once, reducing the overall number of training phases. Consistency models (Song et al., 2023) achieve online distillation in a single pass by taking advantage of a carefully designed teacher and distillation loss objective.

Diffusion Fourier neural operator (DFNO) (Zheng et al., 2022) maps the initial Gaussian distribution to the solution trajectory of the reverse diffusion process by inserting the temporal Fourier integral operators in the pretrained U-Net backbone. Meng et al. (2022) propose a two-stage approach to distill classifier-free guided diffusion models into few-step generative models by first distilling a combined conditional and unconditional model, and then progressively distilling the resulting model for faster generation.

Fast sampling methods for diffusion models. While distillation is a predominant approach to speed up sampling rate of existing diffusion models, there are alternate lines of work to reduce the length of sampling chains by considering alternate formulations of diffusion model (Song et al., 2021a; Karras et al., 2022; Watson et al., 2021; Song et al., 2021b; Kong and Ping, 2021), correcting bias and truncation errors in the denoising process (Bao et al., 2022b; San-Roman et al., 2021; Bao et al., 2022a), and through training-free fast samplers at inference (Kong and Ping, 2021; Lu et al., 2022; Zhang and Chen, 2023; Dockhorn et al., 2022; Jolicoeur-Martineau et al., 2021; Liu et al., 2022). Several works like DDIM (Song et al., 2021a), Improved DDPM (Nichol and Dhariwal, 2021), FastDPM (Kong and Ping, 2021), SGM-CLD (Dockhorn et al., 2021), EDM (Karras et al., 2022) modify or optimize the forward diffusion process so that the denoising process can be made more efficient. DPM-Solver (Lu et al., 2022), and GENIE (Dockhorn et al., 2022) are higher-order ODE solvers that generate samples in few steps. There are also works that combine diffusion models with other families of generative models for faster sampling (Xiao et al., 2021; Zheng et al., 2023).

Transformers. Transformers were first proposed by Vaswani et al. (2017) for machine translation and since then have been widely applied in many domains like natural language processing (Devlin et al., 2019; Radford et al., 2018; Roberts et al., 2019; Lewis et al., 2019), reinforcement learning (Parisotto et al., 2020; Chen et al., 2021), self-supervised learning (Caron et al., 2021), vision (Dosovitskiy et al., 2021; Liu et al., 2021), and generative modeling (Hudson and Zitnick, 2021; Ramesh et al., 2022; Peebles and Xie, 2022; Esser et al., 2021). Many design paradigms for architectures of transformers have emerged over years. Notable ones include encoder-only (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019), decoder-only (Radford et al., 2018; 2019; Brown et al., 2020; Wang et al., 2022; Wei et al., 2021), and encoder-decoder architectures (Vaswani et al., 2017; Raffel et al., 2020; Lample and Conneau, 2019). We are interested in scalable transformer-based architectures

for generative modeling. Most relevant to this work are two encoder-only transformer architectures: Vision Transformer (ViT) (Dosovitskiy et al., 2021) and Diffusion Transformer (DiT) (Peebles and Xie, 2022). Vision Transformer (ViT) closely follows the original transformer architecture. It first converts 2D images into patches which are flattened and projected onto an embedding space. Positional encodings are added to patch embeddings to retain positional information. This sequence of embedding vectors is fed into the standard transformer architecture. The resulting architecture is then trained on a downstream task. Diffusion Transformers (DiT) are based on ViT architecture and operate on sequences of patches of image that are projected onto a latent space through an image encoder (Rombach et al., 2022). In addition, DiTs adapt several architectural modifications that enable their use as a backbone for diffusion models and help them scale better to yield better generative models with increasing model size. Some of these architecture design choices include adaptive Layer Normalization (AdaLN) (Dhariwal and Nichol, 2021; Brock et al., 2018; Perez et al., 2018; Karras et al., 2019), zero-initializing the final convolutional layer in each DiT block (Goyal et al., 2017), and use of timestep embedding (Ho et al., 2020; Song et al., 2021a).

Deep equilibrium models. Deep Equilibrium models (DEQs) (Bai et al., 2019) solve for a fixed point in the forward pass. Specifically, given an input $\mathbf{x}$ and a layer or a block f_θ, DEQ approximates an infinite-depth representation of f_θ by solving for its fixed point $z^\star$: $z^\star = f_\theta(z^\star; \mathbf{x})$. For the backward pass, one can differentiate analytically through $z^\star$ by the implicit function theorem. DEQs do not have any convergence guarantees and can be highly unstable to train (Bai et al., 2021). As a result, recent efforts focus on addressing these issues by designing variants of DEQs with provable guarantees (Winston and Kolter, 2020; Revay et al., 2020), or through optimization techniques such as Jacobian regularization (Bai et al., 2021), and fixed-point correction (Bai et al., 2022). DEQs have been successfully applied on a wide range of tasks such as image classification (Bai et al., 2020), semantic segmentation (Bai et al., 2020; Zhang et al., 2022), optical flow estimation (Bai et al., 2022), landmark detection (Micaelli et al., 2023), out-of-distribution generalization (Anil et al., 2022), language modelling (Bai et al., 2019), unsupervised learning (Tsuchida and Ong, 2023), and generative modelling (Lu et al., 2021; Pokle et al., 2022).

B. Addition details of GET architecture

Details of injection interface in transformer blocks. First, we reiterate the design of the transformer block used in GET. The transformer block is built upon the standard Pre-LN transformer block (Xiong et al., 2020; Dosovitskiy et al., 2021; Peebles and Xie, 2022), as shown below:

$$\begin{aligned}\mathbf{z} &= \mathbf{z} + \text{Attention}\left(\text{LN}\left(\mathbf{z}\right), \mathbf{u}\right)\\ \mathbf{z} &= \mathbf{z} + \text{FFN}\left(\text{LN}\left(\mathbf{z}\right)\right)\end{aligned}$$

Here, $\mathbf{z} \in \mathbb{R}^{N\times D}$ represents the latent token, $\mathbf{u} \in \mathbb{R}^{N\times 3D}$ is the input injection, LN, FFN, and Attention stand for Layer Normalization (Ba et al., 2016), a 2-layer Feed-Forward Network with a hidden dimension of size $D \times E$, and an attention (Vaswani et al., 2017) layer with an injection interface, respectively.

For blocks in the injection transformer, $\mathbf{u}$ is equal to the class embedding token $\mathbf{c} \in \mathbb{R}^{1\times 3D}$ for conditional image generation, i.e., $\mathbf{u} = \mathbf{c}$ for conditional models, and $\mathbf{u} = \mathbf{0}$ otherwise. In contrast, for blocks in the equilibrium transformer, $\mathbf{u}$ is the broadcast sum of noise injection $\mathbf{n} \in \mathbb{R}^{N\times 3D}$ and class embedding token $\mathbf{c} \in \mathbb{R}^{1\times 3D}$, i.e., $\mathbf{u} = \mathbf{n} + \mathbf{c}$ for conditional models and $\mathbf{u} = \mathbf{n}$ otherwise.

We modify the standard transformer attention layer to incorporate an additive injection interface before the query $\mathbf{q} \in \mathbb{R}^{N\times D}$, key $\mathbf{k} \in \mathbb{R}^{N\times D}$, and value $\mathbf{v} \in \mathbb{R}^{N\times D}$,

$$\begin{aligned}\mathbf{q}, \mathbf{k}, \mathbf{v} &= \mathbf{z}\mathbf{W}_i + \mathbf{u}\\ \mathbf{z} &= \text{MHA}\left(\mathbf{q}, \mathbf{k}, \mathbf{v}\right)\\ \mathbf{z} &= \mathbf{z}\mathbf{W}_o\end{aligned}$$

where $\mathbf{W}_i \in \mathbb{R}^{D\times 3D}$, $\mathbf{W}_o \in \mathbb{R}^{D\times D}$. The injection interface enables interactions between the latent tokens and the input injection in the multi-head dot-product attention (MHA) operation,

$$\begin{aligned}\mathbf{q}\mathbf{k}^\top &= (\mathbf{z}\mathbf{W}_q + \mathbf{u}_q)(\mathbf{z}\mathbf{W}_k + \mathbf{u}_k)^\top\\ &= \mathbf{z}\mathbf{W}_q\mathbf{W}_k^\top\mathbf{z}^\top + \mathbf{z}\mathbf{W}_q\mathbf{u}_k^\top + \mathbf{u}_q\mathbf{W}_k^\top\mathbf{z}^\top + \mathbf{u}_q^\top\mathbf{u}_k,\end{aligned}$$

where $\mathbf{W}_q, \mathbf{W}_k \in \mathbb{R}^{D\times D}$ are slices from $\mathbf{W}_i$, and $\mathbf{u}_q, \mathbf{u}_k \in \mathbb{R}^{N\times D}$ are slices from $\mathbf{u}$. This scheme adds no more computational cost compared to the standard MHA operation, yet it achieves a similar effect as cross-attention and offers good stability during training.

C. Experimental Setup

Data Collection. For unconditional image generation on CIFAR-10 (Krizhevsky, 2009), we generate 1M noise/image pairs from the pretrained unconditional EDM Karras et al. (2022). This dataset is denoted as `EDM-Uncond-1M`. As in EDM, we sample 1M images using Heun's second-order deterministic solver (Ascher and Petzold, 1998). Generating a batch of images takes 18 steps or 35 NFEs (Number of Function Evaluations). Overall, this dataset takes up around 29 GB of disk space. The entire process of data generation takes about *4 hours* on 4 NVIDIA A6000 GPUs using Pytorch (Paszke et al., 2019) Distributed Data Parallel (DDP) and a batch size of 128 per GPU. In addition

to unconditional image generation, we sample 1M noise-label/image pairs from the conditional VP-EDM Karras et al. (2022) using the same settings. This dataset is denoted as `EDM-Cond-1M`. Both the datasets will be released for future studies.

Training Details. We use AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 1e-4, a batch size of 128 (denoted as 1×BS), and 800k training iterations, which are identical to Progressive Distillation (PD) (Salimans and Ho, 2022). For conditional models, we adopt a batch size of 256 (2×BS). No warm-up, weight decay, or learning rate decay is applied. We convert input noise to patches of size 2×2. We use 6 steps of fixed point iterations in the forward pass of GET-DEQ with fixed point correction (Bai et al., 2022) and differentiate through it. For the $\mathcal{O}(1)$ memory mode, we utilize gradient checkpointing (Chen et al., 2016) for DEQ's computational graph. We set the EMA momentum to 0.9999 for all the models.

Evaluation Metrics. We measure image sample quality for all our experiments via Frechet inception distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016) computed on 50k images. We include other relevant metrics such as FLOPs, training speed, memory, sampling speed, and the Number of Function Evaluations (NFEs), wherever necessary.

Model Configuration. The configuration of different GET architectures are listed in Table 4. Here, L_i and L_e denote the number of transformer blocks in the Injection transformer and Equilibrium transformer, respectively. D denotes the width of the network. E corresponds to the expanding factor of the FFN layer in the Equilibrium transformer, which results in the hidden dimension of $E \times D$. For the injection transformer, we always adopt an expanding factor of 4.

Table 4. Details of configuration for GET architectures.

Model	Params	L_i	L_e	D	E
GET-Tiny	8.9M	6	3	256	6
GET-Mini	19.2M	6	3	384	6
GET-Small	37.2M	6	3	512	6
GET-Base	62.2M	1	3	768	12
GET-Base+	83.5M	6	3	768	8

Table 5. Details of configuration for ViT architectures.

Model	Params	L	D
ViT-B	85.2M	12	768
ViT-L	302.6M	24	1024

We have listed relevant model configuration details of ViT in Table 5. The model configurations are adopted from DiT (Peebles and Xie, 2022), whose effectiveness was tested for learning diffusion models. In this table, L denotes the number of transformer blocks in ViT. D stands for the width of the network. We always adopt an expanding factor of 4 following the common practice (Vaswani et al., 2017; Dosovitskiy et al., 2021; Peebles and Xie, 2022).

D. Additional Experiments

Class Conditioning. As both GET and ViT share the same class injection interface, we perform an ablation study on ViT. We consider two types of input injection schemes for class labels: 1) additive injection scheme 2) injection with adaptive layer normalization (AdaLN-Zero) as used in DiT (Peebles and Xie, 2022). We summarize the results in Table 6. Despite using almost the same parameters as unconditional ViT-B, the class-conditional ViT-B using additive injection interface has an FID of 12.43 at 200k, while the ViT-B w/ AdaLN-Zero class embedding (Peebles and Xie, 2022) set up an FID of 17.19 at 200k iterations. Another surprising observation is that ViT-B w/ AdaLN-Zero class embedding performs worse than unconditional ViT in terms of FID score. Therefore, it seems that adaptive layer normalization might not be useful when used only with class embedding.

Table 6. Ablation on class conditioning.

Model	FID↓	IS↑	Params↓
ViT-Uncond	15.20	8.27	85.2M
ViT-AdaLN-Zero	17.19	8.38	128.9M
ViT-Inj-Interface	12.43	8.69	85.2M

Why Scaling Laws for Implicit Models? As a prospective study, we preliminarily investigate the scaling properties of Deep Equilibrium models using GET. The scaling law is an attractive property, as it enables us to predict models' performance at extremely large compute based on the performance of tiny models. This predictive capability allows us to select the most efficient model given the constraints of available training budget (Brown et al., 2020; Hoffmann et al., 2022; OpenAI, 2023). While the scaling law for explicit networks has been extensively studied, its counterpart for implicit models remains largely unexplored. Implicit models are different from explicit models as they utilize more computation through weight-tying under similar parameters and model designs. Therefore, it is natural to question whether their scaling laws align with those of their explicit counterparts.

Scaling Model Size. We conduct extensive experiments to understand the trends of sample quality as we scale the

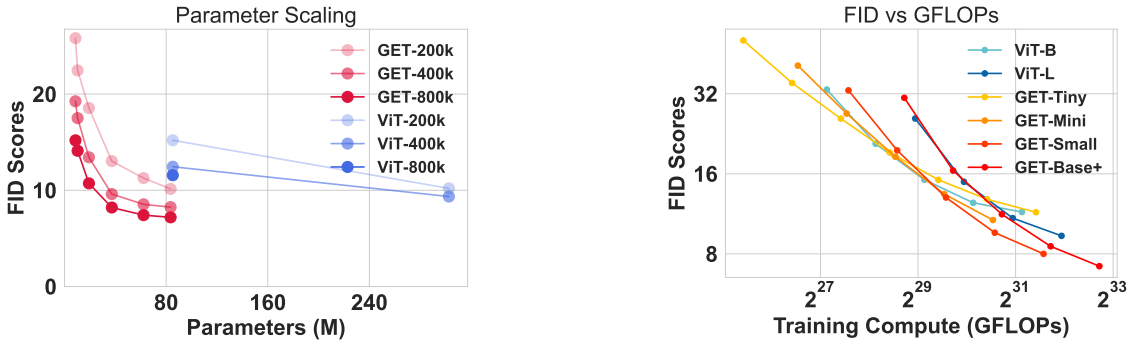

Figure 6. (a) (*Left*) Smaller GETs can achieve better FID scores than larger ViTs, demonstrating DEQ's parameter efficiency. Each curve in this plot connects models of different sizes within the same model family at identical training iterations, as indicated by the numbers after the model names in the legend. (b) (*Right*) **Compute efficiency of GET**: Larger GET models use training compute more efficiently. For a given GET, the training budget is calculated from training iterations. Refer to Table 8 for the exact size of GET models.

Table 7. Benchmarking GET against ViT on unconditional image generation on CIFAR-10. For the first time, implicit layers *strictly* surpass explicit networks in all metrics. Results are benchmarked on 4 A6000 GPUs using a batch size of 128, 800k iterations, and PyTorch (Paszke et al., 2019) distributed training protocol. Training Mem stands for training memory consumed per GPU. $\mathcal{O}(1)$ symbolizes the $\mathcal{O}(1)$ training memory mode, which differs only in training memory and speed.

Model	FID↓	IS↑	Params↓	FLOPs↓	Training Mem↓	Training Speed↑
ViT-Base	11.49	8.61	85.2M	23.0G	10.1GB	4.83 iter/sec
GET-Mini	10.72	8.69	19.2M	15.2G	9.2GB	5.79 iter/sec
GET-Mini-$\mathcal{O}(1)$	-	-	-	-	5.0GB	4.53 iter/sec

Table 8. Performance of GETs on unconditional CIFAR-10.

Models	Params	NFE ↓	FID ↓	IS ↑
GET-Tiny	8.9M	1	15.19	8.37
GET-Mini	19.2M	1	10.72	8.69
GET-Small	37.2M	1	8.00	9.03
GET-Base	62.2M	1	7.42	9.16
GET-Base+	83.5M	1	7.19	9.09
More Training				
GET-Tiny-4×Iters	8.9M	1	11.47	8.64
GET-Base-2×BS	62.2M	1	6.91	9.16

model size of GET. Table 8 provides a summary of our findings on single-step unconditional image generation. We find that even small GET models with 10-20M parameters can generate images with sample quality on par with NAS-derived AutoGAN (Gong et al., 2019). In general, sample quality improves with the increase in model size.

Scaling Training Compute. Our experimental results support the findings of Peebles and Xie (2022) for explicit models (DiT) and extend them to implicit models. Specifically, for both implicit and explicit models, larger models are better at exploiting training FLOPs. Figure 6 shows that larger models eventually outperform smaller models when the training compute increases. For implicit models, there also exists a "sweet spot" in terms of model size under a fixed training budget, *e.g.,* GET-Small outperforms both smaller and larger GETs at 2^{31} training GFLOPs. Furthermore, because of the internal dynamics of implicit models, they can match a much larger explicit model in terms of FLOPs and training compute while using fewer parameters. This underscores the potential of implicit models as candidates for compute-optimal models (Hoffmann et al., 2022) with significantly better parameter efficiency. For example, at 2^{31} training GFLOPs, Figure 6(b) suggests that one should choose GET-Small (31.2M) among implicit models for the best performance, which is much more parameter efficient and faster in sampling than the best-performing explicit model, ViT-L (302M), at this training budget.

Benchmarking GET against ViT. Table 7 summarizes key metrics for unconditional image generation for ViT and GET. Our experiments indicate that a smaller GET (19.2M) can generate higher-quality images faster than a much larger ViT (85.2M) while utilizing less training memory and fewer FLOPs. GET also demonstrates substantial parameter efficiency over ViTs as shown in Figure 6(b).