ON DISTILLING GENERATOR MATCHING MODELS

Shiv Shankar
University of Massachusetts

ABSTRACT

Generator Matching (GM) is a new framework which encompasses the current workhorse generative modeling methods. However GM suffers from the computationally intensive sampling process common to these ODE/SDE based models. We introduce "Implicit Generator Matching" (IGM), a general framework for one-step distillation of generator matching models. Our method generalizes the recently proposed one-step diffusion distillation (Zhou et al., 2024; Luo et al., 2024b) methods to Generator Matching. We present promising initial results on image generation.

1 INTRODUCTION

ODE/SDE-based generative models have revolutionized the generation of images (Rombach et al., 2022; Saharia et al., 2022; Podell et al., 2023), videos (Brooks et al., 2024; Gupta et al., 2024), and audio (Liu et al., 2023; Evans et al., 2024). At the core of this success lies the use of continuous-time processes that simulate the transformation from noise to data, such as diffusion models (Song et al., 2020; Ho et al., 2020) and flow matching (Peluchetti, 2022; Lipman et al., 2022). Researchers have further extended these methods to handle diverse data types, including discrete data (Campbell et al., 2022; Gat et al., 2024), graphs (Kong et al., 2023), manifolds (Huang et al., 2022; Chen and Lipman, 2024), and tabular data (Jolicoeur-Martineau et al., 2024).

While the training processes for these generative models vary—ranging from score matching (Song et al., 2020) and denoising diffusion (Ho et al., 2020) to flow matching (Lipman et al., 2022)—they share a common feature: the emulation of a Markovian process. Starting with an initial sample, these methods iteratively construct new samples by applying a functional transform that depends solely on the current sample. Recognizing this similarity, Holderrieth et al. (2024) unified these ideas into a single framework called Generator Matching (GM). GM provides a scalable, simulation-free approach to training parameterized approximations of generators for arbitrary Markov processes.

Despite its strengths, GM models inherit a key challenge from diffusion and flow matching methods: slow inference. Specifically, generating samples requires simulating an ODE (or SDE) using a numerical solver, where each step involves evaluating a deep neural network. Moreover, because the sample paths are non-linear, small step sizes are necessary for accurate simulation, as larger steps can lead to accumulating discretization errors (Song et al., 2023). Improving the sampling efficiency of these models is therefore critical for broadening their practical applications.

2 RELATED WORK

Researchers have proposed various approaches to accelerate sampling in diffusion and flow matching methods. A prominent family of techniques involves distribution distillation (Luo et al., 2024a; Salimans and Ho, 2022; Gu et al., 2023; Fan and Lee, 2023; Aiello et al., 2023), which aims directly match the output distribution of a fast (few-step) generative model with a pre-trained teacher diffusion model. One prominent example is the Score-Identity method (Zhou et al., 2024), which enables one-shot distillation of diffusion models. Recently,Luo et al. (2024b) extended the method of (Zhou et al., 2024), and achieved SoTA distillation results. However, these methods rely on the score-projection identity (Zhou et al., 2024; Vincent, 2011), limiting their applicability to score based generators.

Inspired by the distribution distillation methods (Zhou et al., 2024; Luo et al., 2024b; Huang et al., 2024), we propose a general framework called Implicit Generator Matching (IGM) for one-step distillation of any generator matching model. This framework extends the benefits of distillation

beyond diffusion models, offering a versatile solution for improving sampling efficiency across a broader range of generative frameworks.

3 BACKGROUND

3.1 GENERATOR MATCHING

Let $\mathbf{x}_t$ denote a set of time t indexed multivariate random variables. We denote by p_0 the target distribution for which we want to learn a generative model.

If $(\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_t, \mathbf{x}_{t+1}, ..)$ is a Markov process then $\mathbf{x}_{t+h}$ is independent of any variables $\mathbf{x}_{t-\delta}$ when conditioned on $\mathbf{x}_t$. A Markov process can be identified by its transition kernel, $(k_{t+h|t})$. From the transition kernel k one can obtain an operator $\mathcal{L}_t$, known as the generator defined as $\mathcal{L}_t := \frac{d}{dh}\big|_{h=0} k_{t+h|t} - k_{t|t}$ Under certain regularity assumptions, there is a direct correspondence between Markov processes and their generators (Ethier and Kurtz, 2009; Rüschendorf et al., 2016).

Holderrieth et al. (2024) propose a broad recipe for generative modelling of data based on the idea of generators of markov processes. Specifically, the current workhorses of many generative models, diffusion models (Ho et al., 2020) and flow matching (Lipman et al., 2022), can be cast into this framework. Holderrieth et al. (2024) propose learning parameterized generators $\mathcal{L}_{\theta,t}$ to match the generator $\mathcal{L}$ of the Markov process. by optimizing a generator matching loss $\mathcal{D}_{GM} = \mathbb{E}[D(F_t, F_{\theta,t})]$ where D is a Bregman divergence and F_t is a natural parameterization of $\mathcal{L}_t$. However the above objective is intractable without access to $\mathcal{L}_t / F_t$.

Inspiring from Lipman et al. (2022), Holderrieth et al. (2024) propose using a generator linearly parameterized by conditional generators viz $F_t(\mathbf{x}_t) = \mathbb{E}_{t, p_{\mathbf{x}_0|\mathbf{x}_t}}[F_t^{\mathbf{x}_0}(\mathbf{x}_t)]$, and show that the following conditional GM objective which uses conditional generators has the same minima as the GM objective.

$$\mathcal{D}_{CGM} = \mathbb{E}_{t, \mathbf{x}_t \sim p_t} D(F_t^{\mathbf{x}_1}(\mathbf{x}_t), F_{\theta,t}(\mathbf{x}_t))$$

Comparing to the standard flow matching problem, we see that F corresponds to the velocity field u and $F_\theta = v_\theta$ is a neural network used to parameterize the flow objective. Then the FM loss and the CFM loss correspond naturally to $\mathcal{D}_{GM}$ and $\mathcal{D}_{CGM}$ respectively. [1]

4 IMPLICIT GENERATOR MATCHING

Our goal is to train a model M_θ, which in one step maps a random noise $\epsilon \sim p_\epsilon$ to obtain a sample $\mathbf{x} = M_\theta(\epsilon)$. Let $p_{\theta,0}$ denote the distribution of the student model over the generated sample $\mathbf{x}$, and $p_{\theta,t}$ denote the marginal probability path transitioned with $k_{t|0}(.|\mathbf{x}_0)$, i.e.,

$$p_{\theta,t}(\mathbf{x}_t) = \int k_{t|0}(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,0}(\mathbf{x}_0) d\mathbf{x}_0$$

This marginal probability path implicitly defines a generator $F_{\theta,t}(\mathbf{x}_t)$. Further note, that with such a choice of p_θ, t, we do not need to consider how θ influences p_θ when differentiating any expectation over p_θ i.e. $\mathbb{E}_{p_{\theta,t}}$ as the reparameterization trick applies in this case(Kingma, 2013). Instead we can differentiate wrt θ the empirical expectations by differentiating through the samples $\mathbf{x}_t$ directly. Thus depending on context we may use $\mathbf{x}_t(\theta)$ to highlight this. We also denote by ρ_t the coupling induced by $k_{t|0}$ i.e. it is the joint distribution of x_t, x_0. Finally as is common in distillation literature, we will assume access to a pre-trained GM trained model F_t for the target data p_0. Note that we do not require access to samples from p_0.

We propose to minimize the Generator matching loss D_{GM} between the implicit generator $F_{\theta,t}$ and the pre-trained generator F_t, which writes

$$\mathcal{D}_{GM}(\theta) \coloneqq \mathbb{E}_{t, \mathbf{x}_t \sim p_{\theta,t}} D(F_t(\mathbf{x}_t), F_{\theta,t}(\mathbf{x}_t)) \tag{1}$$

It is clear to see that the $\mathcal{D}_{GM} = 0$ if and only if all induced generator are the same, i.e. $F_{\theta,t}(\mathbf{x}_t) = F_t(\mathbf{x}_t)$ with respect to the support of $p_{\theta,t}$. Unfortunately, minimizing objective (1) directly is intractable because we do not have direct access to the induced generator $F_{\theta,t}(\mathbf{x}_t)$.

[1] FM/CFM uses the L^2 loss which is a Bregman Divergence

4.1 TRACTABLE OBJECTIVE

Our goal is to optimize the parameter θ to minimize the objective (1). A natural option is to consider gradient based optimization. However, consider the gradient of the $\mathcal{D}_{GM}$ objective:

$$\begin{aligned}\frac{\partial}{\partial\theta}\mathcal{D}_{GM}(\theta) &= \frac{\partial}{\partial\theta}\mathbb{E}_{t,\mathbf{x}_t\sim p_{\theta,t}} D(F_t(\mathbf{x}_t), F_{\theta,t}(\mathbf{x}_t)) \\ &= \mathbb{E}_{t,\mathbf{x}_t\sim p_{\theta,t}}\left[\frac{\partial}{\partial\mathbf{x}_t} D(F_t(\mathbf{x}_t), F_{\theta,t}(\mathbf{x}_t))\frac{\partial\mathbf{x}_t(\theta)}{\partial\theta}\right] \\ &\quad + \mathbb{E}_{t,\mathbf{x}_t\sim p_{\theta,t}}\langle\frac{\partial}{\partial F_{\theta,t}} D(F_t(\mathbf{x}_t), F_{\theta,t}(\mathbf{x}_t)), \frac{\partial}{\partial\theta}F_{\theta,t}(\mathbf{x}_t)\rangle \end{aligned} \tag{2}$$

Remark 4.1. Note that here when we differentiated wrt θ the expectation $\mathbb{E}_{p_{\theta,t}}$, we were able to move the derivative inside the expected value because of the reparameterization trick(Kingma, 2013).

The direct optimization approach faces two primary obstacles in computing the gradient of the objective function: first, the need to evaluate $F_{\theta,t}$, and second, the need to evaluate its derivative with respect to θ. However, we do not have access to the generator corresponding to $p_{\theta,0}$. Recall that we only have the model M_θ instead which can generate samples from $p_{\theta,0}$, and the generator $F_{\theta,t}$ is implicit. This inherent limitation makes direct minimization of the objective intractable. Furthermore, even if we assume access to an oracle capable of evaluating $F_{\theta,t}$, the challenge of computing its derivative remains unresolved.

Next, we show however that we can replace the derivatives of F with an alternative that only uses oracle access to F. This is formalized in Theorem 4.2.

Theorem 4.2. *Under simple regularity conditions, we have for any smooth function $g(x_t, \theta)$, the generative model $p_\theta(\boldsymbol{x})$ and its generator $F_\theta(\boldsymbol{x})$*

$$\mathbb{E}_{\boldsymbol{x}_t\sim p_{\theta,t}}\langle g(\boldsymbol{x}_t,\theta), \frac{\partial}{\partial\theta}F_{\theta,t}(\boldsymbol{x}_t)\rangle = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{x}_t\sim\rho_t}\frac{\partial}{\partial\boldsymbol{x}_t}\langle g(\boldsymbol{x}_t,\theta), F_t^{\boldsymbol{x}_0}(\boldsymbol{x}_t) - F_{\theta,t}(\boldsymbol{x}_t)\rangle\frac{\partial\boldsymbol{x}_t}{\partial\theta} \tag{3}$$

$$+ \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{x}_t\sim\rho_t}\langle g(\boldsymbol{x}_t,\theta), \frac{\partial}{\partial\boldsymbol{x}_0}F_t^{\boldsymbol{x}_0}(\boldsymbol{x}_t)\rangle\frac{\partial\boldsymbol{x}_0}{\partial\theta} \tag{4}$$

The proof is in the Appendix.

We draw the readers attention to a key property of the above expression: any derivative wrt θ of the implicitly defined F_θ does not exist any more. Instead we are left with evaluation of the generator $F_{\theta,t}$, the conditional generators $F_t^{\mathbf{x}_0}$, and the derivative of the conditional generator $F_t^{\mathbf{x}_0}$ which is independent of θ. Thus in principle we can replace the $F_{\theta,t}$ on the right side with an oracle which can simply give the value of $F_{\theta,t}$ at any given point.

We then propose that instead of an oracle we can use a model $F_{\eta,t}$ that is trained to match $F_{\theta,t}$. This is relatively easy as $F_{\eta,t}$ can be obtained by simply optimizing the Generator matching loss $\mathcal{D}_{GM}$ using the generated samples $p_{\theta,0}$. This then gives the following objective function

$$\begin{aligned}\mathcal{D}_{IGM}(\theta,\eta) = \mathbb{E}_{t,\mathbf{x}_t\sim p_{\theta,t}}\underbrace{D(F_t^{\mathbf{x}_0}(\mathbf{x}_t), F_{\eta,t}(\mathbf{x}_t))}_{\mathcal{A}_1} + \mathbb{E}_{t,\mathbf{x}_t\sim p_{\theta,t}}\underbrace{D(F_t(\mathbf{x}_t), F_{sg(\eta),t}(\mathbf{x}_t))}_{\mathcal{A}_2} \\ + \mathbb{E}_{t,\mathbf{x}_t\sim p_{\theta,t}}\underbrace{\langle\frac{\partial}{\partial F_{\eta,t}}D(F_t, F_{sg(\eta),t}), F_t^{\mathbf{x}_0}(\mathbf{x}_t) - F_{sg(\eta),t}(\mathbf{x}_t)\rangle}_{\mathcal{A}_3}\end{aligned}$$

where sg refers to the stop gradient operator. sg is applied on η because we want η to only learn the induced generator $F_{\theta,t}$ via the standard generator matching loss.

Remark 4.3. Since $F_{\eta,t}$ is supposed to act as the oracle, it should be close $F_{\theta,t}$ before we optimize the terms $\mathcal{A}_{2,3}$. To achieve this we update η for K iterations where K is a hyperparameter, and then do one update of θ.

Figure 1: Samples from the best performing Jump + Flow IGM model

Method	FID ↓
DDPM (Ho et al., 2020)	3.17
VP-SDE (Song et al., 2020)	3.01
EDM (Karras et al., 2022)	1.98
Flow model (Holderrieth et al., 2024)	2.94
Jump model (Holderrieth et al., 2024)	4.23
Jump + Flow (Holderrieth et al., 2024)	2.49
Flow IGM (Our)	3.11
Jump IGM (Ours)	5.22
Jump + Flow IGM (Our)	2.79

Table 1: Experimental results for image generation on CIFAR-10. Euler integration was used for sampling the flow models with NFE=100. All our methods are one-shot generators (NFE=1)

5 EXPERIMENTS

Since our framework is supposed to work for arbitrary generator matching models, instead of working with regular diffusion and flow matching models, we instead focus on jump models, a new class of generative models enabled by Generator Matching.

Holderrieth et al. (2024) show that jump processes with rate kernel Q_t and transition kernel k_t given below satisfy the CondOT path used in Lipman et al. (2022) to connect a given target data with gaussian noise.

$$Q_t(\mathbf{x}';\mathbf{x}|\mathbf{x}_1) = \underbrace{\frac{[k_t(\mathbf{x})]_+}{(1-t)^3}}_{\lambda_t(\mathbf{x})} \underbrace{\frac{[-k_t(\mathbf{x}')]_+ p_t(\mathbf{x}'|\mathbf{x}_1)}{\int [-k_t(\tilde{\mathbf{x}})]_+ p_t(\tilde{\mathbf{x}}|\mathbf{x}_1) d\tilde{\mathbf{x}}}}_{J_t(\mathbf{x}';)}, \quad k_t(\mathbf{x}) = \mathbf{x}^2 - (t+1)\mathbf{x}\mathbf{x}_1 - (1-t)^2 + t\mathbf{x}_1^2 \tag{5}$$

The corresponding generative process can be trained with the following loss

$$\mathcal{D}_\theta = (\sum_{x' \neq x} Q_t^\theta(x';x) - Q_t(x';x|z) \log Q_t^\theta(x';x))$$

where $\log Q_t^\theta$ is the parameterized generator. For modeling Q Holderrieth et al. (2024) parameterized the rate λ, J seperately and combined them according to Equation (5). They then show that these models could be used to generate images. The jump process is parameterized by applying softmax on the output of a U-Net model with $d+1$ channels. Each channel follows its own independent process, however the parameters of the process is determined by all the channels combined.

We follow the same approach and train an initial model on the image data. Then we distill it using our IGM method, and compare generative performance. As is common with image data, the results are evaluated with FID (Heusel et al., 2017) metric. Results on CIFAR-10 are presented in Table 1, with some samples presented in Figure 1. We can see from the results that IGM models are in general close in quality with their teacher models while having and NFE=1. Moreover they can learn not only from a flow matching objective, but also other models like jump model and a combination of different generators.

6 CONCLUSION

We presented a novel framework for distilling generators of general Markov processes using the idea of Generator Matching (Holderrieth et al., 2024). Our framework generalizes the recent and promising score-distillation framework for diffusion models (Luo et al., 2024b); and applies simultaneously to flow matching, diffusion processes as well as jump processes. We show experimentally some promising results for image generation.

REFERENCES

Emanuele Aiello, Diego Valsesia, and Enrico Magli. Fast inference in denoising diffusion models via mmd finetuning. *ArXiv*, abs/2301.07969, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

Stewart N Ethier and Thomas G Kurtz. *Markov processes: characterization and convergence*. John Wiley & Sons, 2009.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. *arXiv preprint arXiv:2407.14358*, 2024.

Ying Fan and Kangwook Lee. Optimizing ddpm sampling with shortcut fine-tuning. *ArXiv*, abs/2301.13362, 2023.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *arXiv preprint arXiv:2407.15595*, 2024.

Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M. Susskind. Boot: Data-free distillation of denoising diffusion models with bootstrapping. *ArXiv*, abs/2306.05544, 2023.

Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*, pages 393–411. Springer, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky T. Q. Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. 2024.

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

Zemin Huang, Zhengyang Geng, Weijian Luo, and Guo jun Qi. Flow generator matching, 2024.

Alexia Jolicoeur-Martineau, Kilian Fatras, and Tal Kachman. Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees. 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B Aditya Prakash, and Chao Zhang. Autoregressive diffusion model for graph generation. In *International conference on machine learning*, pages 17391–17408. PMLR, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024a.

Weijian Luo, Zemin Huang, Zhengyang Geng, J Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. *arXiv preprint arXiv:2410.16794*, 2024b.

Stefano Peluchetti. Non-denoising forward-time diffusions. 2022. URL https://openreview.net/forum?id=oVfIKuhqfC.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

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.

Ludger Rüschendorf, Alexander Schnurr, and Viktor Wolf. Comparison of time-inhomogeneous markov processes. *Advances in Applied Probability*, 48(4):1015–1044, 2016.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

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*, 2020.

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

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *International Conference on Machine Learning*, 2024.

A PROOFS

Lemma A.1. *Under simple regularity conditions, for any function g we have the following:*

$$\mathbb{E}_{\boldsymbol{x}_t \sim p_{\theta,t}} \langle g, F_{\theta,t}(\boldsymbol{x}_t) \rangle = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t \sim \rho_t} \langle g, F_t^{\boldsymbol{x}_0}(\boldsymbol{x}_t) \rangle$$

Proof. By the definition of $p_{\theta,t}$ and $F_{\theta,t}$:

$$p_{\theta,t}(\mathbf{x}_t) = \int k_{t|0}(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,0}(\mathbf{x}_0) d\mathbf{x}_0 \tag{6}$$

$$F_{\theta,t}(\mathbf{x}_t) = \int F_t^{\mathbf{x}_0}(\mathbf{x}_t) p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \tag{7}$$

$$= \int F_t^{\mathbf{x}_0}(\mathbf{x}_t) \frac{k_{t|0}(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,0}(\mathbf{x}_0)}{p_{\theta,t}(\mathbf{x}_t)} d\mathbf{x}_0. \tag{8}$$

We have

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \langle g, F_{\theta,t}(\mathbf{x}_t) \rangle &= \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \langle g, \int F_t^{\mathbf{x}_0}(\mathbf{x}_t) \frac{k_{t|0}(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,0}(\mathbf{x}_0)}{p_{\theta,t}(\mathbf{x}_t)} d\mathbf{x}_0 \rangle \\
&= \int p_{\theta,t}(\mathbf{x}_t) \langle g, \int F_t^{\mathbf{x}_0}(\mathbf{x}_t) \frac{k_{t|0}(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,0}(\mathbf{x}_0)}{p_{\theta,t}(\mathbf{x}_t)} d\mathbf{x}_0 \rangle d\mathbf{x}_t \\
&= \int \int \langle g, F_t^{\mathbf{x}_0}(\mathbf{x}_t) k_{t|0}(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,0}(\mathbf{x}_0) d\mathbf{x}_0 \rangle d\mathbf{x}_t \\
&= \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \langle g, F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle
\end{aligned} \tag{9}$$

□

Note that this a more general form of the score projection identity (Zhou et al., 2024). One can obtain the score projection identity by plugging in the generator for diffusion model given by Holderrieth et al. (2024) into Lemma A.1.

By replacing g with $\partial_\theta g(\mathbf{x}_t, \theta)$ in (9), we also get that, for any differentiable θ dependent function $g(., \theta)$:

$$\mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \langle \frac{\partial}{\partial \theta} g(\mathbf{x}_t, \theta), F_{\theta,t}(\mathbf{x}_t) \rangle = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \langle \frac{\partial}{\partial \theta} g(\mathbf{x}_t, \theta), F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \tag{10}$$

A.1 PROOF OF THEOREM 4.2

Proof. Let us put $g = g(\mathbf{x}_t, \theta)$ in (9) and differentiate wrt θ. We get

$$\mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \left(\frac{\partial}{\partial \theta} \langle g(\mathbf{x}_t, \theta), F_{\theta,t}(\mathbf{x}_t) \rangle + \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \theta} F_{\theta,t}(\mathbf{x}_t) \rangle \right) + \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_{\theta,t}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta}$$

$$= \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \langle \frac{\partial}{\partial \theta} g(\mathbf{x}_t, \theta), F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle + \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta} \tag{11}$$

$$+ \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \mathbf{x}_0} F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_0}{\partial \theta}$$

$$\overset{a}{\Rightarrow} \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \left(\cancel{\langle \frac{\partial}{\partial \theta} g(\mathbf{x}_t, \theta), F_{\theta,t}(\mathbf{x}_t) \rangle} + \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \theta} F_{\theta,t}(\mathbf{x}_t) \rangle \right) + \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_{\theta,t}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta}$$

$$= \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \cancel{\langle \frac{\partial}{\partial \theta} g(\mathbf{x}_t, \theta), F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle} + \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta} \tag{12}$$

$$+ \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \mathbf{x}_0} F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_0}{\partial \theta}$$

$$\Rightarrow \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \theta} F_{\theta,t}(\mathbf{x}_t) \rangle + \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_{\theta,t}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta}$$

$$= \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta} + \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \mathbf{x}_0} F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_0}{\partial \theta} \tag{13}$$

$$\overset{b}{\Rightarrow} \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \theta} F_{\theta,t}(\mathbf{x}_t) \rangle + \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \underline{\frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_{\theta,t}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta}}$$

$$= \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta} + \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \mathbf{x}_0} F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_0}{\partial \theta} \tag{14}$$

$$\Rightarrow \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \theta} F_{\theta,t}(\mathbf{x}_t) \rangle$$

$$= \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \frac{\partial}{\partial \mathbf{x}_t} \langle g(\mathbf{x}_t, \theta), F_t^{\mathbf{x}_0}(\mathbf{x}_t) - F_{\theta,t}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_t}{\partial \theta} + \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \rho_t} \langle g(\mathbf{x}_t, \theta), \frac{\partial}{\partial \mathbf{x}_0} F_t^{\mathbf{x}_0}(\mathbf{x}_t) \rangle \frac{\partial \mathbf{x}_0}{\partial \theta} \tag{15}$$

Here in (a) we used the Equation 10 to cancel the indicated terms. In (b) we used the fact that underlined term is independent of $\mathbf{x}_0$ and so the expectation can be changed from only over $\mathbf{x}_t$ to the coupling ρ. □