Forward-Learned Discrete Diffusion
Learning how to noise to denoise faster

Grigory Bartosh, Teodora Pandeva, Sushrut Karmalkar, Javier Zazo

AMLAB Amsterdam Machine Learning Lab · Universiteit van Amsterdam · Microsoft Research

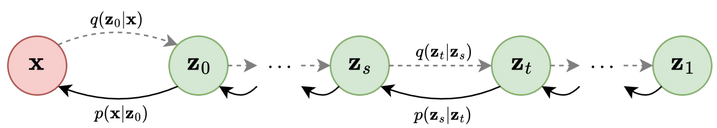
Abstract

Few-step discrete diffusion is hard: the fast factorized sampler must match a non-factorized target.

We **learn the noising process** to make the **target factorized** and **sample faster**.

Motivation

- Markovian forward process:



- Diffusion objective:

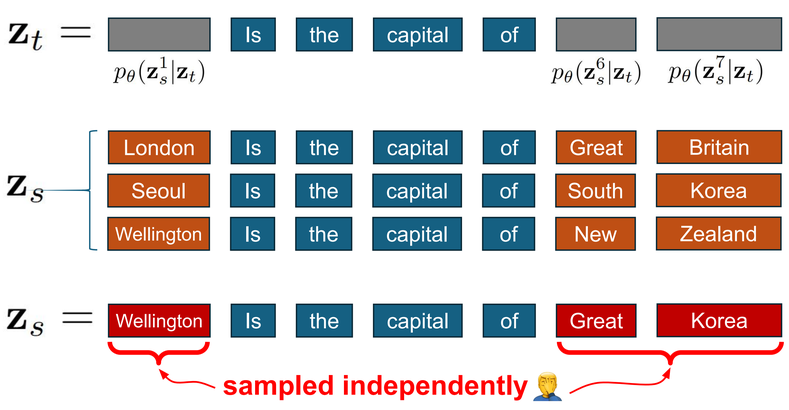
$$\mathbb{E}_{u(t)q(\mathbf{z}_t|\mathbf{x})}[\mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x})|p_\theta(\mathbf{z}_s|\mathbf{z}_t))]$$

- Target reverse process vs parameterisation:

$$q(\mathbf{z}_s|\mathbf{z}_t) = \int q(\mathbf{x}|\mathbf{z}_t)q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x})d\mathbf{x} \quad \approx \quad p_\theta(\mathbf{z}_s|\mathbf{z}_t) = \prod_{i=1}^{D} p_\theta(\mathbf{z}_s^i|\mathbf{z}_t)$$

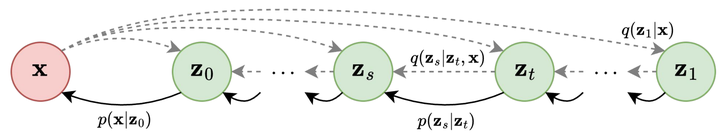
Non-Factorized 😱 Factorized 😰

- Problem with masked diffusion:



Method

- Non-Markovian forward process:



- Forward process:

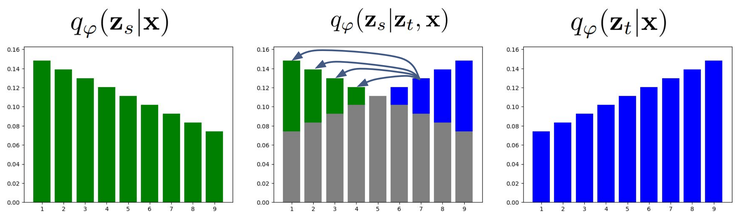
$$q_\varphi(\mathbf{z}_{0:1}|\mathbf{x}) = q_\varphi(\mathbf{z}_1|\mathbf{x}) \prod_{t=\frac{1}{T}}^{1} q_\varphi(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x})$$

- Parameterisation of the forward marginals:

$$q_\varphi(\mathbf{z}_t|\mathbf{x}) = \prod_{i=1}^{D} q_\varphi(\mathbf{z}_t^i|\mathbf{x}) \qquad q_\varphi(\mathbf{z}_s|\mathbf{x}) = \int q_\varphi(\mathbf{z}_t|\mathbf{x})q_\varphi(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x})d\mathbf{z}_t$$

posterior must be consistent with the marginals 🤝

- Maximum Coupling as a forward posterior:



- Same objective function:

$$\mathbb{E}_{u(t)q_\varphi(\mathbf{z}_t|\mathbf{x})}\left[\mathrm{D}_{\mathrm{KL}}\left(q_\varphi(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x})\|p_\theta(\mathbf{z}_s|\mathbf{z}_t)\right)\right]$$

Non-differentiable, need RL to optimize 🤏

Properties

- 🔓 Unlocking few-step discrete diffusion!
- ⚡ No sampling overhead, same reverse process
- ✅ Same likelihood-based objective function
- 🎭 Can learn unmasking for **pretrained MDMs**
- 🎲 Non-differentiable optimization with RL
- 🔄 End-to-end training of noising and generation
- 🌎 Works across domains such as text and molecules

Results

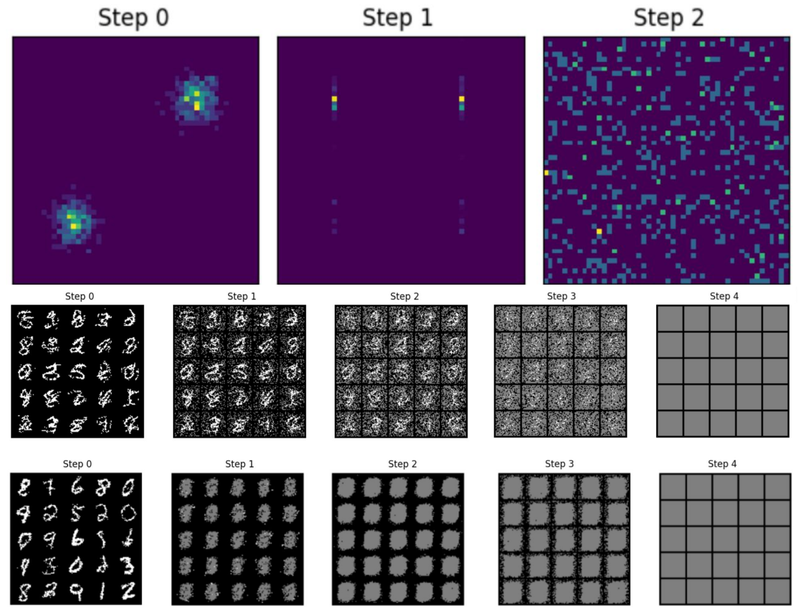


Figure 2: Learned 4-step dynamics on the binarized MNIST dataset. Regular masked diffusion (top) and FLDD (bottom).

Table 1: Results on ROCStories dataset.

	MAUVE ↑	PPL ↓	Div ↑
GPT2 (Radford et al., 2019)	0.789	20.5	0.252
GPT Neo (Black et al., 2021)	0.720	19.9	0.258
AR-Diffusion (Wu et al., 2023)	0.066	41.8	0.101
DiffuSeq (Gong et al., 2022)	0.086	50.5	0.124
SeqDiffuSeq (Yuan et al., 2022)	0.103	29.3	0.137
TESS (Mahabadi et al., 2023)	0.061	22.4	0.163
SEDD (Lou et al., 2023)	0.598	70.8	0.336
LD4LG (Lovelace et al., 2023)	0.716	30.6	0.331
COSMOS (Meshchaninov et al., 2025)	0.940	26.3	0.346
FLDD, $T = 100$	0.538	55.2	0.280
FLDD, $T = 10$	0.511	60.5	0.285

Table 2: Molecular generation results.

	QM9			ZINC250k		
	Valid ↑	Unique ↑	FCD ↓	Valid ↑	Unique ↑	FCD ↓
MoFlow (Zang & Wang, 2020)	91.36	98.65	4.467	63.11	99.99	20.931
EDP-GNN (Niu et al., 2020)	47.52	99.25	2.680	82.97	99.79	16.737
GraphEBM (Liu et al., 2021)	8.22	97.90	6.143	5.29	98.79	35.471
GDSS (Jo et al., 2022)	95.72	98.46	2.900	97.01	99.64	14.656
Digress Vignac et al. (2022)	99.00	96.20	-	-	-	-
Flow Matching (Lipman et al., 2023)	94.10	98.20	5.155	94.01	96.68	18.764
Dirichlet FM (Stark et al., 2024)	99.10	98.15	0.888	97.52	99.20	14.222
CatFlow (Eijkelboom et al., 2024)	99.81	99.95	0.441	99.21	100.00	13.211
LO-ARM (Wang et al., 2025)	99.85	98.85	0.240	96.70	100.00	3.229
FLDD, $T = 100$	99.67	98.93	0.328	97.79	100.00	8.487
FLDD, $T = 10$	99.08	99.95	0.385	96.77	100.00	10.414