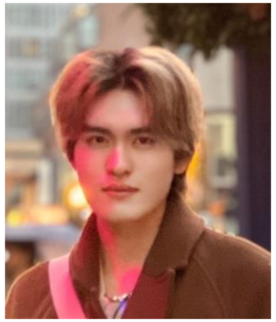
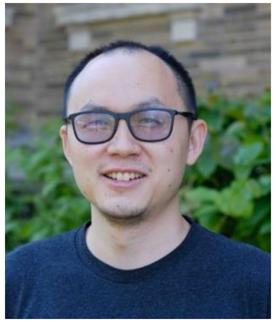
SymmetricDiffusers

Learning Discrete Diffusion on Finite Symmetric Groups

Yongxing (Nick) Zhang[1,3], Donglin Yang[2,3], Renjie Liao[2,3,4]

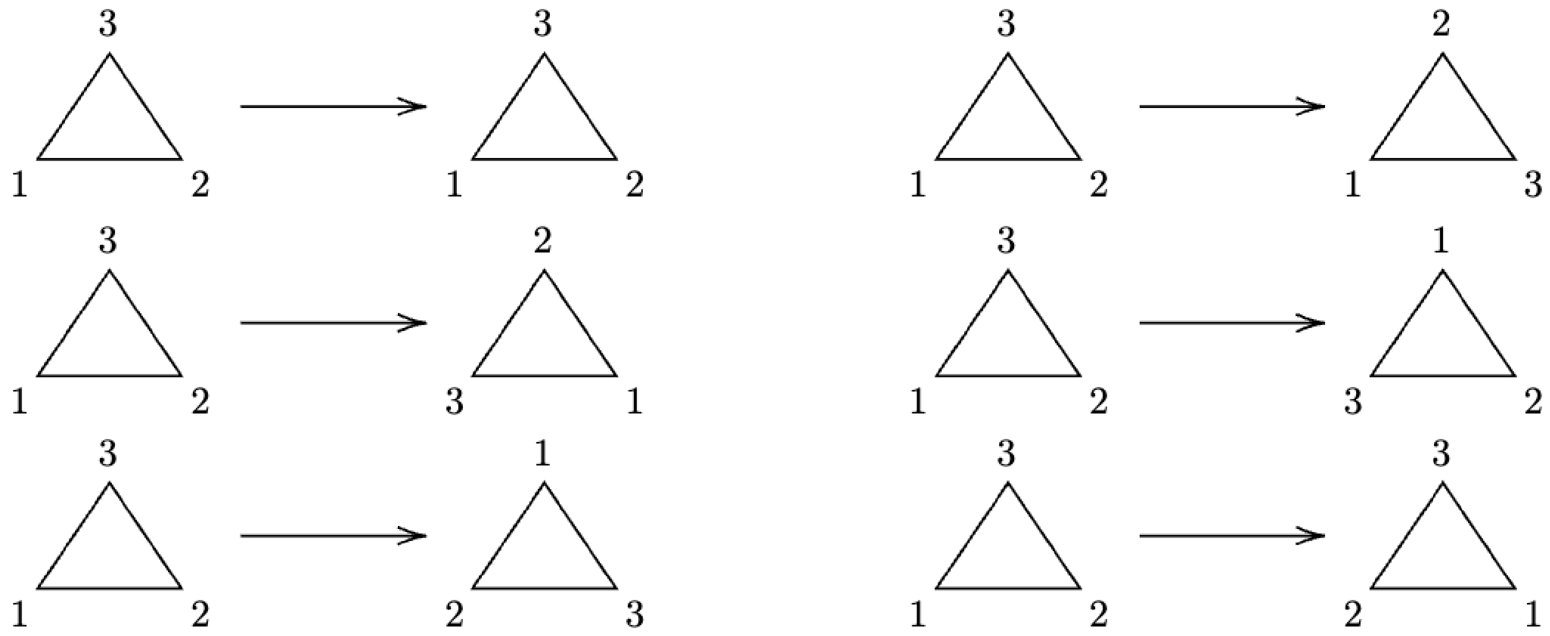
1 UNIVERSITY OF WATERLOO
2 UBC
3 VECTOR INSTITUTE | INSTITUT VECTEUR
4 CIFAR

Code

Finite Symmetric Groups

The finite symmetric group S_n:

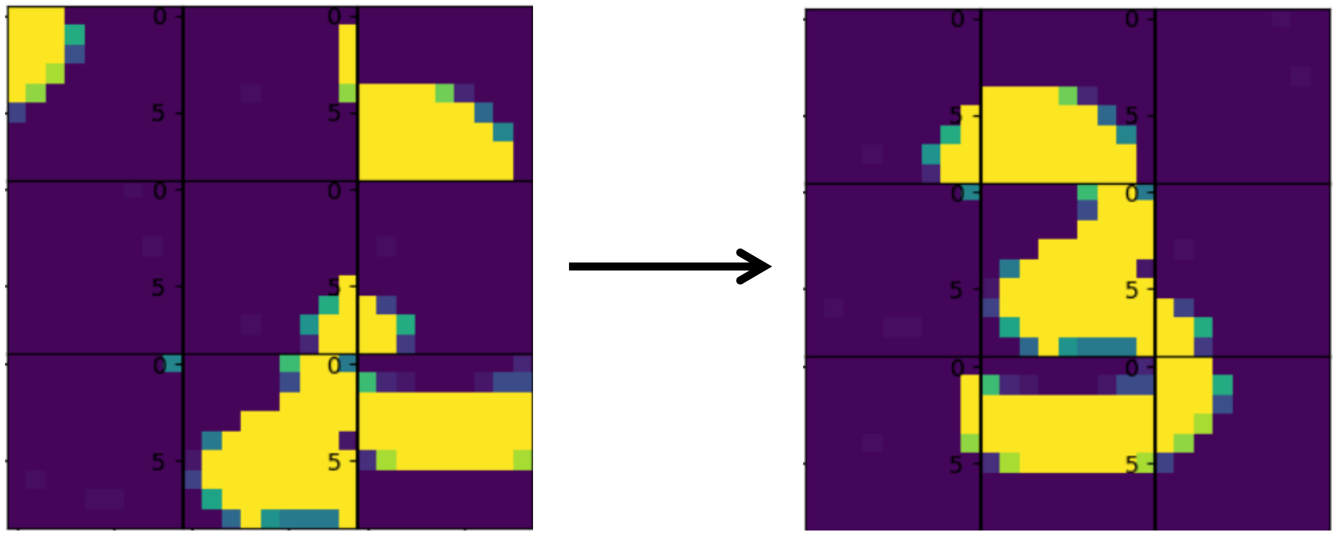
- **Bijections** from a set of n elements to itself
- Group operation is **function composition**.



Finite Symmetric Groups

The finite symmetric group S_n:

- **Bijections** from a set of n elements to itself
- Group operation is **function composition**.



Learning a Distribution over S_n

Challenges:

- **Factorial** size;

Learning a Distribution over S_n

Challenges:

- **Factorial** size;
- **Discrete** structure;

Learning a Distribution over S_n

Challenges:

- **Factorial** size;
- **Discrete** structure;
- Thus, difficulties in:
  - Designing **expressive probabilistic** modelling;
  - **Gradient**-based learning.

Learning a Distribution over S_n via Discrete Diffusion

We use **discrete diffusion** to model over S_n

- **Decomposing** learning a complicated distribution **into a sequence of simpler problems**.

Learning a Distribution over S_n via Discrete Diffusion

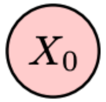
We use **discrete diffusion** to model over S_n

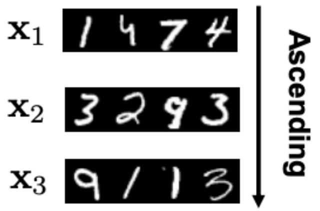
- **Decomposing** learning a complicated distribution **into a sequence of simpler problems**.

We propose:

- Shuffling methods as the **forward process** based on theories of random walks on finite groups;
- Transitions and parameterizations for the **reverse process** with provable expressiveness.

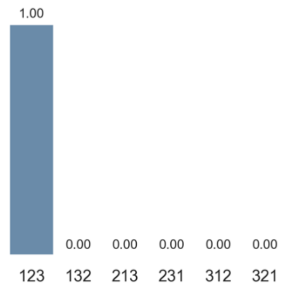
Learning a Distribution over S_n via Discrete Diffusion

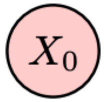


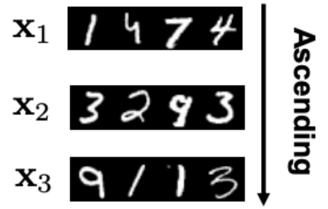


Three 4-digit numbers

Learning a Distribution over S_n via Discrete Diffusion

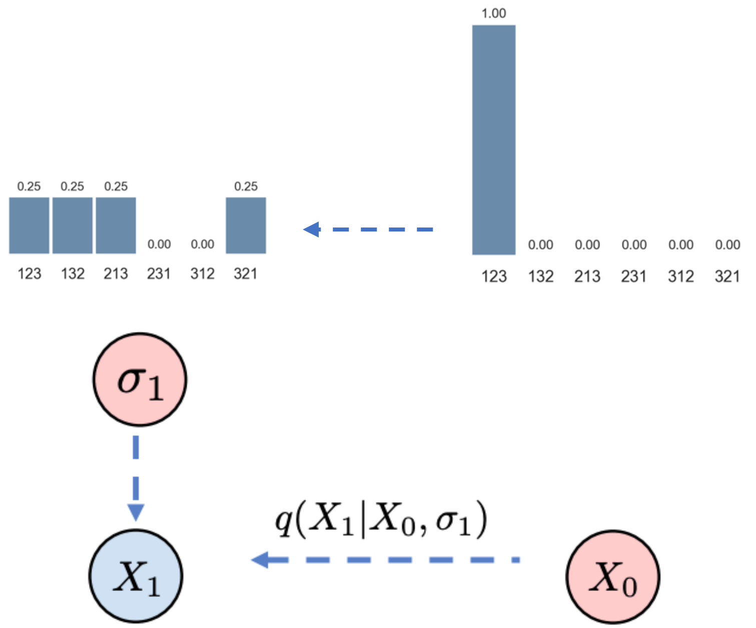


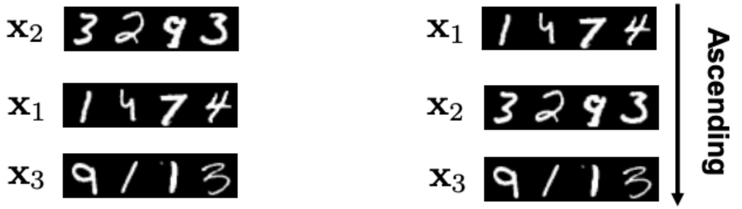




Three 4-digit numbers

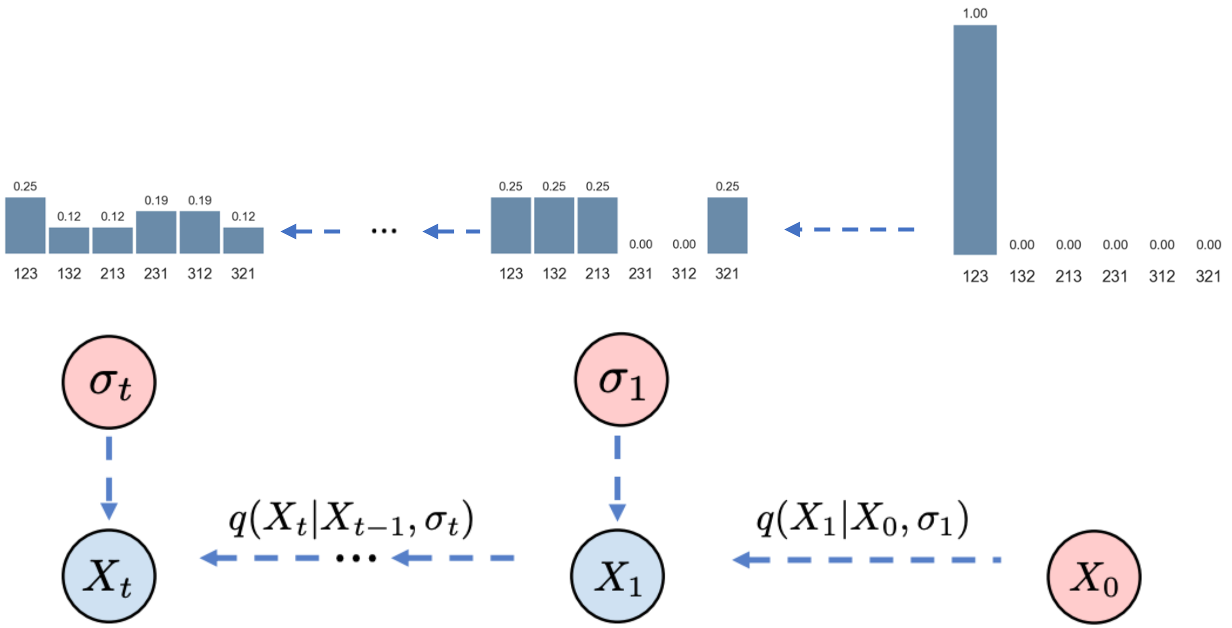
Learning a Distribution over S_n via Discrete Diffusion

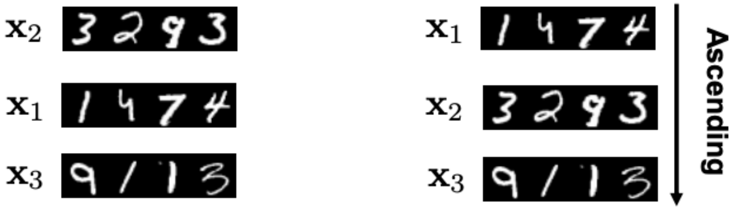




Three 4-digit numbers

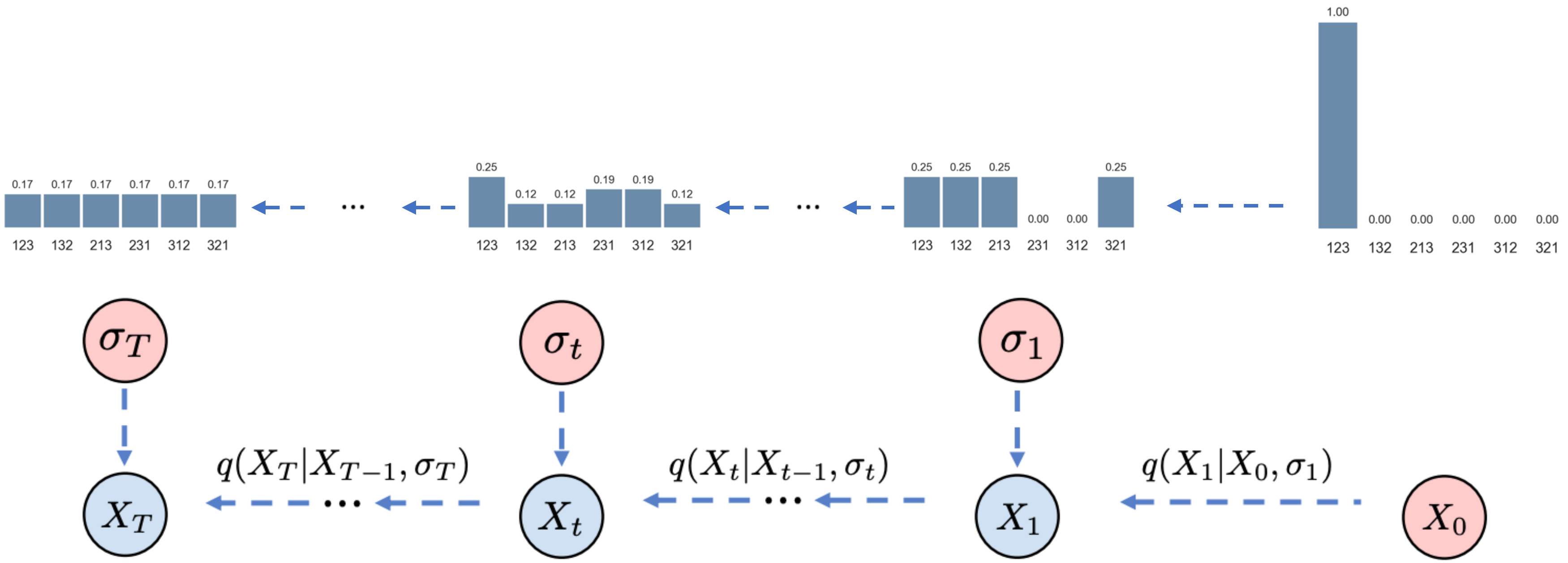
Learning a Distribution over S_n via Discrete Diffusion





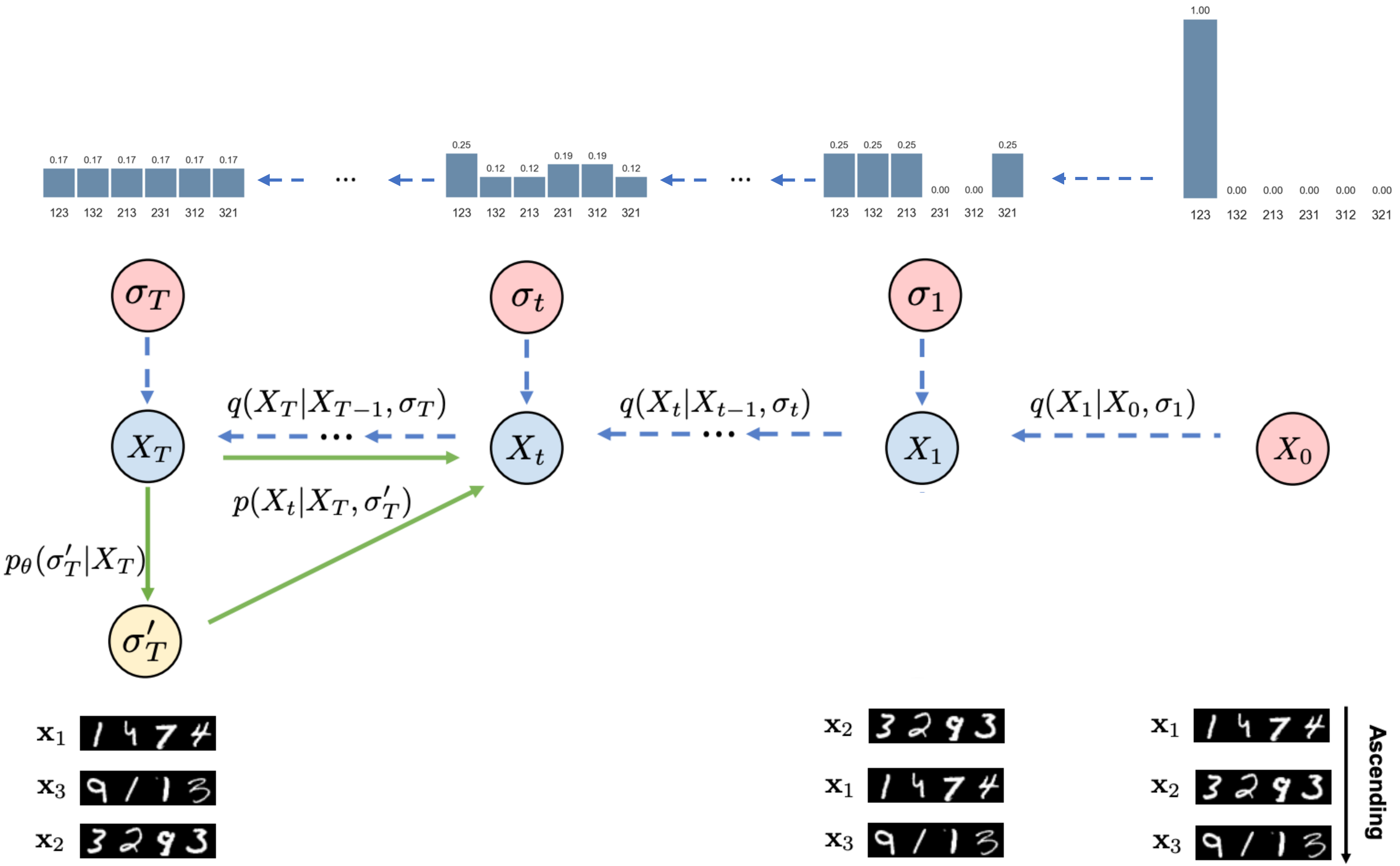
Three 4-digit numbers

Learning a Distribution over S_n via Discrete Diffusion



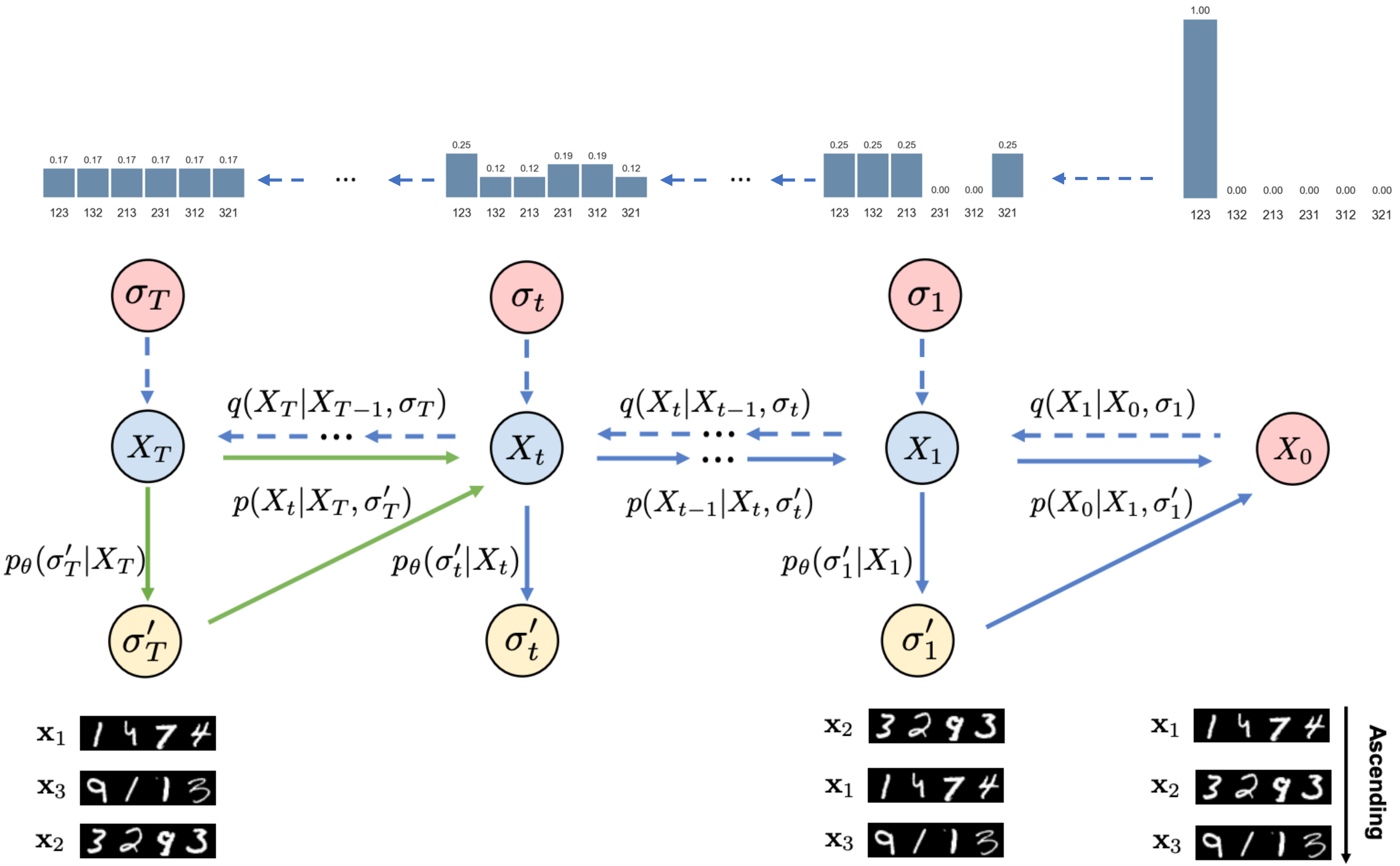
Three 4-digit numbers

Learning a Distribution over S_n via Discrete Diffusion



Three 4-digit numbers

Learning a Distribution over S_n via Discrete Diffusion



Three 4-digit numbers

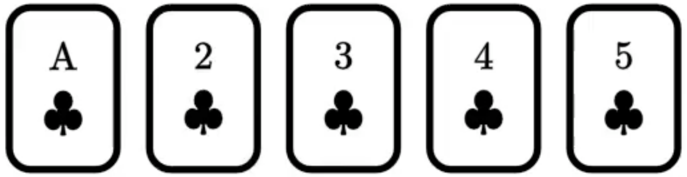
The Forward Process

Shuffling methods:

- *Random Transpositions*
- *Random Insertions*
- *Riffle Shuffles*

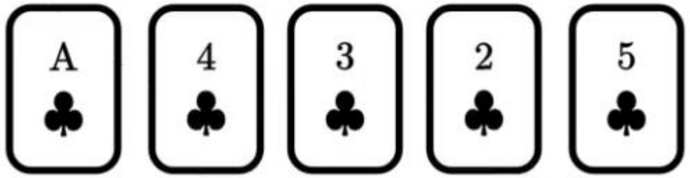
Shuffling Methods for the Forward Process

Random Transpositions

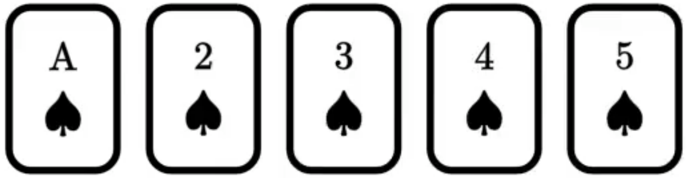


Shuffling Methods for the Forward Process

Random Transpositions



Random Insertions



Shuffling Methods for the Forward Process

- **Riffle shuffles:** similar to how we shuffle cards in card games

Sampling the Forward Process

- **DDPM style models:** one step $q(X_t \mid X_0)$.

Sampling the Forward Process

- **DDPM style models:** one step $q(X_t \mid X_0)$.
- Not possible for *most* shuffling methods.
- Run the whole forward Markov chain to obtain the state at time t.

Sampling the Forward Process

- **DDPM style models:** one step $q(X_t \mid X_0)$.
- Not possible for *most* shuffling methods.
- Run the whole forward Markov chain to obtain the state at time t.
- Riffle shuffles do admit efficient sampling at arbitrary time-step.

Mixing Time and the Cut-off Phenomenon

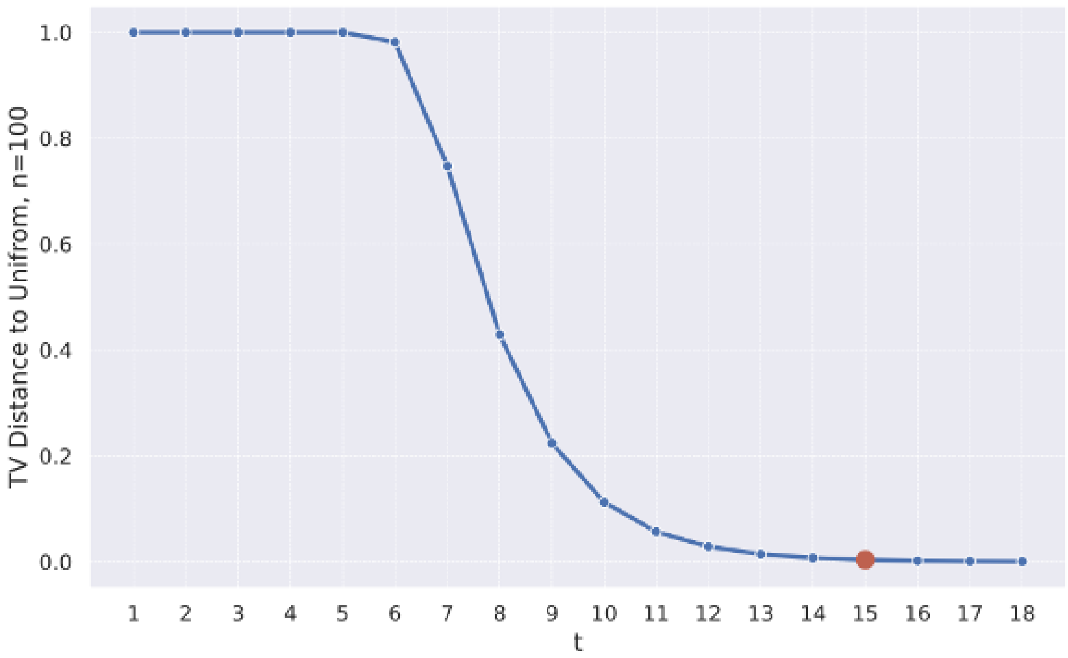
- **Stationary distribution:** uniform for all shuffling methods.
- **Mixing time:** time until the Markov chain is close in *TV distance* to stationary.

$$D_{\mathrm{TV}}(p,q) = \sup_{A \text{ measurable}} |p(A) - q(A)| \overset{\text{discrete}}{=} \frac{1}{2}\sum_x |p(x) - q(x)|$$

Mixing Time and the Cut-off Phenomenon

Cut-off phenomenon:

- TV distance to stationary first stays around 1;
- But then **abruptly drops to close to 0**.



Mixing Time and the Cut-off Phenomenon

Cut-off phenomenon:

- TV distance to stationary first stays around 1;
- But then **abruptly drops to close to 0**.

Shuffling Methods	**Asymptotic Cut-off Time**
Random Transposition	$\frac{n}{2}\log n$
Random Insertion	$n\log n$
Riffle Shuffle	$\frac{3}{2}\log_2 n$

Reference: Laurent Saloff-Coste. Random Walks on Finite Groups.

The Reverse Process

Different **distribution parameterizations**:

- *Inverse Card Shuffling:* undo the shuffling;
- *The PL Distribution;*
- *The Generalized PL Distribution.*

Previous Work: The Plackett-Luce Distribution

Given scores $s_1, s_2, \ldots, s_n$ and a permutation $\sigma \in S_n$:

$$p_{\mathrm{PL}}(\sigma) = \frac{\exp\left(s_{\sigma(1)}\right)}{\exp\left(s_{\sigma(1)}\right) + \exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)}$$

.

Source:

R. L. Plackett. The analysis of permutations. Applied Statistics, 24(2):193 – 202, 1975.

R. D. Luce. Individual Choice Behavior. John Wiley, 1959.

Previous Work: The Plackett-Luce Distribution

Given scores $s_1, s_2, \ldots, s_n$ and a permutation $\sigma \in S_n$:

$$p_{\mathrm{PL}}(\sigma) = \frac{\exp\left(s_{\sigma(1)}\right)}{\exp\left(s_{\sigma(1)}\right) + \exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)}$$

$$\cdot \frac{\exp\left(s_{\sigma(2)}\right)}{\exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)}$$

$$\cdot$$

Source:

R. L. Plackett. The analysis of permutations. Applied Statistics, 24(2):193 – 202, 1975.

R. D. Luce. Individual Choice Behavior. John Wiley, 1959.

Previous Work: The Plackett-Luce Distribution

Given scores $s_1, s_2, \ldots, s_n$ and a permutation $\sigma \in S_n$:

$$
\begin{aligned}
p_{\mathrm{PL}}(\sigma) = \; & \frac{\exp\left(s_{\sigma(1)}\right)}{\exp\left(s_{\sigma(1)}\right) + \exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)} \\
& \cdot \frac{\exp\left(s_{\sigma(2)}\right)}{\exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)} \\
& \cdot \ldots \\
& \cdot \frac{\exp\left(s_{\sigma(n-1)}\right)}{\exp\left(s_{\sigma(n-1)}\right) + \exp\left(s_{\sigma(n)}\right)} \\
& \cdot \frac{\exp\left(s_{\sigma(n)}\right)}{\exp\left(s_{\sigma(n)}\right)}
\end{aligned}
$$

Source:
R. L. Plackett. The analysis of permutations. Applied Statistics, 24(2):193 – 202, 1975.
R. D. Luce. Individual Choice Behavior. John Wiley, 1959.

Previous Work: The Plackett-Luce Distribution

Given scores $s_1, s_2, \ldots, s_n$ and a permutation $\sigma \in S_n$:

$$
\begin{aligned}
p_{\mathrm{PL}}(\sigma) = &\frac{\exp\left(s_{\sigma(1)}\right)}{\exp\left(s_{\sigma(1)}\right) + \exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)} \\
&\cdot \frac{\exp\left(s_{\sigma(2)}\right)}{\exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)} \\
&\cdot \ldots \\
&\cdot \frac{\exp\left(s_{\sigma(n-1)}\right)}{\exp\left(s_{\sigma(n-1)}\right) + \exp\left(s_{\sigma(n)}\right)} \\
&\cdot \frac{\exp\left(s_{\sigma(n)}\right)}{\exp\left(s_{\sigma(n)}\right)}
\end{aligned}
$$

Problem: The PL distribution is not expressive enough, e.g. **cannot** represent a delta distribution.

The Generalized PL (GPL) Distribution

We propose:

- Parameterized using n^2 scores $(s_{ij})_{1 \leq i,j \leq n}$

PL

$$[s_1, s_2, \ldots, s_n]$$

GPL

$$\begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ s_{n,1} & s_{n,2} & \cdots & s_{n,n} \end{bmatrix}$$

The Generalized PL (GPL) Distribution

We propose:

- Parameterized using n^2 scores $(s_{ij})_{1 \le i,j \le n}$
- Each slot in the permutation uses **different** scores.

PL

$$p_{\mathrm{PL}}(\sigma) = \frac{\exp\left(s_{\sigma(1)}\right)}{\exp\left(s_{\sigma(1)}\right) + \exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)} \cdot \frac{\exp\left(s_{\sigma(2)}\right)}{\exp\left(s_{\sigma(2)}\right) + \cdots + \exp\left(s_{\sigma(n)}\right)} \cdots \cdot \frac{\exp\left(s_{\sigma(n-1)}\right)}{\exp\left(s_{\sigma(n-1)}\right) + \exp\left(s_{\sigma(n)}\right)} \cdot \frac{\exp\left(s_{\sigma(n)}\right)}{\exp\left(s_{\sigma(n)}\right)}$$

GPL

$$p_{\mathrm{GPL}}(\sigma) = \frac{\exp\left(s_{1,\sigma(1)}\right)}{\exp\left(s_{1,\sigma(1)}\right) + \exp\left(s_{1,\sigma(2)}\right) + \cdots + \exp\left(s_{1,\sigma(n)}\right)} \cdot \frac{\exp\left(s_{2,\sigma(2)}\right)}{\exp\left(s_{2,\sigma(2)}\right) + \cdots + \exp\left(s_{2,\sigma(n)}\right)} \cdots \cdot \frac{\exp\left(s_{n-1,\sigma(n-1)}\right)}{\exp\left(s_{n-1,\sigma(n-1)}\right) + \exp\left(s_{n-1,\sigma(n)}\right)} \cdot \frac{\exp\left(s_{n,\sigma(n)}\right)}{\exp\left(s_{n,\sigma(n)}\right)}$$

The Generalized PL (GPL) Distribution

We propose:

- Parameterized using n^2 scores $(s_{ij})_{1 \le i,j \le n}$
- Each slot in the permutation uses **different** scores.

Theorem. The reverse process using GPL can model any data distribution.

Training Objective

We maximize the ELBO:

$$\mathbb{E}_{p_{\text{data}}(X_0,\mathcal{X})}\Big[\log p_\theta(X_0|\mathcal{X})\Big] \geq \mathbb{E}_{p_{\text{data}}(X_0,\mathcal{X})q(X_{1:T}|X_0,\mathcal{X})}\left[\log p(X_T|\mathcal{X}) + \sum_{t=1}^{T}\log\frac{p_\theta(X_{t-1}|X_t)}{q(X_t|X_{t-1})}\right].$$

Training Objective

We maximize the ELBO:

$$\mathbb{E}_{p_{\text{data}}(X_0,\mathcal{X})}\Big[\log p_\theta(X_0|\mathcal{X})\Big] \geq \mathbb{E}_{p_{\text{data}}(X_0,\mathcal{X})q(X_{1:T}|X_0,\mathcal{X})}\left[\log p(X_T|\mathcal{X}) + \sum_{t=1}^{T}\log\frac{p_\theta(X_{t-1}|X_t)}{q(X_t|X_{t-1})}\right].$$

- Not the usual objective in other diffusion models.
- Since $q(X_t \mid X_0)$ and the KL divergences have no analytical form.

Denoising Schedule via Merging Reverse Steps

For PL and GPL, **merge** some reverse steps:

- **Faster** and **more memory-efficient** sampling**.**

Denoising Schedule via Merging Reverse Steps

For PL and GPL, **merge** some reverse steps:

- **Faster** and **more memory-efficient** sampling**.**

Denoising schedule: $0 \leq t_0 < \cdots < t_k = T$ not necessarily consecutive

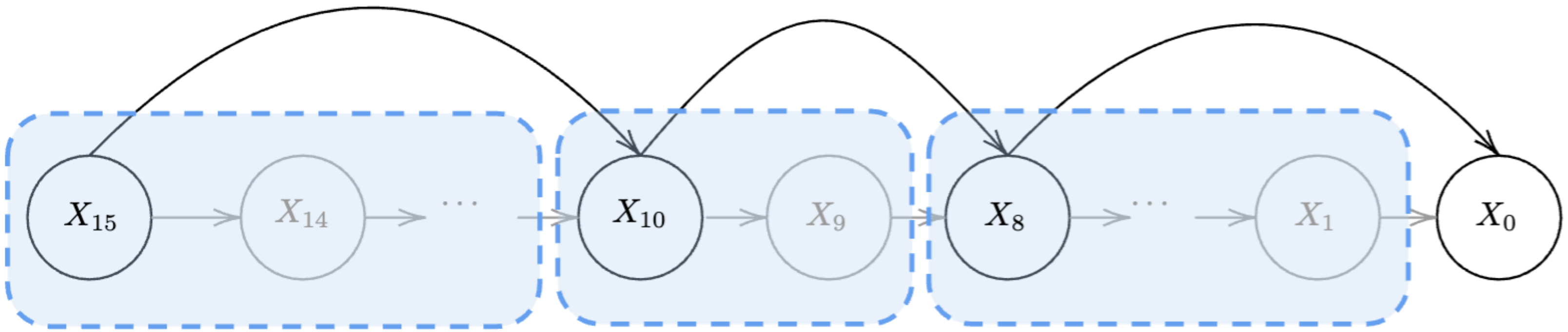
Denoising Schedule via Merging Reverse Steps

For PL and GPL, **merge** some reverse steps:

- **Faster** and **more memory-efficient** sampling**.**

Denoising schedule: $0 \leq t_0 < \cdots < t_k = T$ not necessarily consecutive

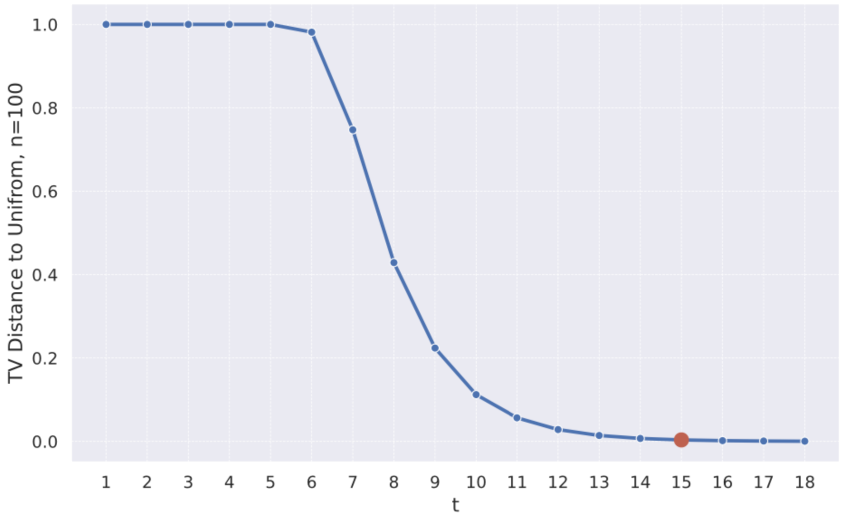
$$\mathcal{L}(\theta) = \mathbb{E}_{p_{\text{data}}(X_0, \mathcal{X})} \mathbb{E}_{q(X_{1:T} | X_0, \mathcal{X})} \left[-\log p(X_T | \mathcal{X}) - \sum_{i=1}^{k} \log \frac{p_\theta(X_{t_{i-1}} | X_{t_i})}{q(X_{t_i} | X_{t_{i-1}})} \right].$$



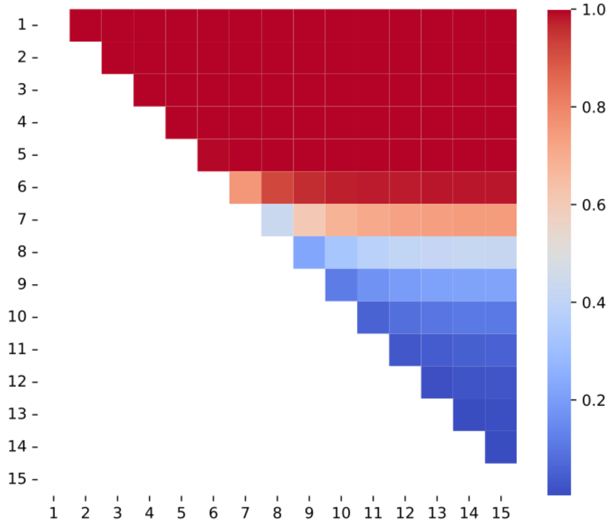
Denoising Schedule via Merging Reverse Steps

Intuitions:

- Use the cut-off phenomenon to determine T.
- Merge steps to keep a moderate "jump distance" in terms of TV distance.



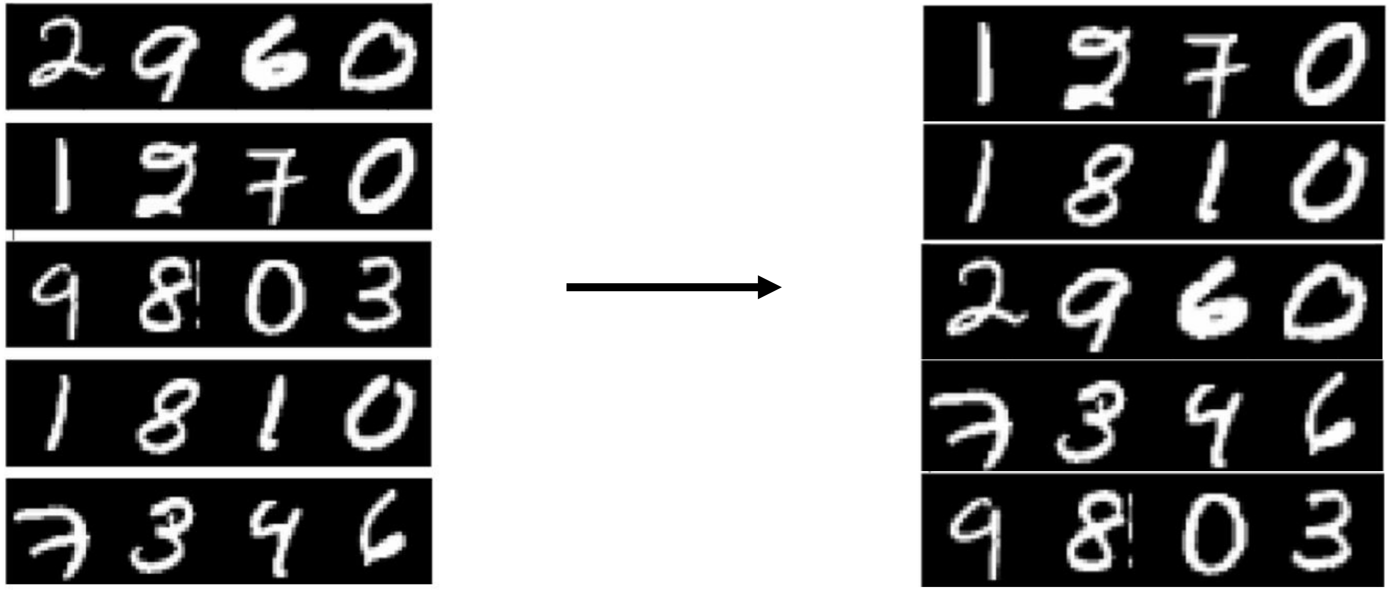
TV to uniform



TV between steps

Sorting 4-Digit MNIST Images

- We have n 4-digit images, where each 4-digit image is constructed by concatenating 4 individual images from MNIST.
- **Task:** Sort the n numbers.



Picture adapted from Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic optimization of sorting networks via continuous relaxations. In International Conference on Learning Representations, 2018.

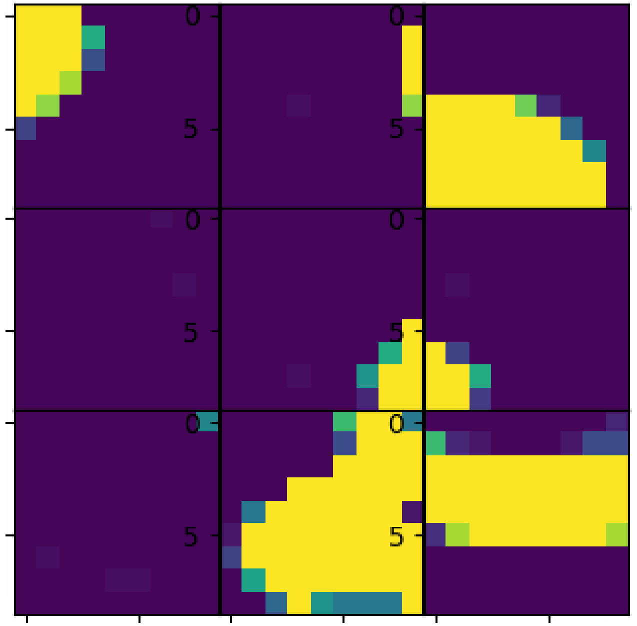
Sorting 4-Digit MNIST Images

- We have n 4-digit images, where each 4-digit image is constructed by concatenating 4 individual images from MNIST.
- **Task:** Sort the n numbers.

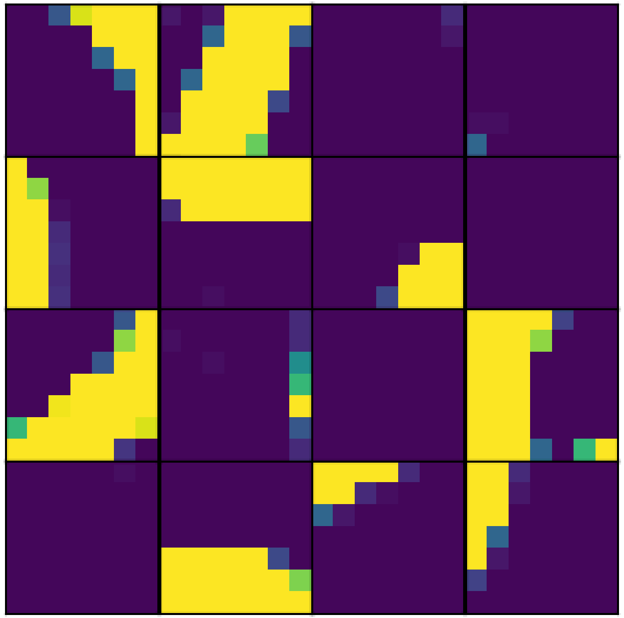
Method	**Metrics**	**Sequence Length**								
		3	5	7	9	15	32	52	100	200
DiffSort (Petersen et al., 2022)	Kendall-Tau ↑	0.930	0.898	0.864	0.801	0.638	0.535	0.341	0.166	0.107
	Accuracy (%)	93.8	83.9	71.5	52.2	10.3	0.2	0.0	0.0	0.0
	Correct (%)	95.8	92.9	90.1	85.2	82.3	61.8	42.8	23.2	15.3
Error-free DiffSort (Kim et al., 2024)	Kendall-Tau ↑	0.974	**0.967**	**0.962**	**0.952**	**0.938**	**0.879**	0.170	0.140	0.002
	Accuracy (%)	97.7	95.3	92.9	89.6	**83.1**	**57.1**	0.0	0.0	0.0
	Correct (%)	98.4	**97.7**	**97.2**	**96.3**	**95.1**	**90.1**	24.2	20.1	0.8
Symmetric Diffusers (Ours)	Kendall-Tau ↑	**0.976**	**0.967**	0.959	0.950	0.932	0.858	**0.786**	**0.641**	**0.453**
	Accuracy (%)	**98.0**	**95.5**	**92.9**	**90.0**	82.6	55.1	**27.4**	**4.5**	**0.1**
	Correct (%)	**98.5**	97.6	96.8	96.1	94.5	88.3	**82.1**	**69.3**	**52.2**

Jigsaw Puzzle

Chop up an image into patches, and recover the original image.



3x3, random transposition



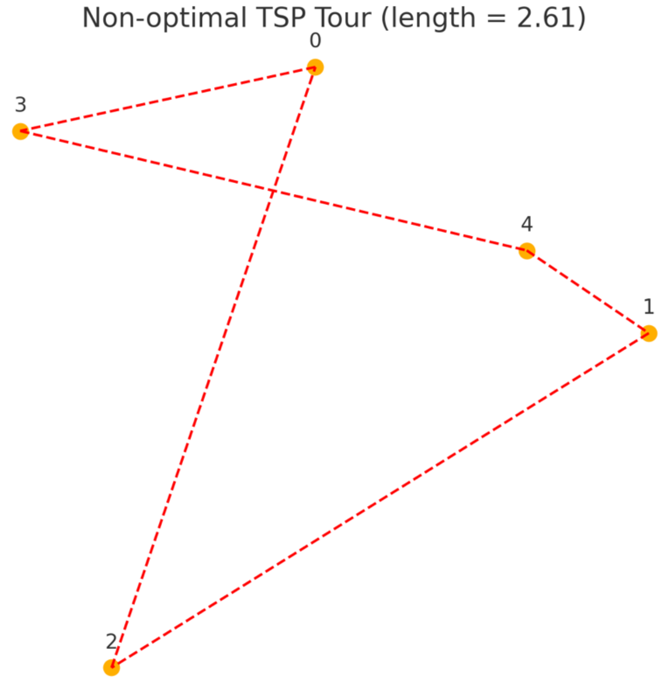
4x4, riffle shuffle

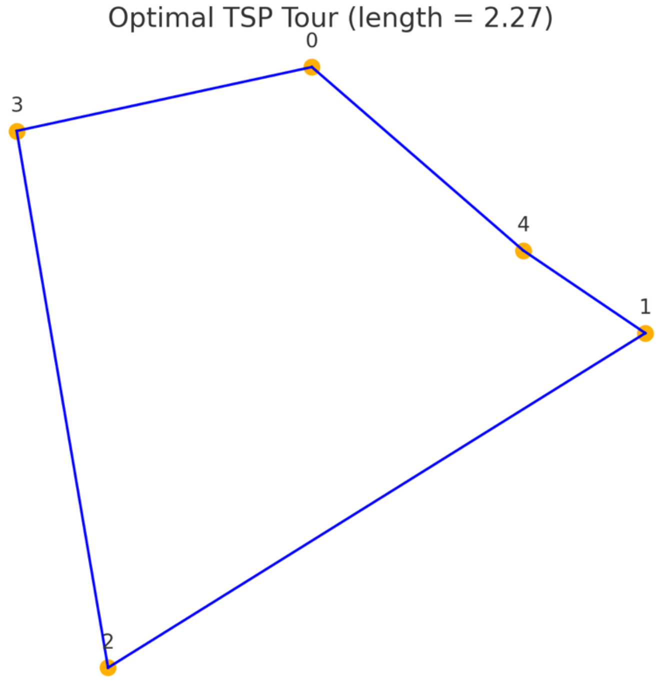
Jigsaw Puzzle

Method	Metrics	Noisy MNIST					CIFAR-10	
		2×2	3×3	4×4	5×5	6×6	3×3	4×4
Gumbel-Sinkhorn Network (Mena et al., 2018)	Kendall-Tau ↑	0.9984	0.6908	0.3578	0.2430	**0.1755**	0.5044	**0.4016**
	Accuracy (%)	99.81	44.65	00.86	0.00	0.00	6.07	0.21
	Correct (%)	99.91	80.20	49.51	26.94	14.91	43.59	25.31
DiffSort (Petersen et al., 2022)	Kendall-Tau ↑	0.9931	0.3054	0.0374	0.0176	0.0095	0.1460	0.0490
	Accuracy (%)	99.02	5.56	0.00	0.00	0.00	0.96	0.00
	Correct (%)	99.50	42.25	10.77	6.39	3.77	27.87	12.27
Error-free DiffSort (Kim et al., 2024)	Kendall-Tau ↑	0.9899	0.2014	0.0100	0.0034	-0.0021	0.1362	0.0318
	Accuracy (%)	98.62	0.82	0.00	0.00	0.00	0.68	0.00
	Correct (%)	99.28	32.65	7.40	4.39	2.50	26.75	10.33
Symmetric Diffusers (Ours)	Kendall-Tau ↑	**0.9992**	**0.8126**	**0.4859**	**0.2853**	0.1208	**0.8363**	0.2518
	Accuracy (%)	**99.88**	**57.38**	**1.38**	0.00	0.00	**70.94**	**0.64**
	Correct (%)	**99.94**	**86.16**	**58.51**	**37.91**	**18.54**	**86.84**	**34.69**

The (Euclidean) Travelling Salesman Problem

- Let $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^2$. We need to find some $\sigma \in S_n$ to minimize the tour length $\sum_{i=1}^{n} \|v_{\sigma(i)} - v_{\sigma(i+1)}\|_2$, where we let $\sigma(n+1) := \sigma(1)$.





The (Euclidean) Travelling Salesman Problem

- Let $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^2$. We need to find some $\sigma \in S_n$ to minimize the tour length $\sum_{i=1}^{n} \|v_{\sigma(i)} - v_{\sigma(i+1)}\|_2$, where we let $\sigma(n+1) := \sigma(1)$.

Method		TSP-20		TSP-50	
		Tour Length ↓	Optimality Gap (%) ↓	Tour Length ↓	Optimality Gap (%) ↓
OR Solvers	Concorde	**3.84**	**0.00**	**5.69**	**0.00**
	LKH-3	**3.84**	**0.00**	**5.69**	**0.00**
	2-Opt	4.02	4.64	5.86	2.95
Learning-Based Models	GCN	**3.85***	0.21*	5.87	3.10
	DIFUSCO	3.88*	1.07*	**5.70**	**0.10**
	Ours	**3.85**	**0.18**	5.71	0.41

Thank You!