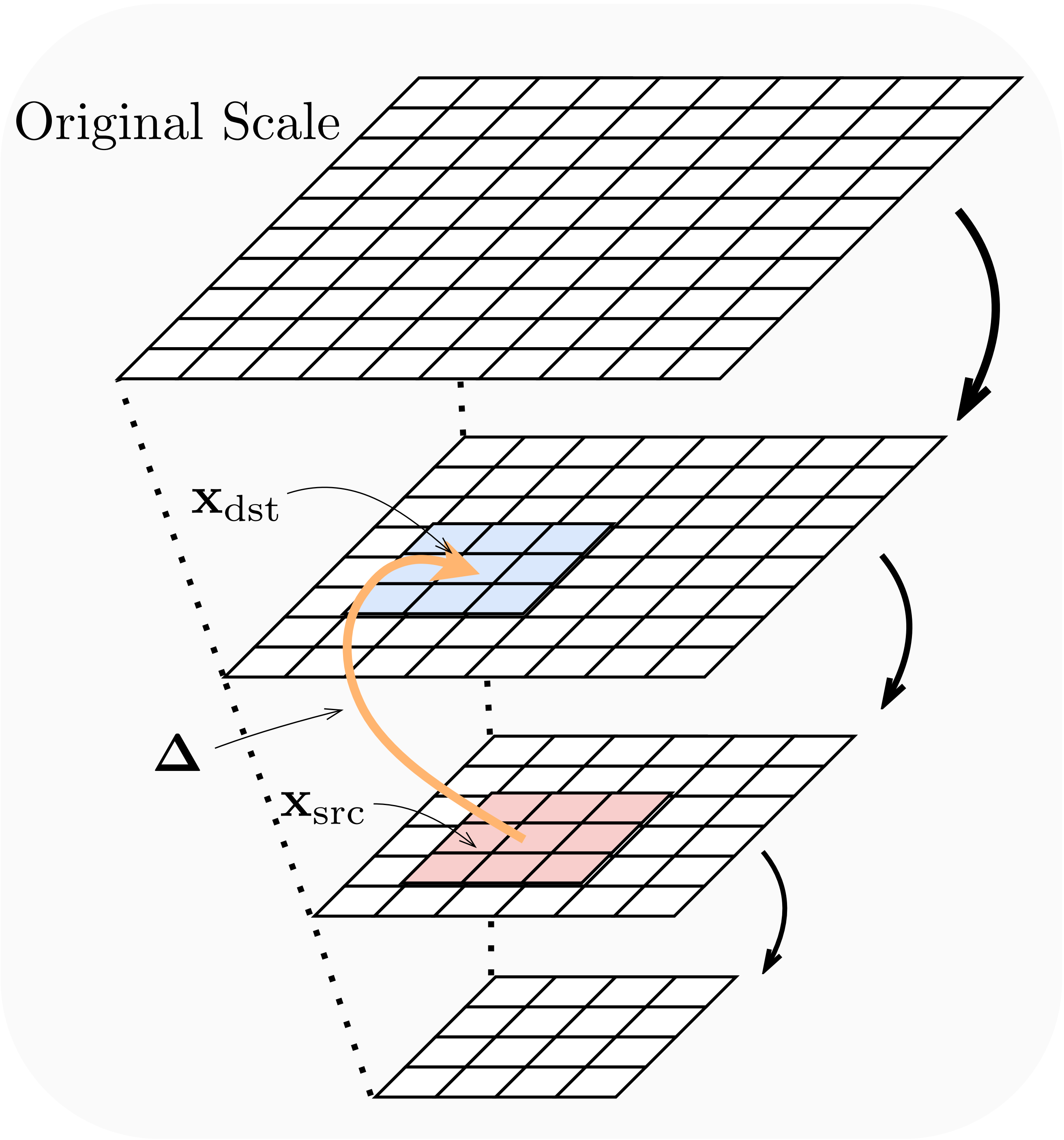
Internal Learning with Diffusion Model

Haojun Qiu
(joint work with Wenzheng Chen, David Lindell, Kyros Kutulakos)

1. Given a single image

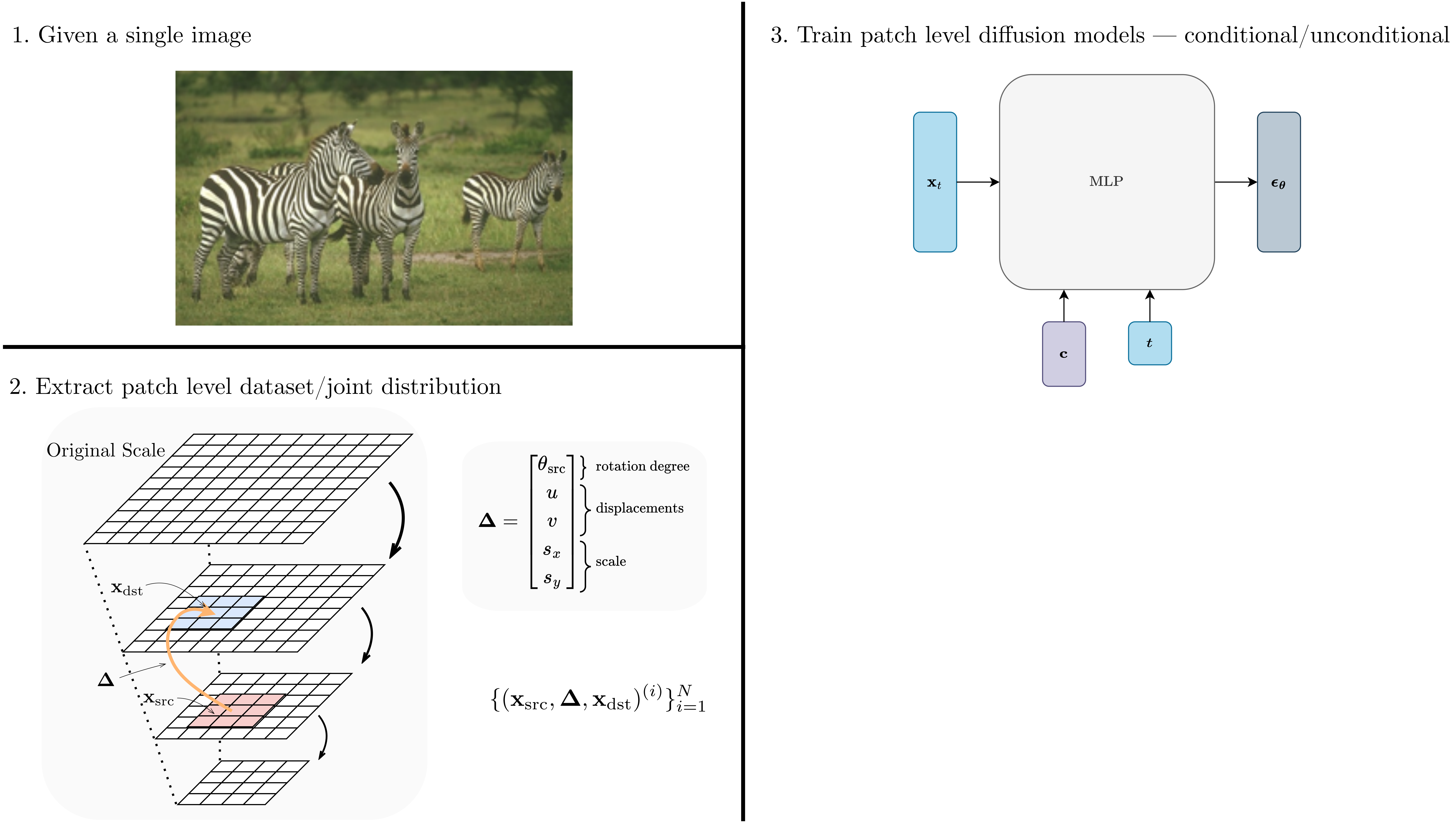
1. Given a single image

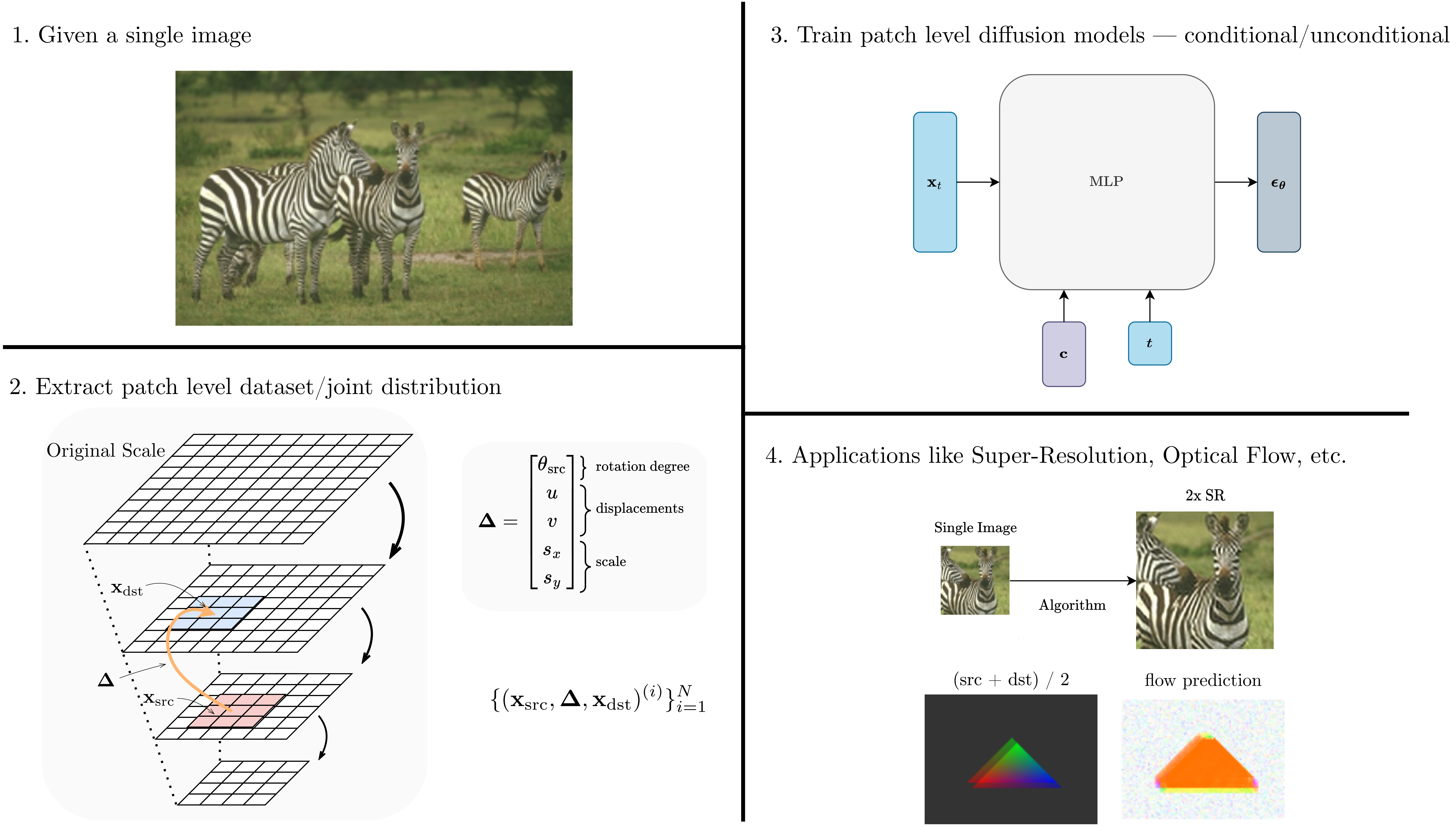
2. Extract patch level dataset/joint distribution



$$\boldsymbol{\Delta} = \begin{bmatrix} \theta_{\mathrm{src}} \\ u \\ v \\ s_x \\ s_y \end{bmatrix} \begin{array}{l} \}\ \text{rotation degree} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\}\ \text{displacements} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\}\ \text{scale} \end{array}$$

$$\{(\mathbf{x}_{\mathrm{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\mathrm{dst}})^{(i)}\}_{i=1}^{N}$$


1. Given a single image
2. Extract patch level dataset/joint distribution
Original Scale
$\mathbf{x}_{\mathrm{dst}}$
$\boldsymbol{\Delta}$
$\mathbf{x}_{\mathrm{src}}$
$\boldsymbol{\Delta} = \begin{bmatrix} \theta_{\mathrm{src}} \\ u \\ v \\ s_x \\ s_y \end{bmatrix}$
rotation degree
displacements
scale
$\{(\mathbf{x}_{\mathrm{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\mathrm{dst}})^{(i)}\}_{i=1}^{N}$
3. Train patch level diffusion models — conditional/unconditional
$\mathbf{x}_t$
MLP
$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$
$\mathbf{c}$
t


1. Given a single image
2. Extract patch level dataset/joint distribution
Original Scale
$\mathbf{x}_{\text{dst}}$
$\boldsymbol{\Delta}$
$\mathbf{x}_{\text{src}}$
$\boldsymbol{\Delta} = \begin{bmatrix} \theta_{\text{src}} \\ u \\ v \\ s_x \\ s_y \end{bmatrix}$
rotation degree
displacements
scale
$\{(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\text{dst}})^{(i)}\}_{i=1}^{N}$
3. Train patch level diffusion models — conditional/unconditional
$\mathbf{x}_t$
MLP
$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$
$\mathbf{c}$
t
4. Applications like Super-Resolution, Optical Flow, etc.
Single Image
Algorithm
2x SR
(src + dst) / 2
flow prediction

Overview

- Objectives (what we do)
- Significance (why we do it)

Overview

Objectives (what we do):

- **Internal Learning with Diffusion Models.** Learning *patch-level* distributions (conditional, unconditional) from a single image, by training diffusion models from scratch.

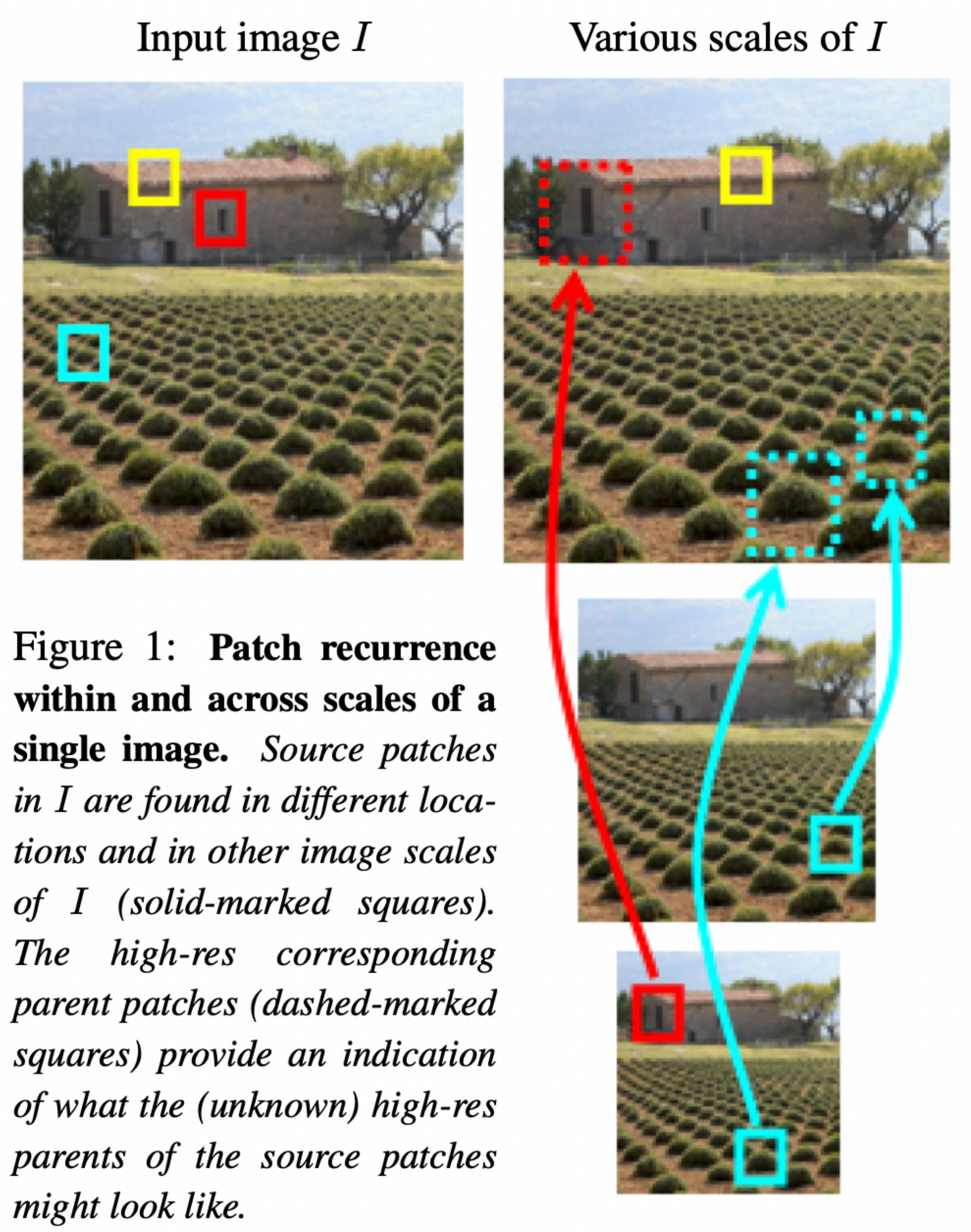
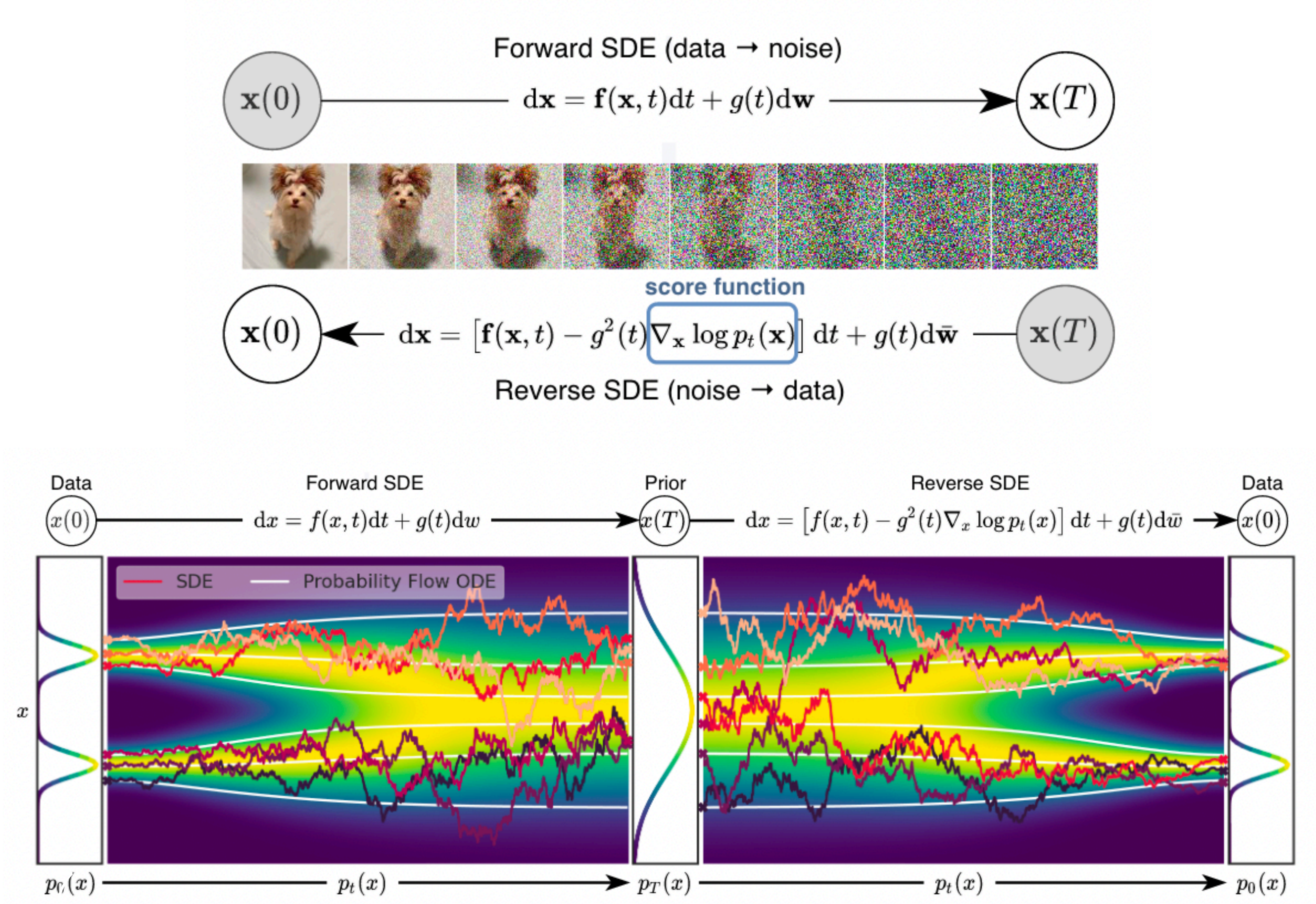


Figure 1: **Patch recurrence within and across scales of a single image.** *Source patches in I are found in different locations and in other image scales of I (solid-marked squares). The high-res corresponding parent patches (dashed-marked squares) provide an indication of what the (unknown) high-res parents of the source patches might look like.*



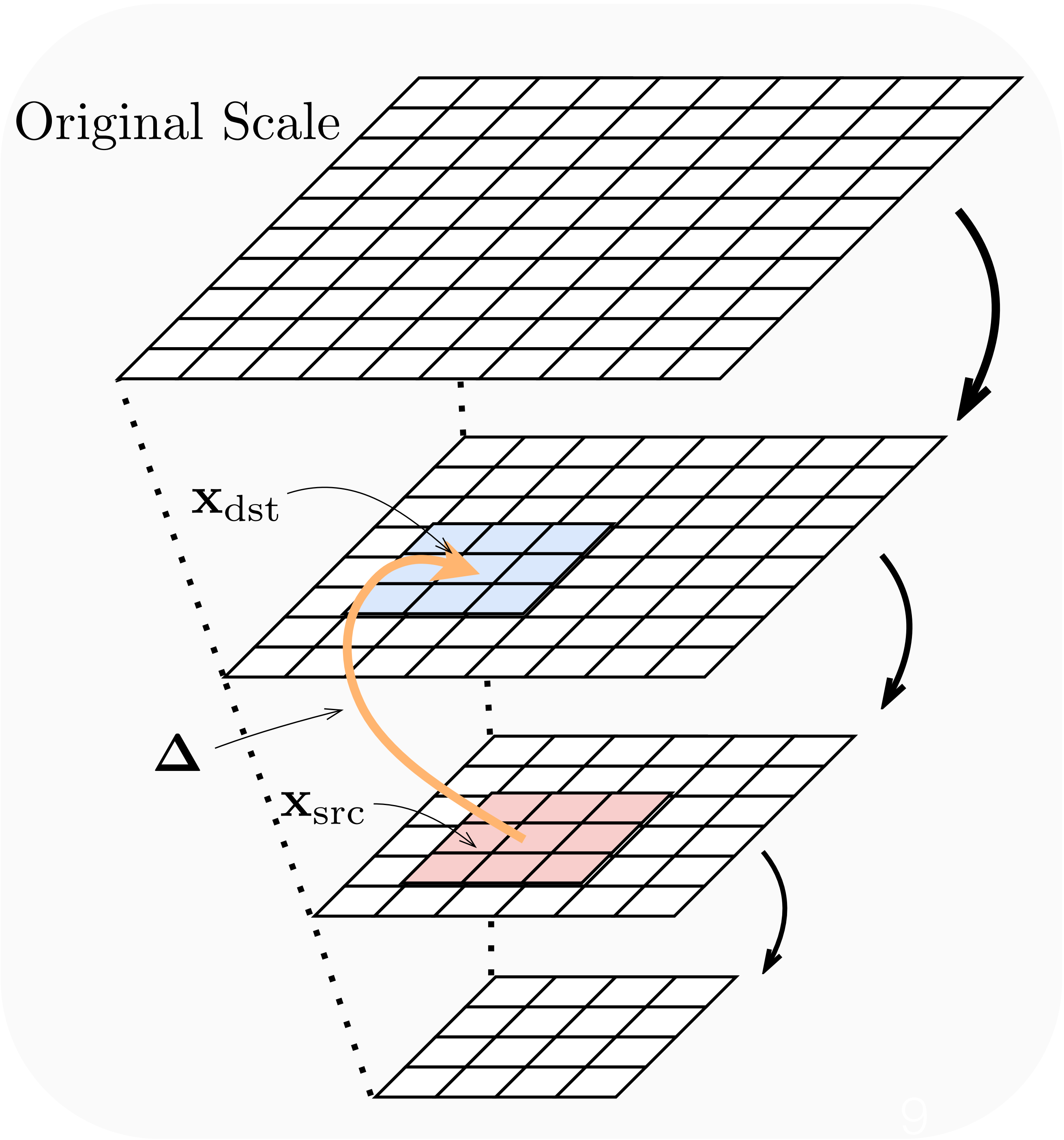
Overview

- Objectives (what we do)
- Significance (why we do it)

Overview

Significances (Why we do it?):

- We make a difference by analyzing patches <u>joint distributions</u> <u>p(patch1, patch2, relationship)</u>



$$\boldsymbol{\Delta} = \begin{bmatrix}\theta_{\text{src}} \\ u \\ v \\ s_x \\ s_y\end{bmatrix} \begin{array}{l} \} \ \text{rotation degree} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{displacements} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{scale} \end{array}$$

$$\boxed{\{(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\text{dst}})^{(i)}\}_{i=1}^{N}}$$

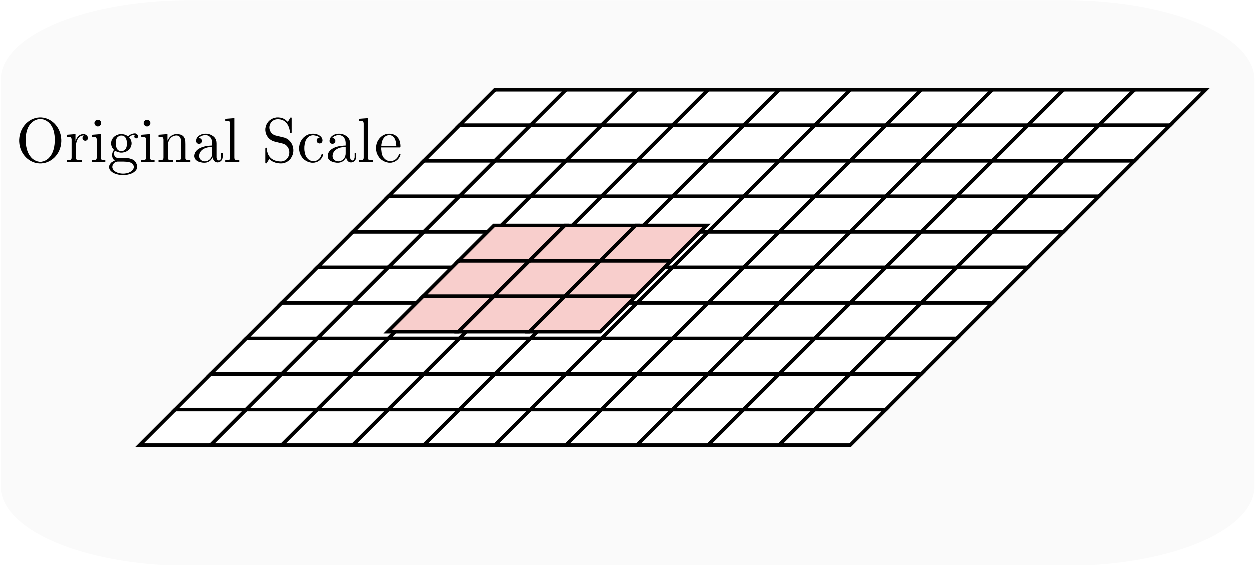
Our Approach

- Data representation (what is the distribution)
- Adapt diffusion model for our data representation

Our Approach — Data Representation

Distribution of patch

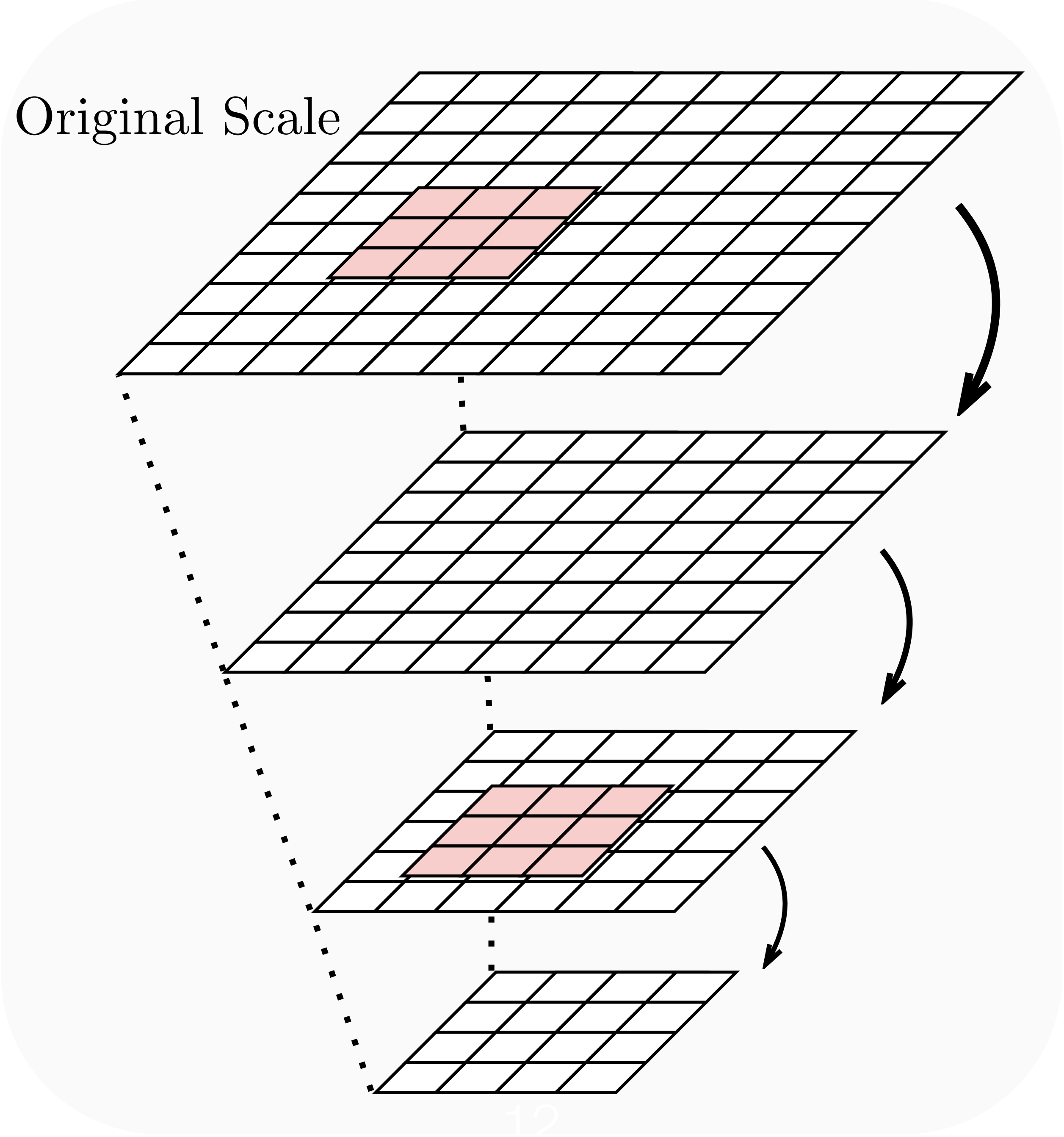
- Given a single image (in experiment mostly use sized 250 x 250), we crop small patches 7 x 7



Our Approach — Data Representation

Distribution of patch

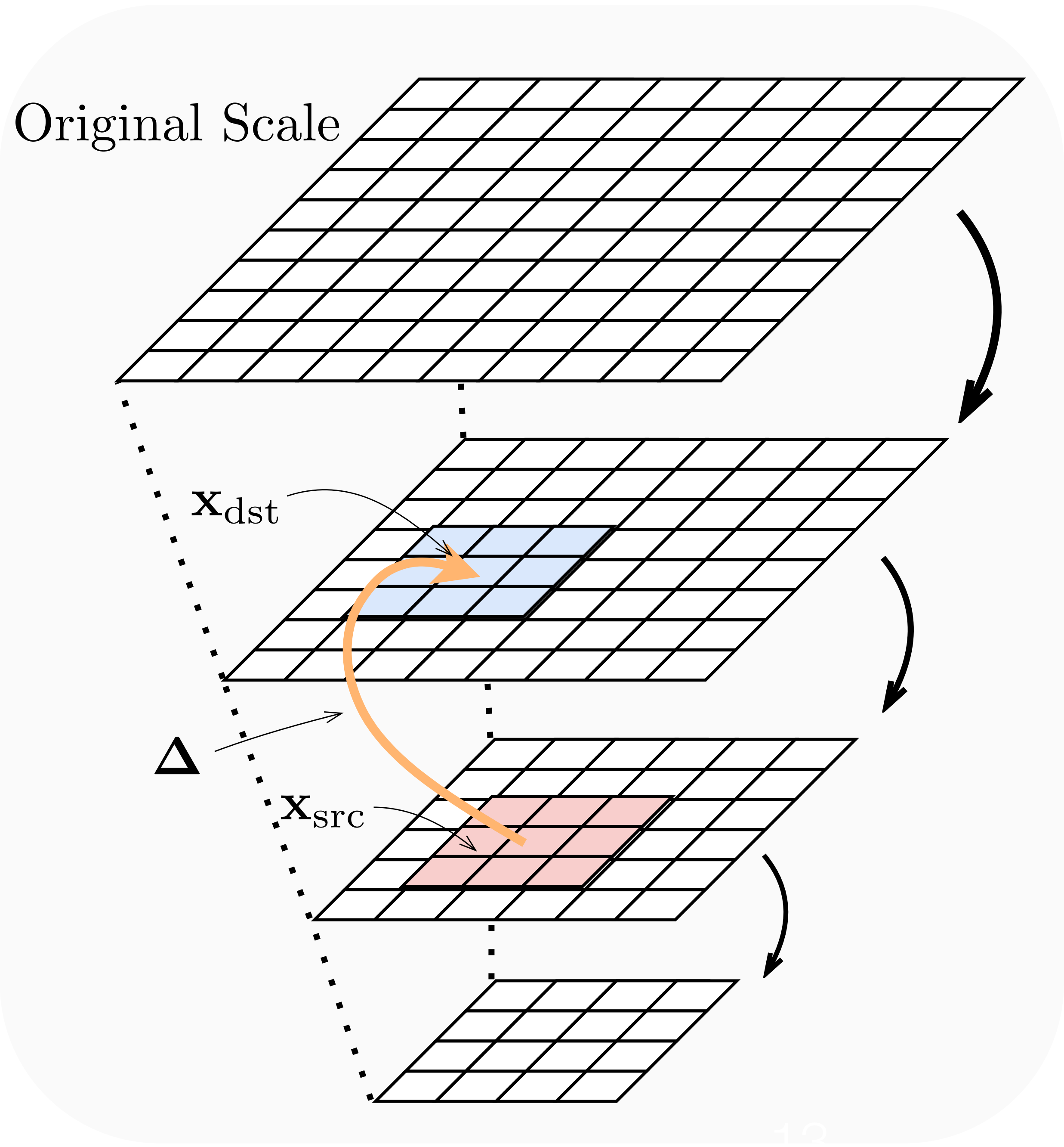
- Over-scales too



Our Approach — Data Representation

Joint distribution of (patch1, patch2, relationship)

- **A core difference** from prior work is the joint distribution of two patches and their 'relationship'.



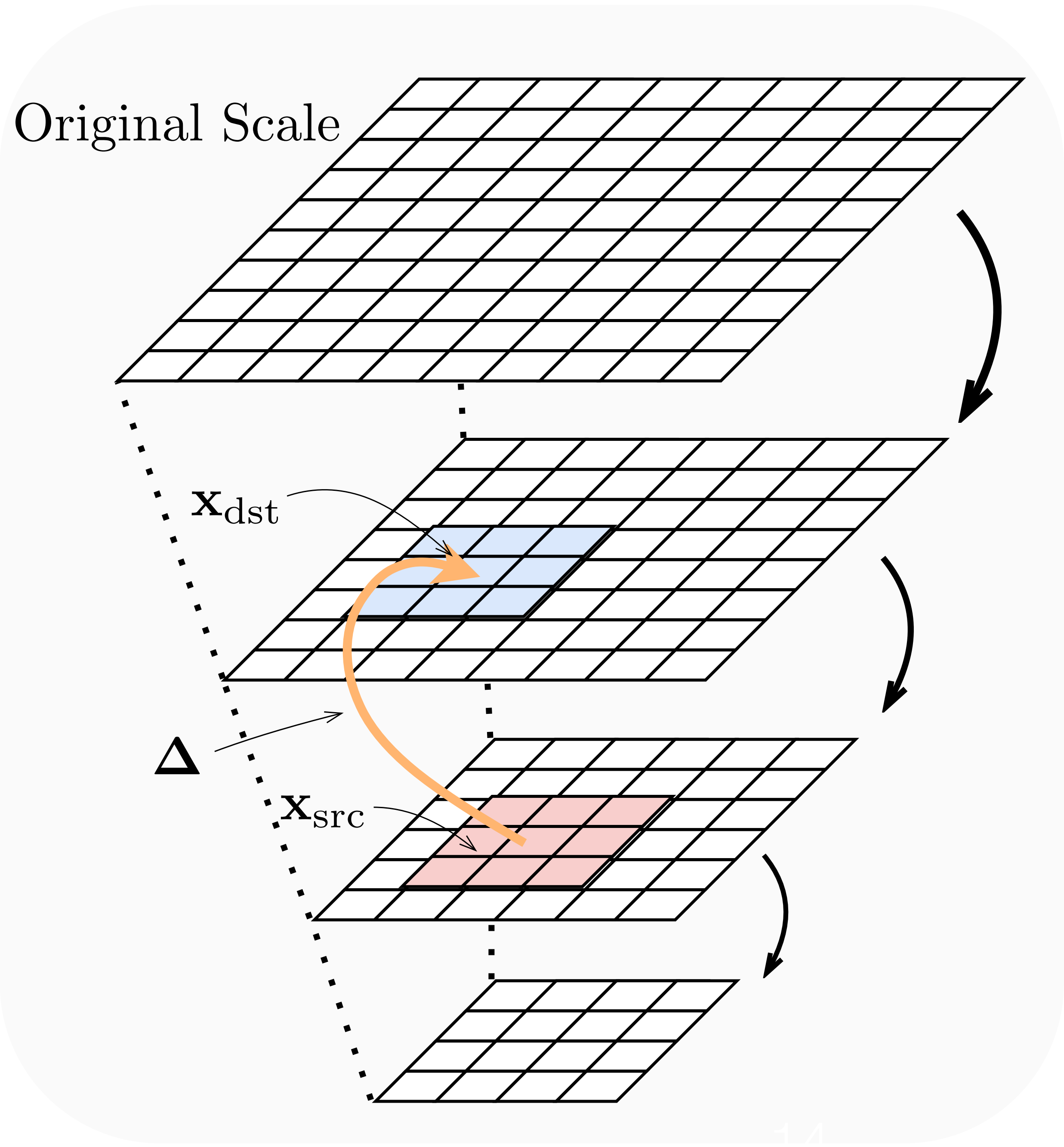
$$\mathbf{\Delta} = \begin{bmatrix} \theta_{\mathrm{src}} \\ u \\ v \\ s_x \\ s_y \end{bmatrix} \begin{array}{l} \} \text{ rotation degree} \\ \left.\begin{array}{l} \\ \end{array}\right\} \text{displacements} \\ \left.\begin{array}{l} \\ \end{array}\right\} \text{scale} \end{array}$$

$$(\mathbf{x}_{\mathrm{src}}, \mathbf{\Delta}, \mathbf{x}_{\mathrm{dst}})$$

Our Approach — Data Representation

Joint distribution of (patch1, patch2, relationship)

- What makes **a core difference** from prior work is the joint distribution of two patches and their 'relationship'.



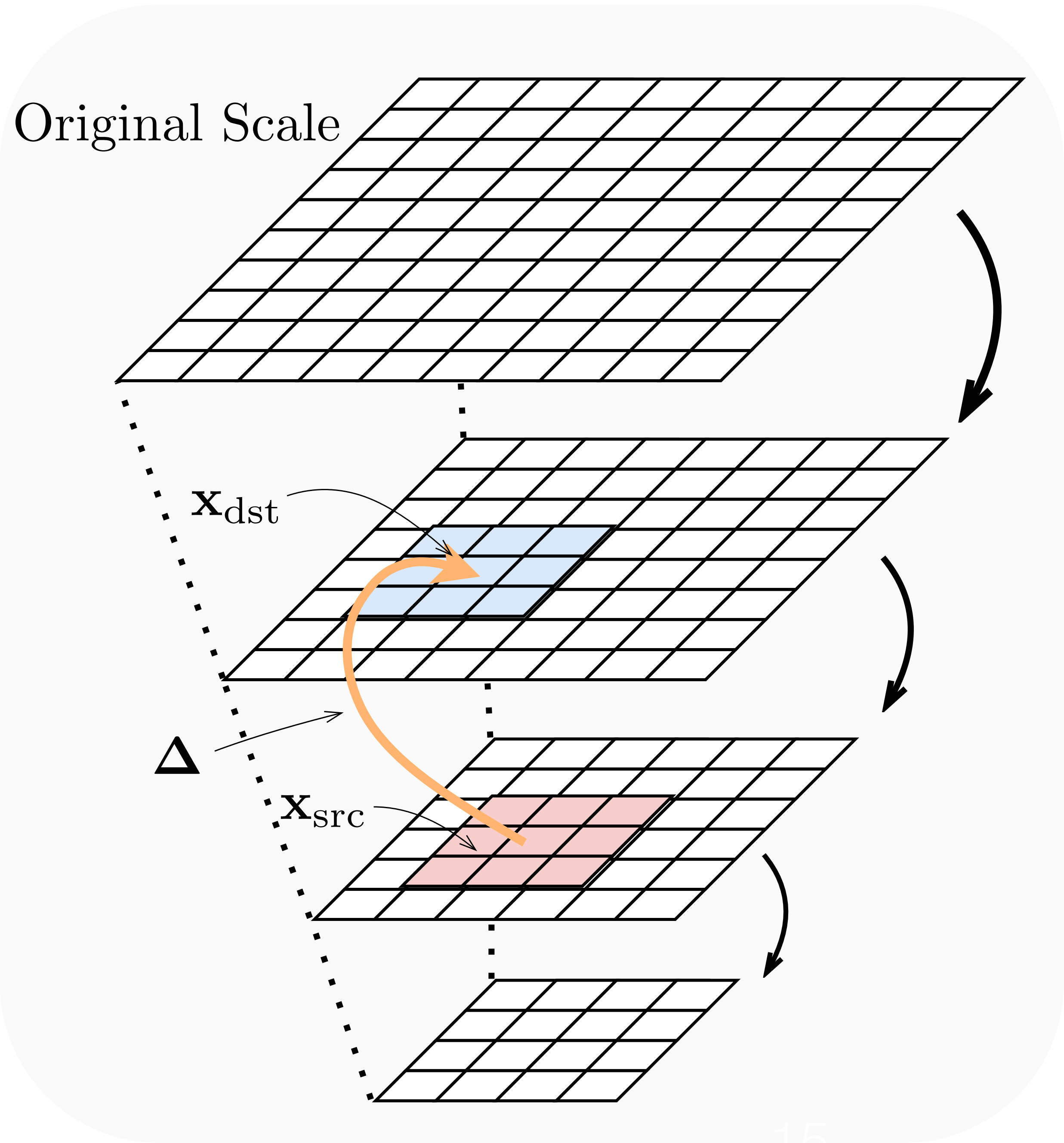
$$\boldsymbol{\Delta} = \begin{bmatrix} \theta_{\mathrm{src}} \\ u \\ v \\ s_x \\ s_y \end{bmatrix} \begin{array}{l} \} \text{ rotation degree} \\ \left.\begin{array}{l} \\ \end{array}\right\} \text{ displacements} \\ \left.\begin{array}{l} \\ \end{array}\right\} \text{ scale} \end{array}$$

$$\boxed{(\mathbf{x}_{\mathrm{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\mathrm{dst}})}$$

Our Approach — Data Representation

Joint distribution of (patch1, patch2, relationship)

- What makes **a core difference** from prior work is the joint distribution of two patches and their 'relationship'.



$$\boldsymbol{\Delta} = \begin{bmatrix}\theta_{\text{src}}\\ u\\ v\\ s_x\\ s_y\end{bmatrix}\begin{matrix}\}\ \text{rotation degree}\\ \left.\begin{matrix}\\ \\ \end{matrix}\right\}\ \text{displacements}\\ \left.\begin{matrix}\\ \\ \end{matrix}\right\}\ \text{scale}\end{matrix}$$

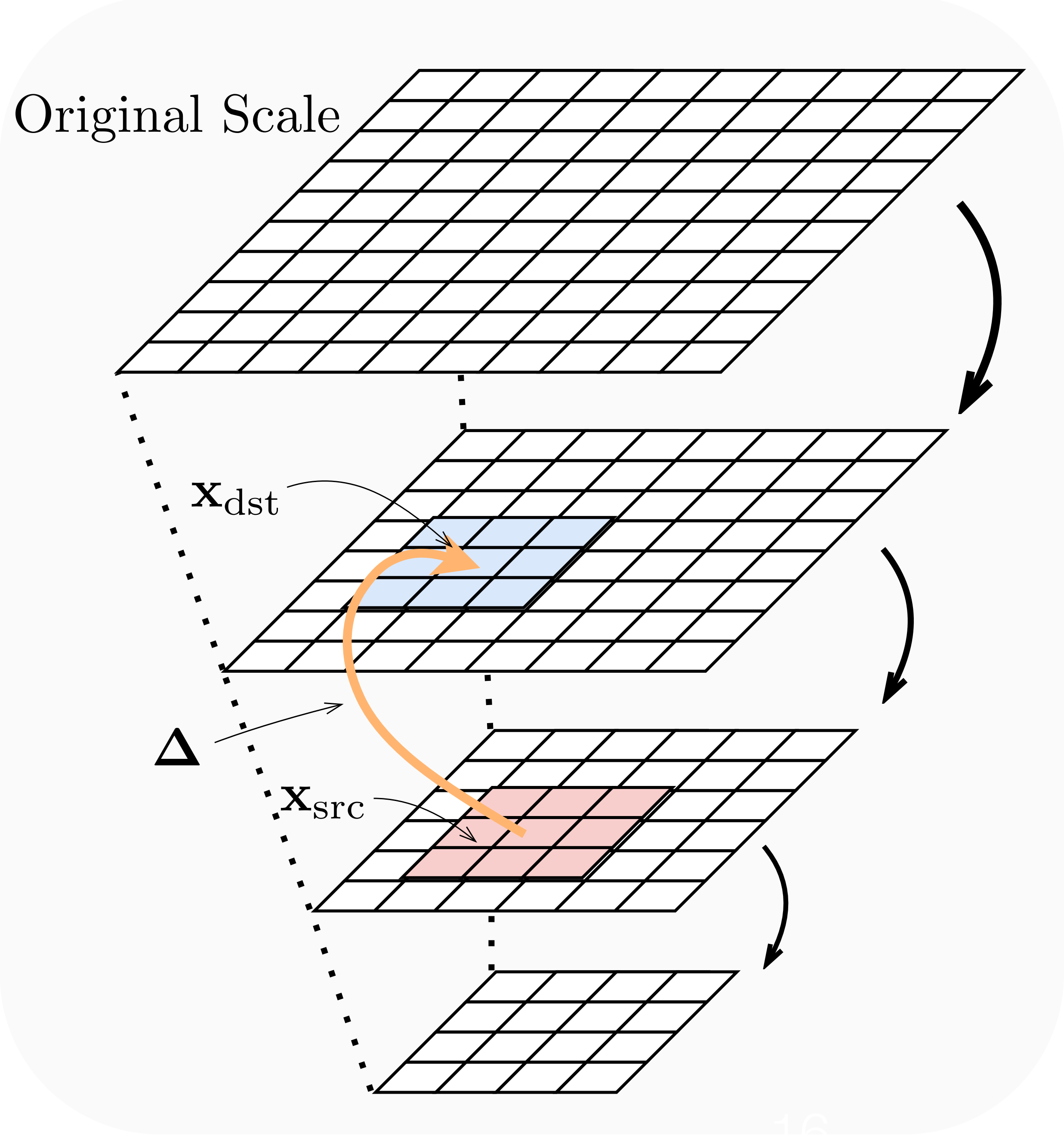
$$(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\text{dst}})$$

$$\boxed{(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}) \mapsto \mathbf{x}_{\text{dst}} \quad p(\mathbf{x}_{\text{dst}} \mid \mathbf{x}_{\text{src}}, \boldsymbol{\Delta})}$$

Our Approach — Data Representation

Joint distribution of (patch1, patch2, relationship)

- What makes **a core difference** from prior work is the joint distribution of two patches and their 'relationship'.



$$\boldsymbol{\Delta} = \begin{bmatrix} \theta_{\text{src}} \\ u \\ v \\ s_x \\ s_y \end{bmatrix} \begin{array}{l} \} \text{ rotation degree} \\ \left.\begin{array}{l} \\ \end{array}\right\} \text{displacements} \\ \left.\begin{array}{l} \\ \end{array}\right\} \text{scale} \end{array}$$

$$(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\text{dst}})$$

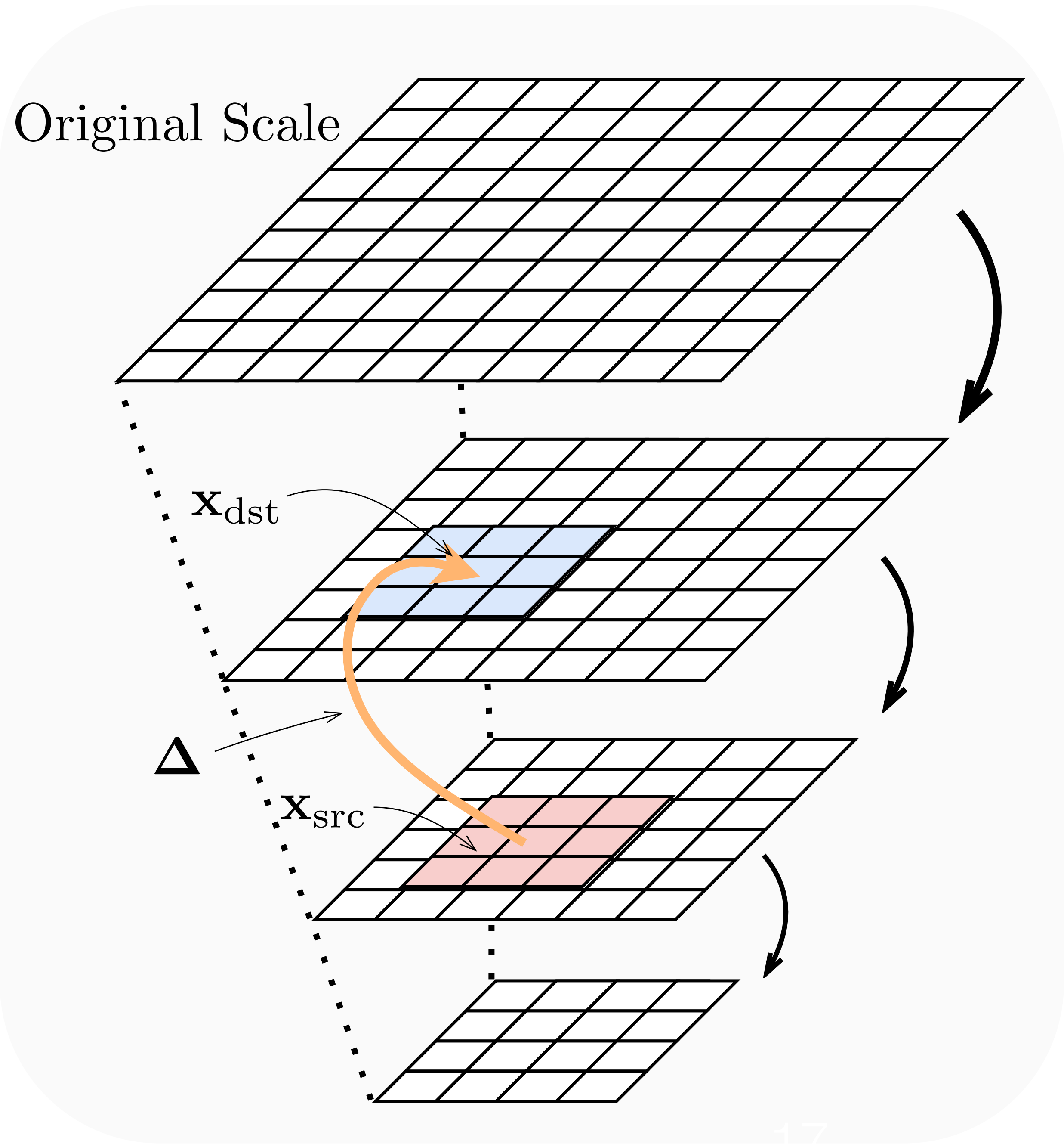
$$(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}) \mapsto \mathbf{x}_{\text{dst}} \quad p(\mathbf{x}_{\text{dst}} \mid \mathbf{x}_{\text{src}}, \boldsymbol{\Delta})$$

$$\boxed{(\mathbf{x}_{\text{src}}, \mathbf{x}_{\text{dst}}) \mapsto \boldsymbol{\Delta} \quad p(\boldsymbol{\Delta} \mid \mathbf{x}_{\text{src}}, \mathbf{x}_{\text{dst}})}$$

Our Approach — Data Representation

Joint distribution of (patch1, patch2, relationship)

- What makes **a core difference** from prior work is the joint distribution of two patches and their 'relationship'.



$$\boldsymbol{\Delta} = \begin{bmatrix} \theta_{\mathrm{src}} \\ u \\ v \\ s_x \\ s_y \end{bmatrix} \begin{matrix} \}\ \text{rotation degree} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\}\ \text{displacements} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\}\ \text{scale} \end{matrix}$$

$$(\mathbf{x}_{\mathrm{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\mathrm{dst}})$$

$$(\mathbf{x}_{\mathrm{src}}, \boldsymbol{\Delta}) \mapsto \mathbf{x}_{\mathrm{dst}} \quad p(\mathbf{x}_{\mathrm{dst}} \,|\, \mathbf{x}_{\mathrm{src}}, \boldsymbol{\Delta})$$

$$(\mathbf{x}_{\mathrm{src}}, \mathbf{x}_{\mathrm{dst}}) \mapsto \boldsymbol{\Delta} \quad p(\boldsymbol{\Delta} \,|\, \mathbf{x}_{\mathrm{src}}, \mathbf{x}_{\mathrm{dst}})$$

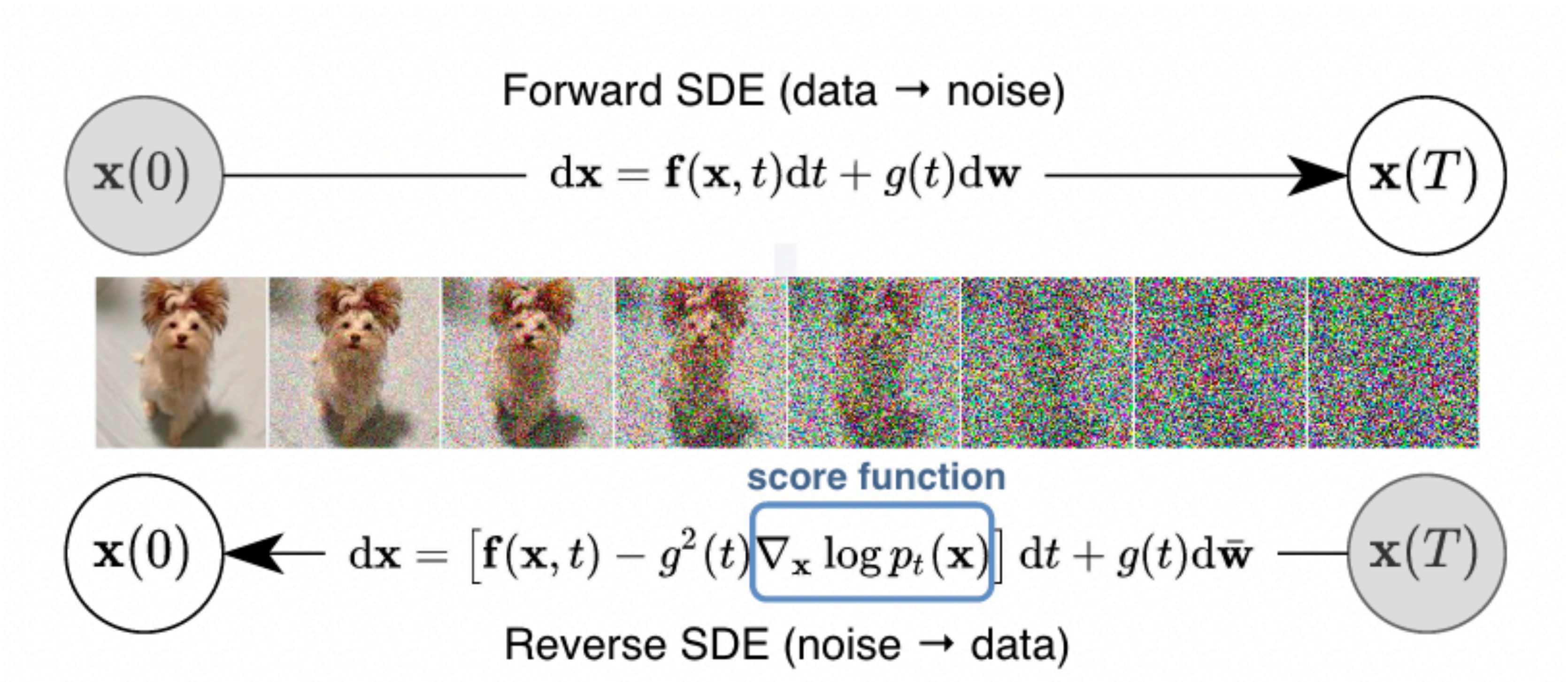
Our Approach

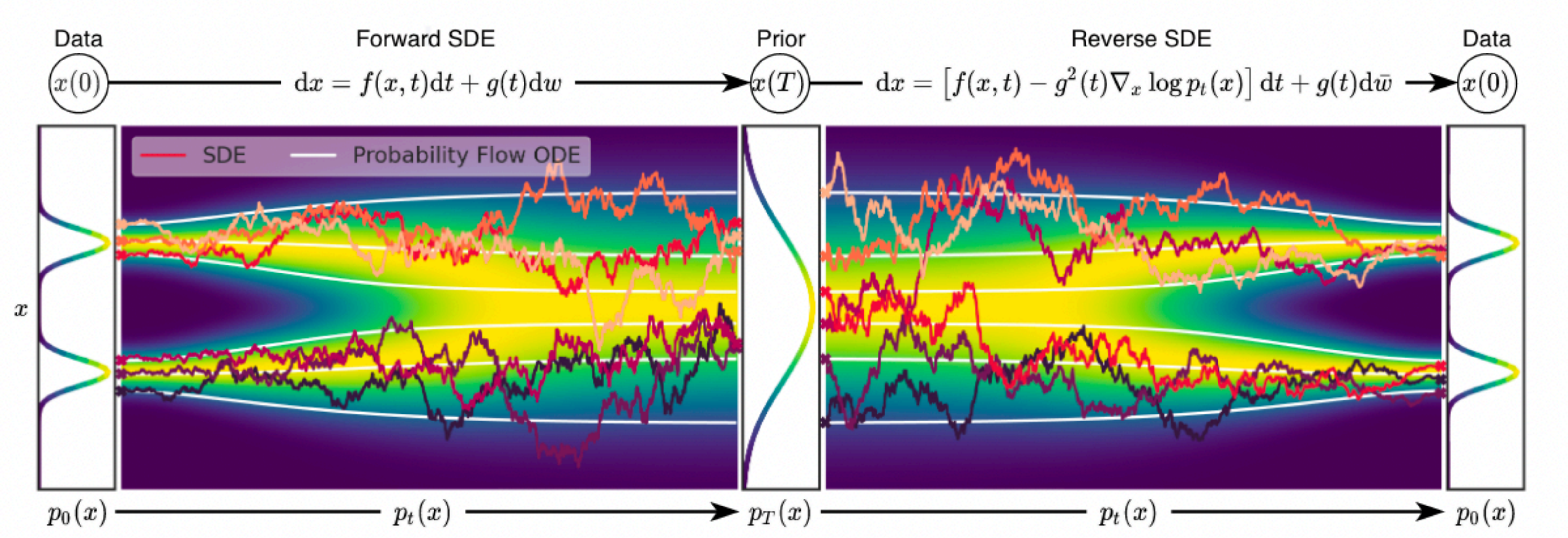
- Data representation (what is the distribution)
- Adapt diffusion model for our data representation

Our Approach — Diffusion Model on Vectors

Mathematical framework

- Choose one of the framework out-of-the-box, DDPM, SDE, ...





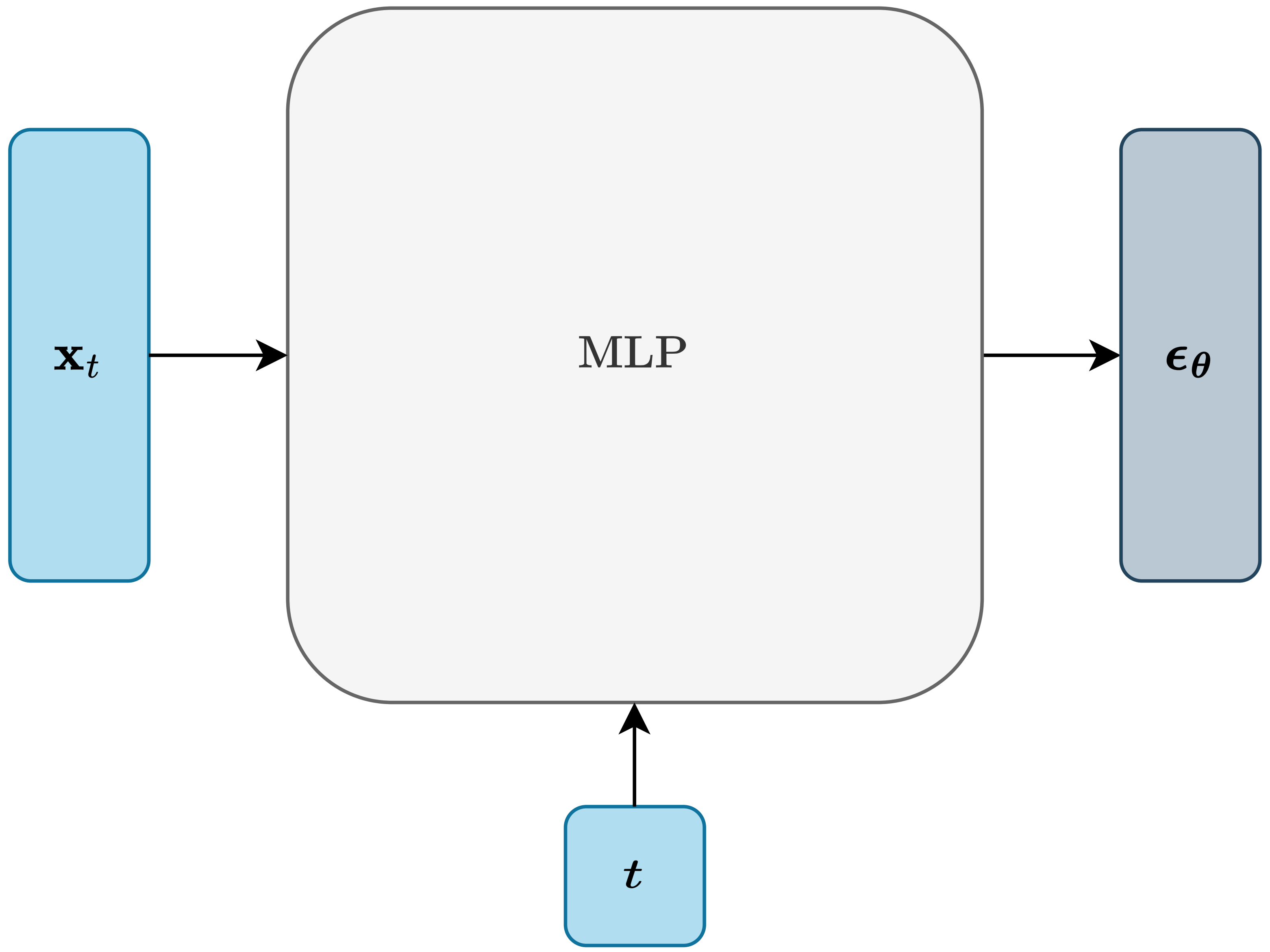
Our Approach — Diffusion Model on Vectors

Change the denoiser

- U-net is for big 2D images.
- Given our data are mostly 'vector-like' (small patch is flattened as a vector), we just use MLP for denoiser

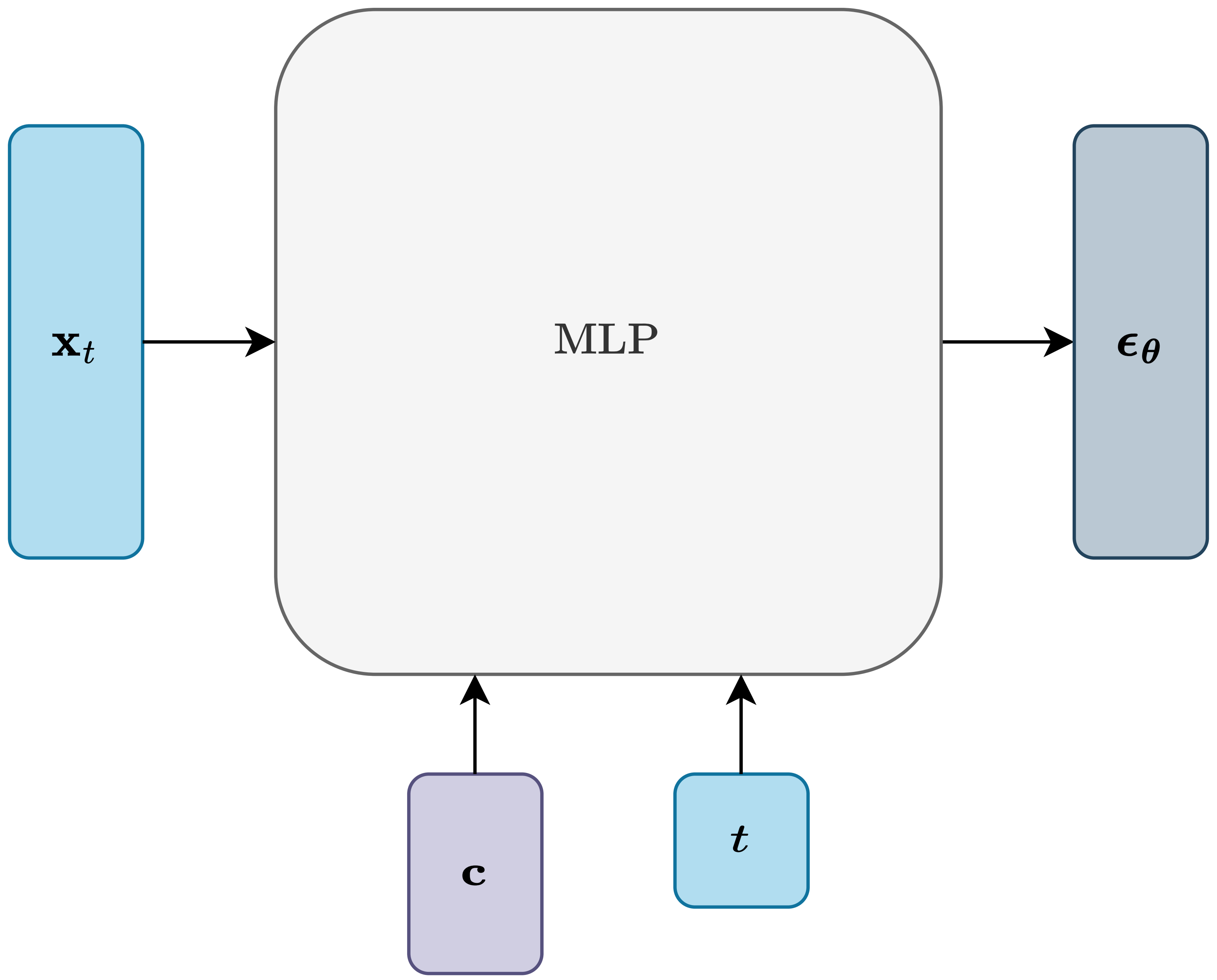
Our Approach — Diffusion Model on Vectors

Unconditional model — $p(\mathbf{x})$



Our Approach — Diffusion Model on Vectors

Conditional model — $p(\mathbf{x} \mid \mathbf{c})$

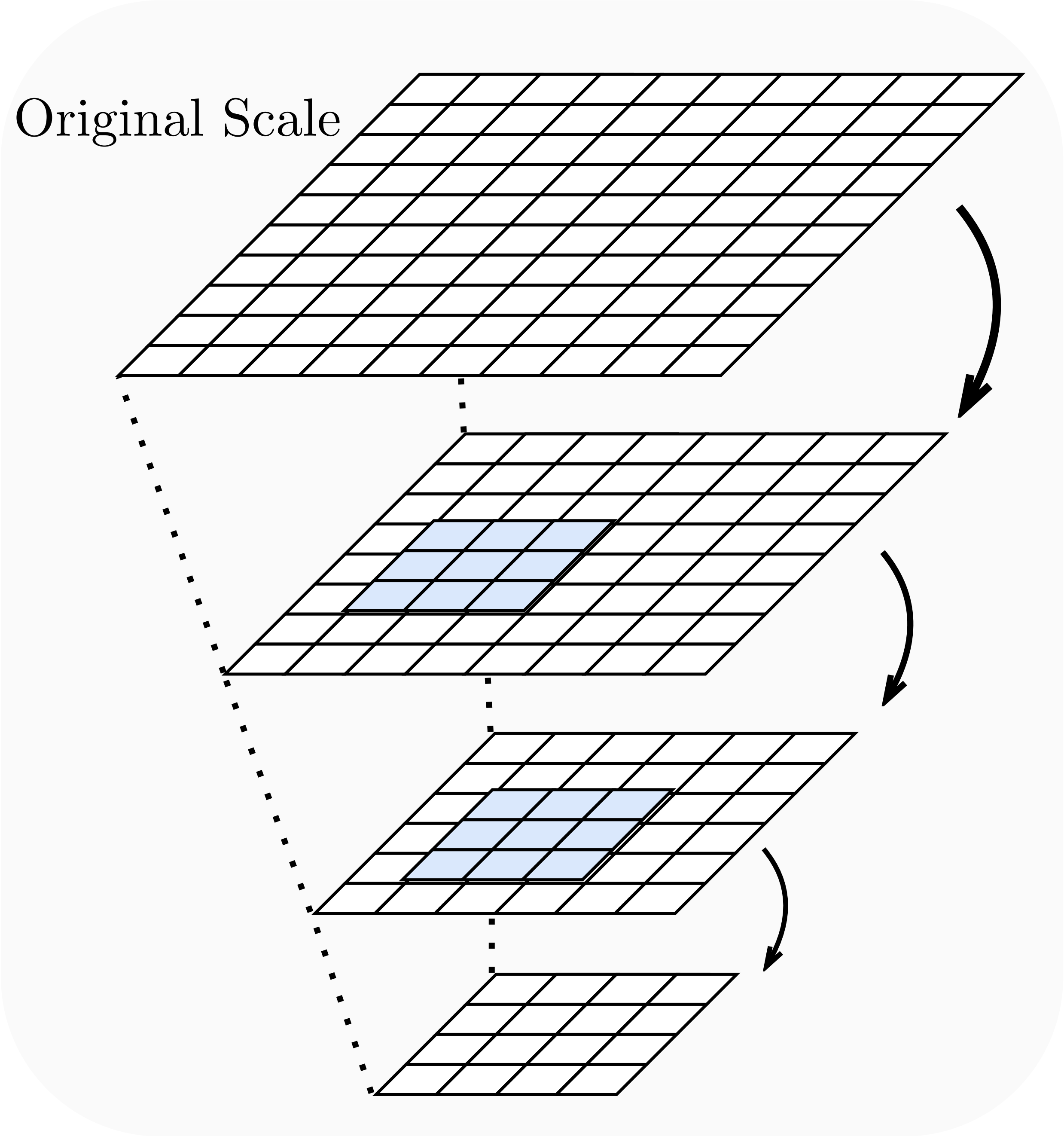


Results

Unconditional Generation

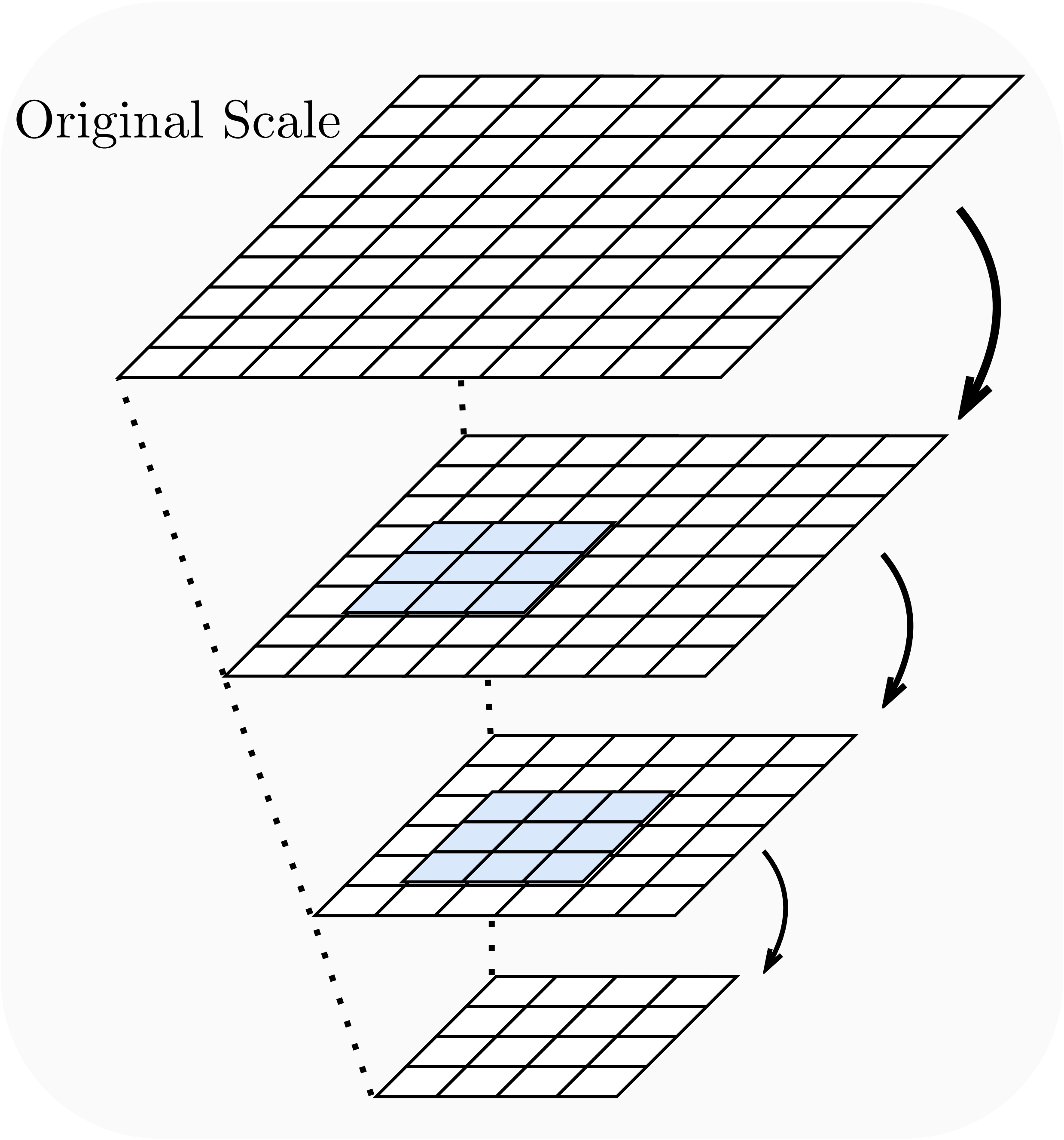
Results

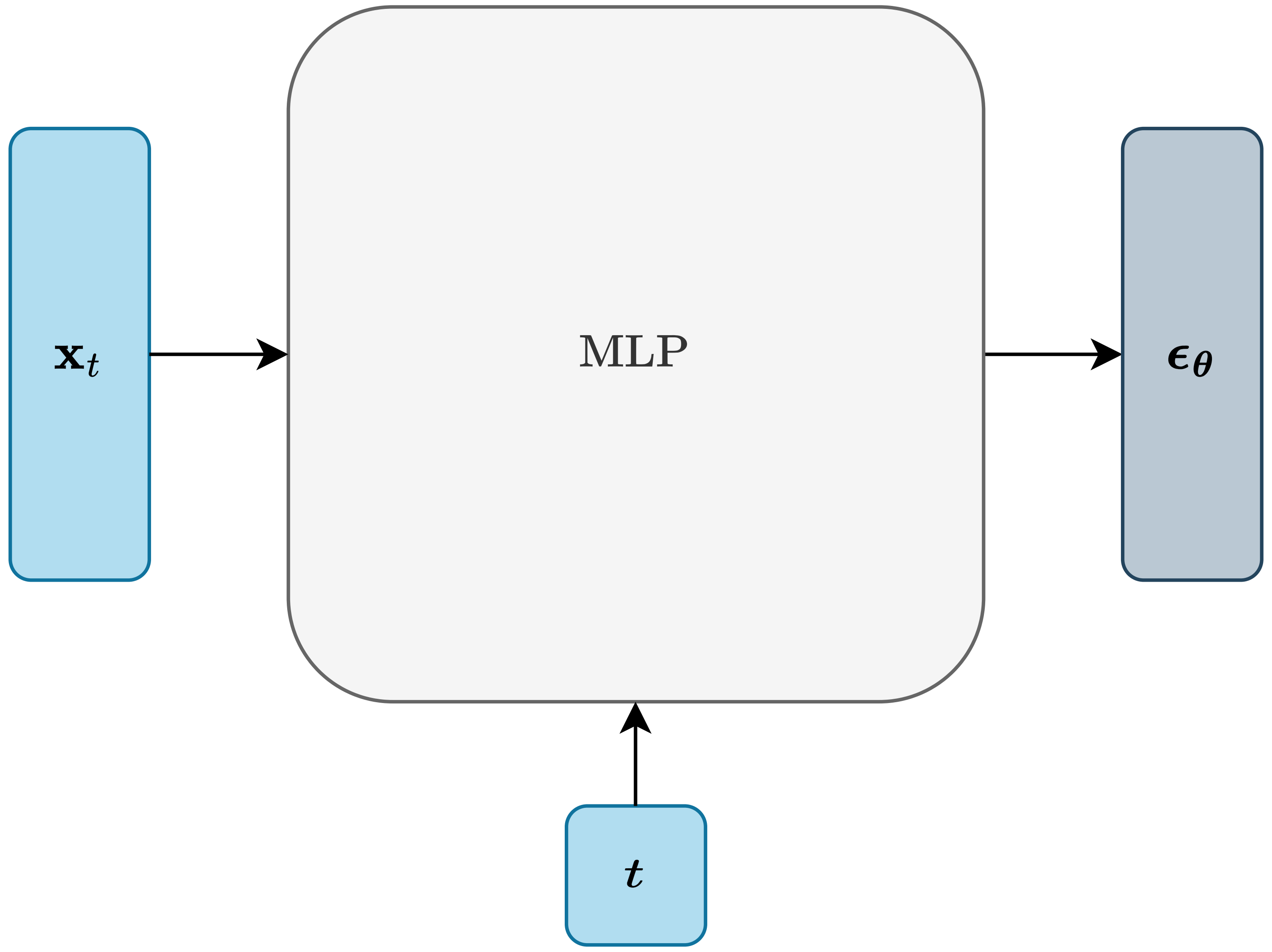
Unconditional Generation



Results

Unconditional Generation

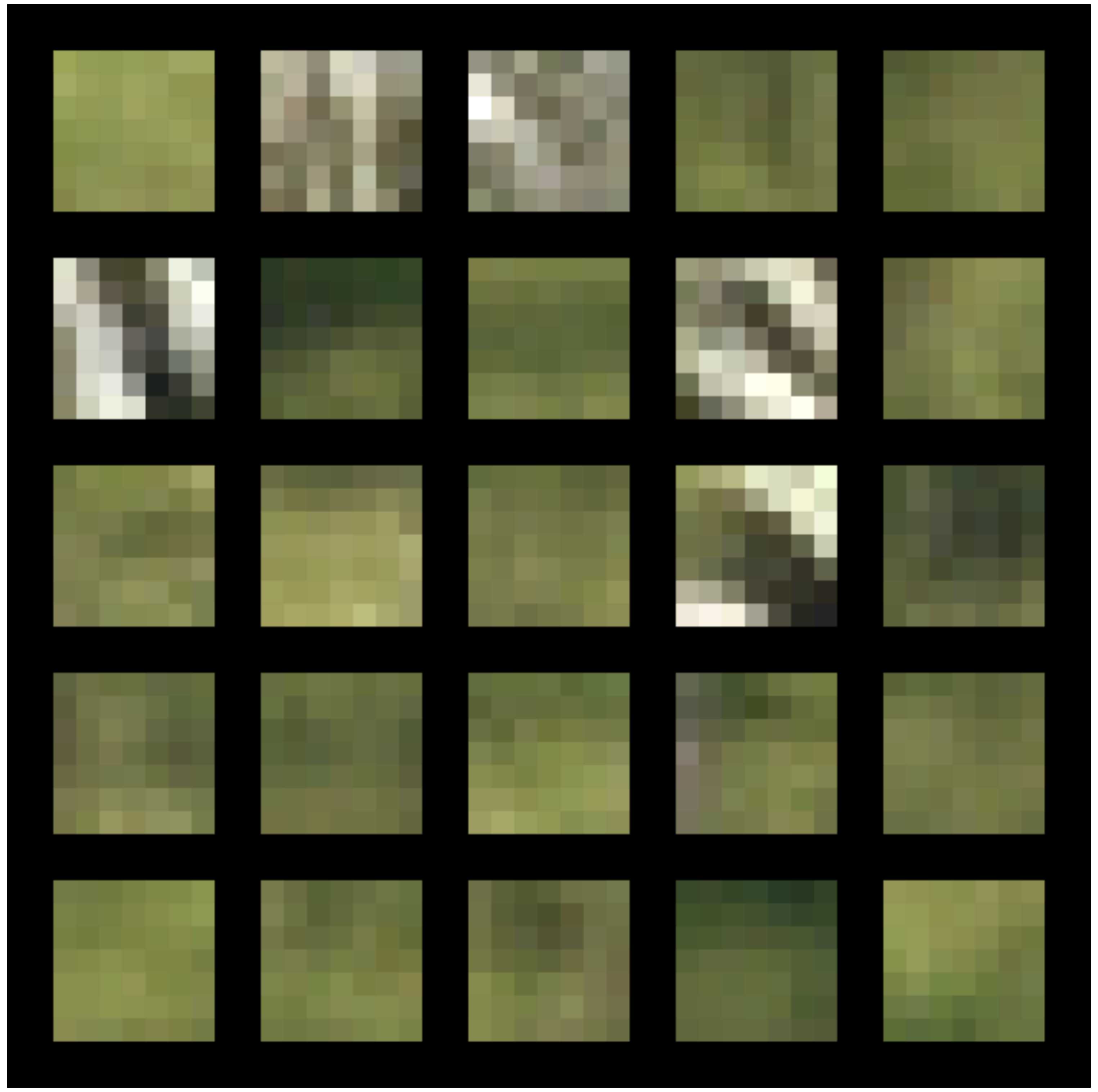




Results

Unconditional Generation

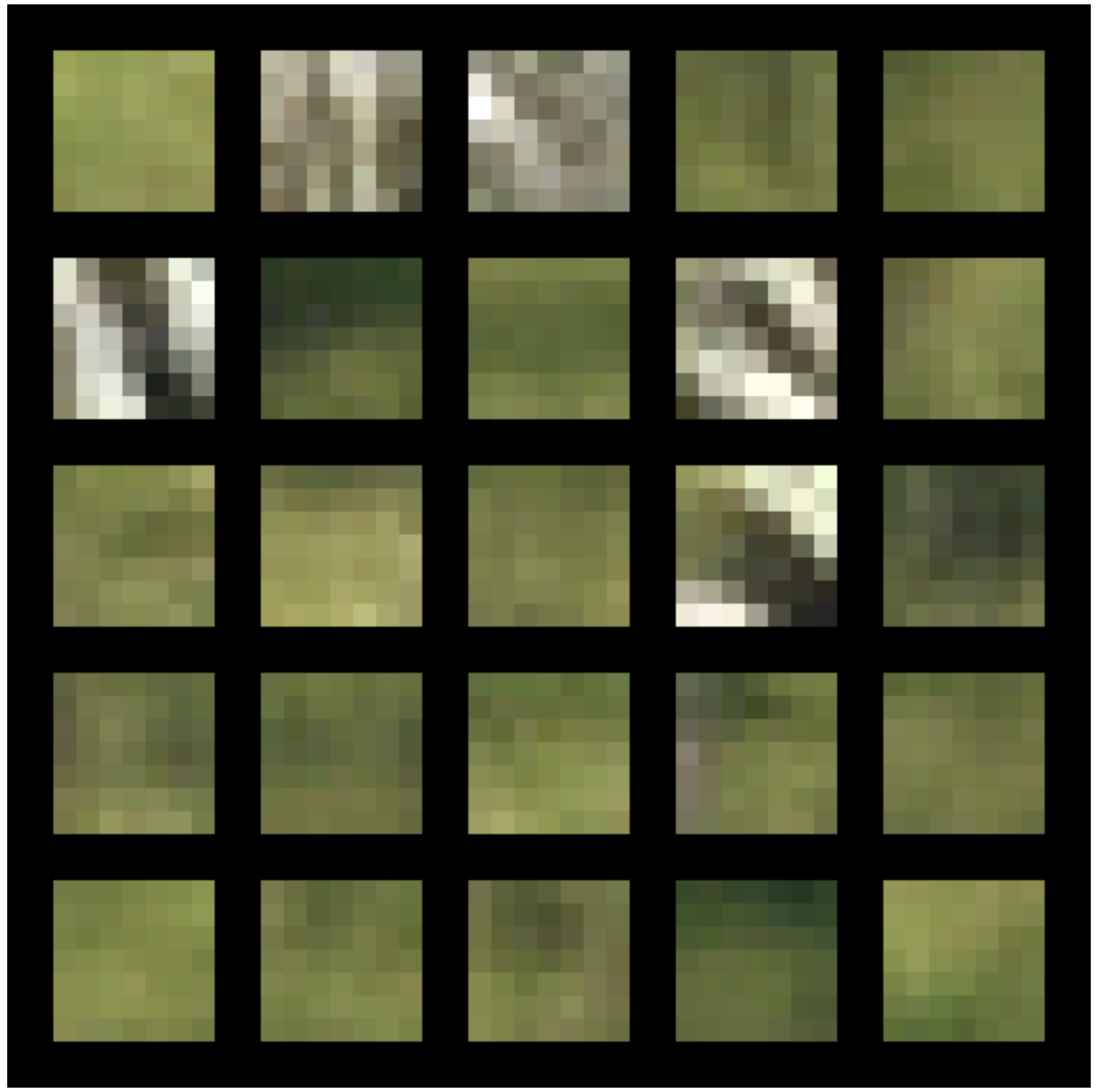
Generated



Results

Unconditional Generation

Generated



Real

Results

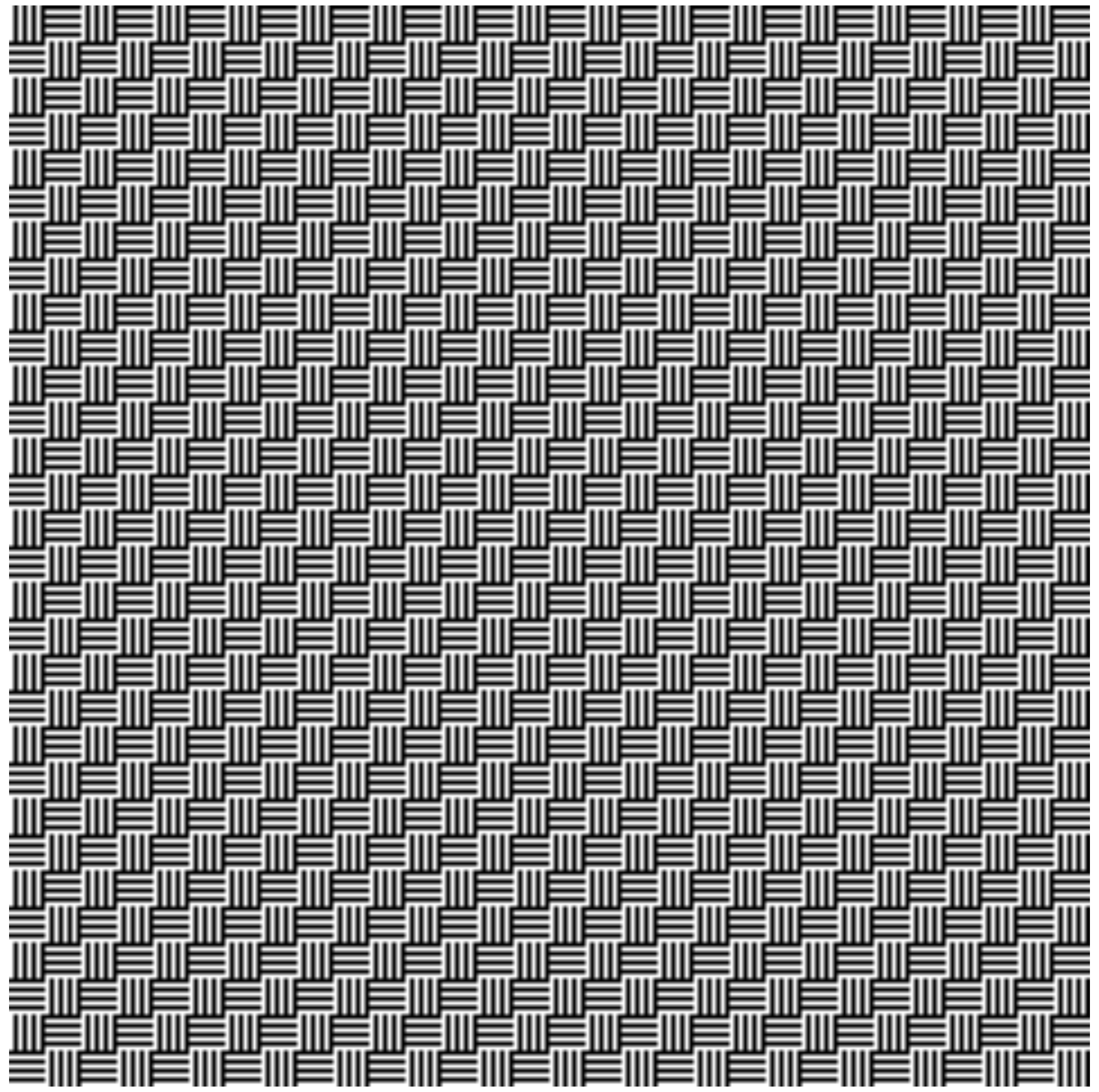
Conditional generation (patch) — synthetic data

$$\boxed{p(\mathbf{x}_{\mathrm{dst}} \mid \mathbf{x}_{\mathrm{src}}, \mathbf{\Delta})}$$

0.5 Scale

model learns conditional distribution (at patch level) →

1.0 Scale



Results

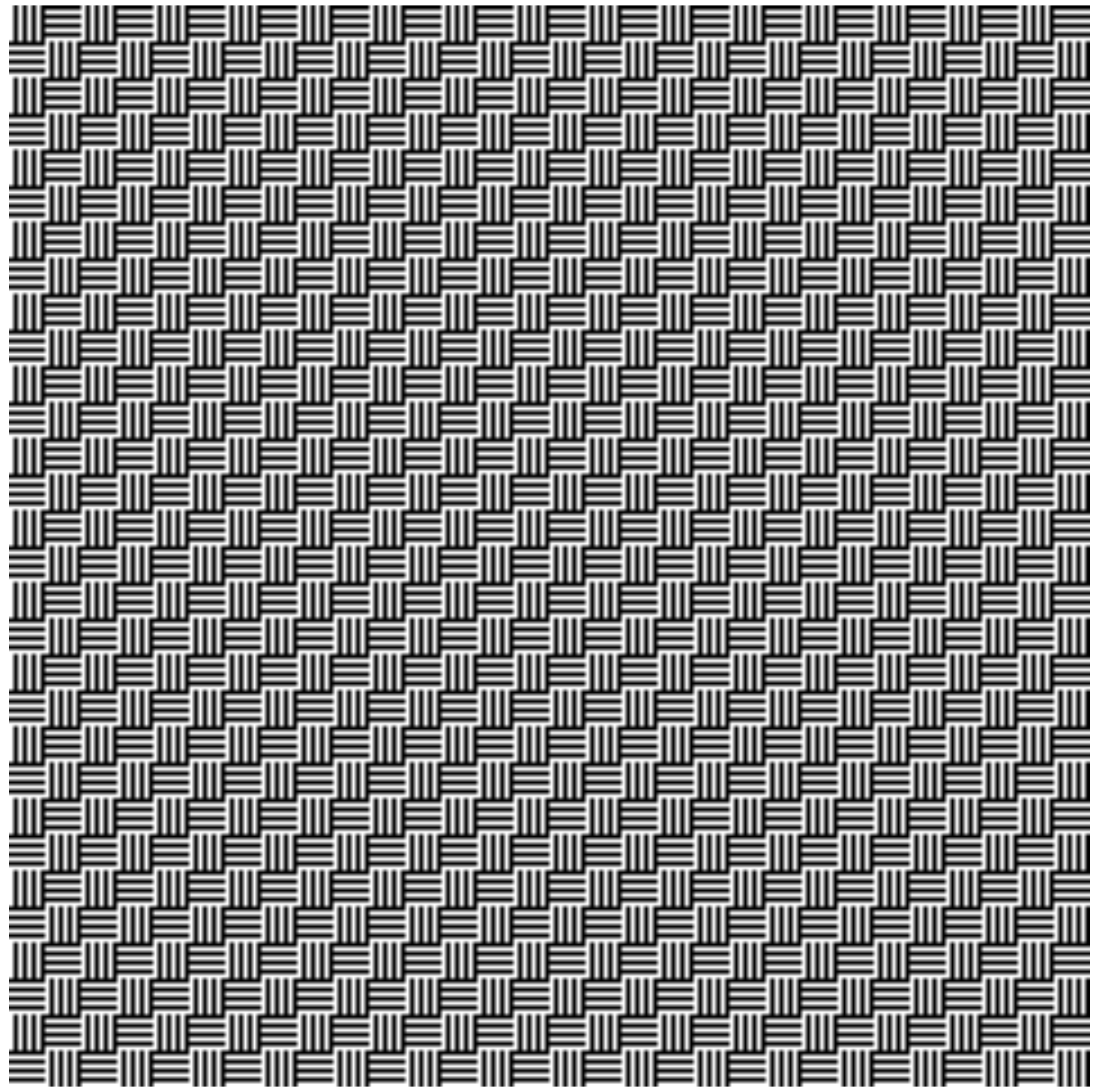
Conditional generation (patch) — synthetic data

$\boxed{p(\mathbf{x}_{\mathrm{dst}} \mid \mathbf{x}_{\mathrm{src}}, \boldsymbol{\Delta})}$

0.5 Scale

model learns conditional distribution (at patch level) $\longrightarrow$

1.0 Scale



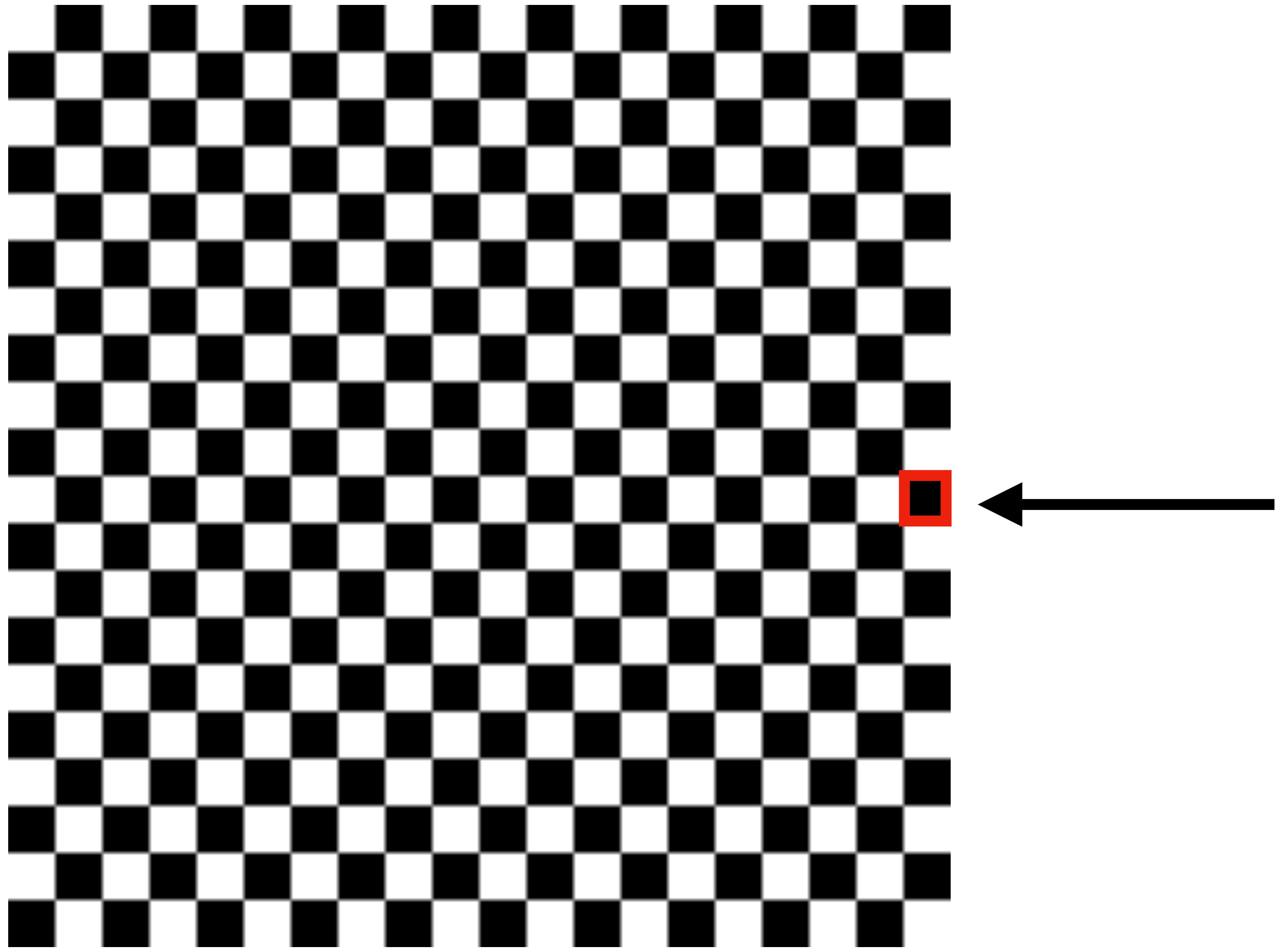
$\mathbf{x}_{\mathrm{src}}$ $\quad$ $\overbrace{\qquad\qquad\qquad}^{\text{sampled } \mathbf{x}_{\mathrm{dst}}}$

$\boldsymbol{\Delta} = \text{scale up 2x}$

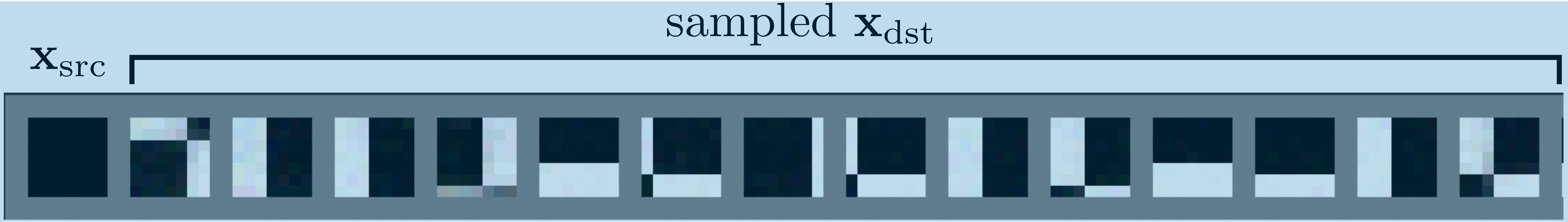
Results

Conditional generation (patch) — synthetic data

$\boxed{p(\mathbf{x}_{\text{dst}} \mid \mathbf{x}_{\text{src}}, \boldsymbol{\Delta})}$



Each block is 10 by 10



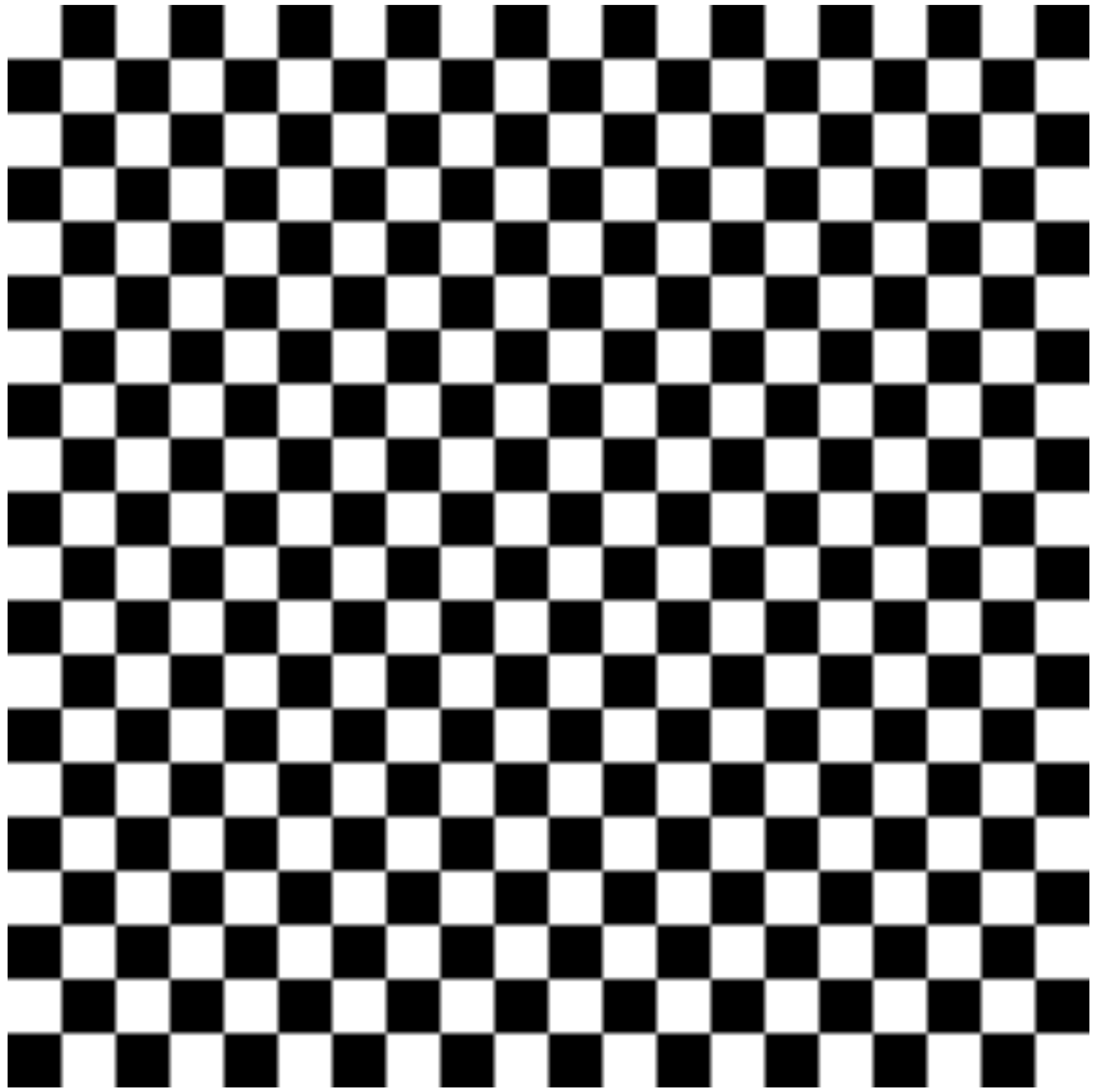
$\boldsymbol{\Delta}$ = displace by (3, 3)

Results

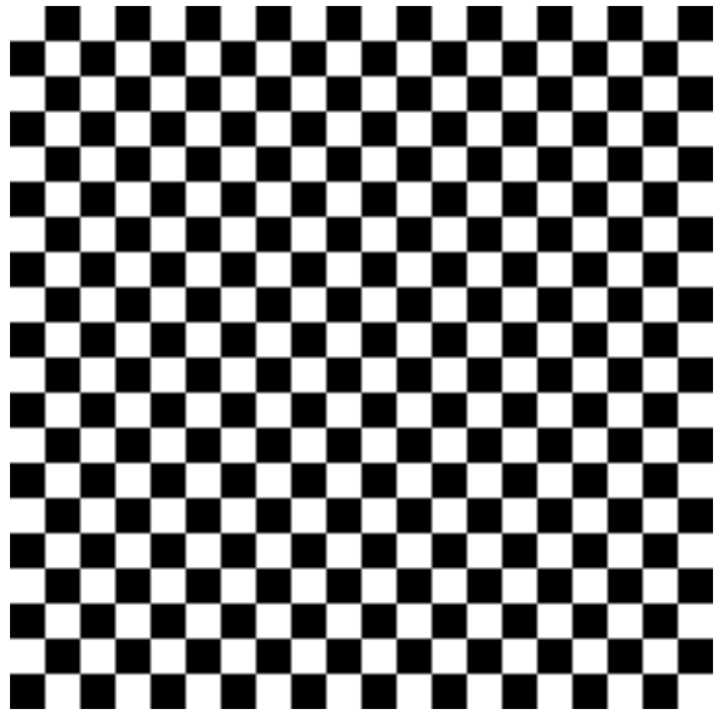
Conditional generation (delta) — synthetic data

$\boxed{p(\boldsymbol{\Delta} \mid \mathbf{x}_{\mathrm{src}}, \mathbf{x}_{\mathrm{dst}})}$

- Checkerboard each block 10 x 10.
- Patch size 7 x 7.
- displacements $= \{-3, -2, -1, 0, 1, 2, 3\}^2$



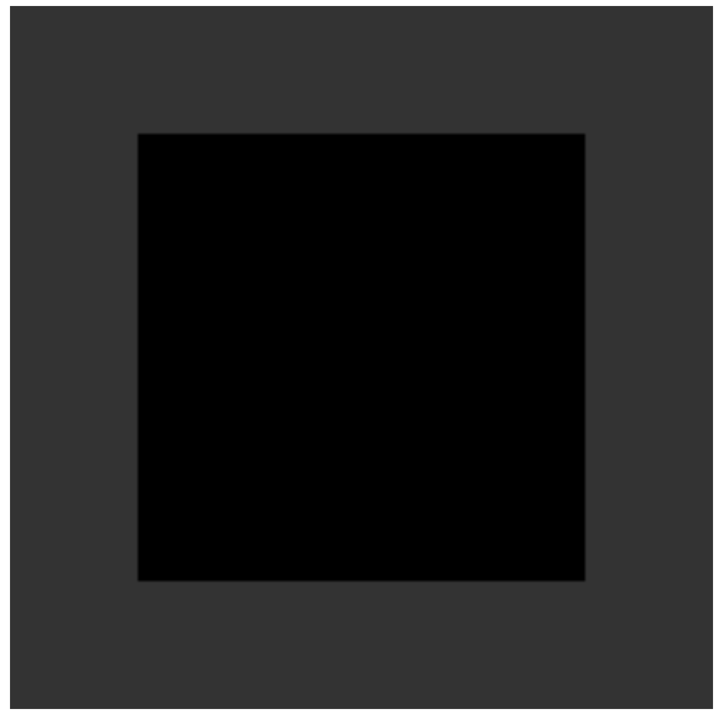
Results



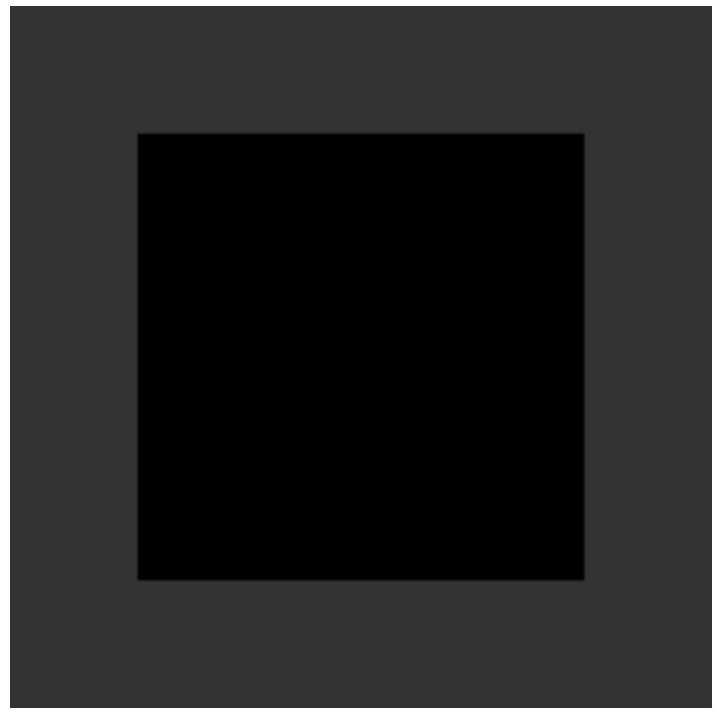
Conditional generation (displacement) — synthetic data

$\boxed{p(\mathbf{\Delta} \,|\, \mathbf{x}_{\mathrm{src}}, \mathbf{x}_{\mathrm{dst}})}$

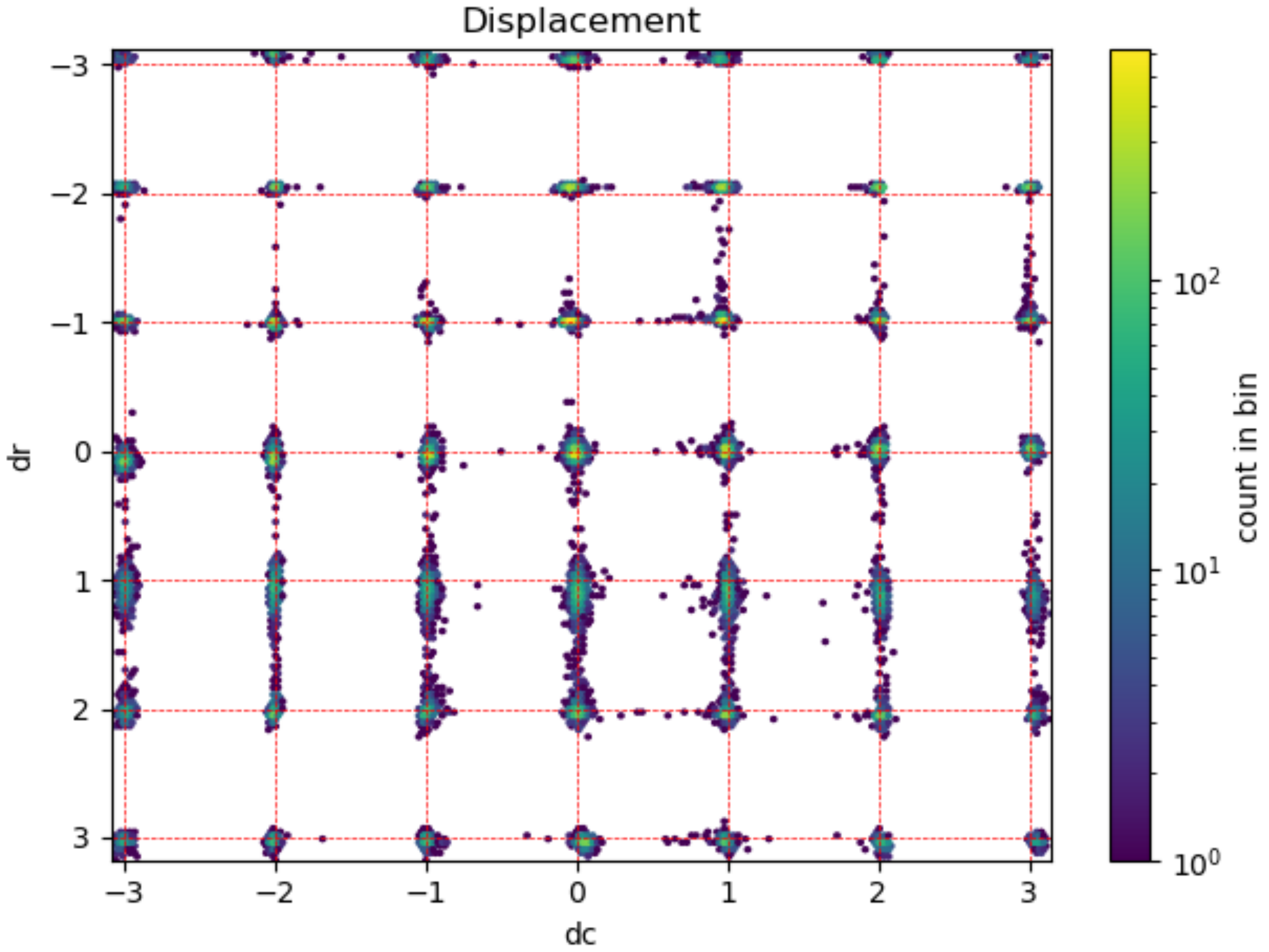
$\mathbf{x}_{\mathrm{src}}$



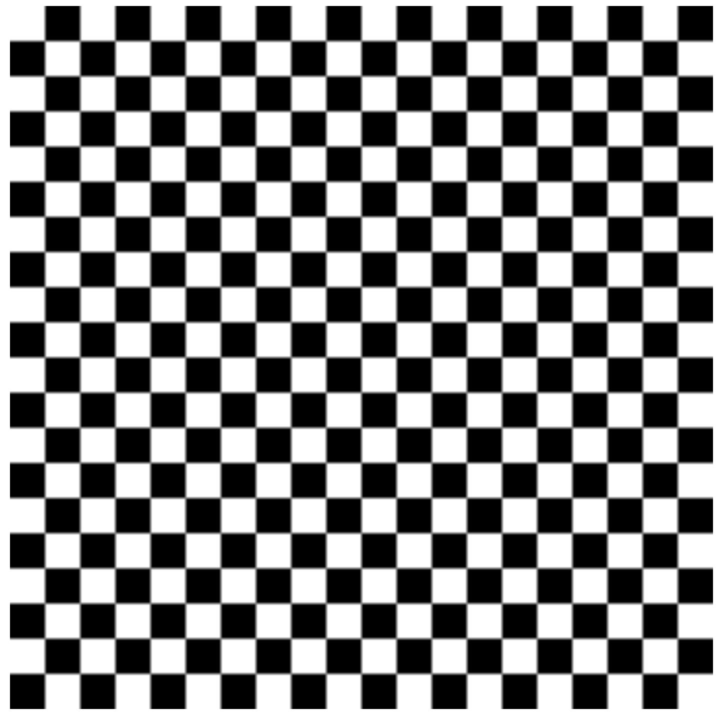
$\mathbf{x}_{\mathrm{dst}}$



2D Histogram of Generated Displacements



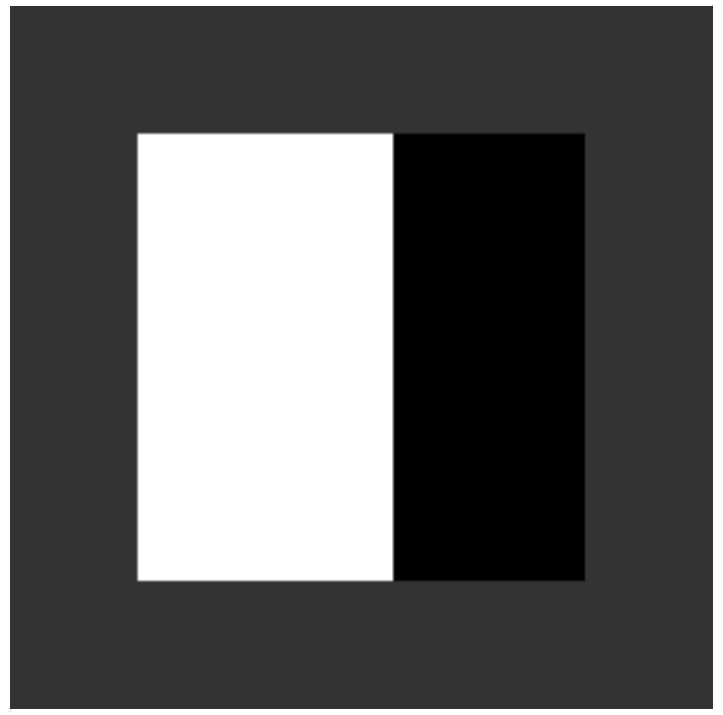
Results



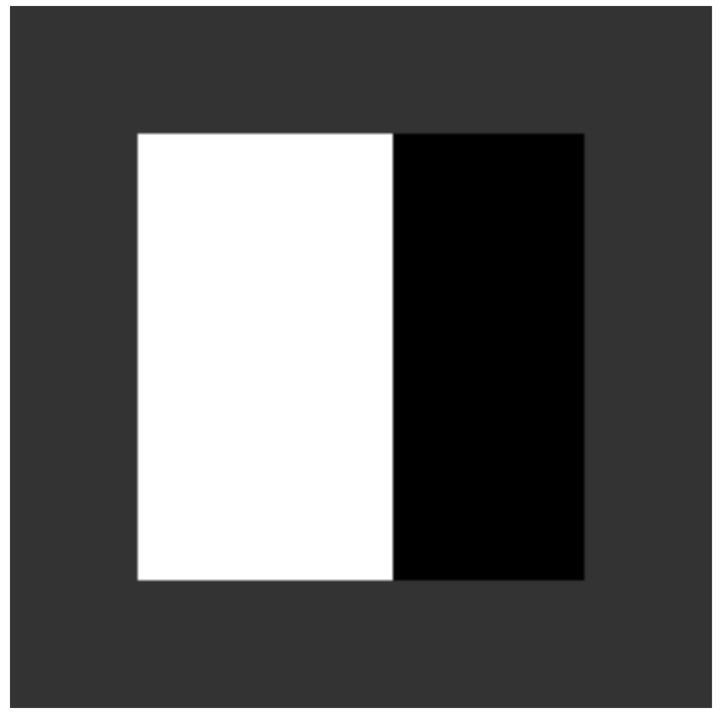
Conditional generation (displacement) — synthetic data

$\boxed{p(\boldsymbol{\Delta} \,|\, \mathbf{x}_{\mathrm{src}}, \mathbf{x}_{\mathrm{dst}})}$

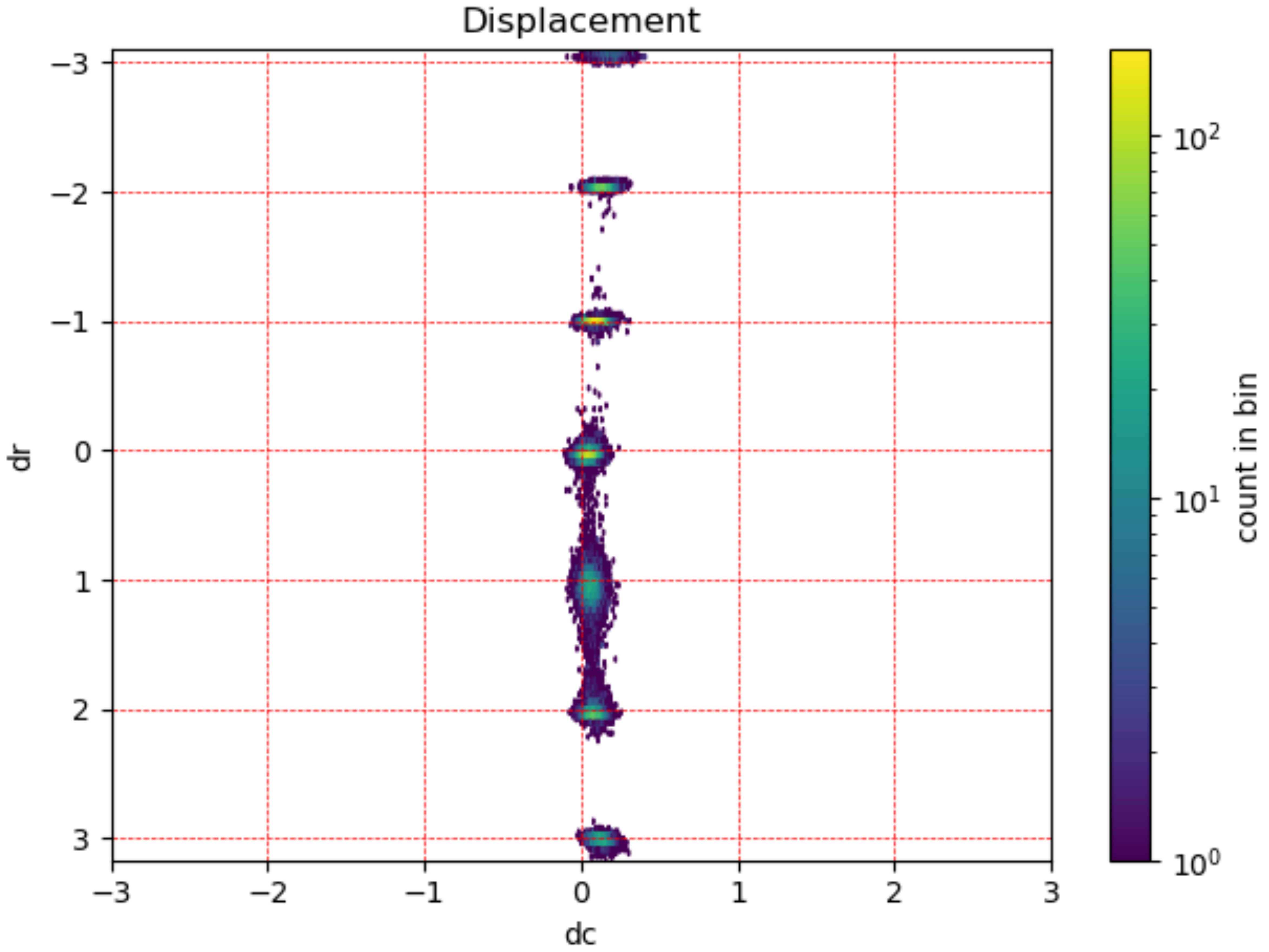
$\mathbf{x}_{\mathrm{src}}$



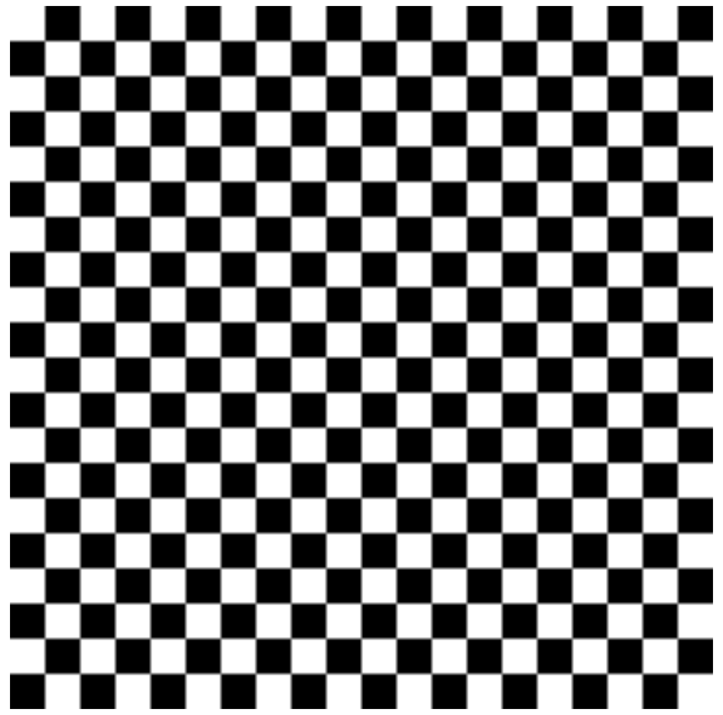
$\mathbf{x}_{\mathrm{dst}}$



2D Histogram of Generated Displacements



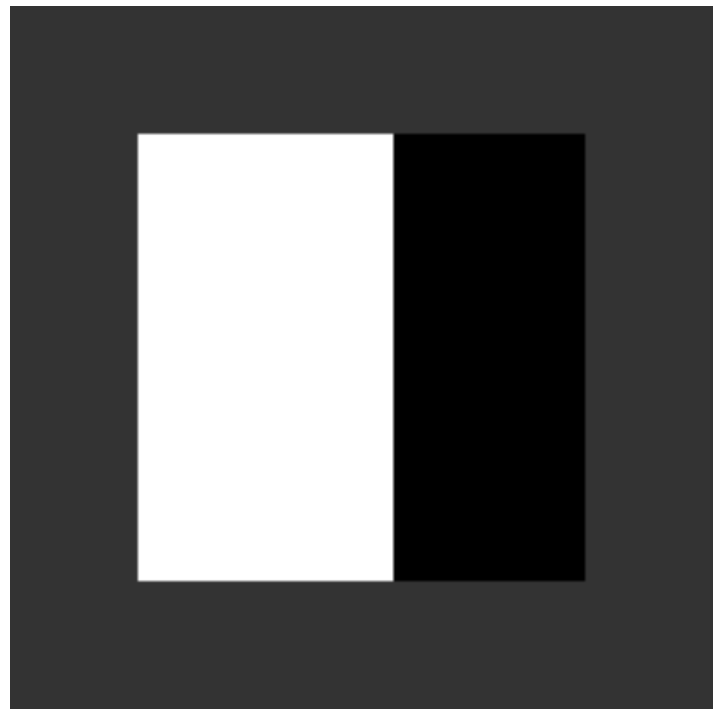
Results



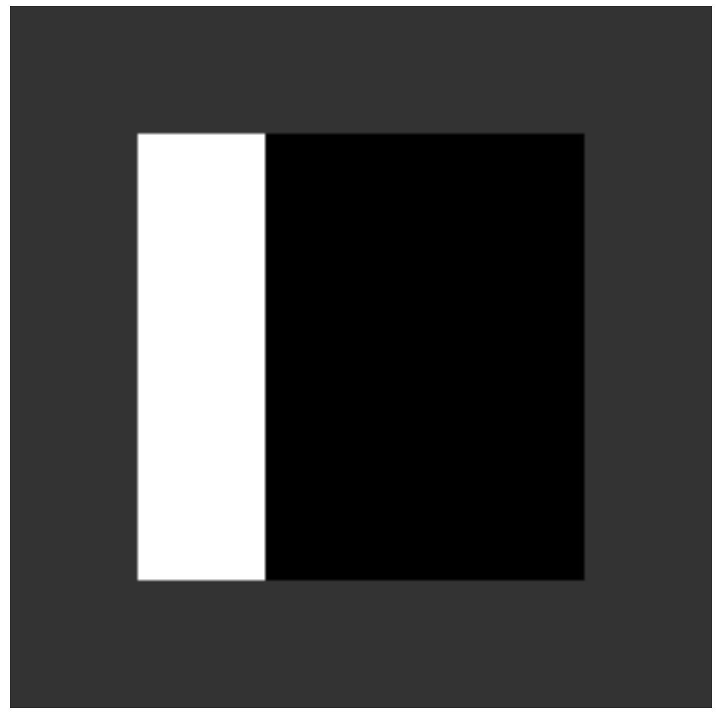
Conditional generation (displacement) — synthetic data

$\boxed{p(\boldsymbol{\Delta} \mid \mathbf{x}_{\mathrm{src}}, \mathbf{x}_{\mathrm{dst}})}$

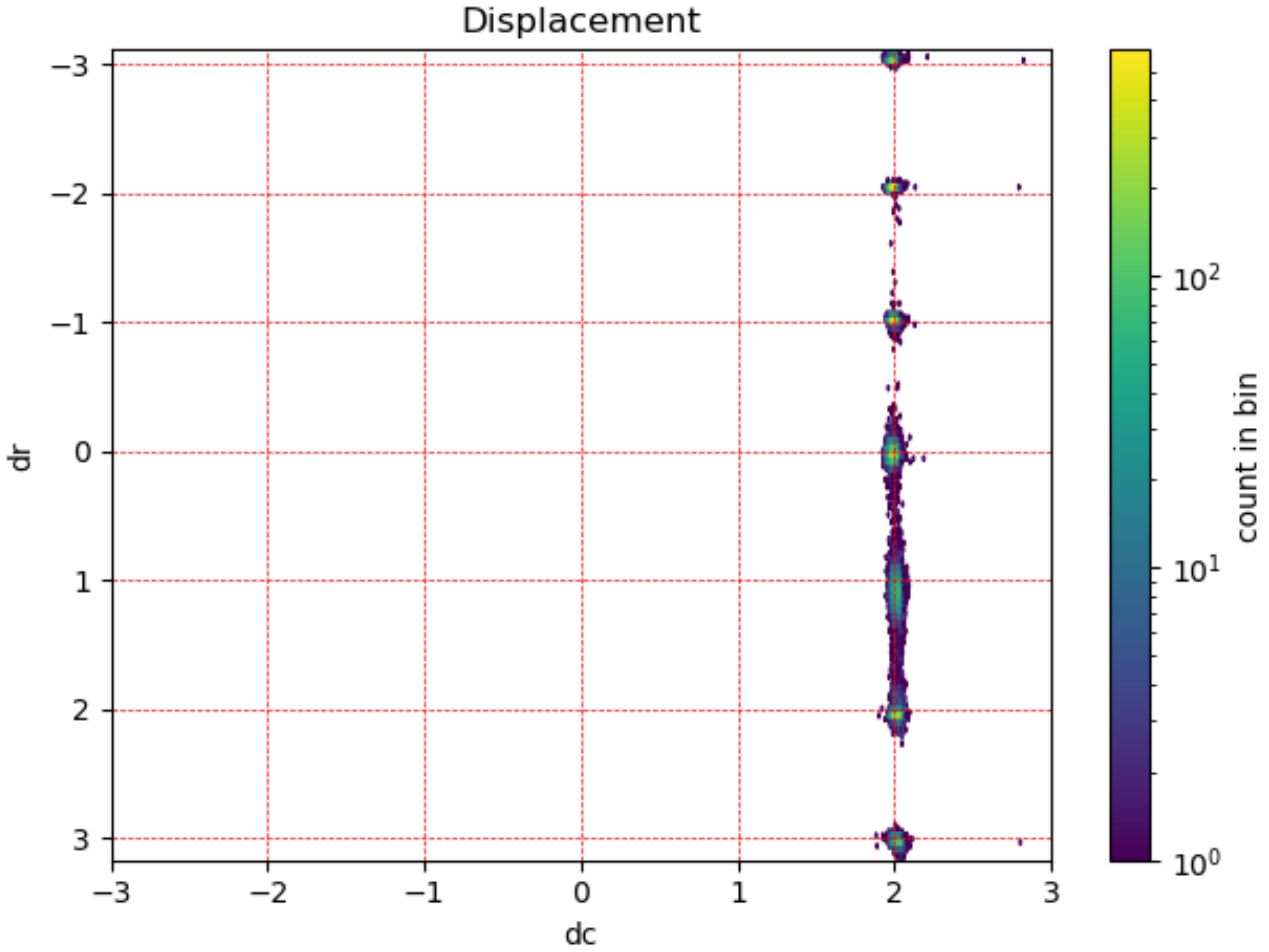
$\mathbf{x}_{\mathrm{src}}$



$\mathbf{x}_{\mathrm{dst}}$



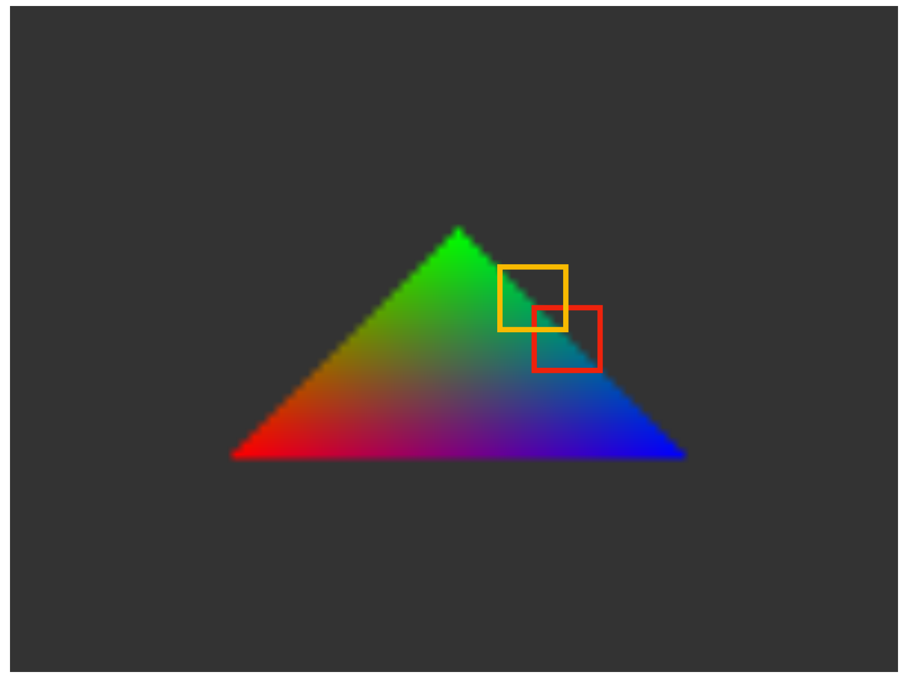
2D Histogram of Generated Displacements



Results

Conditional generation (displacement) — synthetic data

training frame

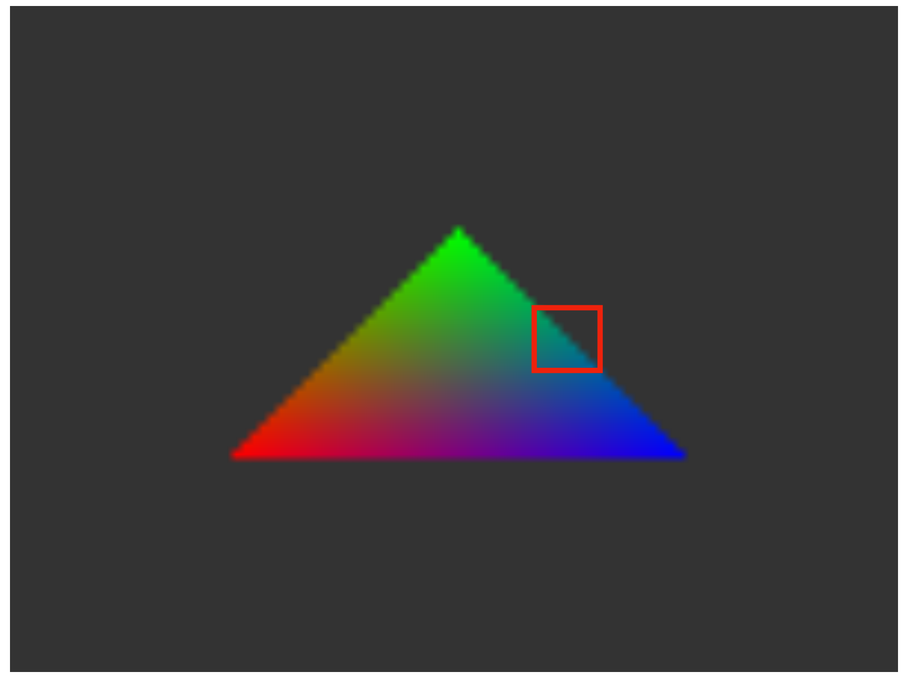


$$p(\underbrace{\boldsymbol{\Delta}}_{\text{disp.}} \mid \mathbf{x}_{\text{src}}, \mathbf{x}_{\text{dst}})$$

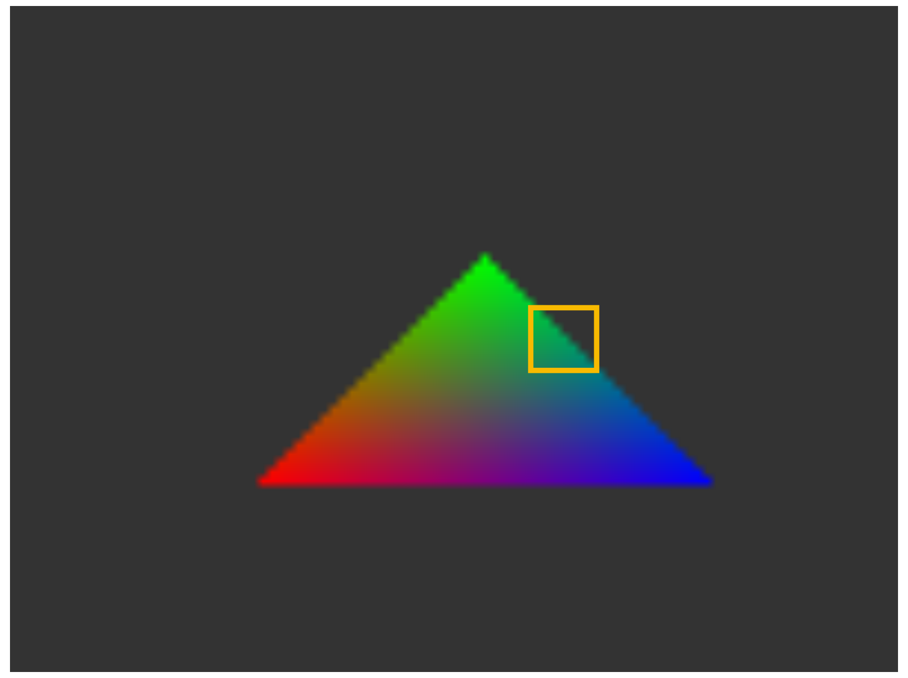
Results

Conditional generation (displacement) — synthetic data

training frame



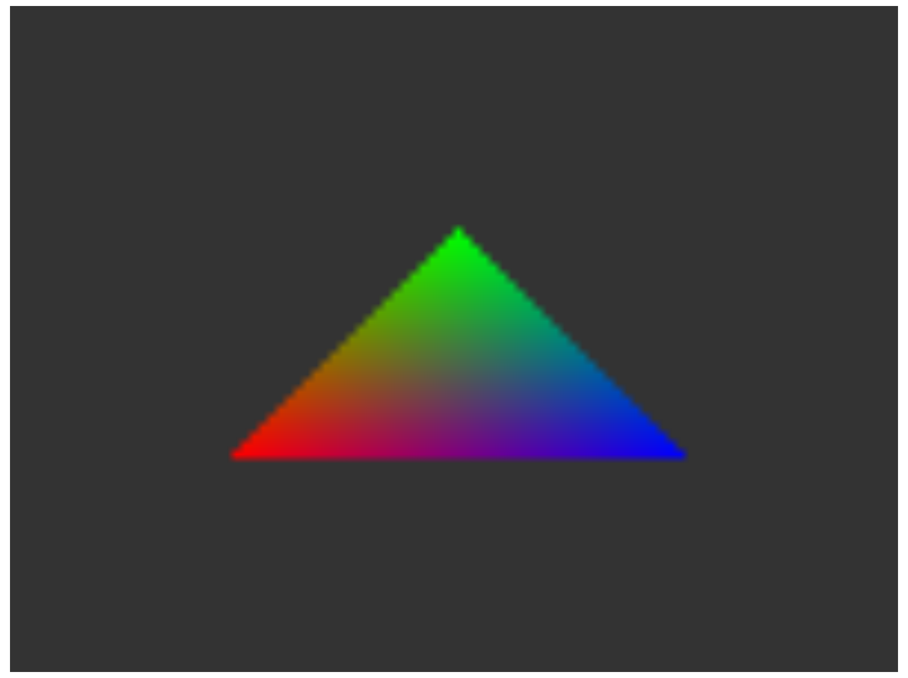
new frame



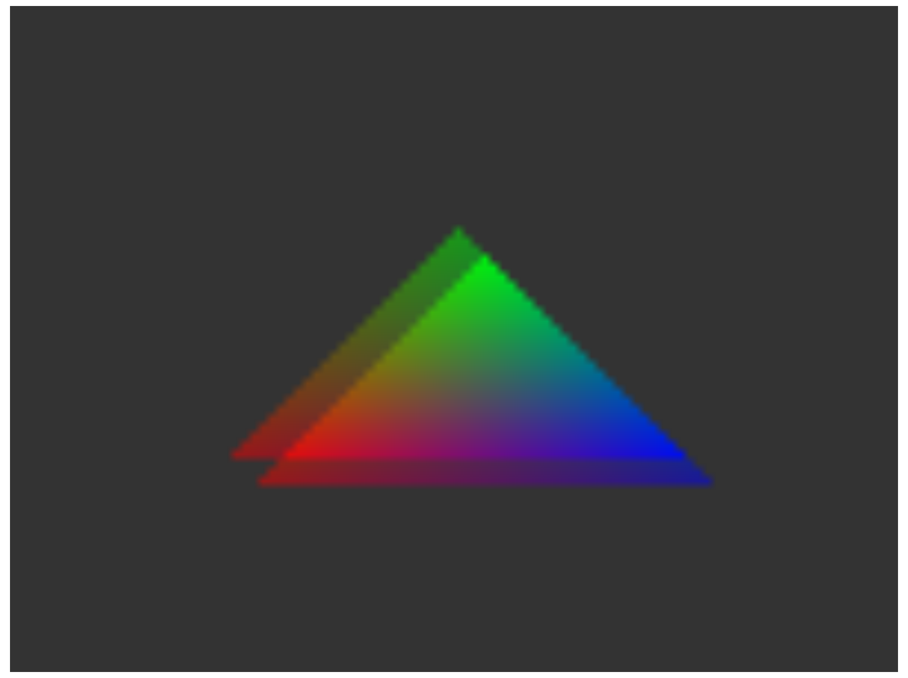
Results

Conditional generation (displacement) — synthetic data

training frame



(train + new) / 2

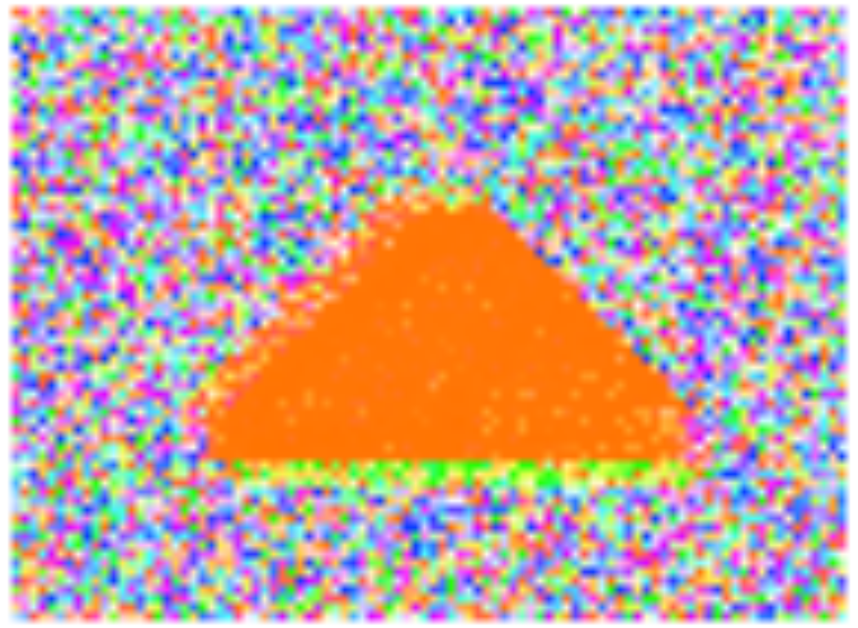


GT flow visualization

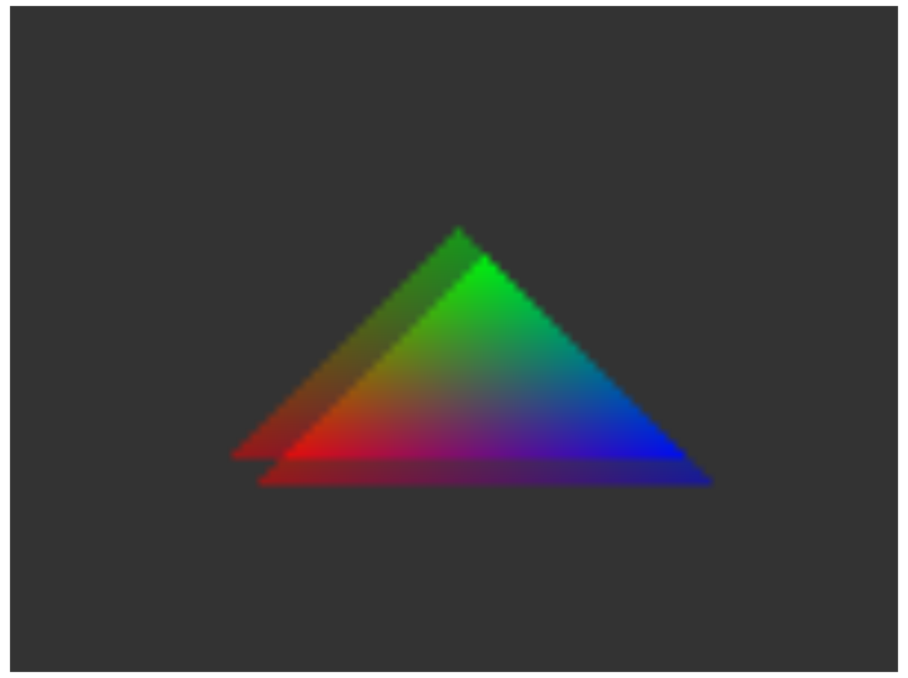
Results

Conditional generation (displacement) — synthetic data

pred (n =1)



(train + new) / 2



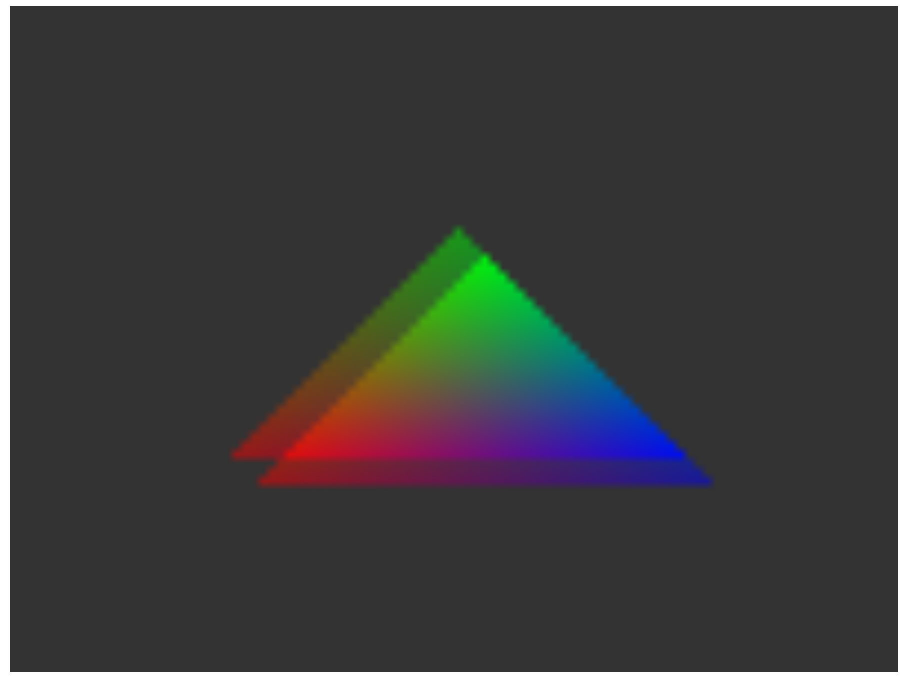
GT flow visualization

Results

Conditional generation (displacement) — synthetic data

mean, n = 10

(train + new) / 2



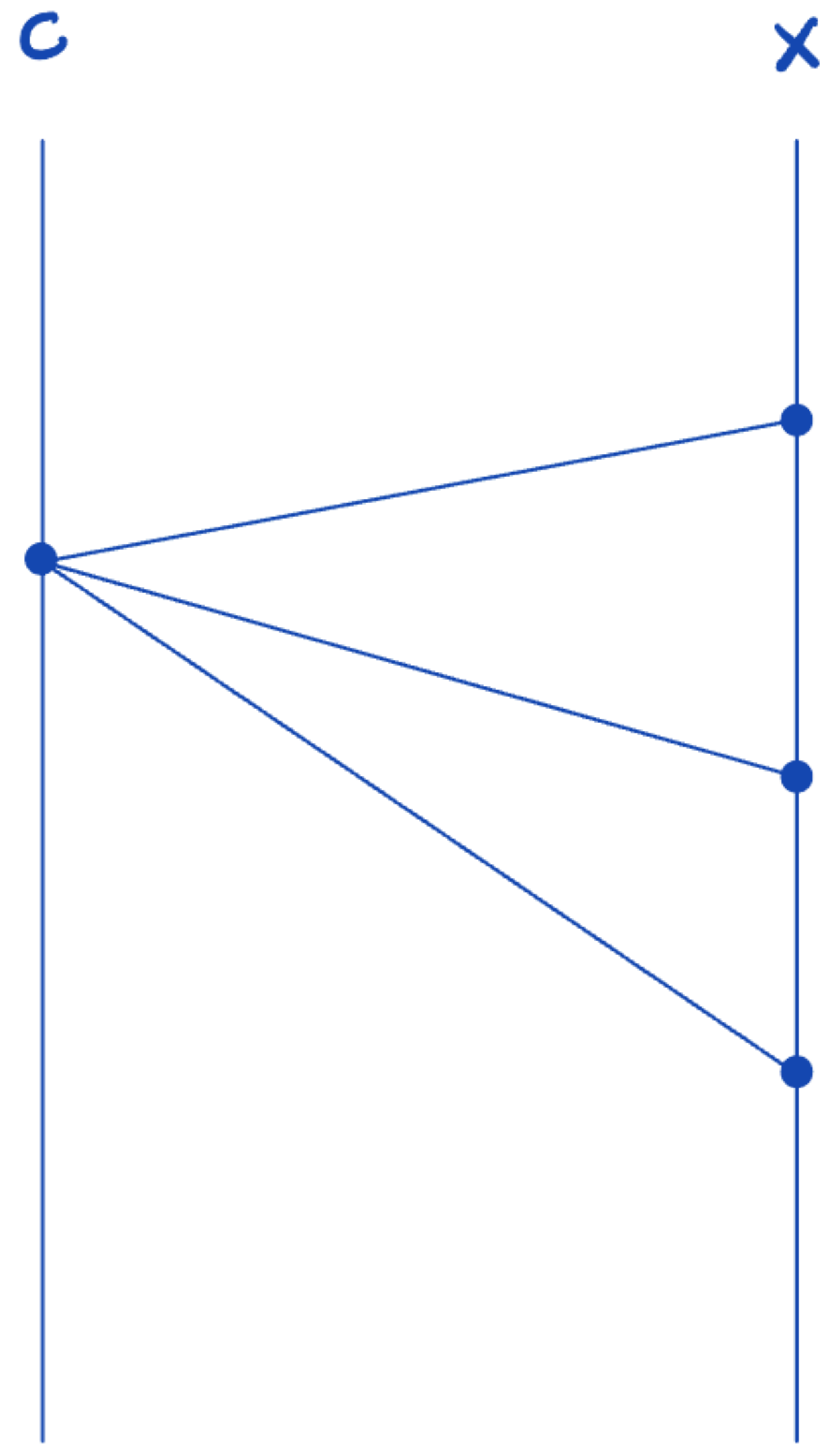
GT flow visualization

Discovery for real image

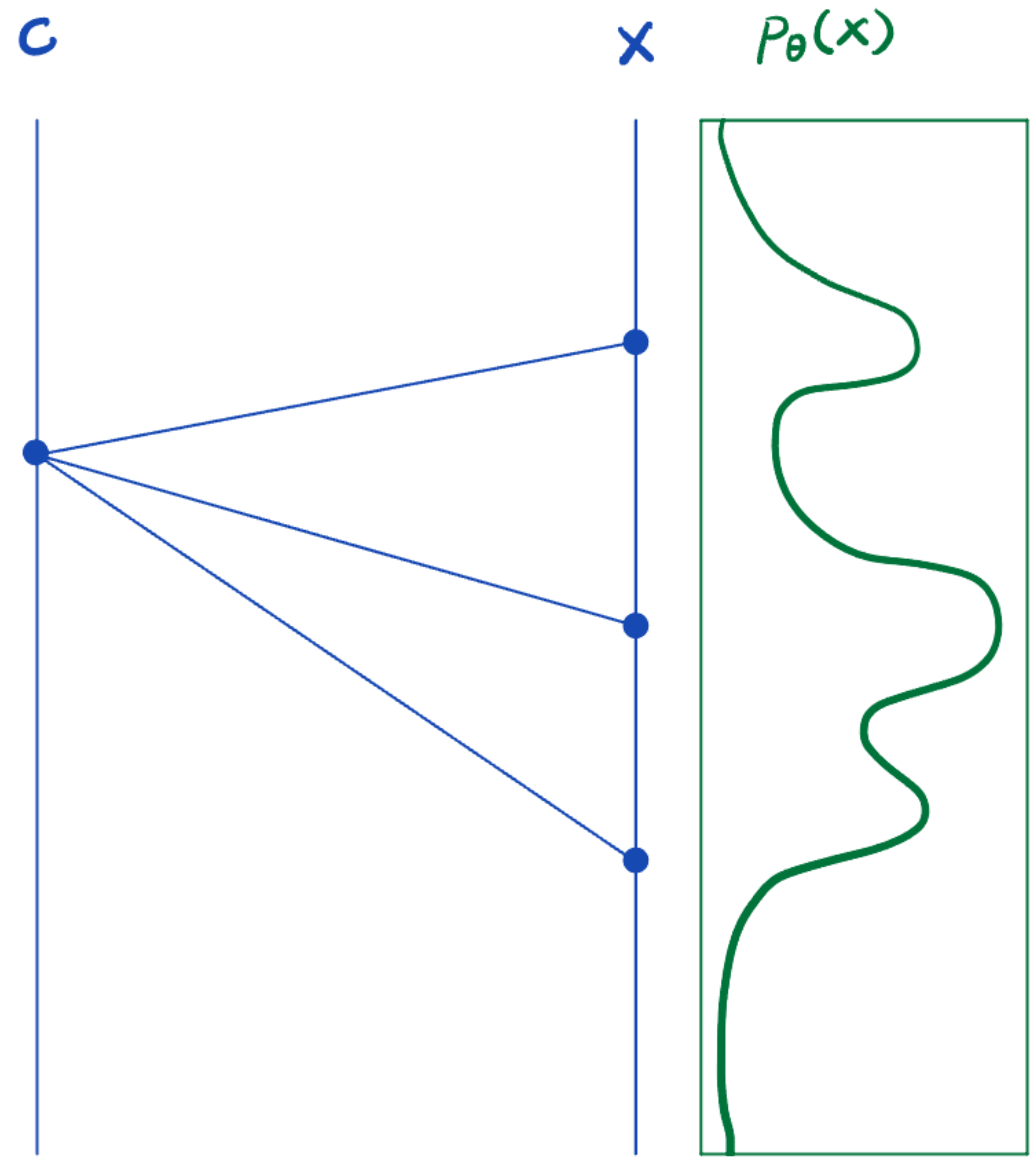
Two problems

- (input space) 'recurrence'/generalization
- (output space) memorization
- both occurs —> memorizing pairs

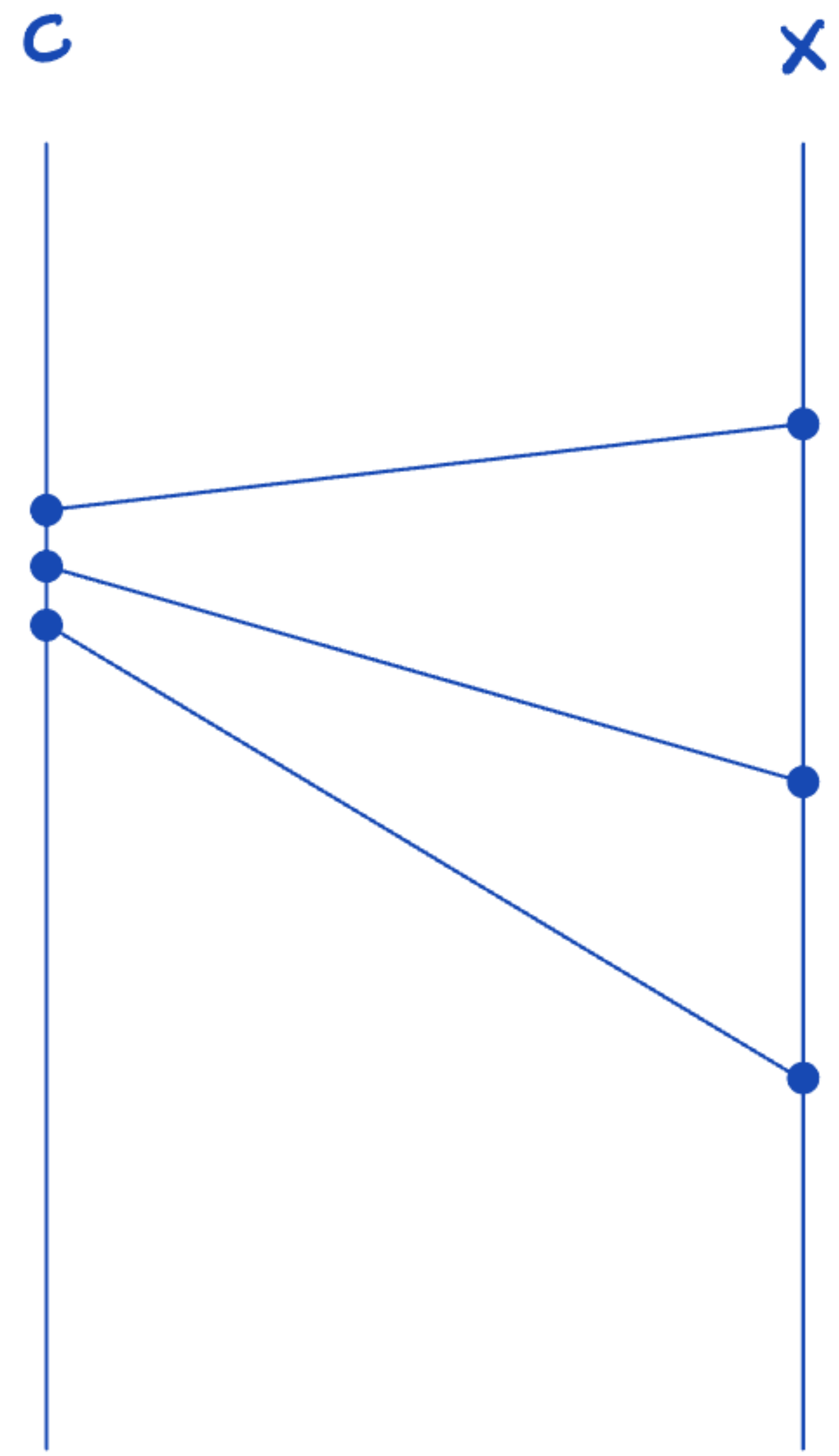
Recurrence & Memorization



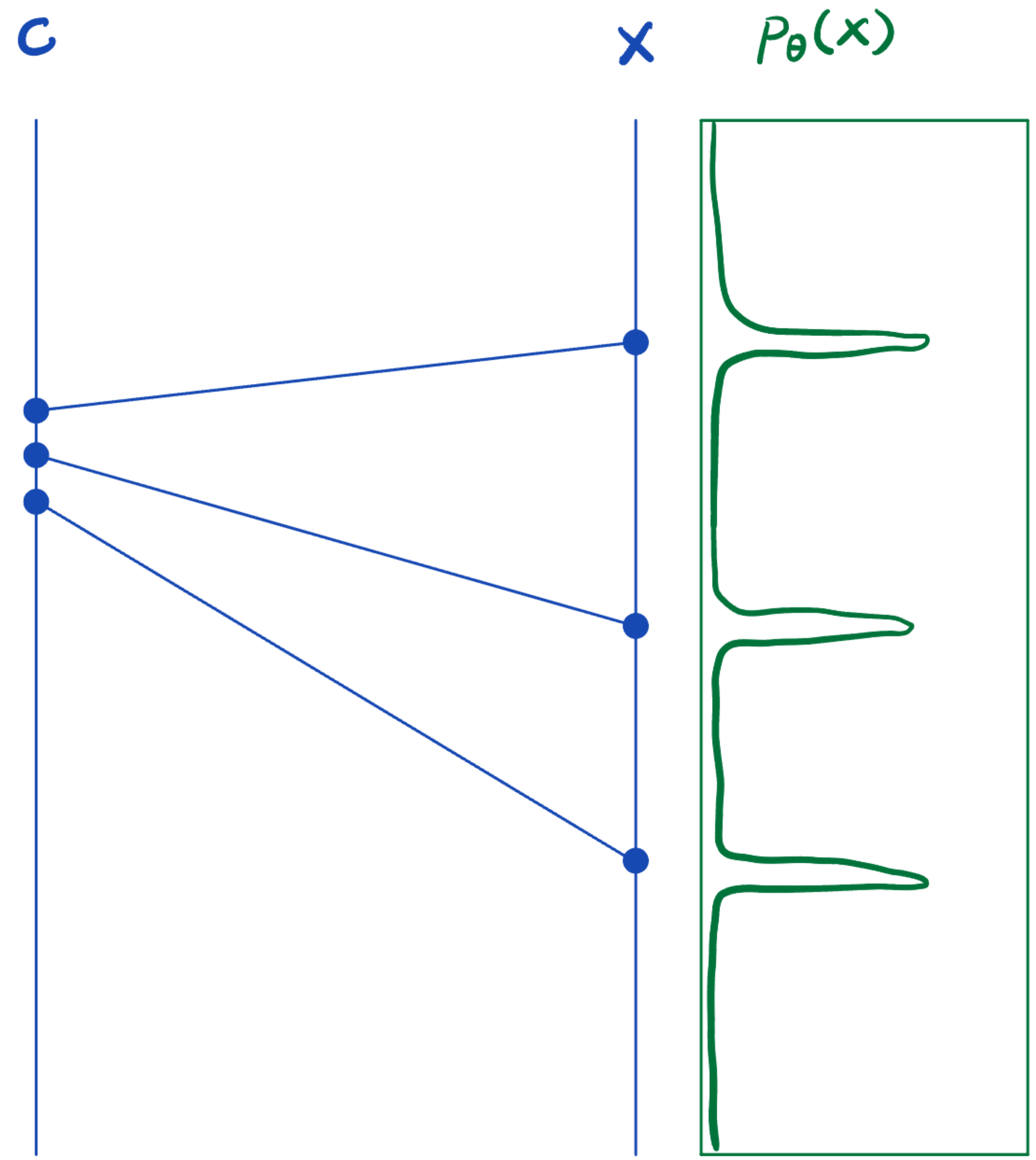
Recurrence & Memorization



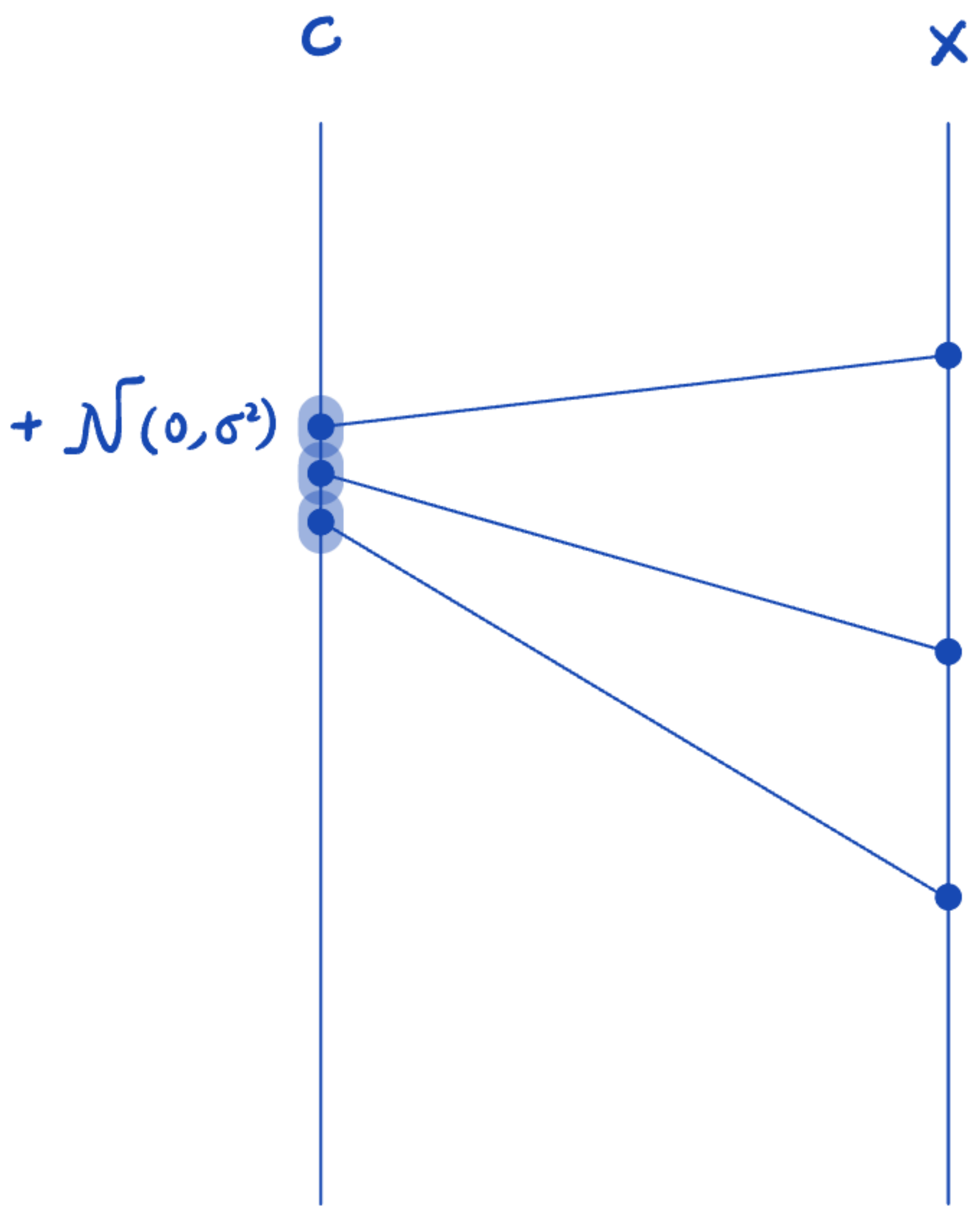
Recurrence & Memorization



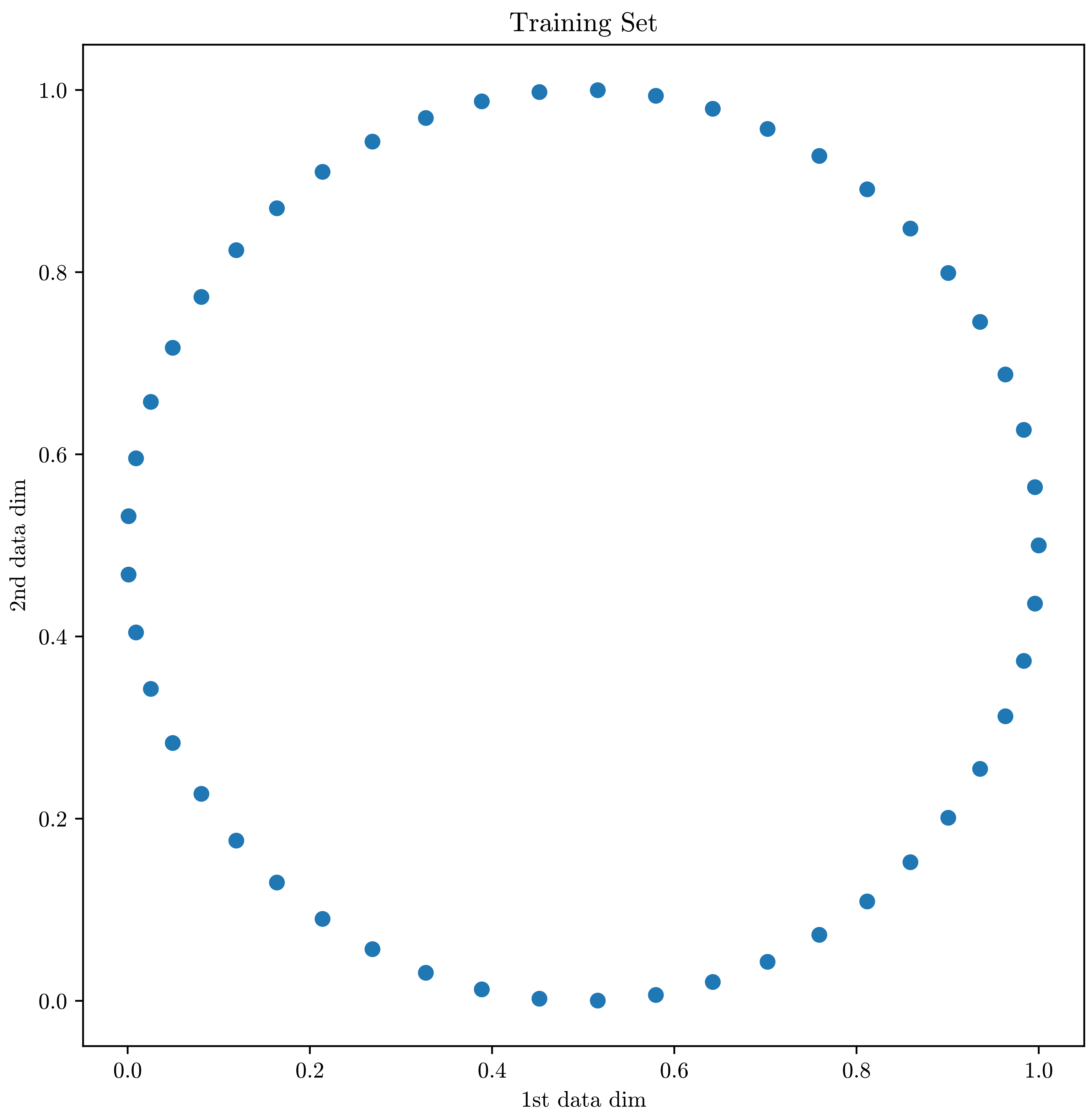
Recurrence & Memorization



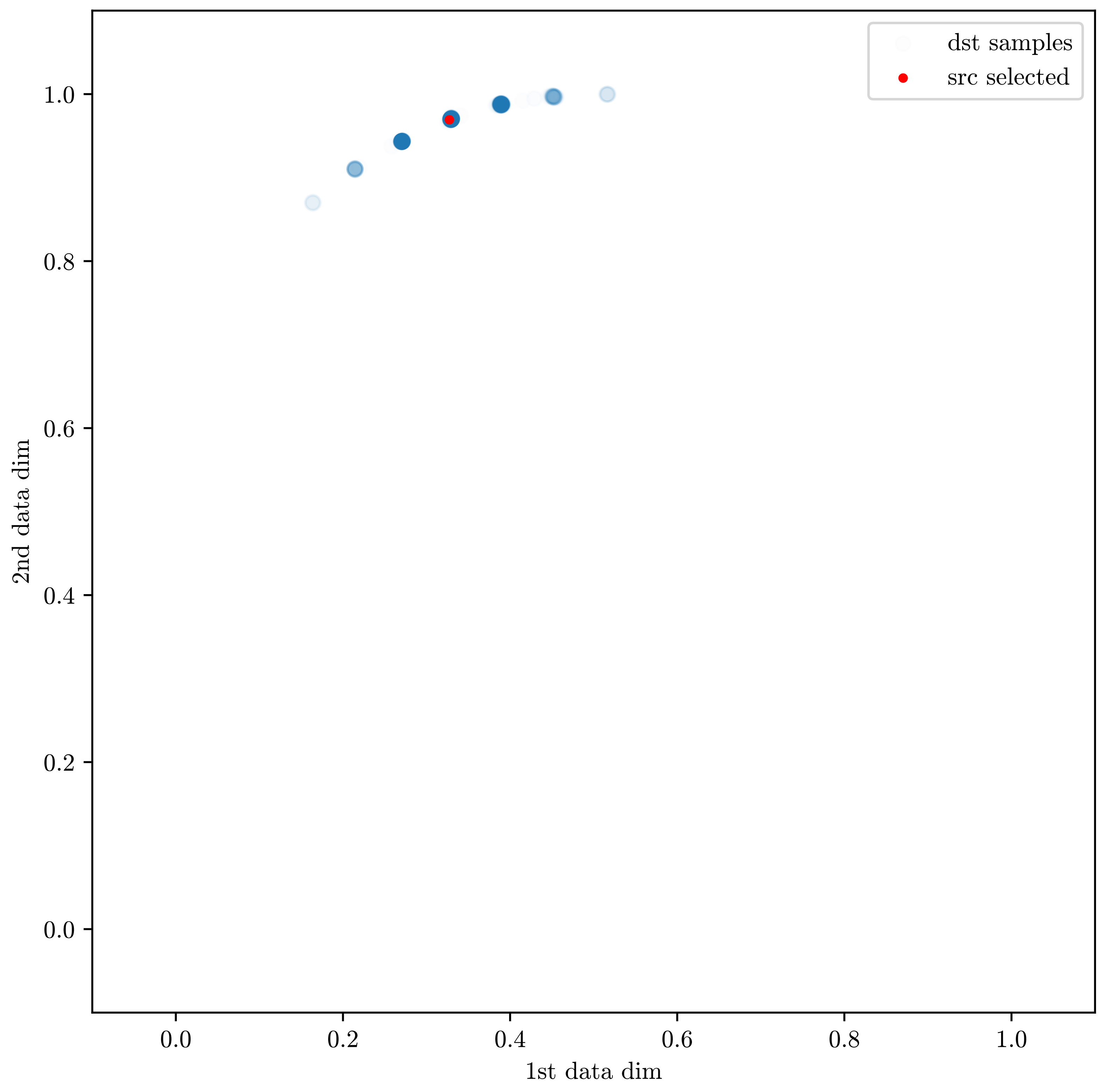
Recurrence & Memorization

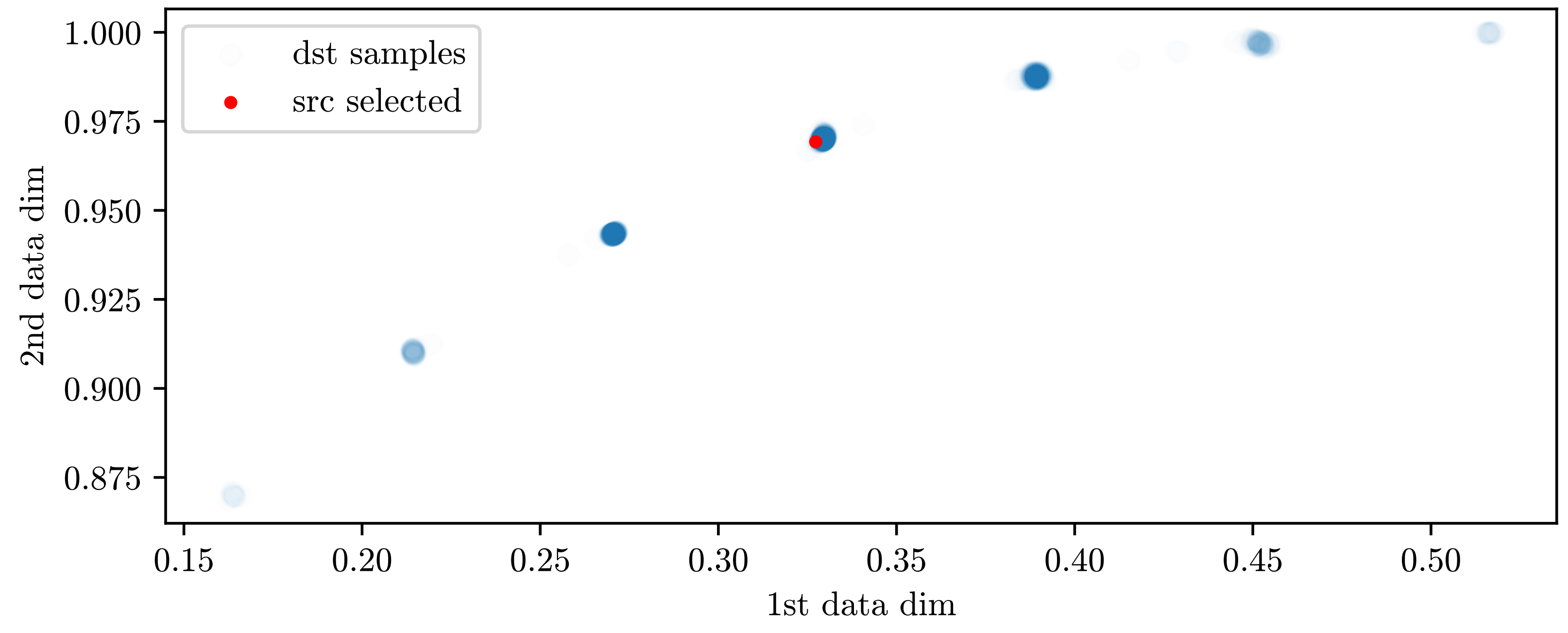


Recurrence & Memorization



Recurrence & Memorization

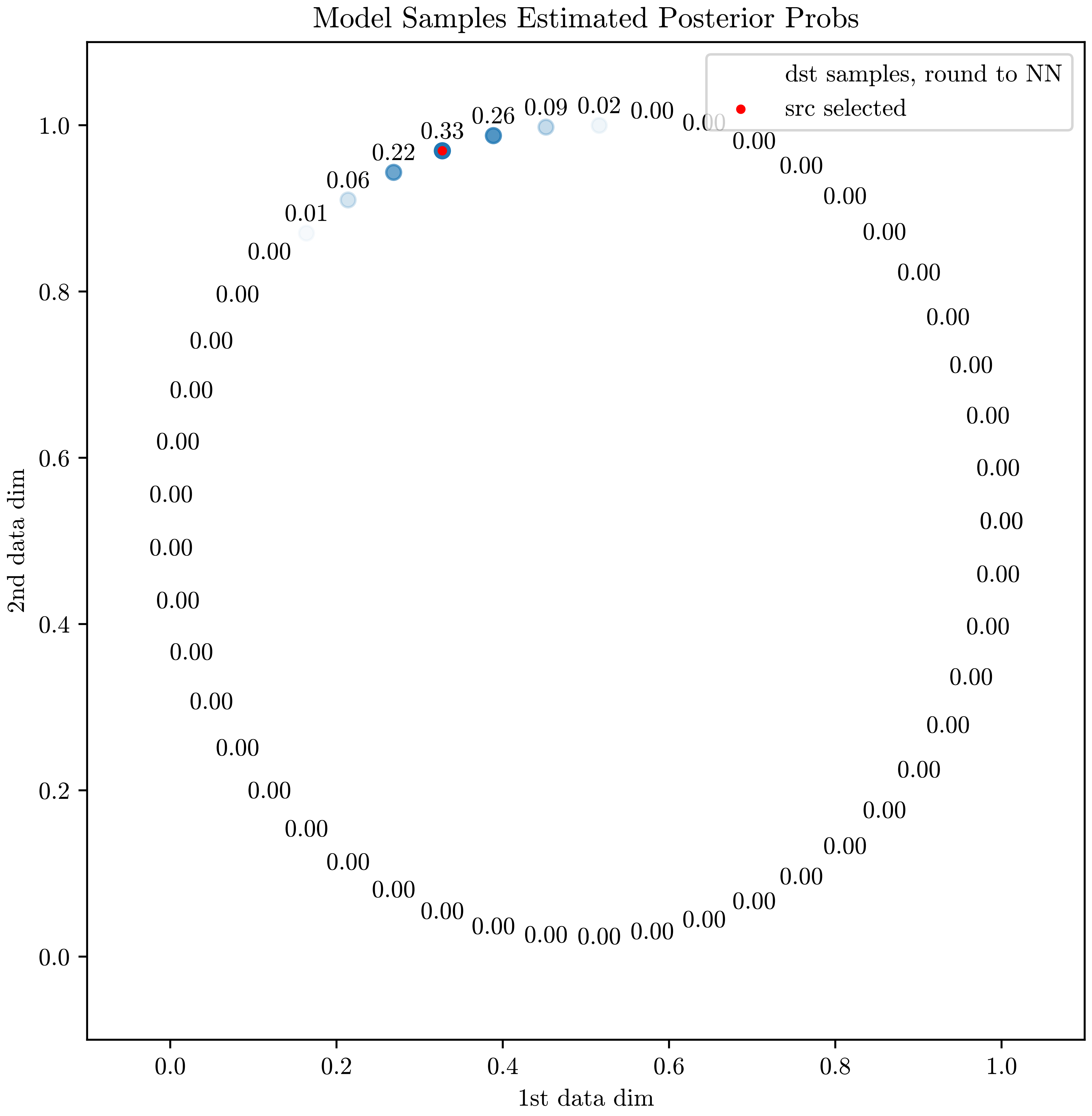


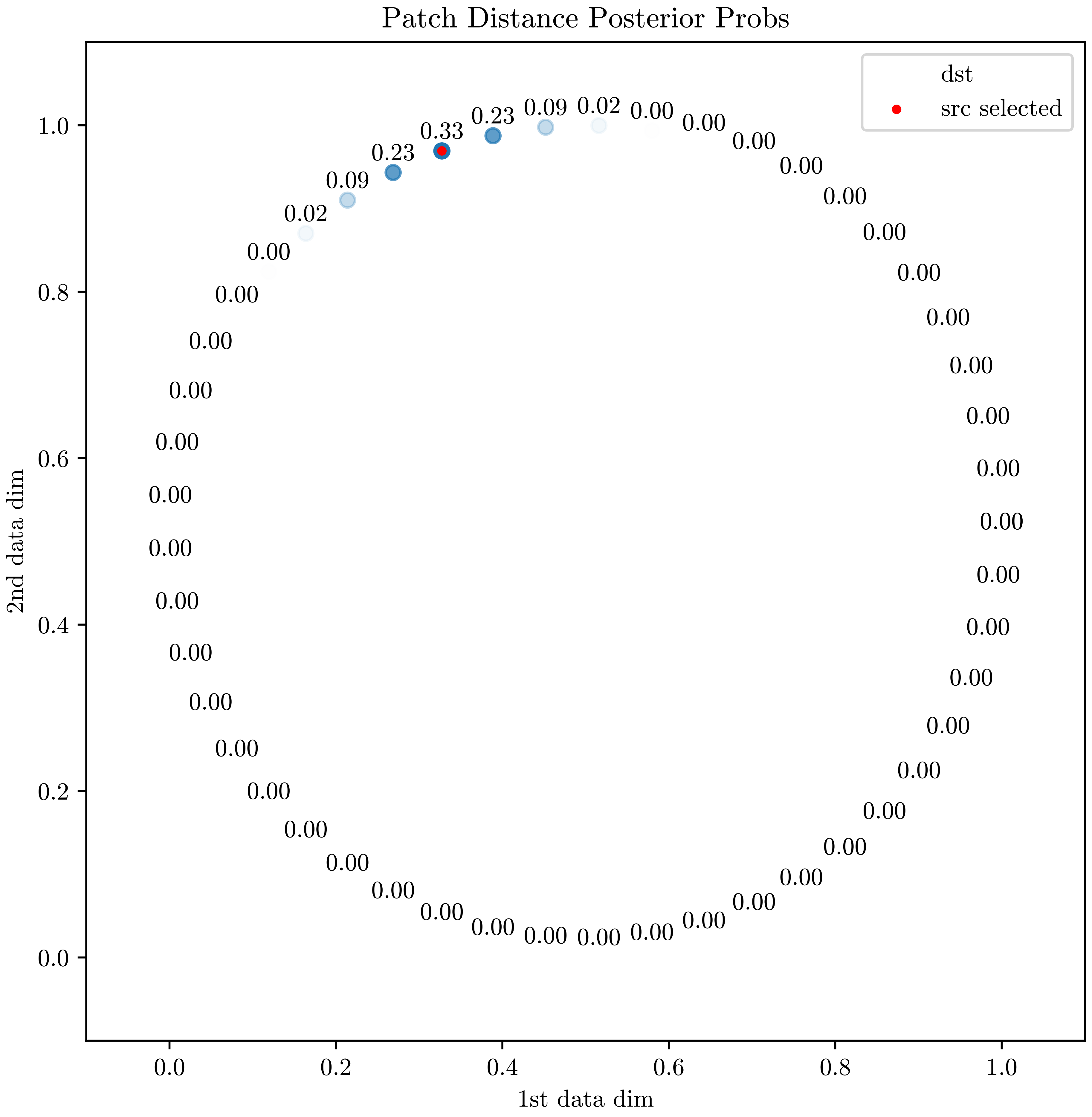


Recurrence & Memorization

$$
\begin{aligned}
\Pr[\mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(i)} \,|\, \mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}'] &= \frac{\Pr[\mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}' \,|\, \mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(i)}] \Pr[\mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(i)}]}{\Pr[\mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}']} \\
&= \frac{\Pr[\mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}' \,|\, \mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(i)}] \Pr[\mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(i)}]}{\sum_{j=1}^{N} \Pr[\mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}' \,|\, \mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(j)}] \Pr[\mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(j)}]} \\
&= \frac{\Pr[\mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}' \,|\, \mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(i)}](1/N)}{\sum_{j=1}^{N} \Pr[\mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}' \,|\, \mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(j)}](1/N)} \\
&= \frac{\Pr[\mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}' \,|\, \mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(i)}]}{\sum_{j=1}^{N} \Pr[\mathbf{x}_{\text{src}} = {\mathbf{x}_{\text{src}}}' \,|\, \mathbf{x}_{\text{dst}} = {\mathbf{x}_{\text{dst}}}^{(j)}]} \\
&= \frac{\mathcal{N}\left({\mathbf{x}_{\text{src}}}'; {\mathbf{x}_{\text{src}}}^{(i)}, {\sigma_{\text{aug}}}^2 \mathbf{I}\right)}{\sum_{j=1}^{N} \mathcal{N}\left({\mathbf{x}_{\text{src}}}'; {\mathbf{x}_{\text{src}}}^{(j)}, {\sigma_{\text{aug}}}^2 \mathbf{I}\right)} \\
&= \frac{\exp\left(-\frac{1}{2{\sigma_{\text{aug}}}^2} \left\| {\mathbf{x}_{\text{src}}}' - {\mathbf{x}_{\text{src}}}^{(i)} \right\|_2^2\right)}{\sum_{j=1}^{N} \exp\left(-\frac{1}{2{\sigma_{\text{aug}}}^2} \left\| {\mathbf{x}_{\text{src}}}' - {\mathbf{x}_{\text{src}}}^{(j)} \right\|_2^2\right)}
\end{aligned}
$$

Recurrence & Memorization





Takeaway

- We are doing (joint) patch distribution modeling using diffusion models
- Our setting of modeling joint distribution p(patch1, patch2, relationship) has not been explored before. There are some evidence from synthetic data that there is hope to make it work well for SR and displacement estimation.
  - More excitedly would be having a cooler killer application, what we are working on
- On the side, we found a connection between classical search-based algorithms versus parametric/deep generative method in our context — the noise augmentation setting. We're still thinking about how could we better incorporate this understanding.