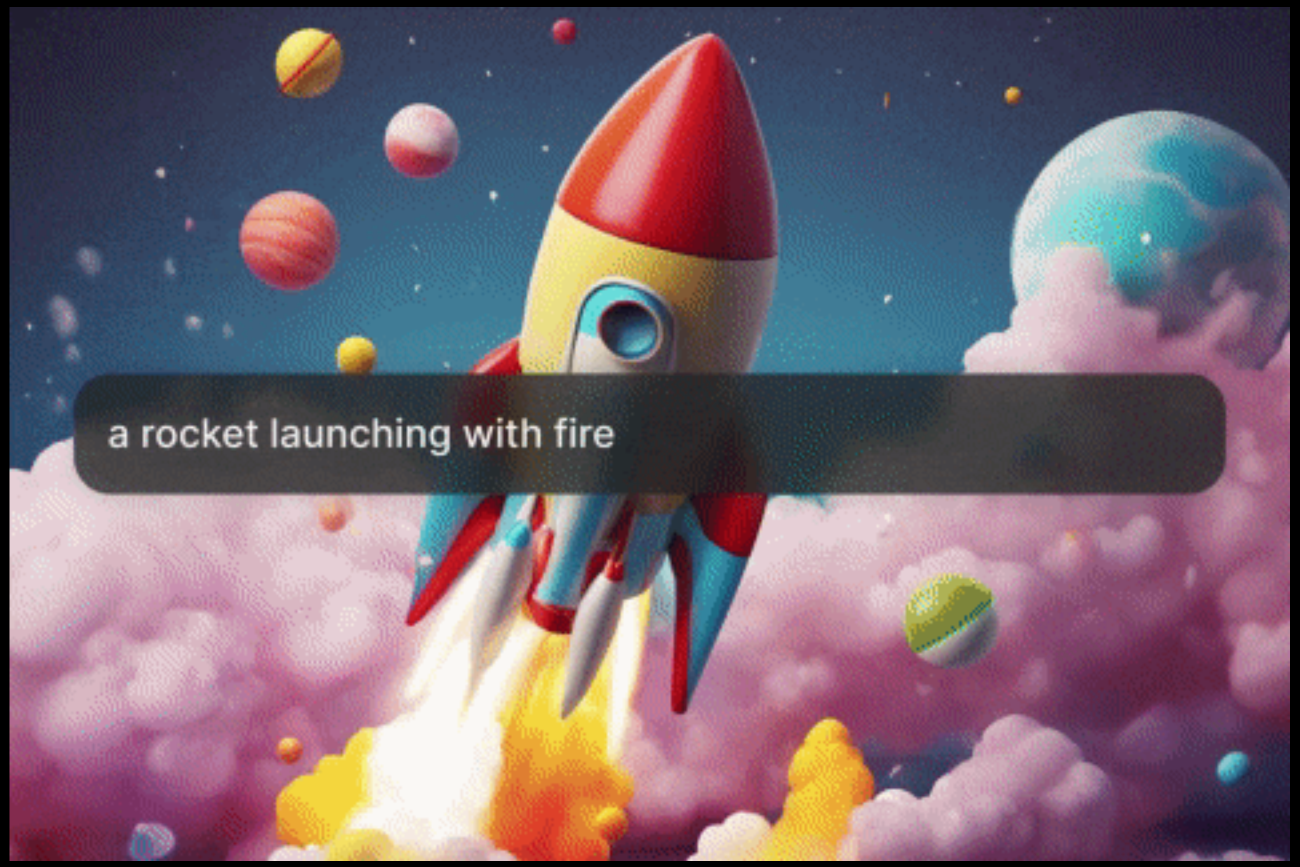
Twisted diffusion sampler for controllable generation with application to motif-scaffolding

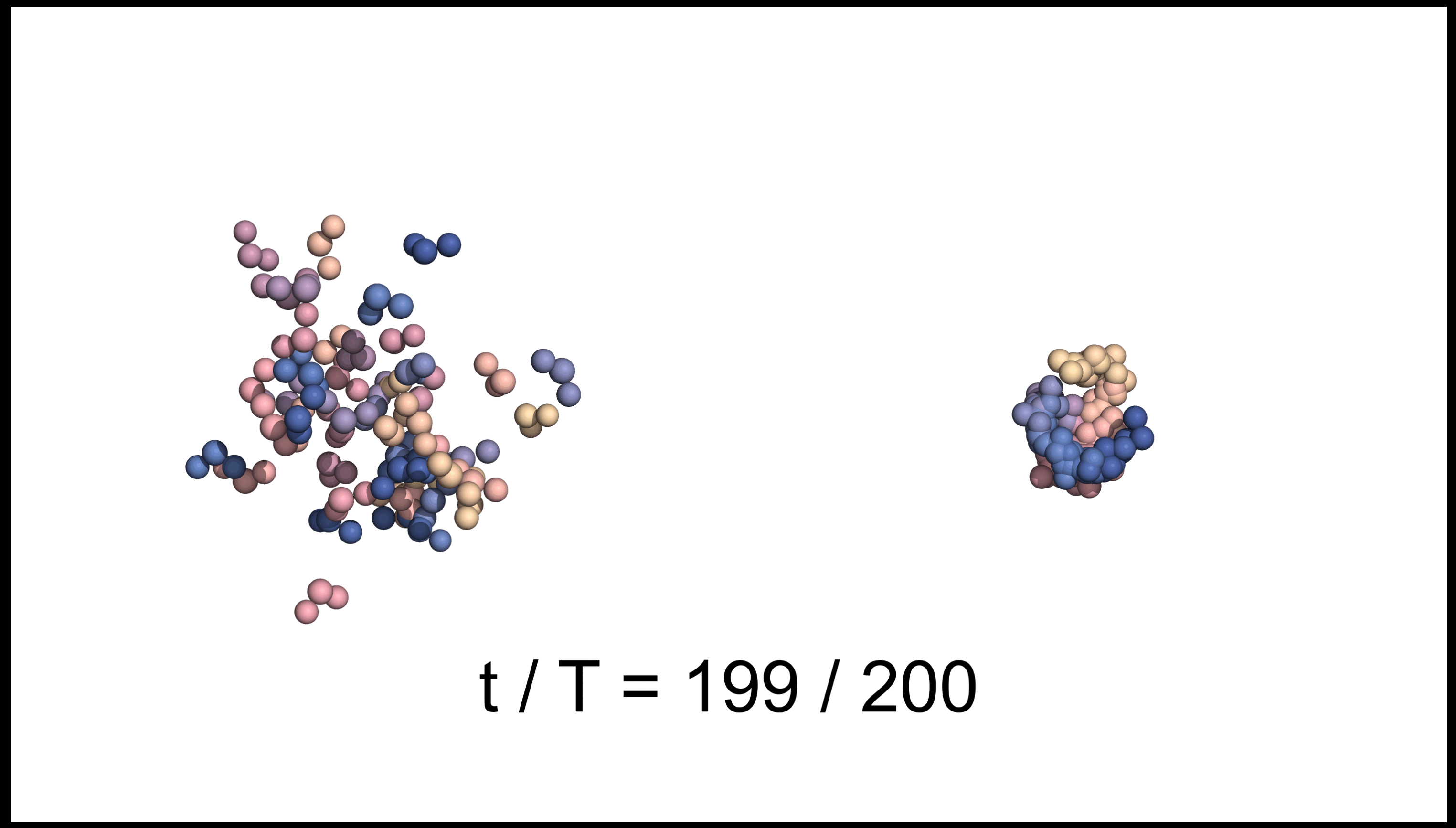
Joint work with Brian Trippe(equal contribution), Christian Nasseth, David Blei, and John Cunningham

Diffusion models have been powerful…



Text to image generation with
Stable Diffusion

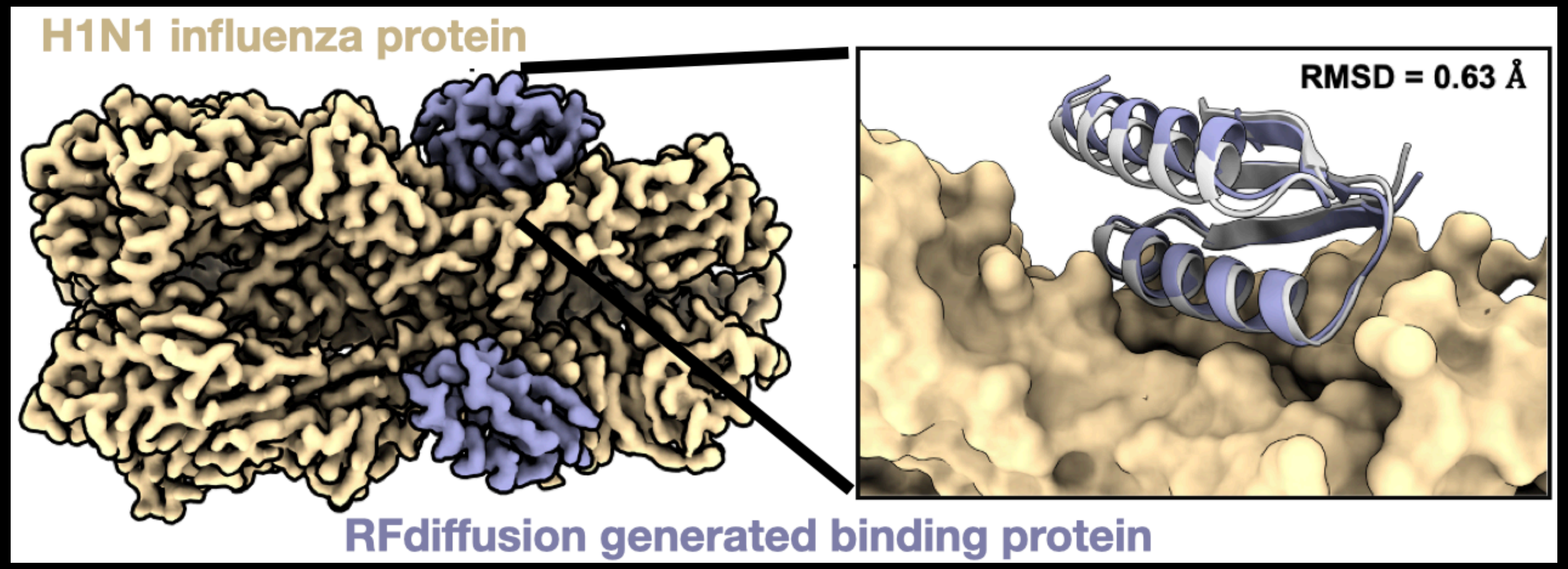
Video generation
with Open AI's Sora

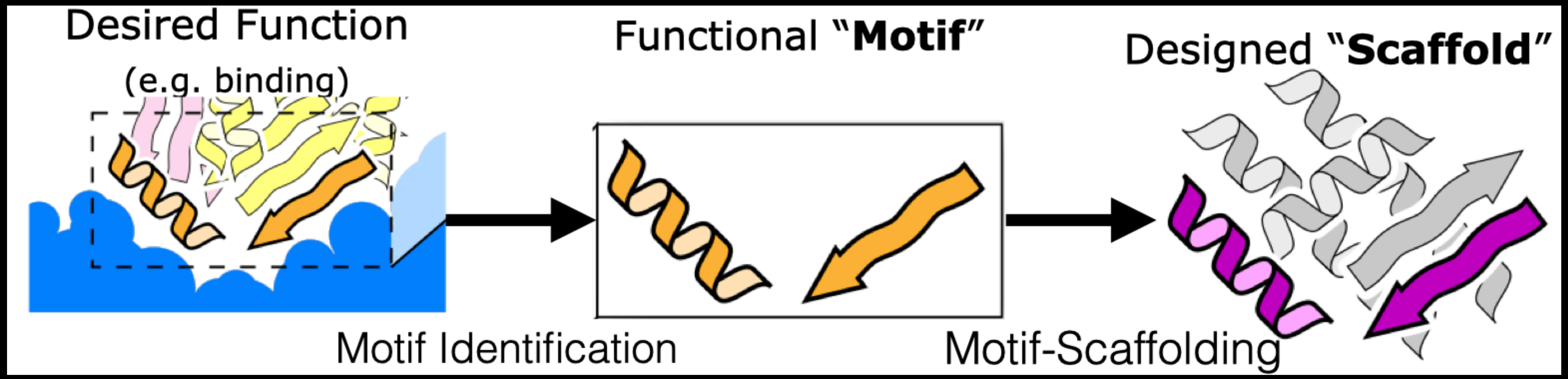


Protein generation
with SE3 FrameDiff*

*SE3 diffusion model with application to protein backbone generation. Yim et al. 2023

- Creating and training large-scale diffusion models from scratch is a massive undertaking
- There are (many) off-the-shelf pre-trained diffusion models
- They provide good-quality *universal* generation
- Practitioners are often more interested in *controllable* generation that is customized to a specific task





source: (1) De novo design of protein structure and function with RFdiffusion. Watson et al. 2023.

(2) Doug Tischer

How can we make use of those powerful, *pre-trained* diffusion models for controllable generation?

There are two common paradigms to adapt pre-trained diffusion models

- **Training-required approach**
  - Finetune on specific tasks
  - Adapt model's architecture to take in additional inputs
  - Pros:
    - fast inference
    - good performance if additional training is "sufficient"
  - Cons:
    - labor- and compute-extensive
    - less flexible. E.g. difficult to adapting to new tasks, or composition of tasks.

- **Inference-time approach**
  - Heuristic method: e.g. guidance
  - More theoretically grounded methods
  - Pros:
    - training-free
    - more flexible
  - Cons:
    - increased inference time and/or compute
    - some heuristic methods may have low-fidelity generation output

There are two common paradigms to adapt pre-trained diffusion models

- **Training-required approach**
  - Finetune on specific tasks
  - Adapt model's architecture to take in additional inputs
  - Pros:
    - fast inference
    - good performance if additional training is "sufficient"
  - Cons:
    - labor- and compute-extensive
    - less flexible. E.g. difficult to adapting to new tasks, or composition of desired properties.

- **Inference-time approach**
  - Heuristic method: e.g. guidance
  - More theoretically grounded methods
  - Pros:
    - training-free
    - more flexible
    - practical and asymptotically accurate
  - Cons:
    - slow inference
    - ~~some heuristic methods may have low-fidelity generation output~~

Roadmap

- **Problem formulation**: controllable generation
- **Background**: diffusion models
- **Method**: Twisted Diffusion Sampler (TDS)
- **Case study**: protein motif-scaffolding

Problem formulation

- **Goal**: generate data x^0 in response to conditioning criteria y
- **Conditional sampling**:
  - given a generative model $p_\theta(x^0)$, a likelihood $p_{y|x^0}(y \mid x^0)$ and and conditional information y
  - sample from the conditional distribution $p_\theta(x^0 \mid y) \propto p_\theta(x^0)p_{y|x^0}(y \mid x^0)$.
  - Example:
    - $p_\theta(x^0)$ is distribution of natural images
    - $p_{y|x^0}(y \mid x^0)$ is an image classifier
    - $p_\theta(x^0 \mid y)$ is the conditional distribution of images given a class label y

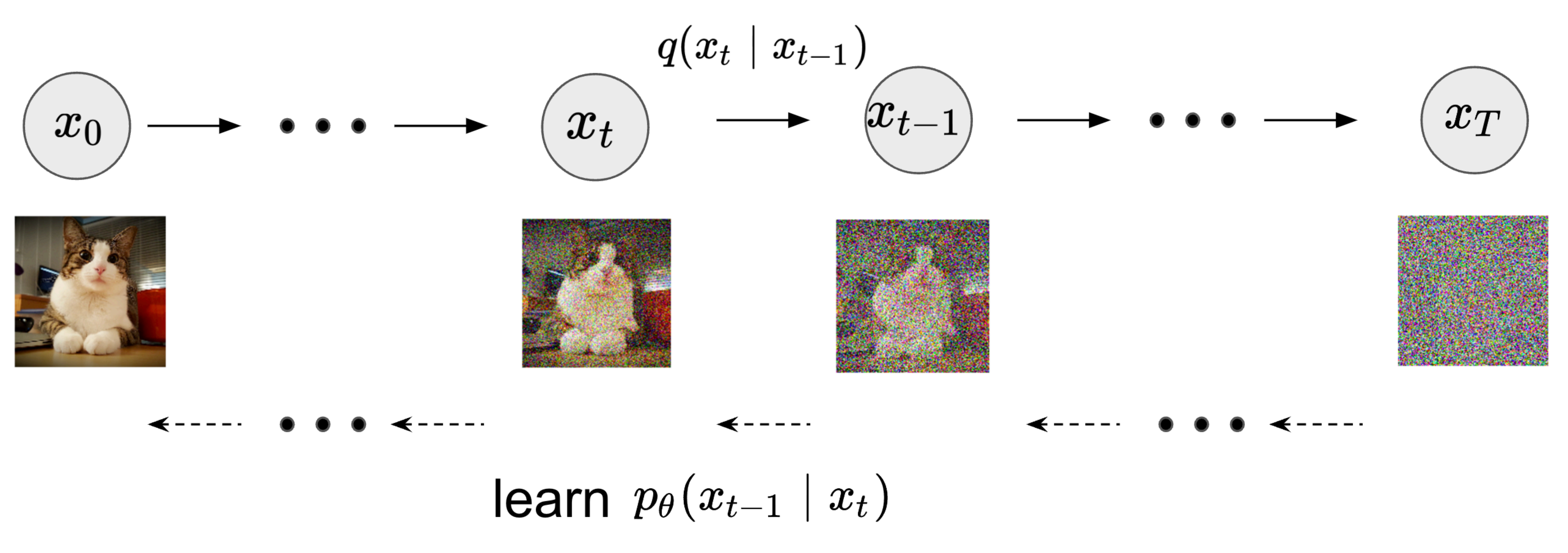
Problem formulation

- **Goal**: generate data x^0 in response to conditioning criteria y
- **Conditional sampling**:
  - given a generative model $p_\theta(x^0)$, a likelihood $p_{y|x^0}(y \mid x^0)$, and conditional information y
  - sample from the conditional distribution $p_\theta(x^0 \mid y) \propto p_\theta(x^0) p_{y|x^0}(y \mid x^0)$.
  - Example:
    - $p_\theta(x^0)$ is distribution of physically realizable proteins
    - $p_{y|x}(y \mid x^0) = \delta_{x^0_M}(y)$ is a Delta distribution at a substructure (e.g. motif)
    - $p_\theta(x^0 \mid y)$ is the conditional distribution of proteins that contain substructure y

Roadmap

- **Problem formulation**: controllable generation
- **Background**: diffusion models
- **Method**: Twisted Diffusion Sampler (TDS)
- **Case study**: protein motif-scaffolding

Diffusion model learns the distribution of data x^0 by gradually adding noise to the data, and learning to reverse the noising process

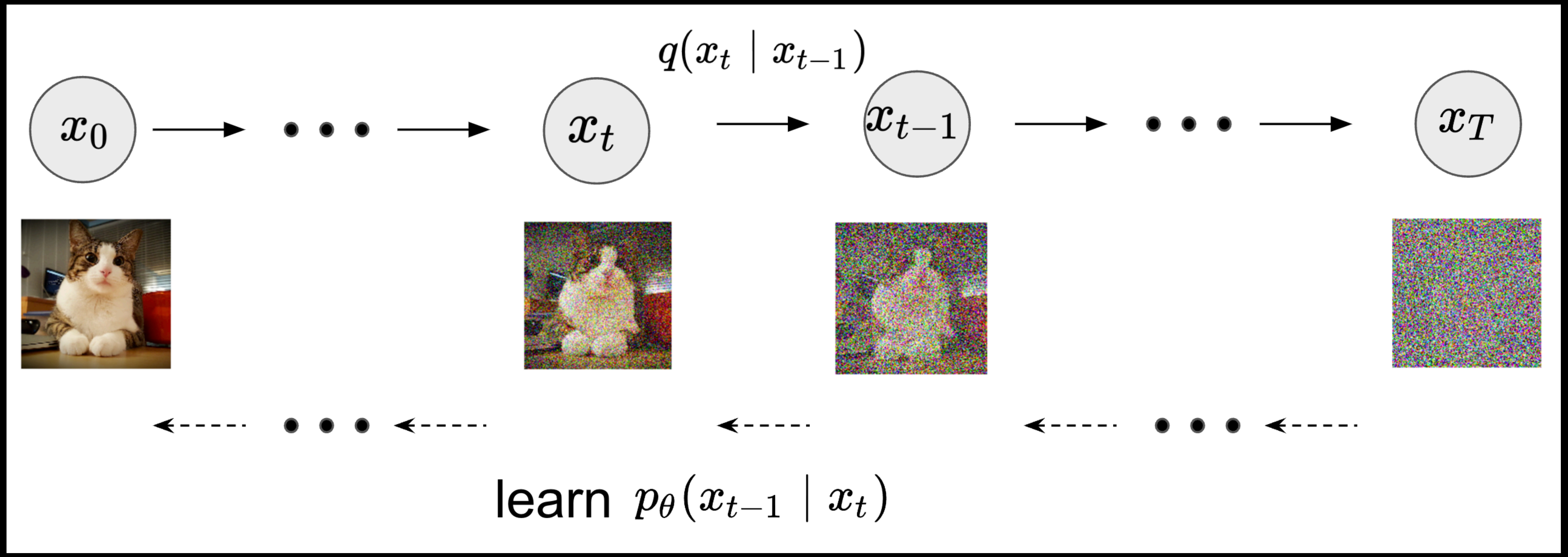


$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t \mid x_{t-1}, \sigma^2)$$

T so large that $q(x^T) \approx \mathcal{N}(x^T \mid 0, T\sigma^2\mathbb{I})$

To learn $p_\theta(x^{t-1} \mid x^t)$ that reverses the noising process:

- Note the true reverse transition is $q(x^{t-1} \mid x^t) \approx \mathcal{N}(x^{t-1} \mid x^t + \sigma^2 \nabla_{x^t} \log q(x^t), \sigma^2)$, $\log q(x^t)$ is score function
- If we have a score network $s_\theta(x^t; t) \approx \nabla_{x^t} \log q(x^t)$, we can parameterize
  $p_\theta(x^{t-1} \mid x^t) := \mathcal{N}(x^{t-1} \mid x^t + \sigma^2 s_\theta(x^t; t), \sigma^2)$
- Can't compute the true score, but $\nabla_{x^t} \log q(x^t) = \frac{\mathbb{E}_q[x^0 \mid x^t] - x^t}{t\sigma^2}$
  - If we can learn to predict the clean data via a denoiser network $\hat{x}^0_\theta(x^t; t) \approx \mathbb{E}_q[x_0 \mid x_t]$
  - Then we can parameterize the score network using denoisier: $s_\theta(x^t; t) := \frac{\hat{x}^0_\theta(x^t; t) - x^t}{t\sigma^2}$

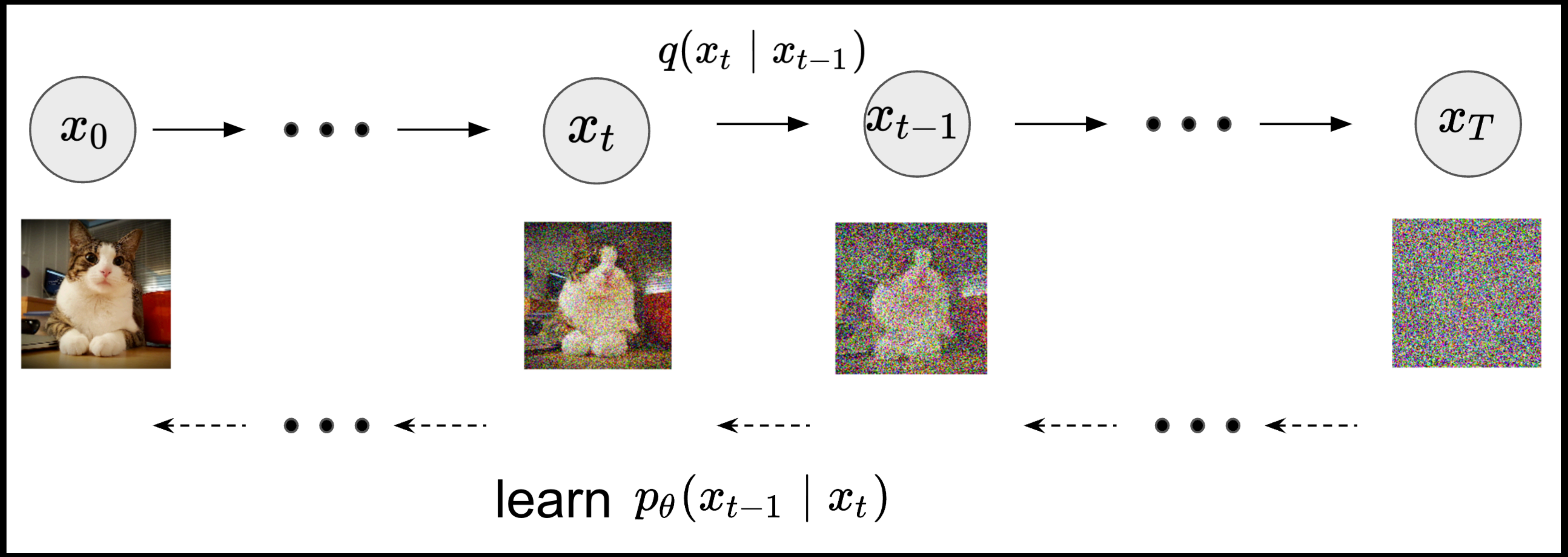


Training is reduced to a series of self-supervision tasks

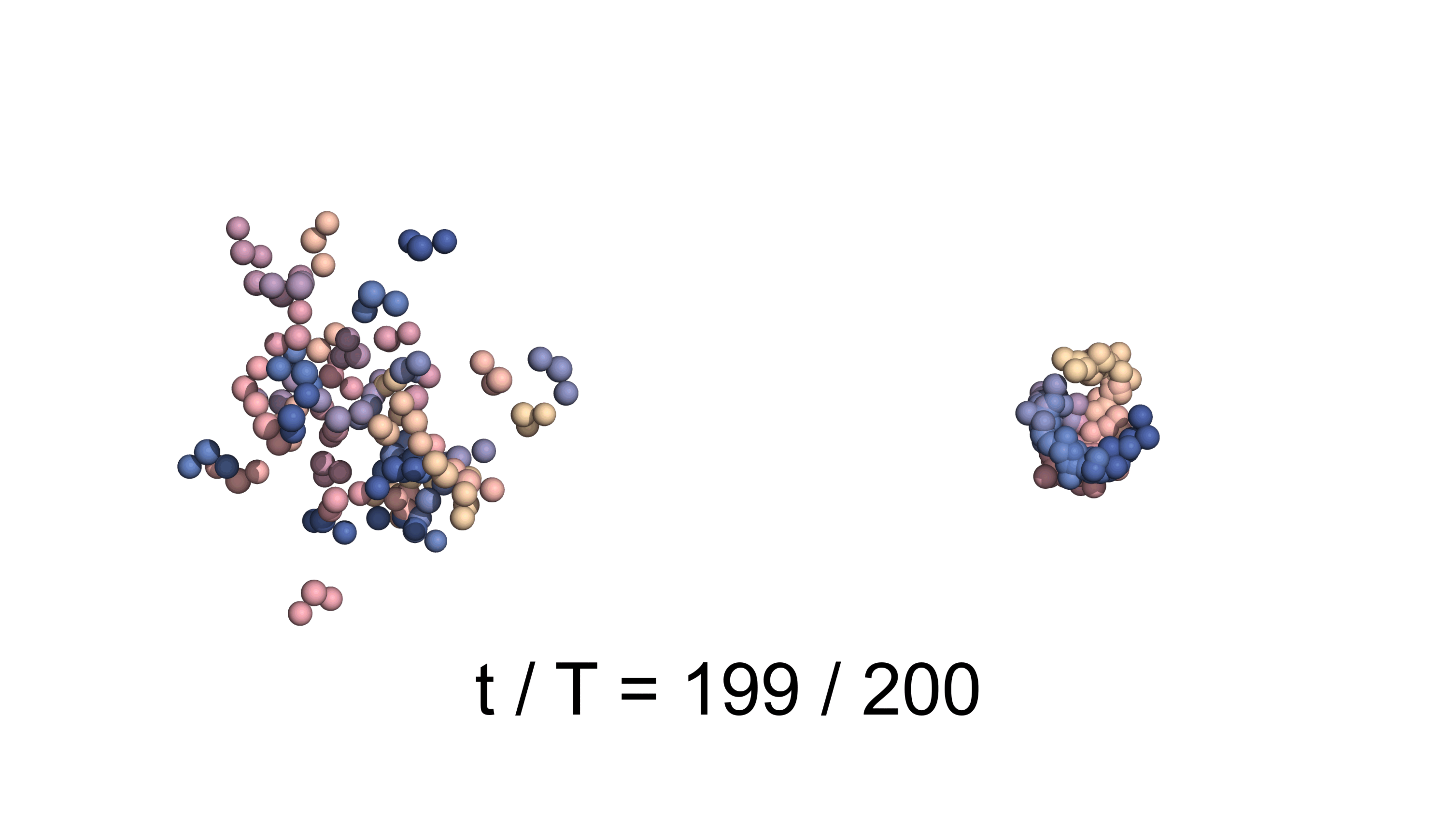
$$\min_\theta \sum_{t=1}^{T} c_t \mathbb{E}_{q(x_t|x_0)} \|x^0 - \hat{x}^0_\theta(x_t; t)\|^2$$

Generation

- Sample random noise x^T
- Iteratively refine the noisy datapoint $x^t \sim p_\theta(x^t \mid x^{t+1})$
  - first predict the clean data $\hat{x}^0_\theta(x_t)$
  - then transform $\hat{x}^0_\theta(x_t)$ to the score $s_\theta(x^t)$ required for transition
- Return the clean data x^0



noisy data x^t denoising pred. $\hat{x}^0_\theta(x_t)$



Roadmap

- **Problem formulation**: controllable generation
- **Background**: diffusion models
- **Method**: Twisted Diffusion Sampler (TDS)
- **Case study**: protein motif-scaffolding

- Recall our task of **conditional sampling**:
  - given a generative model $p_\theta(x^0)$, a likelihood $p_{y|x^0}(y \mid x^0)$, and conditional information y
  - sample from the conditional distribution $p_\theta(x^0 \mid y) \propto p_\theta(x^0)p_{y|x^0}(y \mid x^0)$.

- In **diffusion models**, augment the space to include $x^{1:T}$
  - given a diffusion model $p_\theta(x^{0:T})$, a likelihood $p_{y|x^0}(y \mid x^0)$, and conditional information y
  - sample from the conditional distribution $p_\theta(x^{0:T} \mid y) \propto p_\theta(x^{0:T})p_{y|x^0}(y \mid x^0)$.

- We cannot directly sample from $p_\theta(x^{0:T} \mid y) \propto p_\theta(x^{0:T}) p_{y|x^0}(y \mid x^0)$.
- However, the joint distribution $p_\theta(x^{0:T}) p_{y|x^0}(y \mid x^0)$ is computable
- Naive importance sampling:
  - Proposal: unconditional diffusion model $p_\theta(x^{0:T})$
  - Target (unnormalized): $p_\theta(x^{0:T} \mid y) \propto p_\theta(x^{0:T}) p_{y|x^0}(y \mid x^0)$
  - Weight $= \frac{\text{Target}}{\text{Proposal}}$: $w(x^{0:T}) = \frac{\cancel{p_\theta(x^{0:T})} \; p_{y|x^0}(y \mid x^0)}{\cancel{p_\theta(x^{0:T})}} = p_{y|x^0}(y \mid x^0)$

Generate a bunch of particles (samples) from proposal (diffusion models), and then resample them according to their weights (likelihood values)

- Asymptotically exact
- Low efficiency, e.g. 1 out of 1k sampled images is cat, for 1k possible classes

Method: Twisted Diffusion Sampler

Ideas:

- twist the naive proposal to approach the ideal proposal
- develop intermediate weighting mechanism (Sequential Monte Carlo)

Secret sauce:

- the denoising prediction $\hat{x}^0_\theta(x_t)$

Ideal proposal: $p_\theta(x^{0:T} \mid y) = p_\theta(x^T \mid y) \prod_{t=1}^{T} p_\theta(x^{t-1} \mid x^t, y)$

Let's **approximate** with some $\tilde{p}_\theta(x^{0:T} \mid y)$

- Set $\tilde{p}_\theta(x^T \mid y) := \mathcal{N}(x^T \mid 0, T\sigma^2\mathbb{I}) \approx p_\theta(x^T \mid y)$
- Set $\tilde{p}_\theta(x^{t-1} \mid x^t, y) := \mathcal{N}(x^{t-1} \mid x^t + \sigma^2[s_\theta(x^t) + \nabla_{x_t} \log \tilde{p}_\theta(y \mid x^t)], \sigma^2)$, where

  $\tilde{p}_\theta(y \mid x^t) := p_{y|x^0}(y \mid \hat{x}^0_\theta(x^t)) \approx p_\theta(y \mid x^t)$
- Intuition: we want to refine the sample in the direction of
  1. increasing unconditional marginal density,
  2. increasing likelihood of predicted data.

- Set $\tilde{p}_\theta(x^{t-1} \mid x^t, y) := \mathcal{N}(x^{t-1} \mid x^t + \sigma^2[s_\theta(x^t) + \nabla_{x_t} \log \tilde{p}_\theta(y \mid x^t)], \sigma^2)$, where

  $\tilde{p}_\theta(y \mid x^t) := p_{y|x^0}(y \mid \hat{x}^0_\theta(x^t)) \approx p_\theta(y \mid x^t)$

- **Goal**: show $\tilde{p}_\theta(x^{t-1} \mid x^t, y)$ is a reasonable approximation to the true $p_\theta(x^{t-1} \mid x^t, y)$

$$p_\theta(x^{t-1} \mid x^t, y) = p_\theta(x^{t-1} \mid x^t) p_\theta(y \mid x^{t-1}) / p_\theta(y \mid x^t)$$ Bayes' rule

$$\approx p_\theta(x^{t-1} \mid x^t) \tilde{p}_\theta(y \mid x^{t-1}) / \tilde{p}_\theta(y \mid x^t)$$ Likelihood approximation

$$\approx p_\theta(x^{t-1} \mid x^t) \exp\{(x^t - x^{t-1}) \nabla_{x^t} \log \tilde{p}_\theta(y \mid x^t)\}$$ Taylor expansion

$$= \mathcal{N}(x^{t-1} \mid x^t + \sigma^2[s_\theta(x^t) + \nabla_{x^t} \log \tilde{p}_\theta(y \mid x^t), \sigma^2)$$ Twist and complete the square

$$=: \tilde{p}_\theta(x^{t-1} \mid x^t, y)$$

We have obtained the twisted proposal: $\tilde{p}_\theta(x^{0:T} \mid y) = \tilde{p}_\theta(x^T \mid y) \prod_{t=1}^{T} \tilde{p}_\theta(x^{t-1} \mid x^t, y)$

- We could directly plug it in to the importance sampling procedure.
- But this procedure will accumulate approximation errors in the sequential generation steps, and therefore could still be sample-inefficient.
- **Idea**: design *intermediate target and weight* to correct for intermediate errors
  - roughly: performing IS at every time step
  - more formally: Sequential Monte Carlo (SMC)

- **Idea**: design *intermediate target and weight* to correct for intermediate errors

$$p_\theta(x^{0:T} \mid y) \propto p_\theta(x^T \mid y) \prod_{t=T}^{1} p_\theta(x^{t-1} \mid x^t, y)$$

→ intractable

but we can get a good approx. for free from twisted proposal

Final target

ideal intermediate target up to $t-1$ ⟶ compute intermediate weights

$$p_\theta(x^{t-1} \mid x^t, y) \approx p_\theta(x^{t-1} \mid x^t)\tilde{p}_\theta(y \mid x^{t-1})/\tilde{p}_\theta(y \mid x^t)$$

:= twisted intermediate target

- Set the *intermediate weight* to account for errors

$$w_{t-1} := \frac{p_\theta(x^{t-1} \mid x^t)\tilde{p}_\theta(y \mid x^{t-1})/\tilde{p}_\theta(y \mid x^t)}{\tilde{p}_\theta(x^{t-1} \mid x^t, y)}$$

The Twisted Diffusion Sampler (TDS)

Algorithm 1: Twisted Diffusion Sampler

$x_k^T \sim \mathcal{N}(0, T\sigma^2)\mathbb{I}$ // initialize K particles

$w_k \leftarrow \tilde{p}_k^T = p_{y|x^0}(y \mid \hat{x}_\theta(x_k^T))$

for $t = T, \cdots, 1$ **do**

$\{x_k^t\} \sim \mathrm{Multinomial}\left(\{x_k^t, \tilde{p}_k^t\}; \{w_k\}\right)$ // resample

$x_k^{t-1} \sim \tilde{p}_\theta(\cdot|x_k^t, y) = \mathcal{N}\left(x_k^t + \sigma^2[s_\theta(x_k^t) + \nabla_{x_k^t} \log p_{y|x^0}(y|\hat{x}_\theta(x_k^t))], \sigma^2\right)$ // proposal

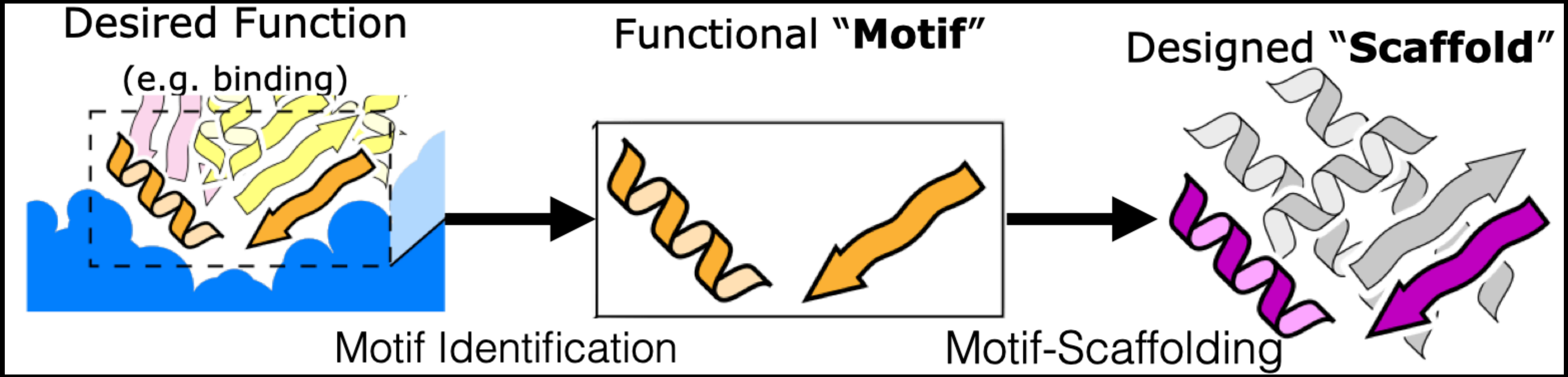
$w_k \leftarrow p_\theta(x_k^{t-1} \mid x_k^t) \cdot \; p_{y|x^0}(y|\hat{x}_\theta(x_k^{t-1}))/[p_{y|x^0}(y|\hat{x}_\theta(x_k^t)) \cdot \tilde{p}_\theta(x_k^{t-1} \mid x_k^t, y)]$ // weight

Return $\{w_k\}, \{x_k^0\}$

We show TDS is asymptotically exact as $K \to \infty$.

Roadmap

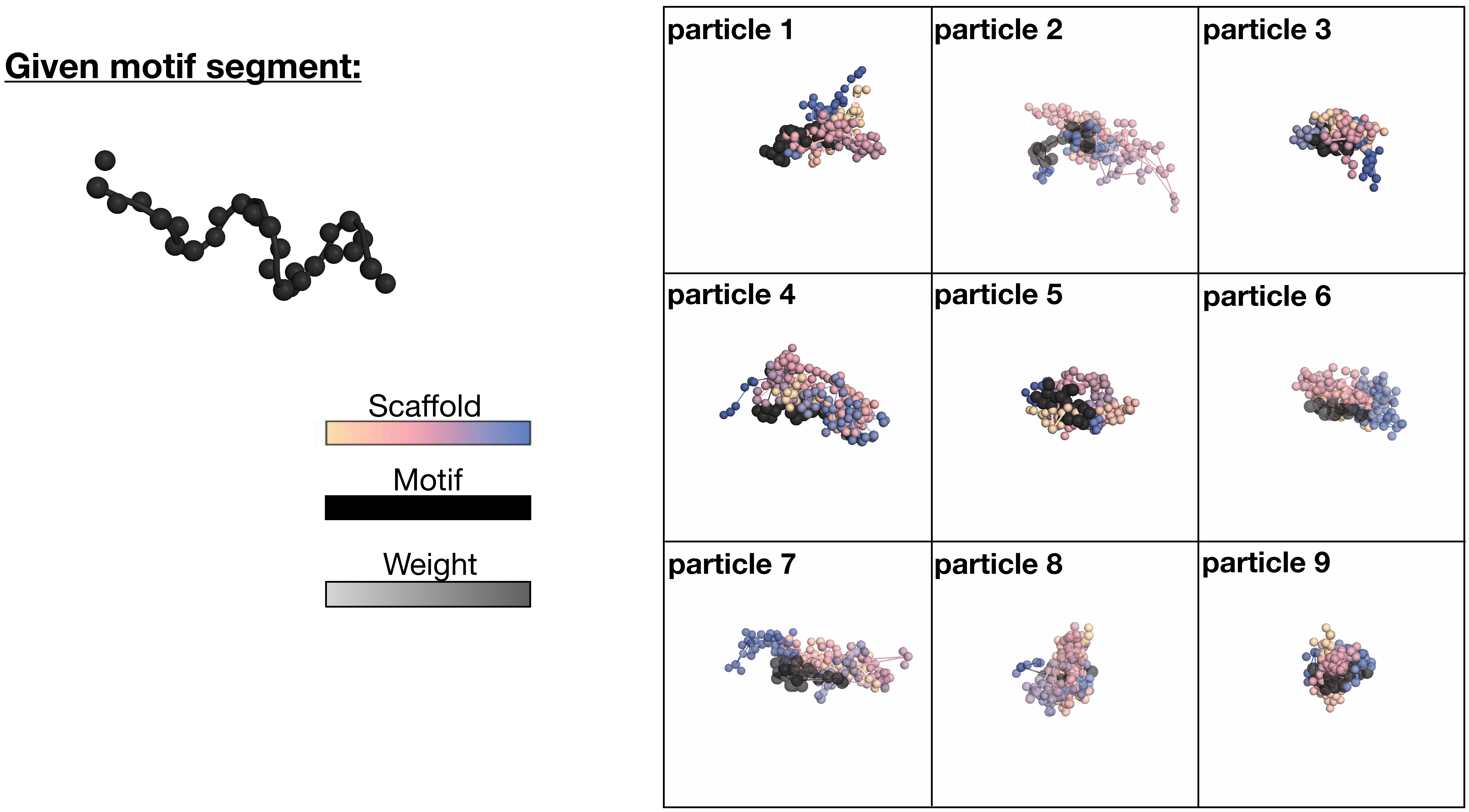
- **Problem formulation**: controllable generation
- **Background**: diffusion models
- **Method**: Twisted Diffusion Sampler (TDS)
- **Case study**: protein motif-scaffolding

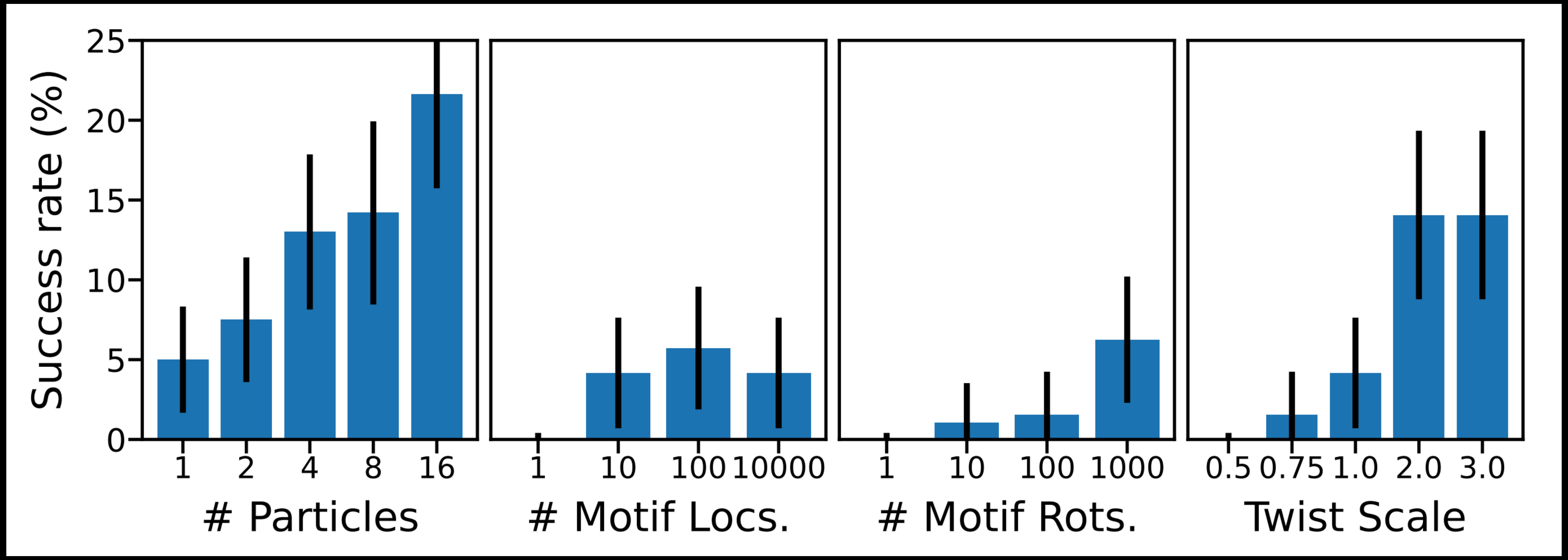


Task: generate the scaffold given a motif

Conditional sampling formulation:

- $x^t = [x_M^t, x_S^t], t = 0, \cdots, T$
- M is the location indices for the motif segment, S for the scaffold
- The likelihood function is $\delta_{x_M^0}(y)$
  - or we can consider additional *degree of freedom* in the locations of motif $\sum_{M \in \mathbf{M}} \delta_{x_M^0}(y)$, where $\mathbf{M}$ are all possible motif locations


Given motif segment:
Scaffold
Motif
Weight
particle 1
particle 2
particle 3
particle 4
particle 5
particle 6
particle 7
particle 8
particle 9



Success rate improves with more particles (motif: 5IUS)

Scaffold size	TDS & FrameDiff	RF diffusion
$<$ 100 res.	9	3
$\geq$ 100 res.	2	8
Overall	11	11

TDS v.s. RF diffusion: Number of problems with higher success rate on benchmark test cases.

TDS achieves the state of the art performance on 9/12 problems with short scaffolds.