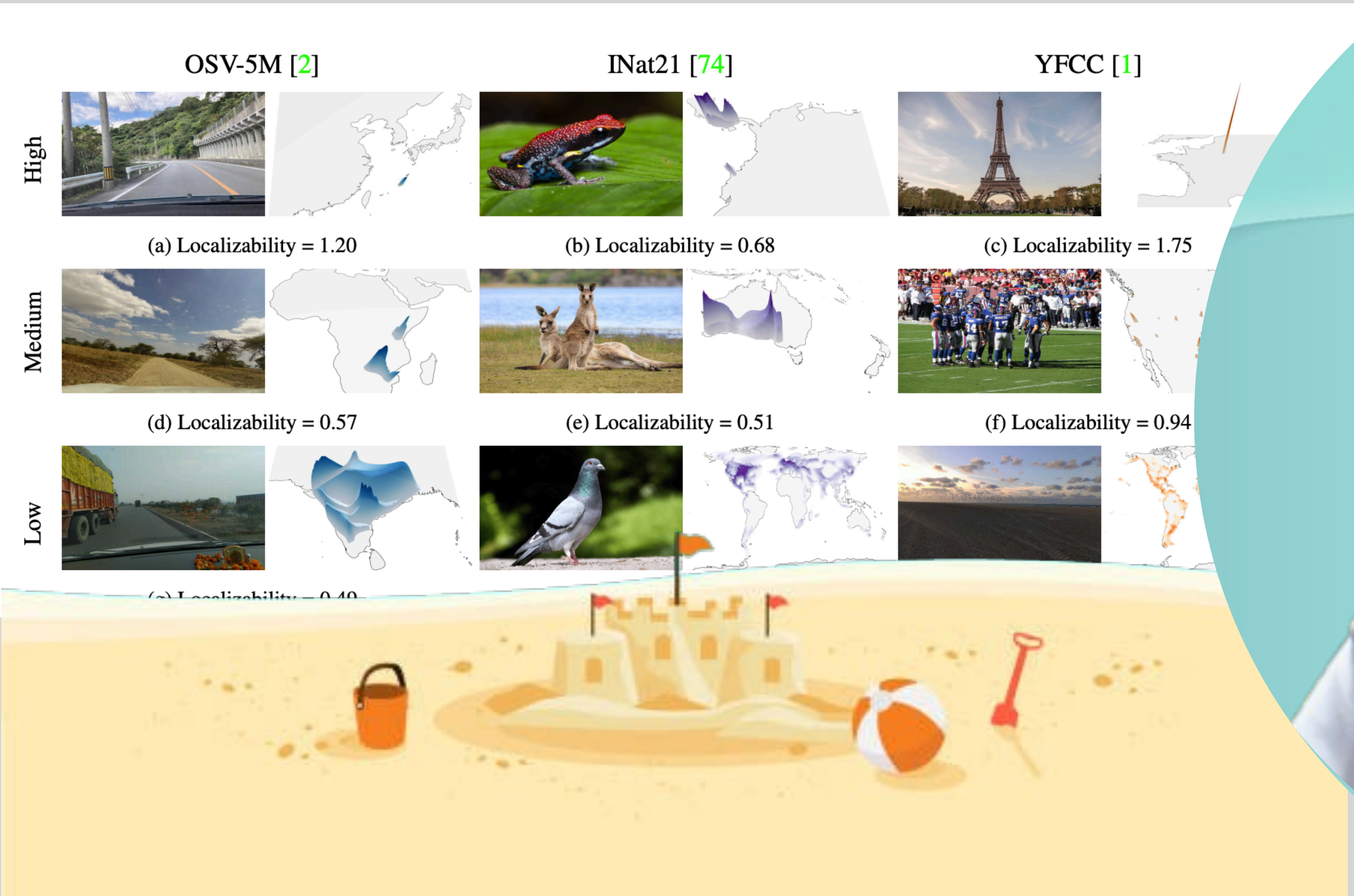
Flow Matching from dummies

Sometimes you gotta go with the flow…

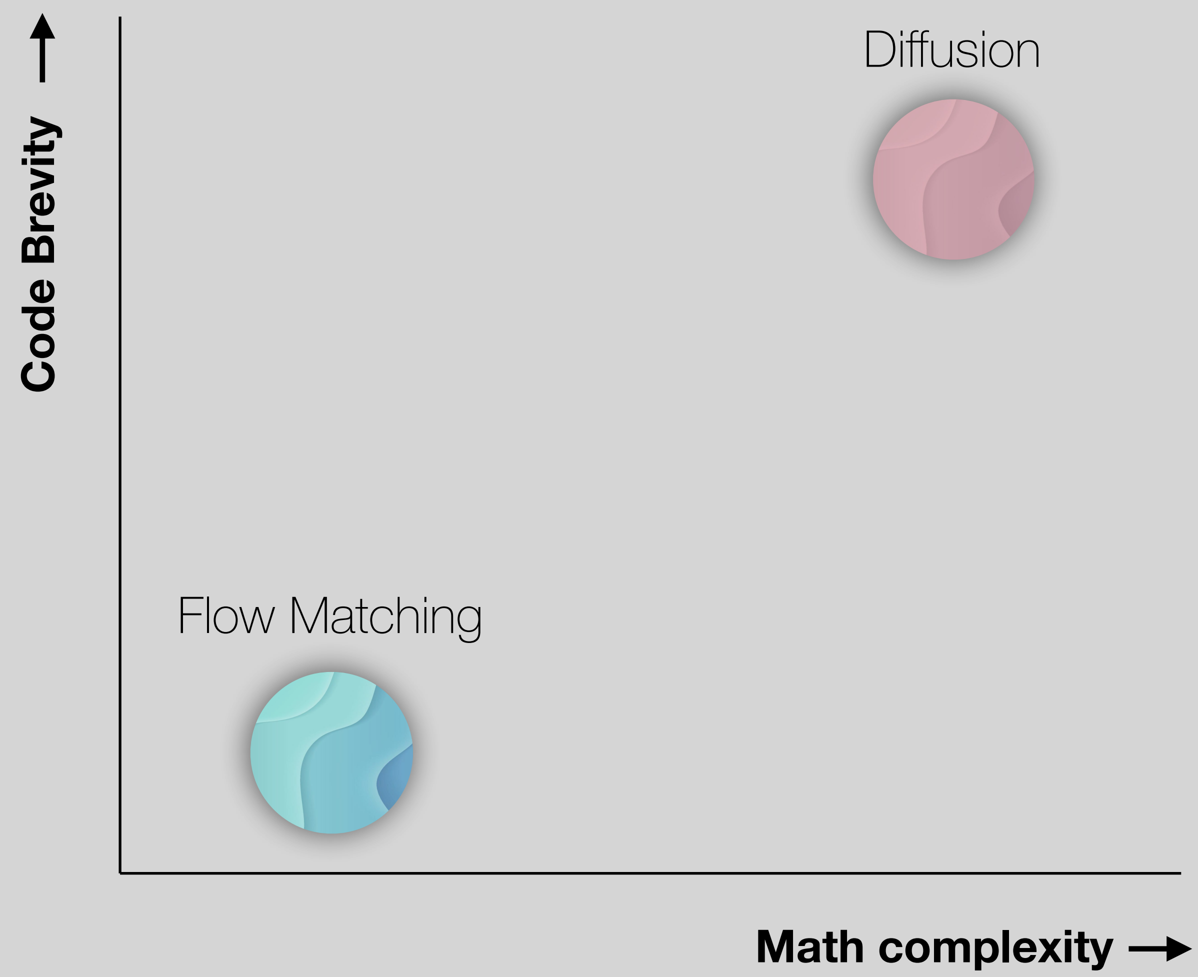
!!! Disclaimer: This presentation contains memes. OMG!

Why do Flow Matching 101…



Dufour, Nicolas, et al. "Around the world in 80 timesteps: A generative approach to global visual geolocation." CVPR 2025.

Why do Flow Matching 102…



Join the Flow Matching Cult

Source: Yaron Lipman

Flow Matching

1

2

Flow Matching

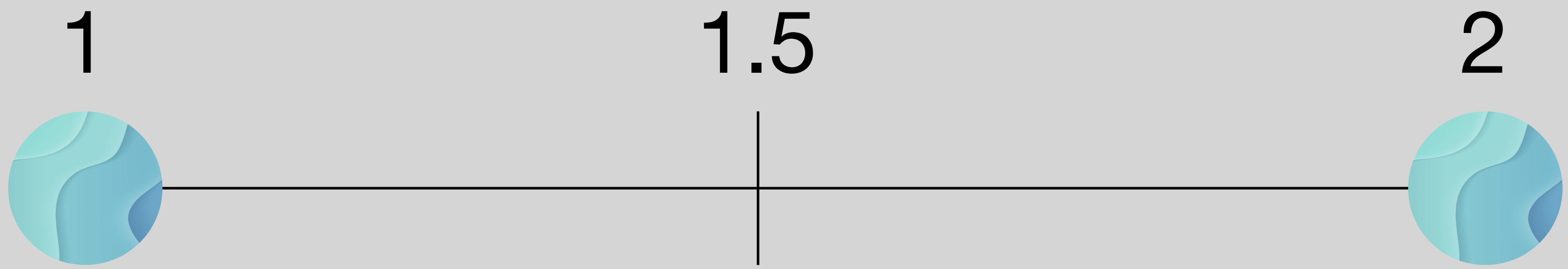
1

3 numbers

2

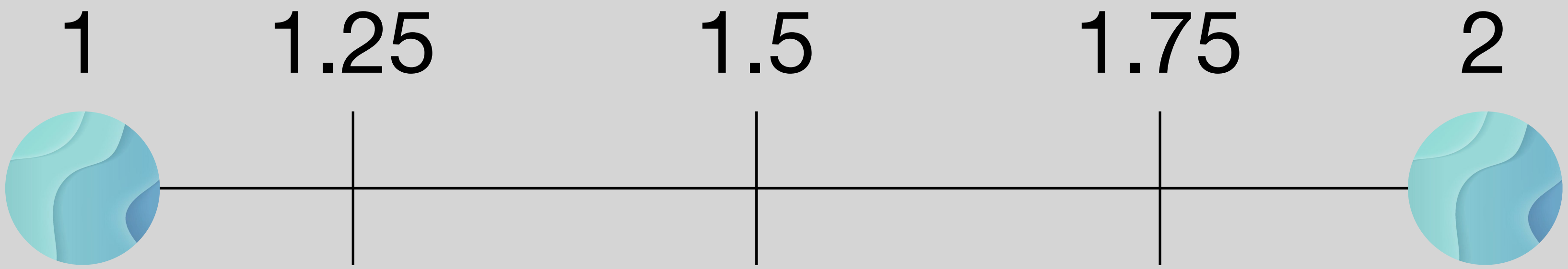
Flow Matching

Flow Matching



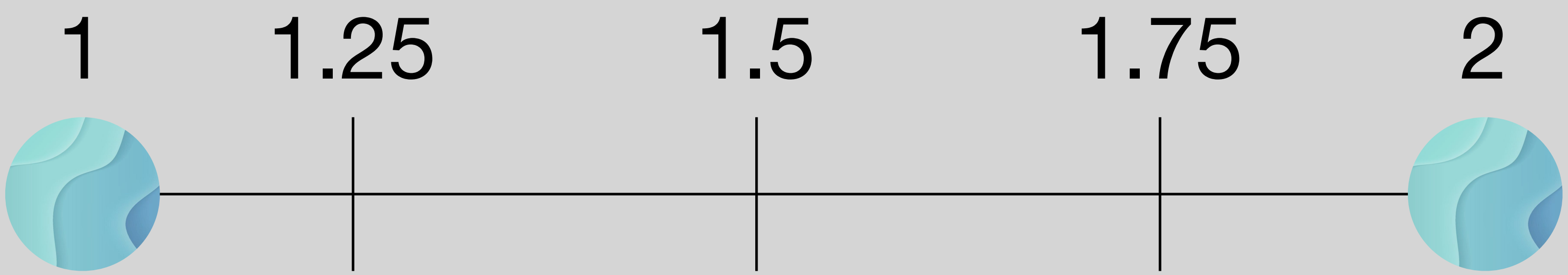
Flow Matching

Flow Matching



Here are the 3 numbers

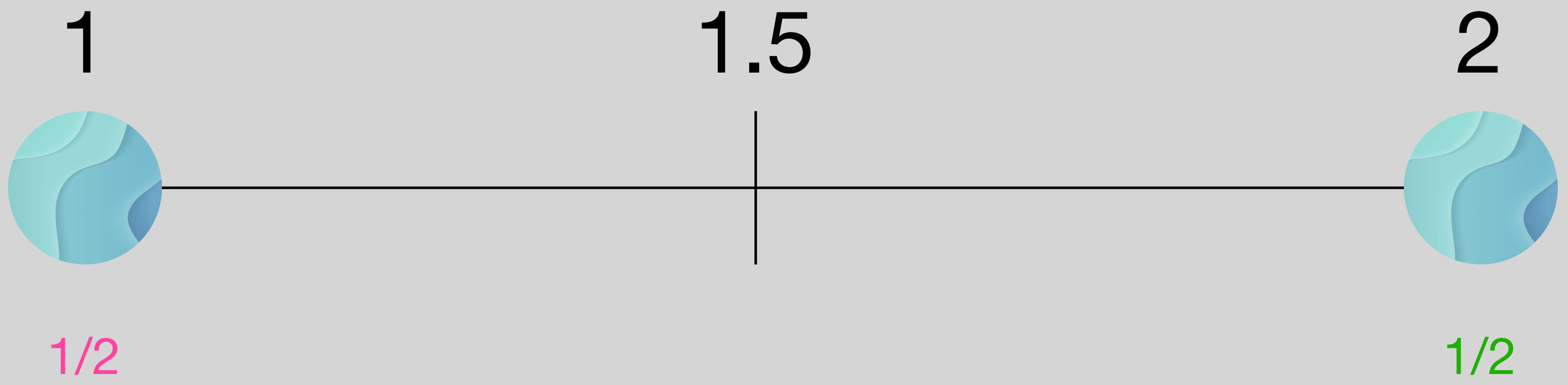
Flow Matching



How do you get them?

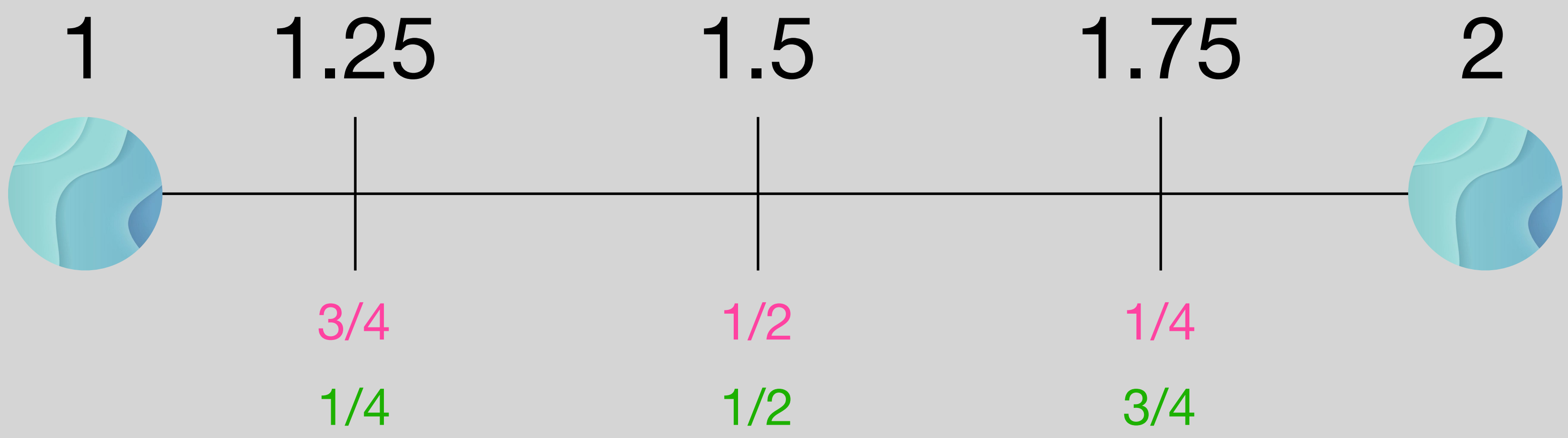
Flow Matching

Flow Matching

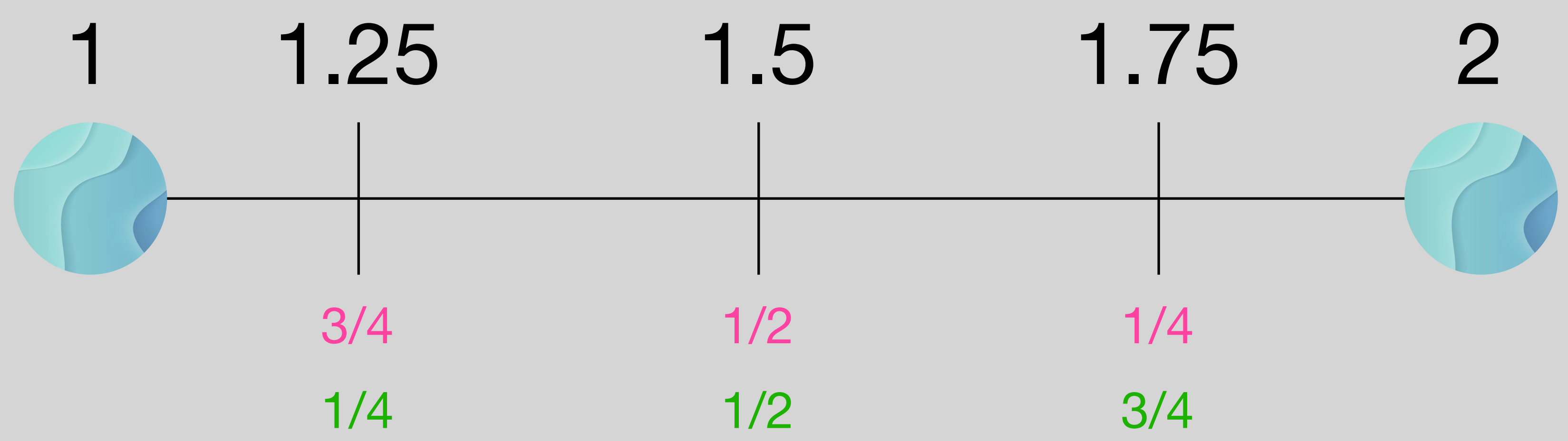


Flow Matching

Flow Matching

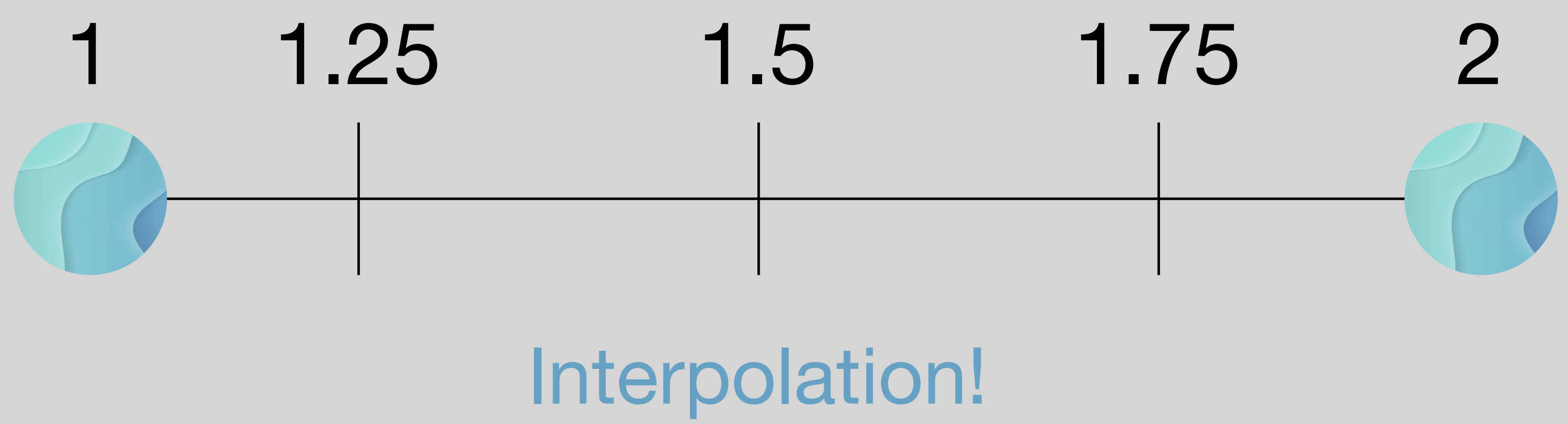


Flow Matching



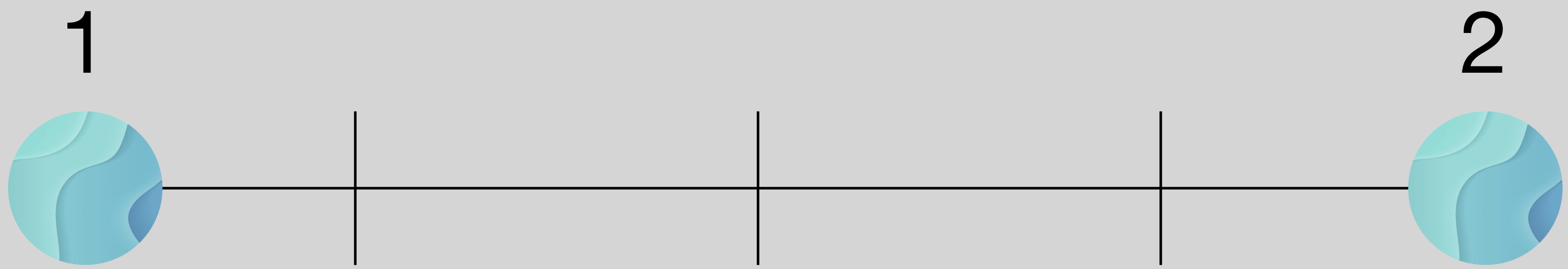
$$\pi_t = t \cdot \pi_1 + (1 - t) \cdot \pi_0$$

Flow Matching



Flow Matching

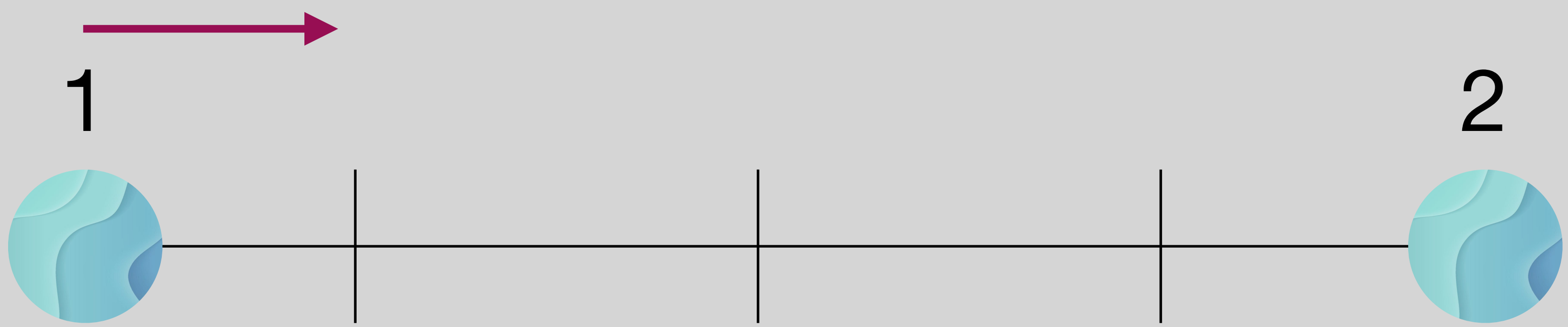
One more way!



Interpolation!

Flow Matching

One more way!



Interpolation!

Flow Matching

One more way!

$$v = \frac{dy}{dt} = 1$$

Interpolation!

Flow Matching

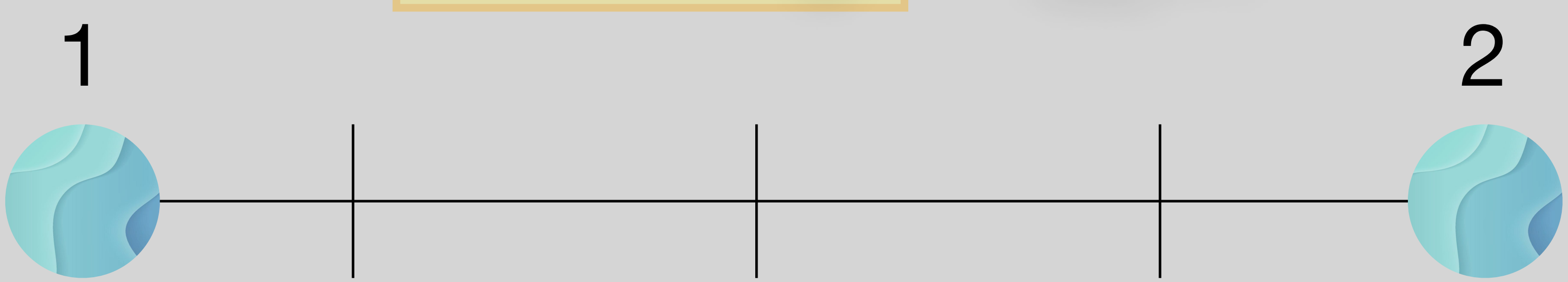
One more way!

Euler's Method

$$y_{new} = y_{old} + h \cdot \frac{dy}{dt}$$

$$\frac{dy}{dt} = 1$$

$$y_{initial} = 1$$



Interpolation!

Flow Matching

One more way!

Initial Value Problem

Unique Solution

Euler's Method

$$y_{new} = y_{old} + h \cdot \frac{dy}{dt}$$

$$\frac{dy}{dt} = 1$$

$$y_{initial} = 1$$

Interpolation!

Source: Picard–Lindelöf theorem

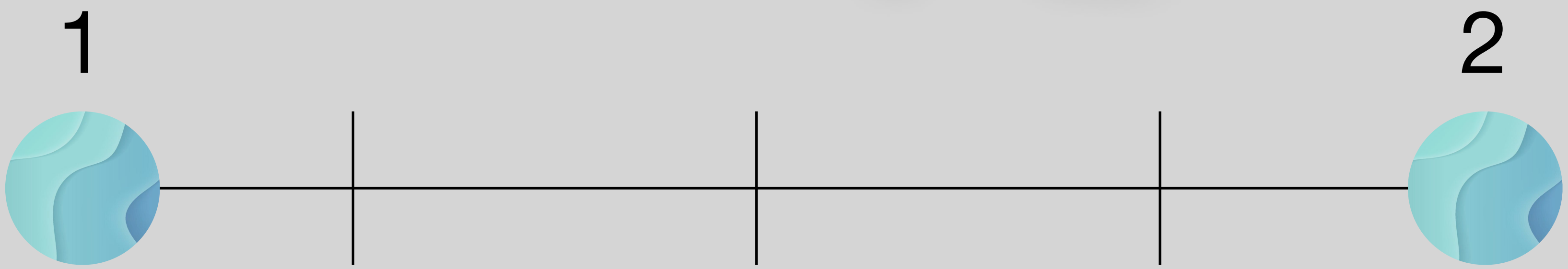
Flow Matching

One more way!

$$\frac{dy}{dt} = 1$$

$$y_1 = y_{initial} + 0.25 \cdot \frac{dy}{dt}$$

$$y_{initial} = 1$$



Interpolation!

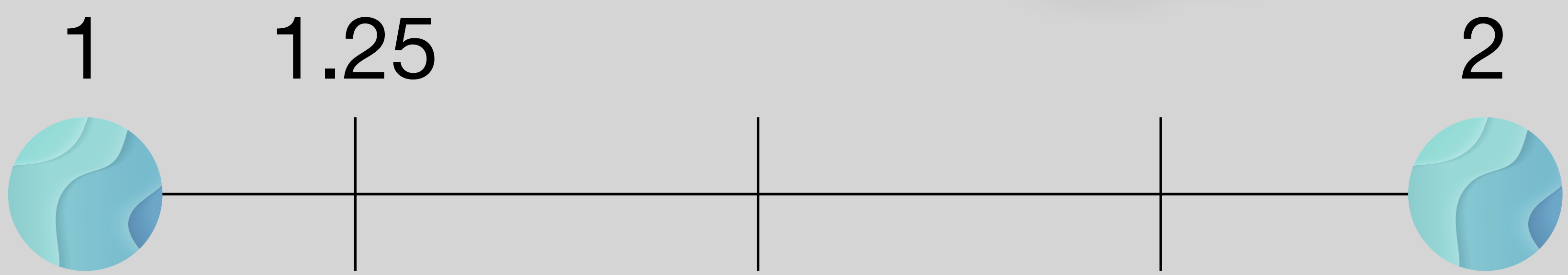
Flow Matching

One more way!

$$y_1 = 1 + 0.25 \cdot 1 = 1.25$$

$$\frac{dy}{dt} = 1$$

$$y_{initial} = 1$$

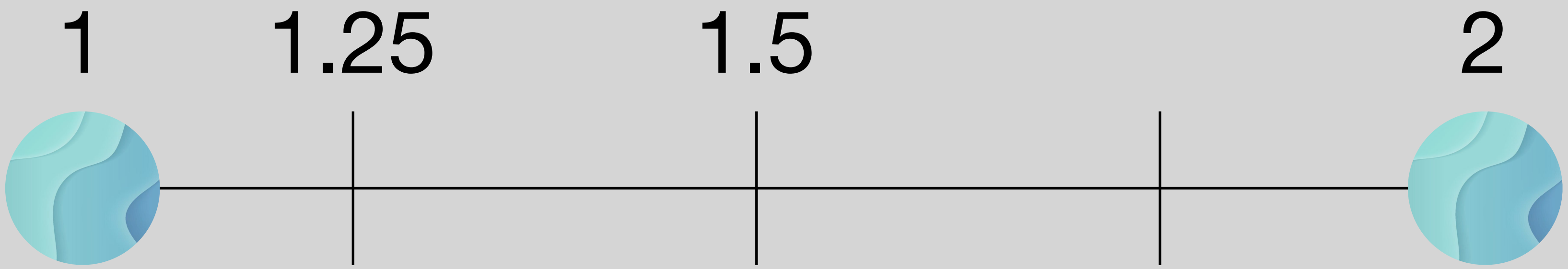


Interpolation!

Flow Matching

One more way!

$$y_2 = 1.25 + 0.25 \cdot 1 = 1.5$$

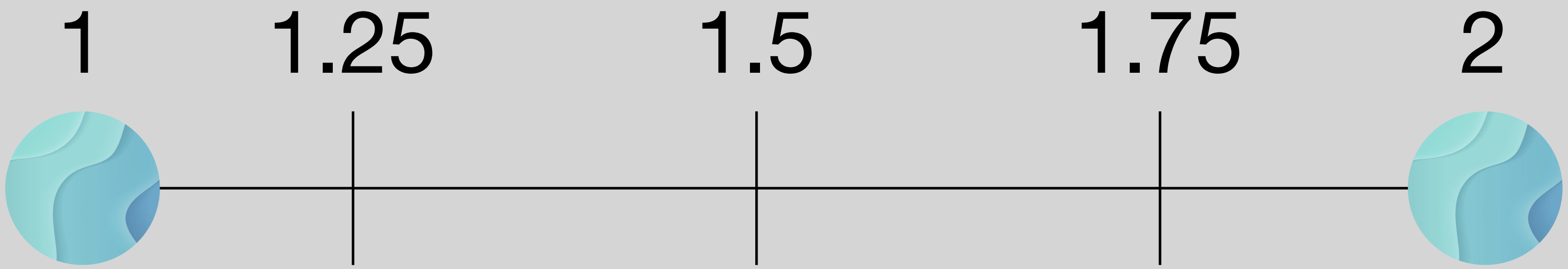


Interpolation!

Flow Matching

One more way!

$$y_3 = 1.5 + 0.25 \cdot 1 = 1.75$$

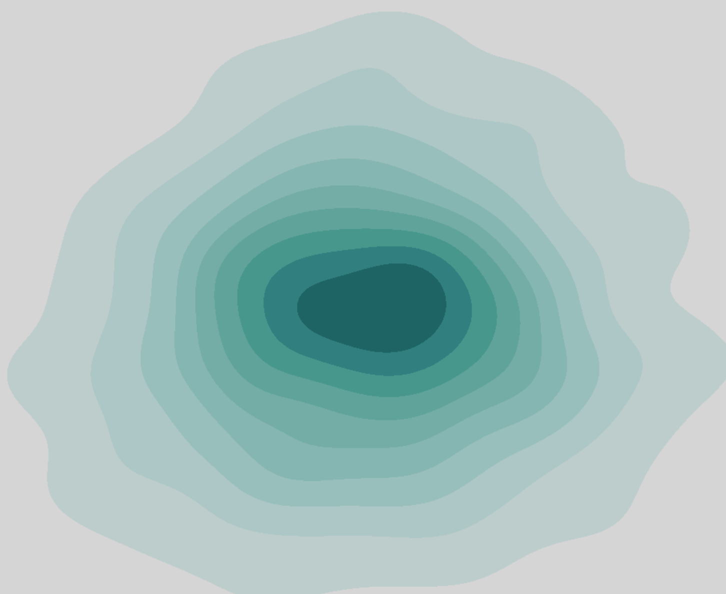


Interpolation!

Flow Matching is literally Interpolation

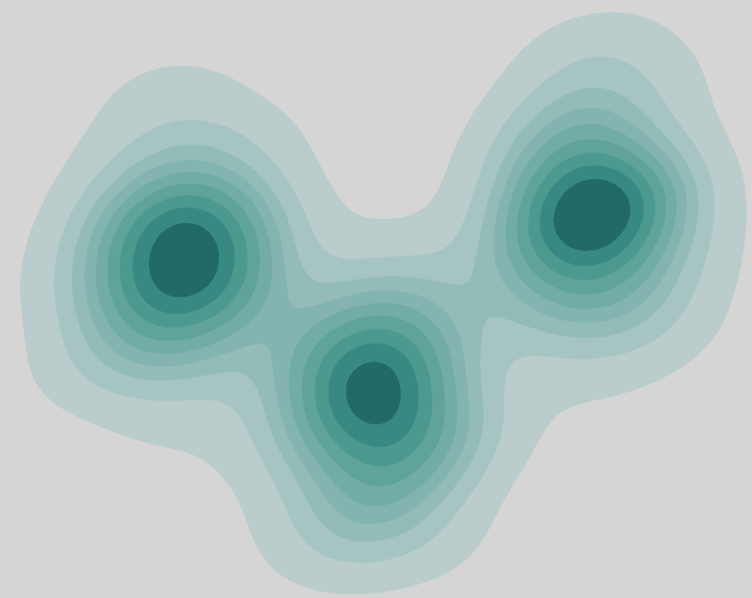
Source: memegenerator.org

Flow Matching is literally Interpolation



Source Distribution

π_0

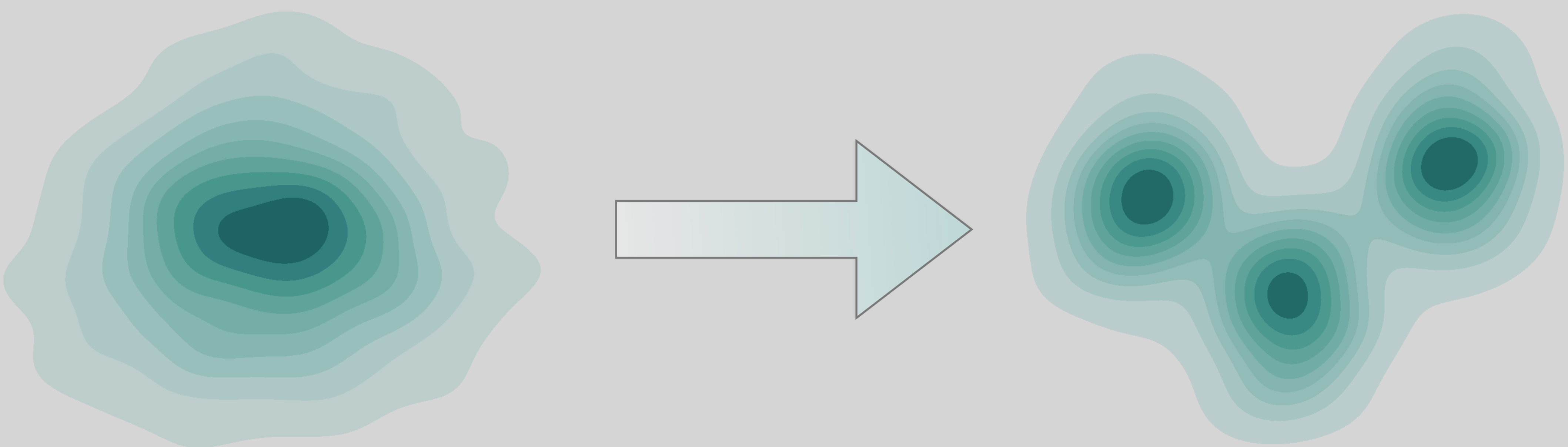


Target Distribution

π_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Flow Matching is literally Interpolation

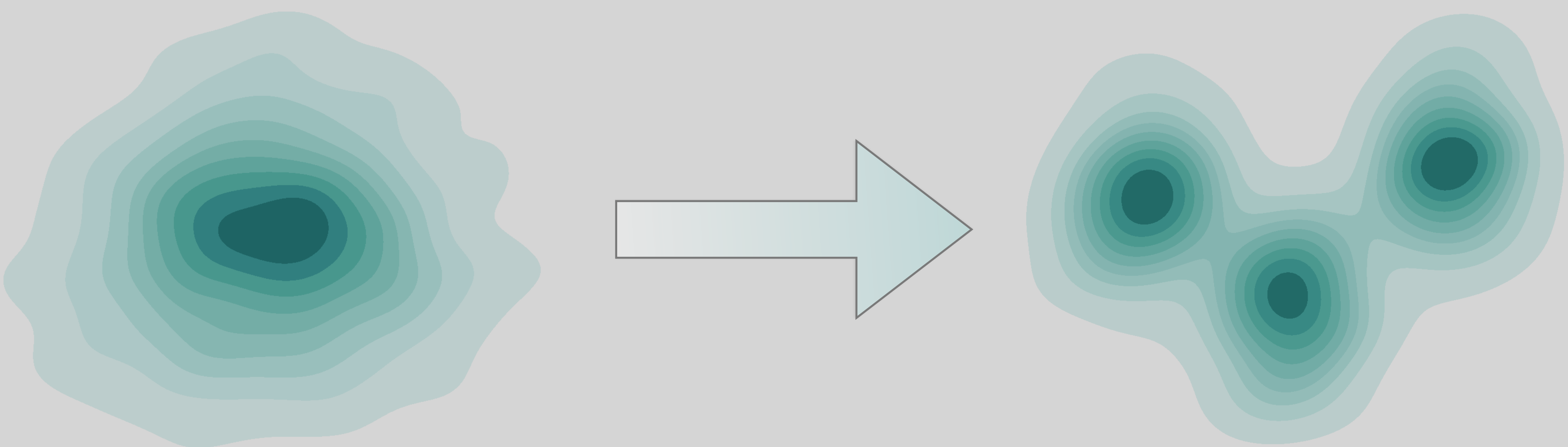


Source Distribution π_0

Target Distribution π_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Flow Matching is literally Interpolation



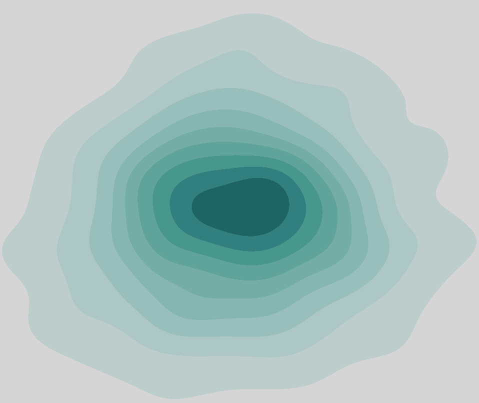
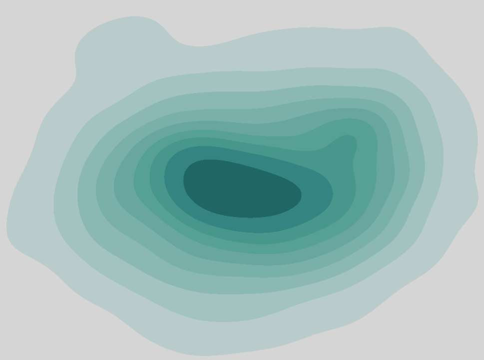
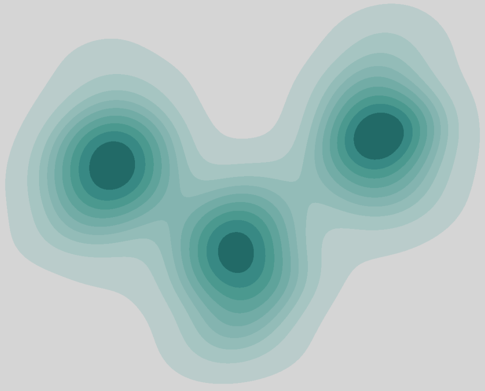
Source Distribution π_0

Target Distribution π_1

$$\pi_t = t \cdot \pi_1 + (1 - t) \cdot \pi_0$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Flow Matching is literally Interpolation

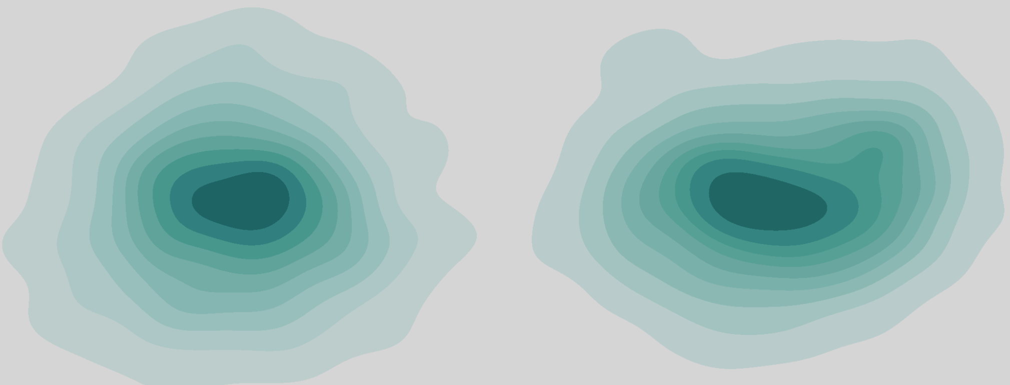
Source Distribution

Target Distribution

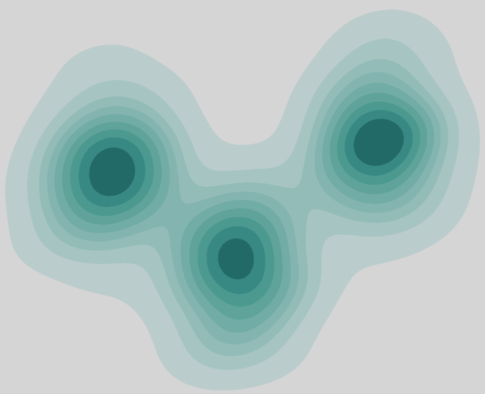
$$\pi_t = \frac{1}{4} \cdot \pi_1 + \frac{3}{4} \cdot \pi_0$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Flow Matching is literally Interpolation



Source Distribution



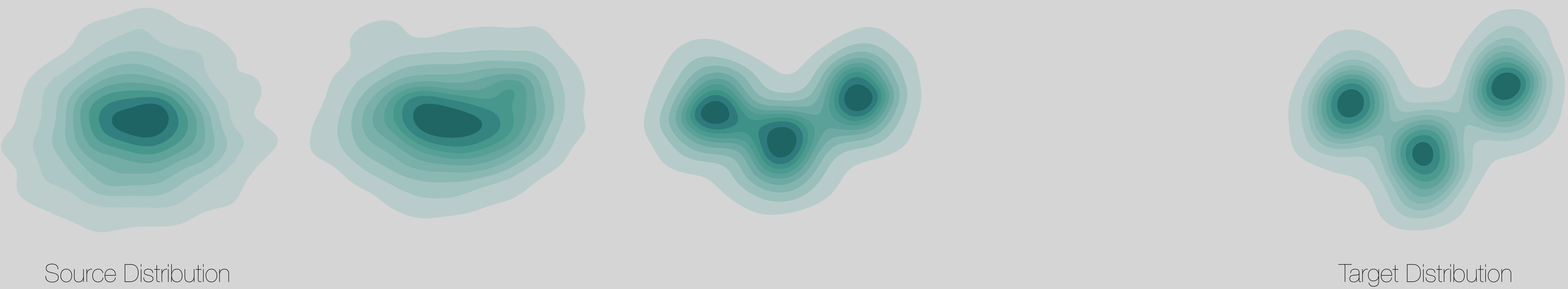
Target Distribution

$$\pi_{\frac{1}{4}} = \frac{1}{4} \cdot \pi_1 + \frac{3}{4} \cdot \pi_0$$

$$\pi_{\frac{1}{4}} = \pi_0 + \frac{1}{4} \cdot \nabla \pi_t$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Flow Matching is literally Interpolation

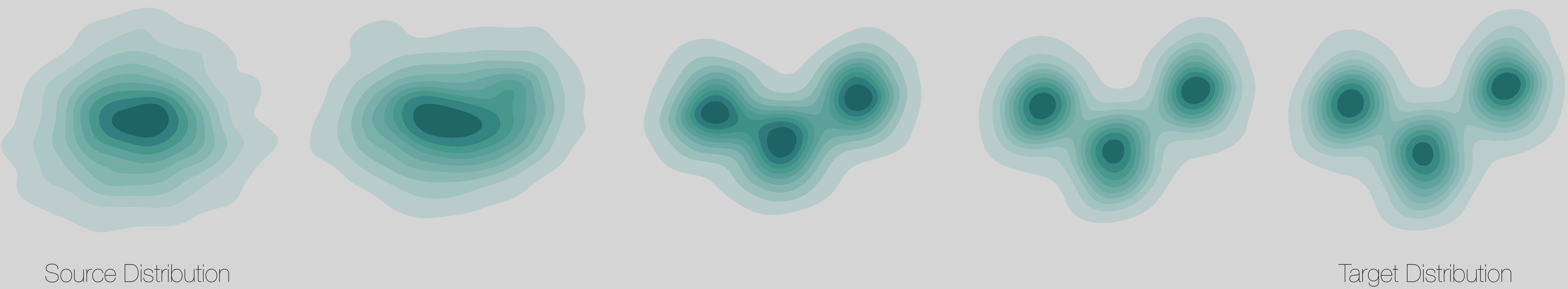


$$\pi_{\frac{1}{4}} = \frac{1}{2} \cdot \pi_1 + \frac{1}{2} \cdot \pi_0$$

$$\pi_{\frac{1}{2}} = \pi_{\frac{1}{4}} + \frac{1}{4} \cdot \nabla \pi_t$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Flow Matching is literally Interpolation

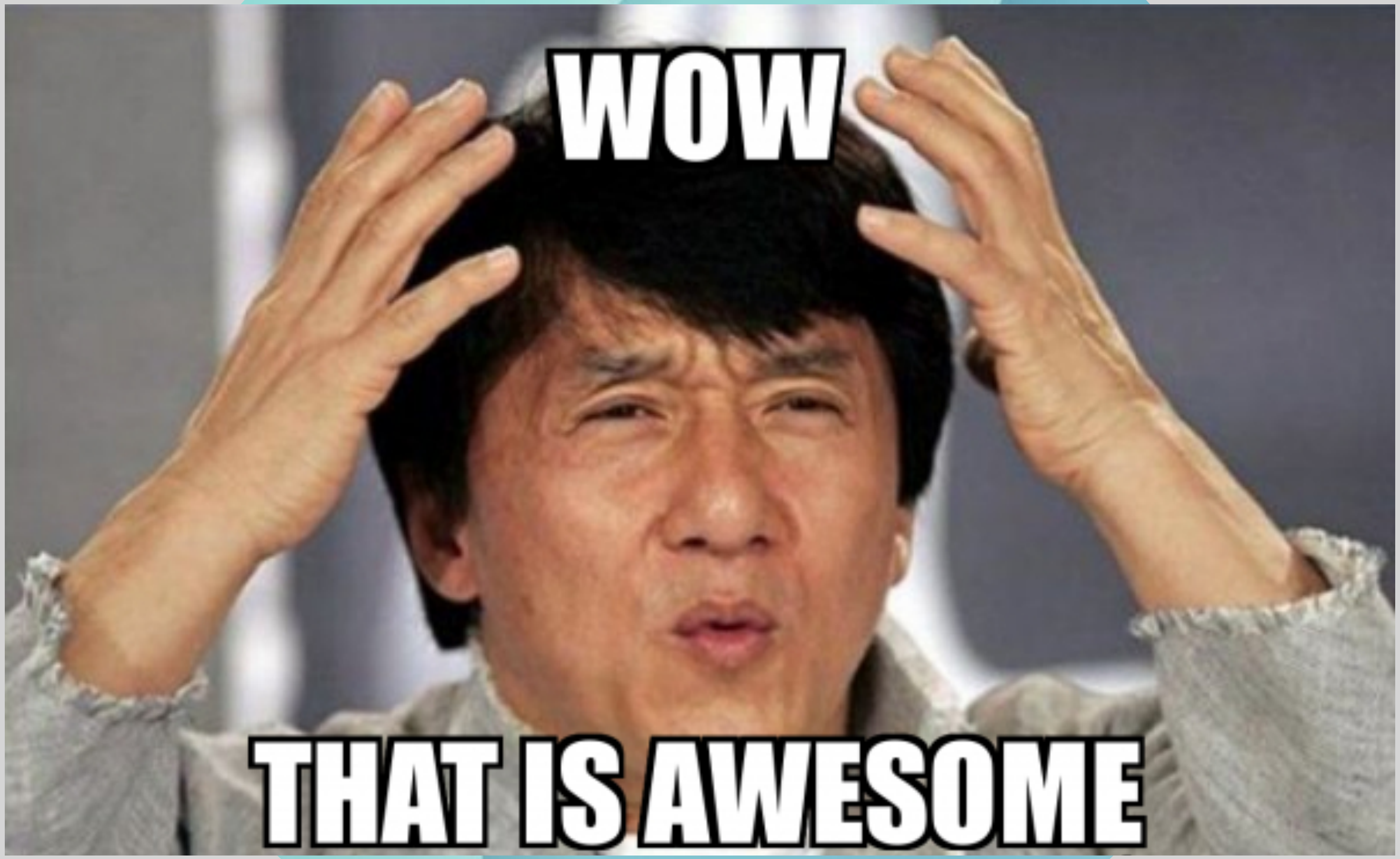


$$\pi_{\frac{3}{4}} = \frac{3}{4} \cdot \pi_1 + \frac{1}{4} \cdot \pi_0$$

$$\pi_{\frac{3}{4}} = \pi_{\frac{1}{2}} + \frac{1}{4} \cdot \nabla \pi_t$$

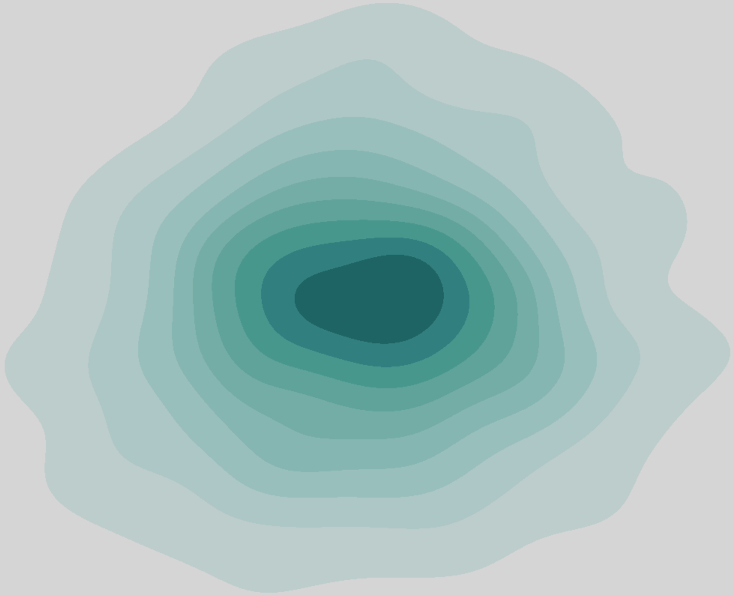
Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Flow Matching is So Simple!

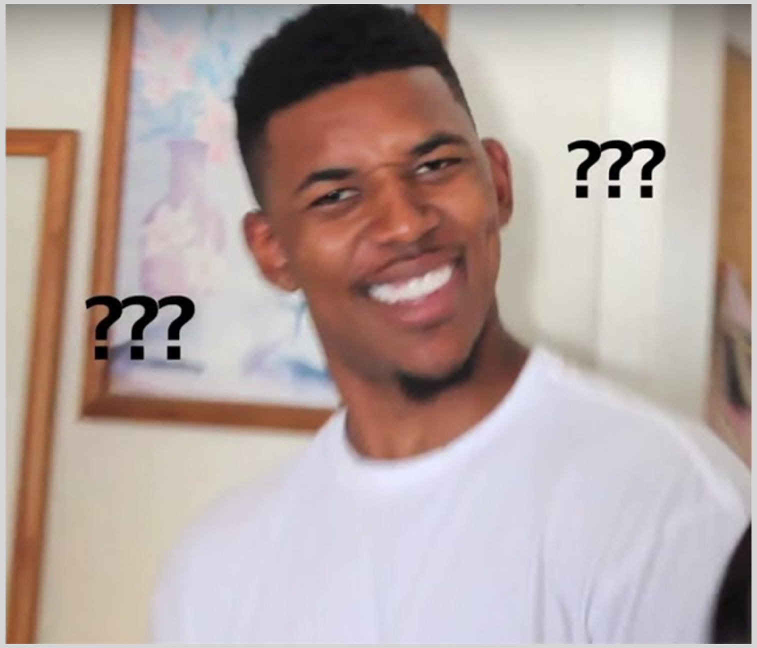


Source: makeameme.org

However…



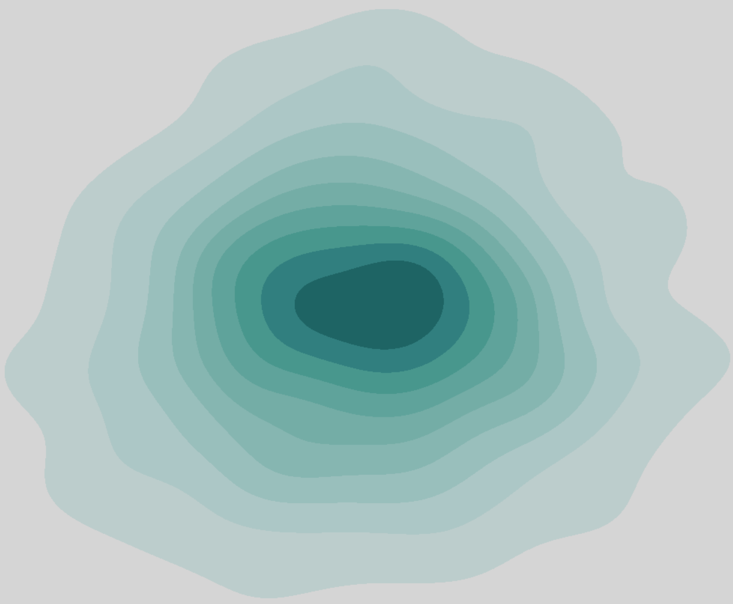
Source Distribution



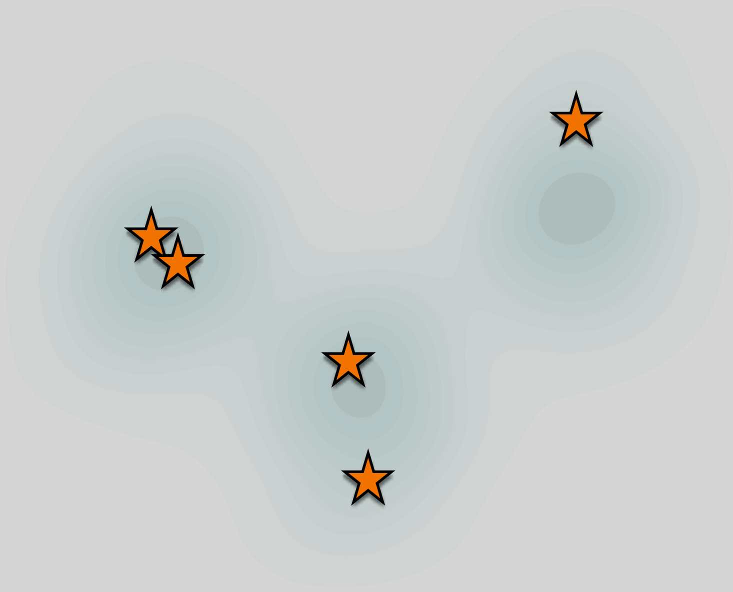
Target Distribution???

Source: Cassy Athena's YouTube episode with Nick Young

However…

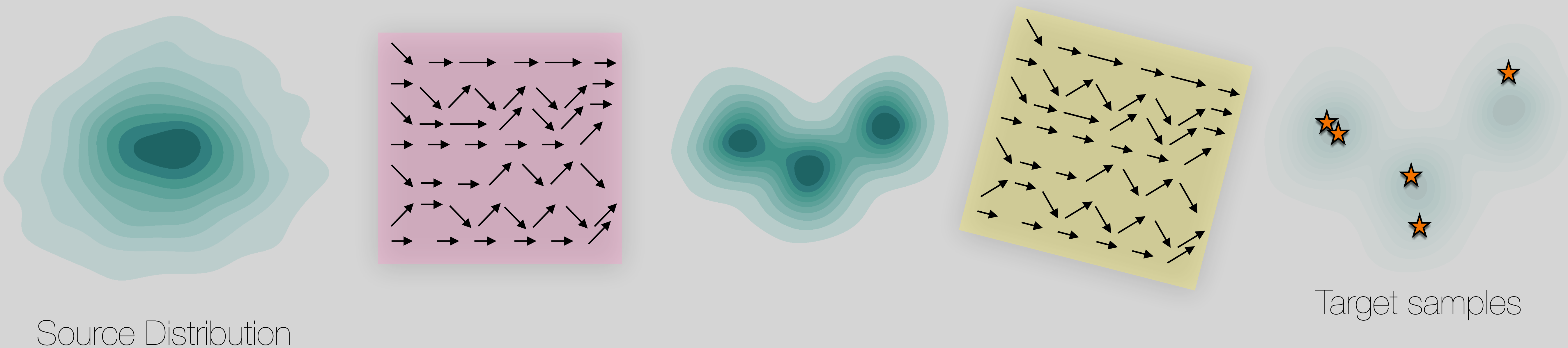


Source Distribution



Target samples

Can we get the velocity to get to intermediates?



Solution 101 : Random coupling

Source Samples

Target samples

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Solution 101 : Random coupling

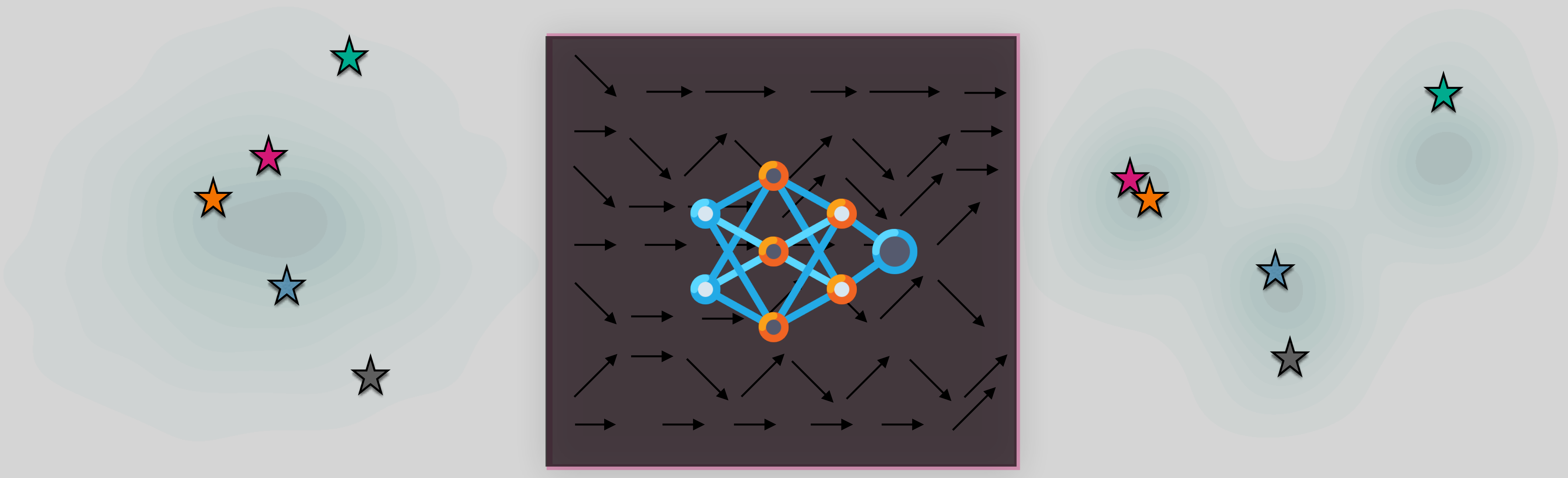
Source Samples

Target samples

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Solution 101 : Intermediate gradient/ velocity

What are the velocities to make the intermediates?



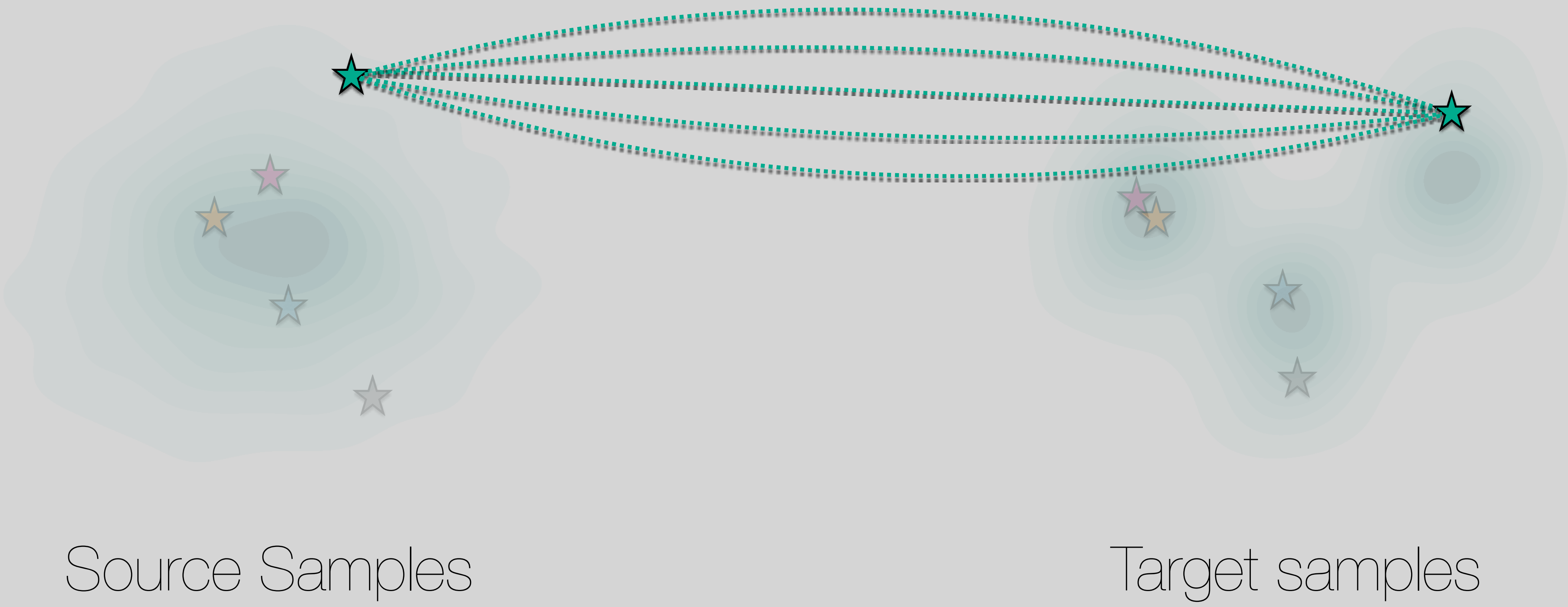
Source Samples

Target samples

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

!!! Velocity depends on Interpolation paths

Which one to pick?



Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Solution 101 : Go linear

Source Samples

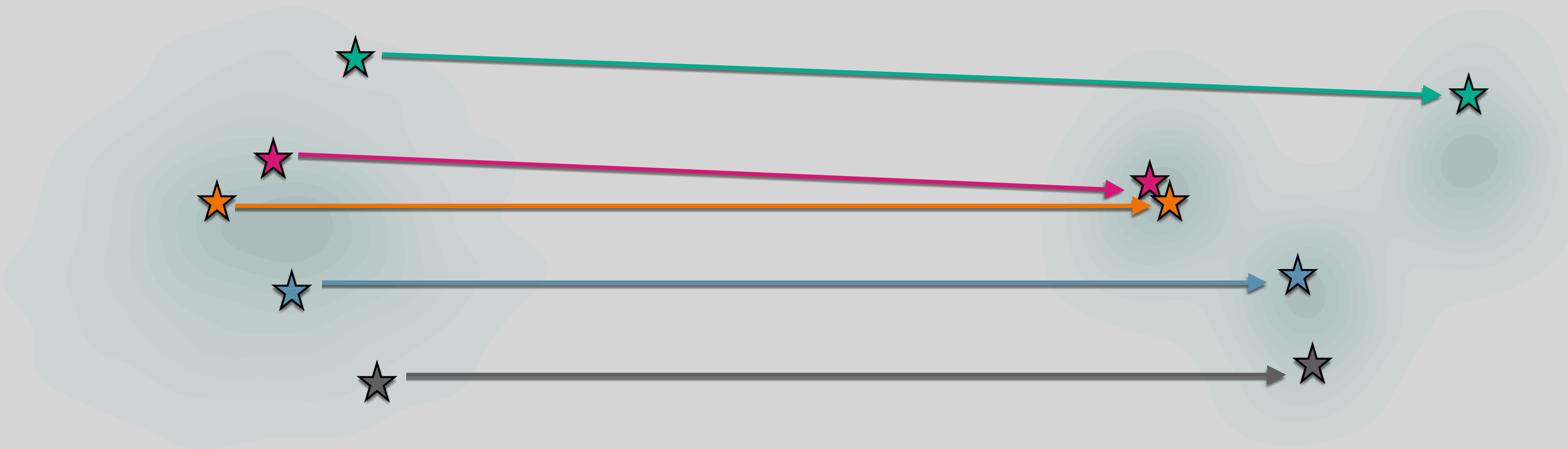
Target samples

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Source: Superman Comics

Solution 101 : Gradients

Model prediction

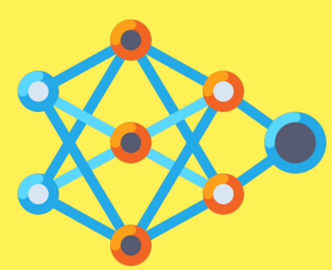


Source Samples

X_0

Target samples

X_1

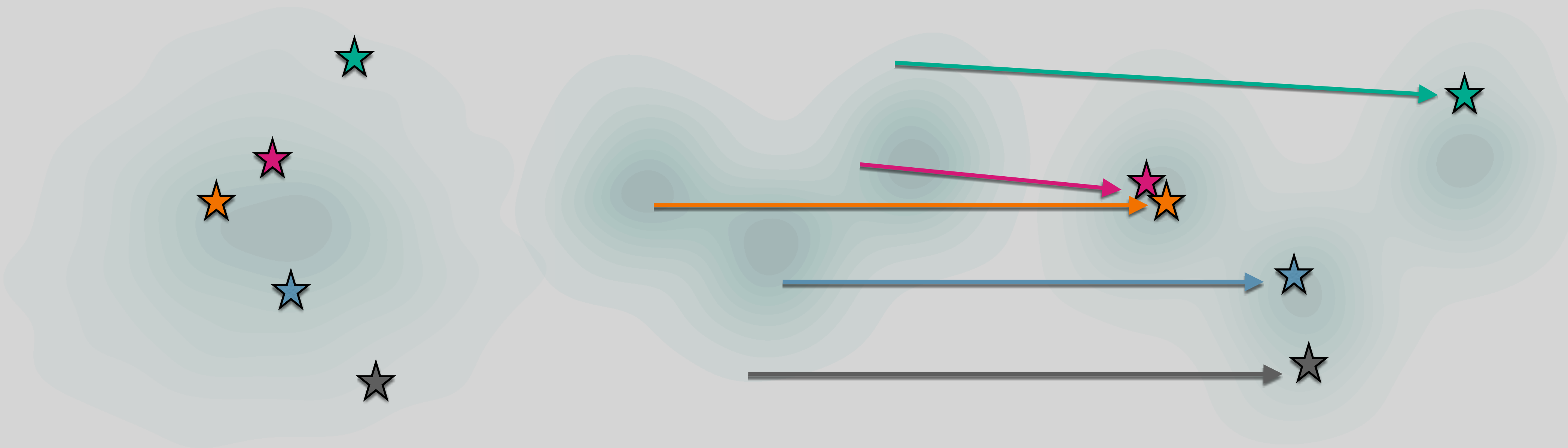


$$\nabla X_t = X_1 - X_0$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Solution 101 : Gradients

Model prediction

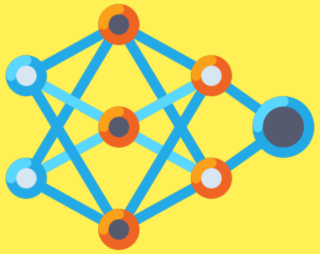


Source Samples

X_0

Target samples

X_1



$$\nabla X_t = X_1 - X_0$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.

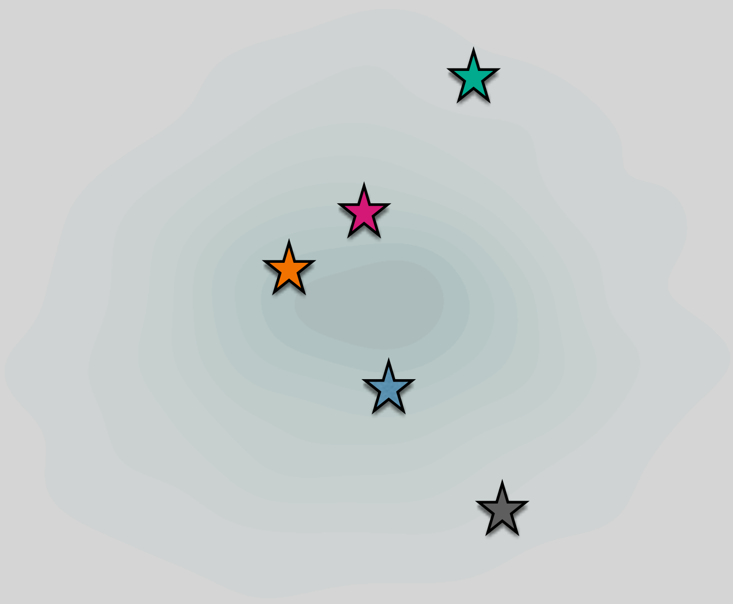
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Overall Training

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

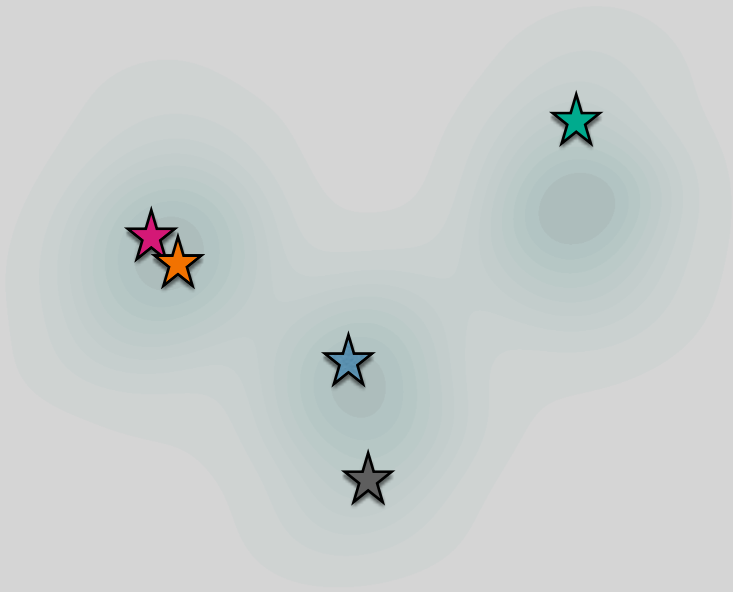
Overall Training

(1) Random Coupling



Source Samples

X_0



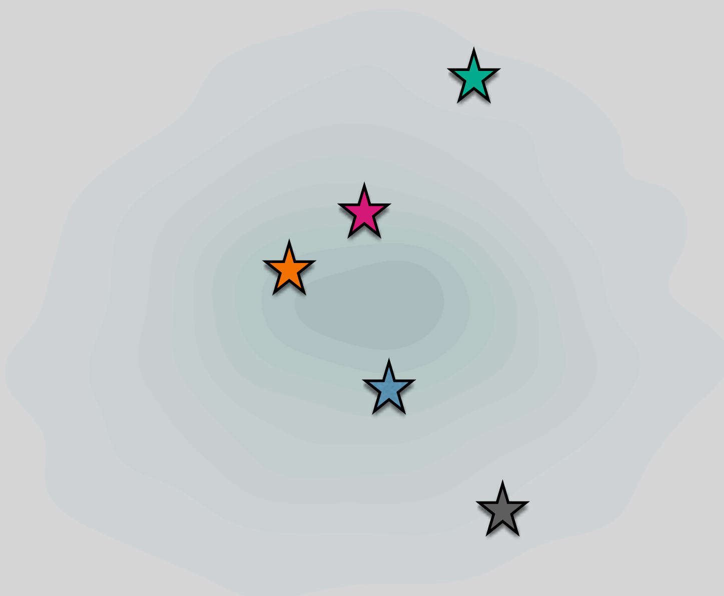
Target samples

X_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

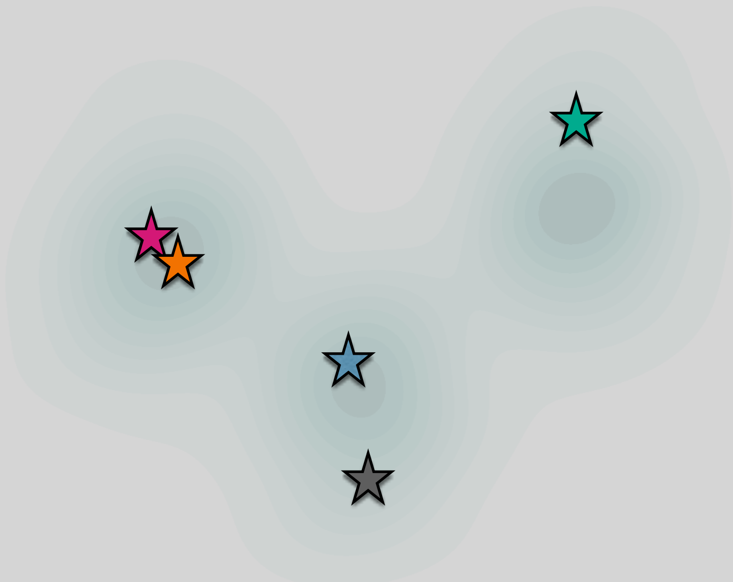
Overall Training

(2) Randomly sample t in [0,1)



Source Samples

X_0



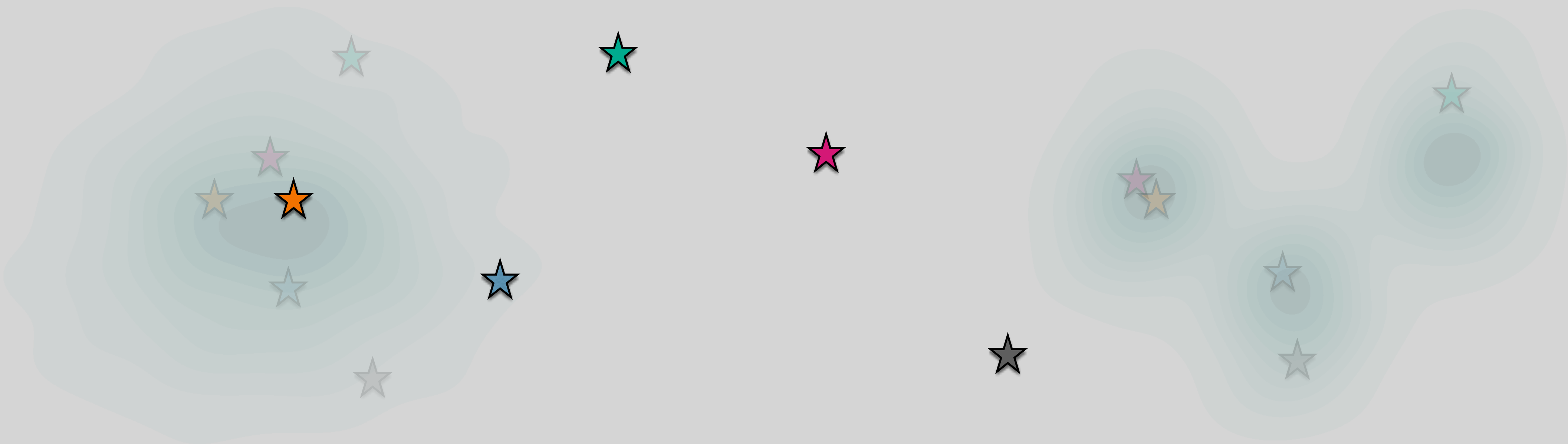
Target samples

X_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Overall Training

(3) Linear interpolation (model input)



Source Samples

X_0

Target samples

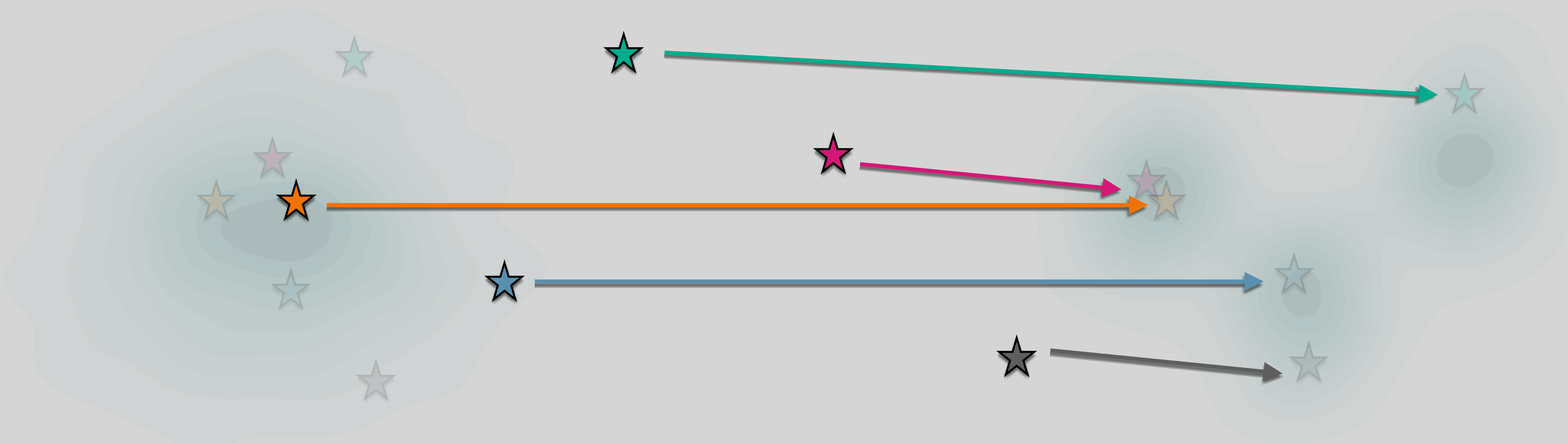
X_1

$$X_t = t \cdot X_1 + (1 - t) \cdot X_0$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Overall Training

(4) Predict Velocity (model output)

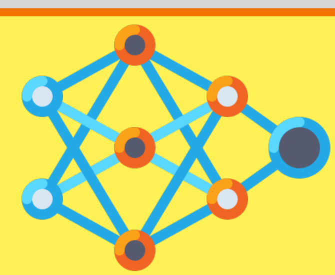


Source Samples

X_0

Target samples

X_1



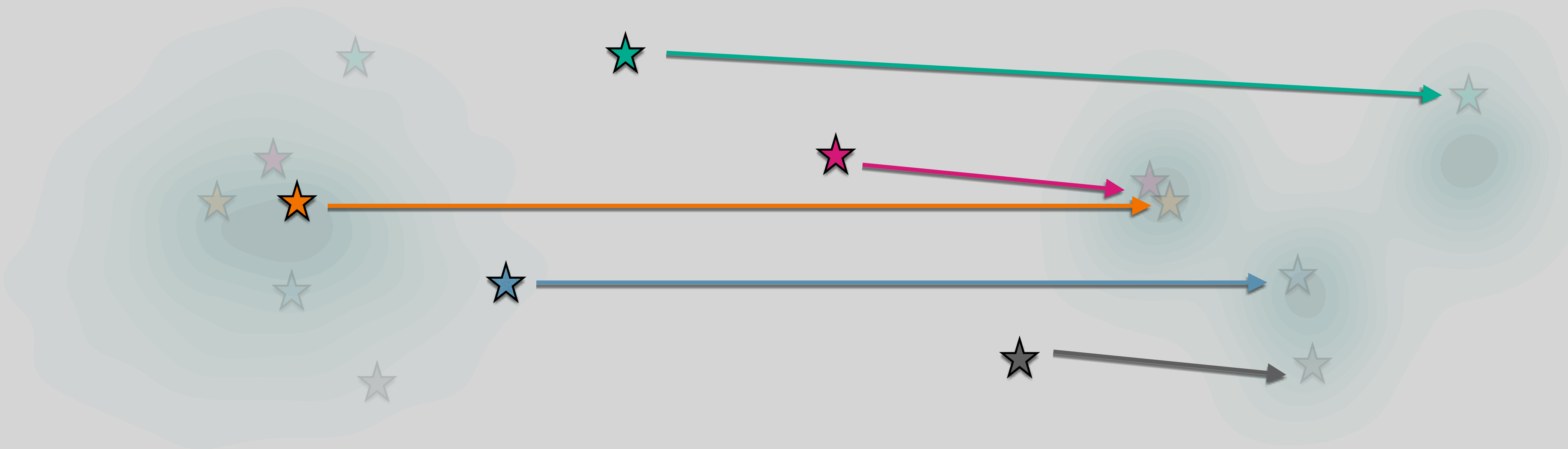
$= \nabla X_t$

$u_\theta(t)$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Overall Training

(5) Optimize



Source Samples

X_0

Target samples

X_1

$$\mathbb{E}[\,||u_\theta(t) - (X_1 - X_0)||^2\,]$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Sampling

Euler's Method

$$x_{next} = x_{old} + \frac{1}{num\_steps} \cdot u_\theta(t)$$

$$x_{initial} \sim \mathcal{N}(0, I)$$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Code 101...

```python
## Setting up the flow matching pipeline ##
class FlowMatchingPipeline:
  """Implements the Flow Matching pipeline"""

  def __init__(self):
    """Constructor"""
    self.model = model()
    self.optim = torch.optim.AdamW(self.model.parameters(), lr=1e-4)
    self.loss = nn.MSELoss()
```

Code 101...

```python
def interpolate(self, x1, t):
  """Interpolate x1 data from x0 data."""

  x0 = torch.randn(*x1.shape, device=x1.device, dtype=x1.dtype)

  xt = t * x1 + (1 - t) * x0

  vel = x1 - x0

  return xt, vel
```

Code 101...

```python
def training_step(self, data):
  """Implements one single training step for the model"""

  data = data.to(self.device)

  t = torch.rand(
      data.shape[0],
      dtype=data.dtype,
      device=self.device
  ).view(-1,1)

  xt, true_vt = self.interpolate(data, t)

  self.optim.zero_grad()

  vt = self.model(xt, t)

  loss = self.loss(true_vt, vt)
  loss_val = loss.clone().detach().cpu().item()

  loss.backward()

  self.optim.step()

  return loss_val
```

Code 101…

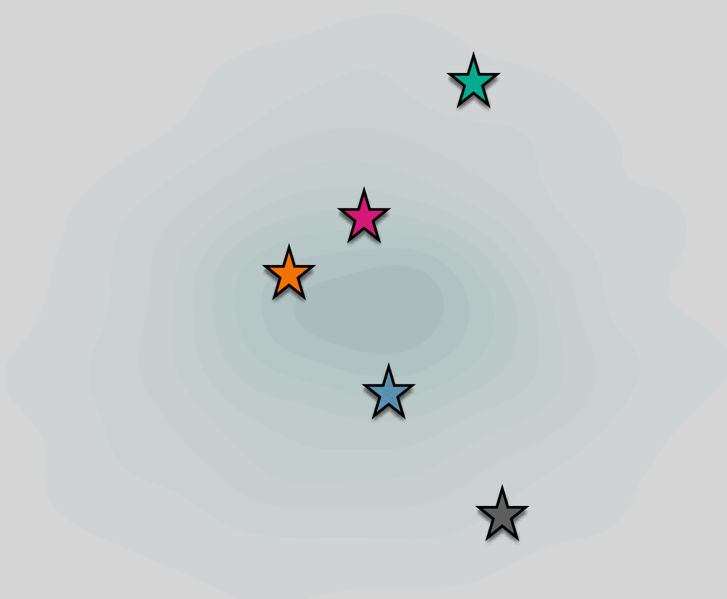
```python
@torch.no_grad()
def sample(self, num_steps, init_val):
  """Samples using euler steps using the model"""

  step_size = 1 / num_steps
  xt = init_val

  for step in range(1, num_steps):
    t = torch.tensor(
        [step / num_steps] * init_val.shape[0],
        dtype=xt.dtype,
        device=self.device
    ).view(-1,1)
    vt = self.model(xt, t)
    xt = xt + step_size * vt

  return xt
```

Gimme the Theory…

Source: https://tapas.io/series/grumpy-codes

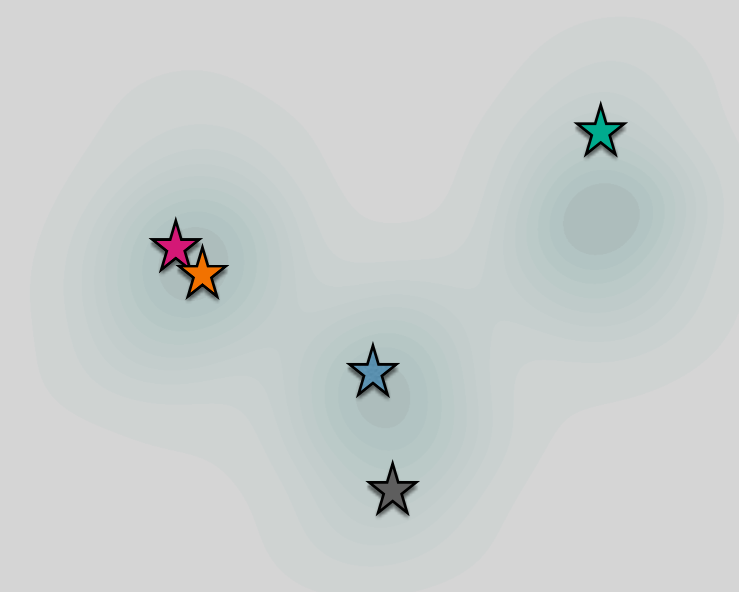
Theory of Flow Matching 101…

Recap: Random Coupling



Source Samples

X_0



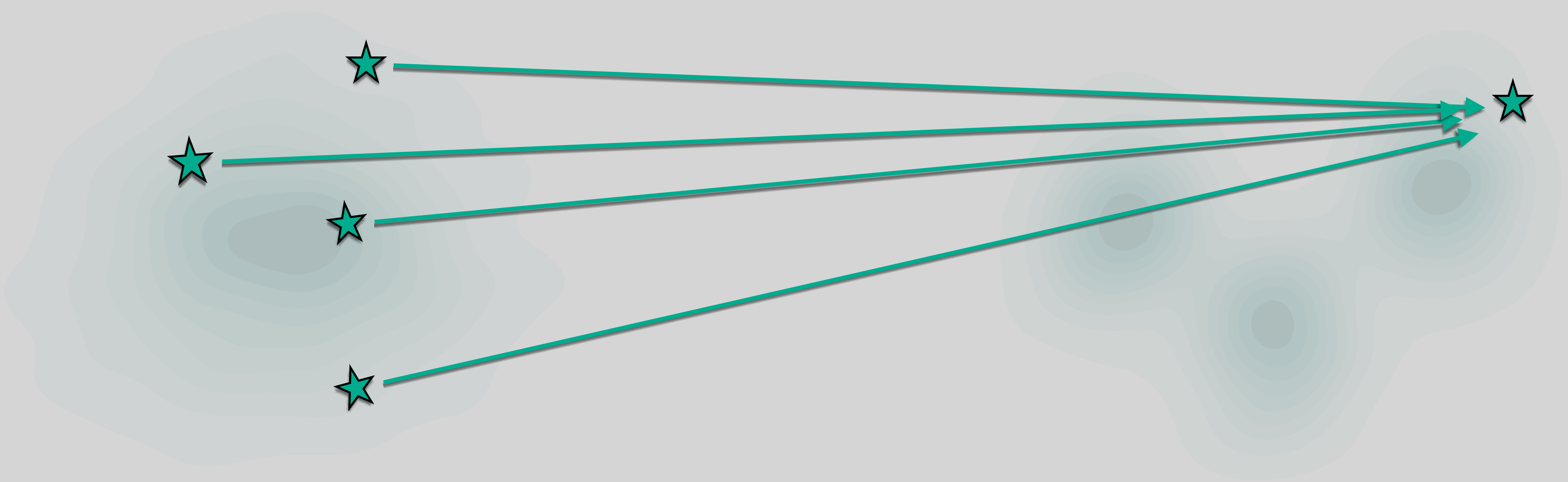
Target samples

X_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Recap: Random Coupling



Source Samples

X_0

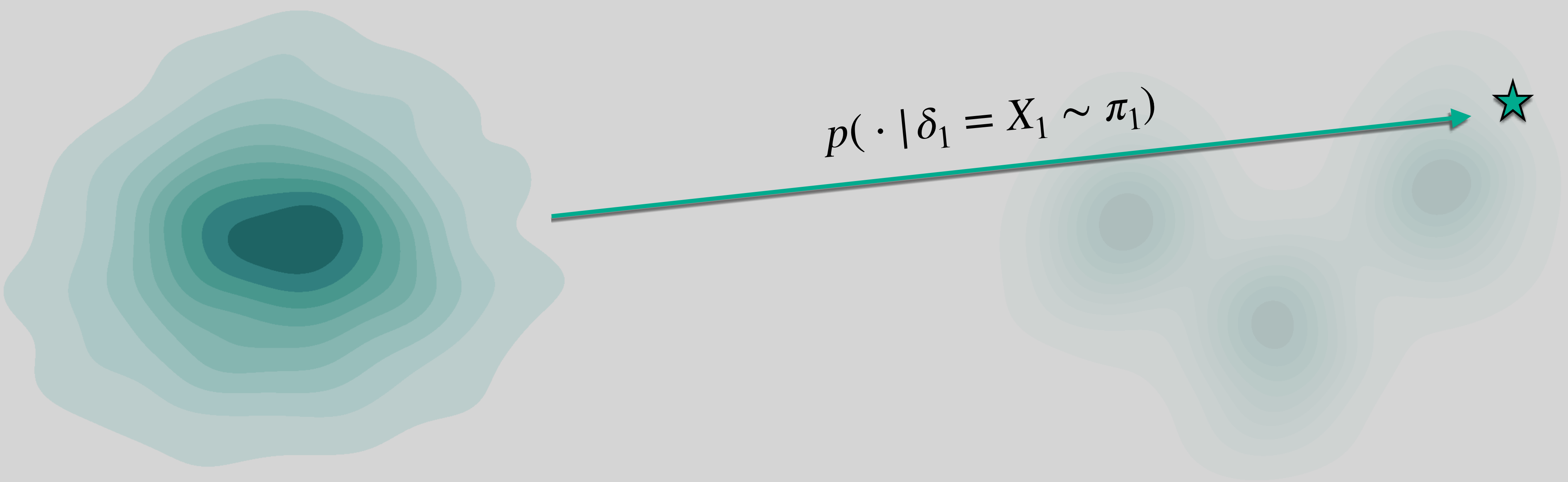
Target samples

X_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Conditional Probability Path



Source Distribution

π_0

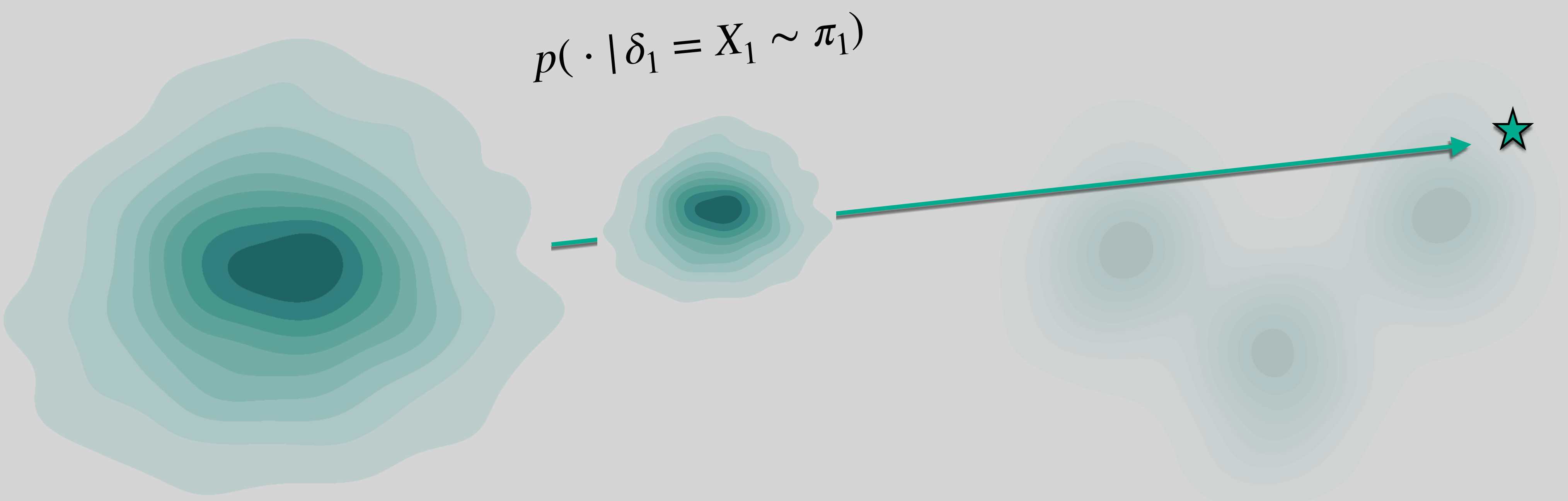
Sample Distribution

$X_1 = \delta_1$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Conditional Probability Path



Source Distribution

π_0

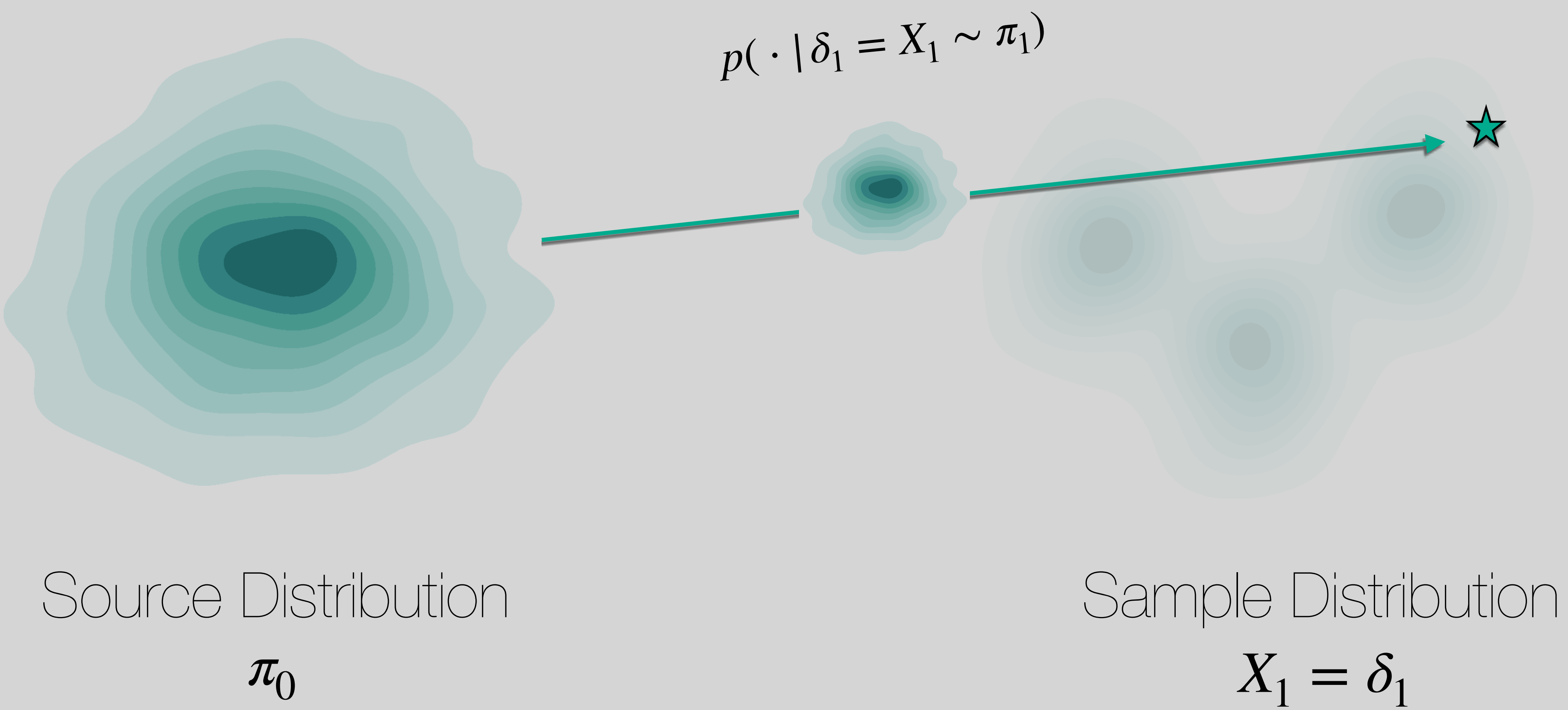
Sample Distribution

$X_1 = \delta_1$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

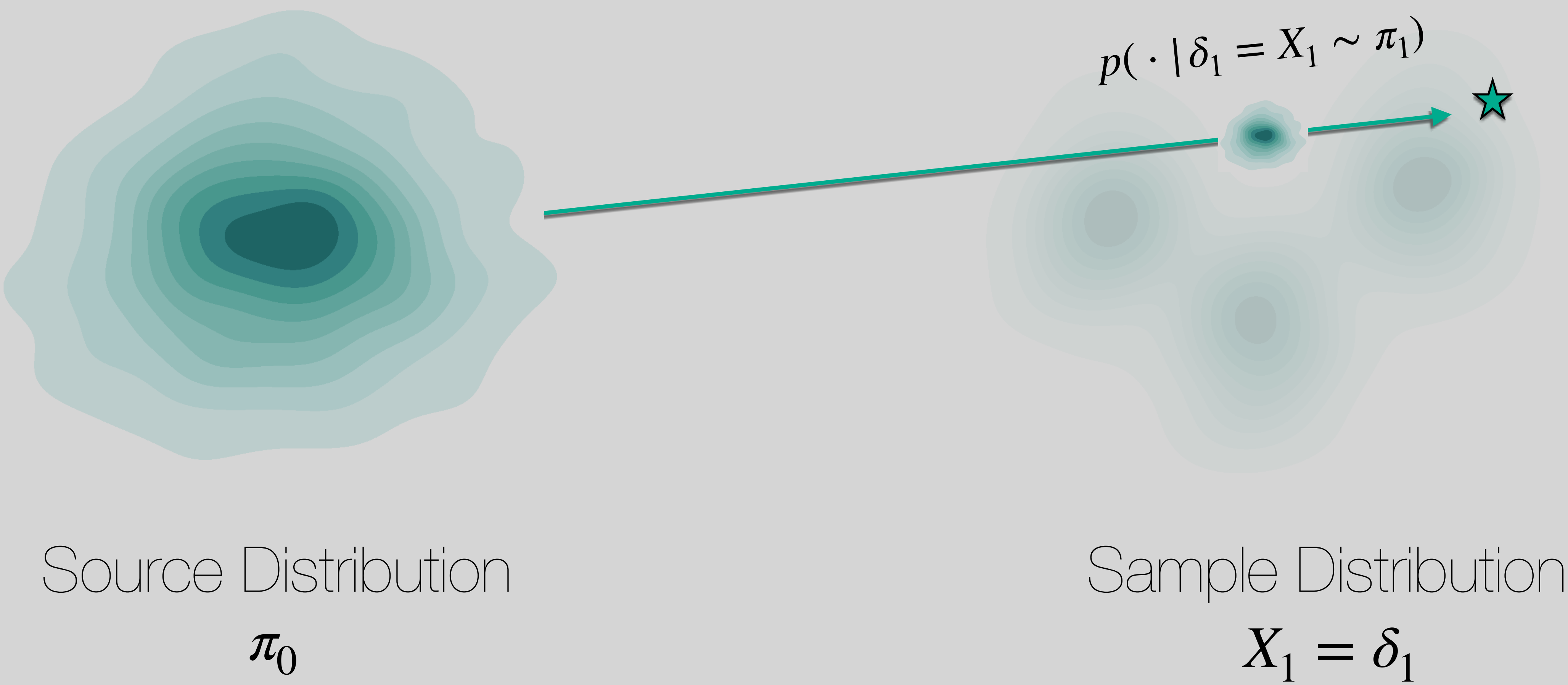
Conditional Probability Path



Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

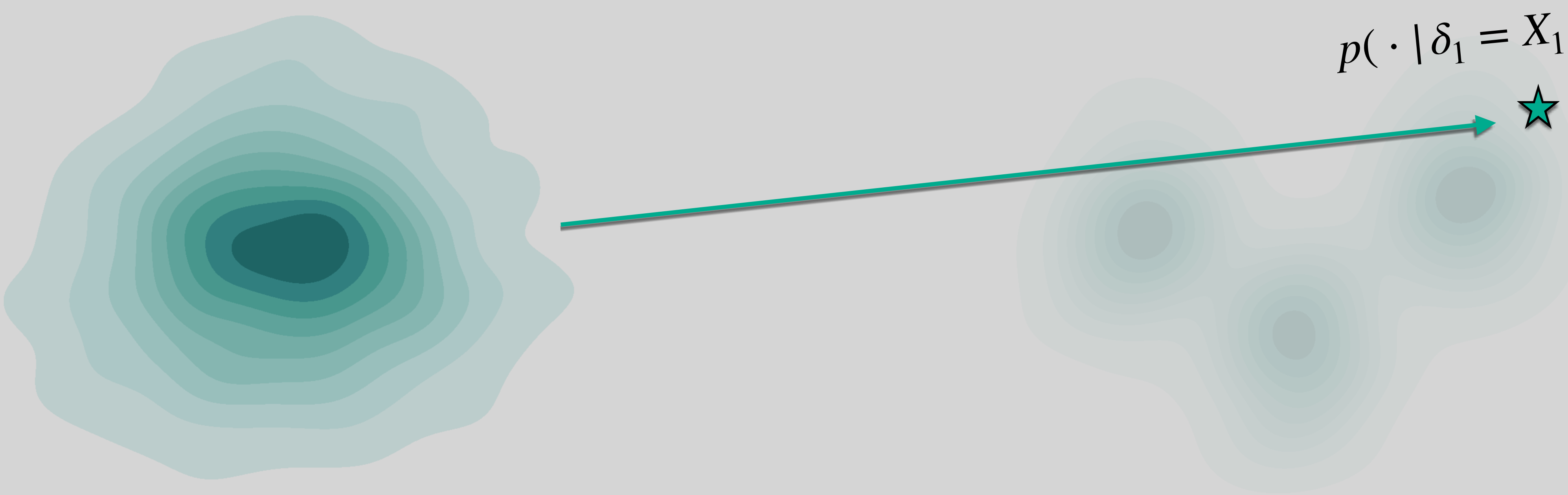
Conditional Probability Path



Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Conditional Probability Path



Source Distribution

π_0

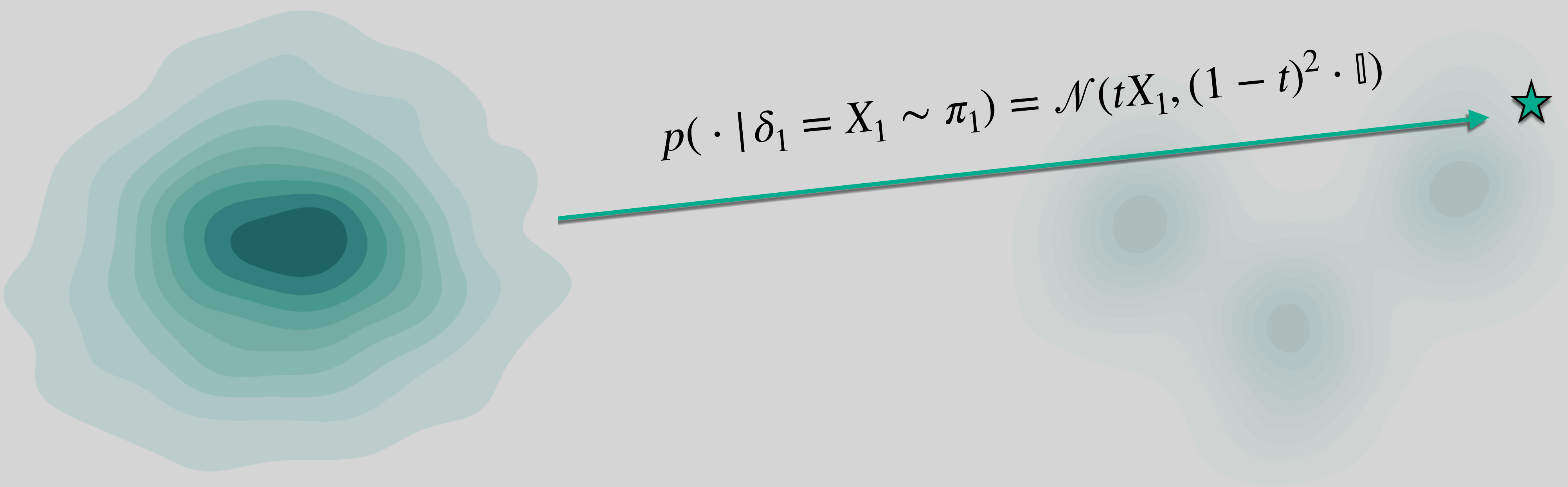
Sample Distribution

$X_1 = \delta_1$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Conditional Probability Path



Source Distribution

π_0

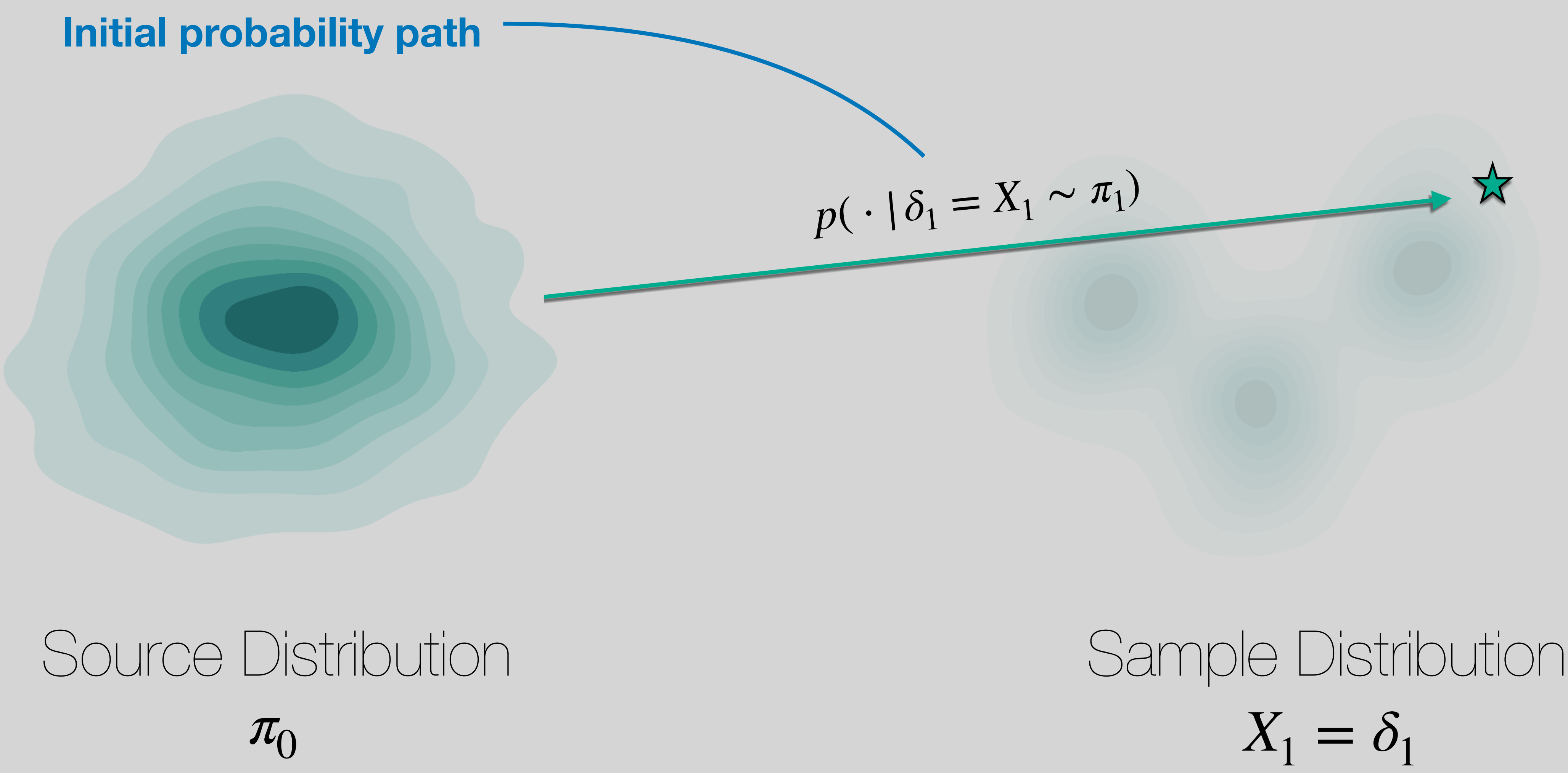
Sample Distribution

$X_1 = \delta_1$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

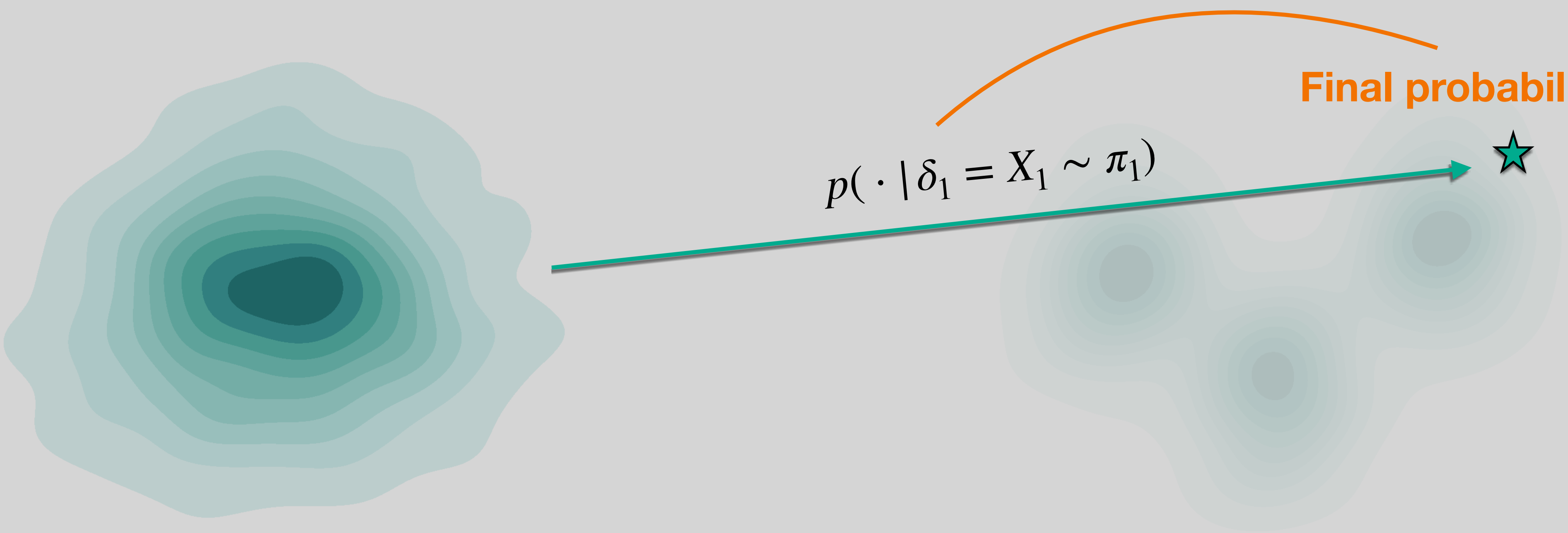
Conditional Probability Path



Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Conditional Probability Path



Source Distribution

π_0

Sample Distribution

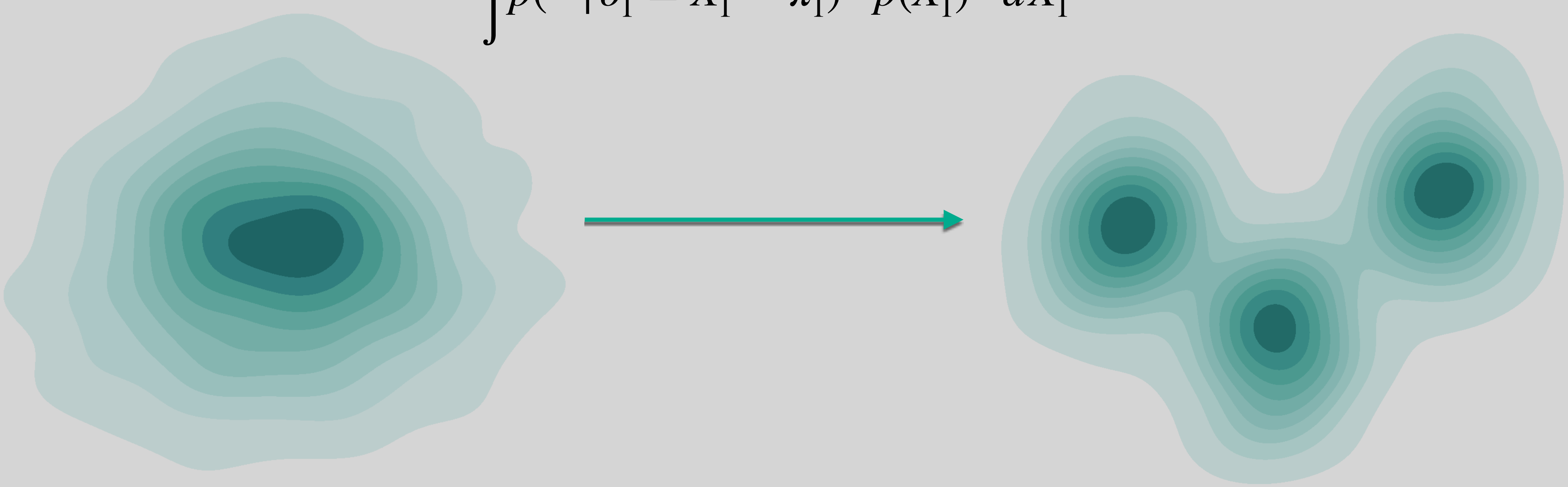
$X_1 = \delta_1$

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Marginal Probability Path

$$\int p(\,\cdot \mid \delta_1 = X_1 \sim \pi_1) \cdot p(X_1) \cdot dX_1$$



Source Distribution

π_0

Target Distribution

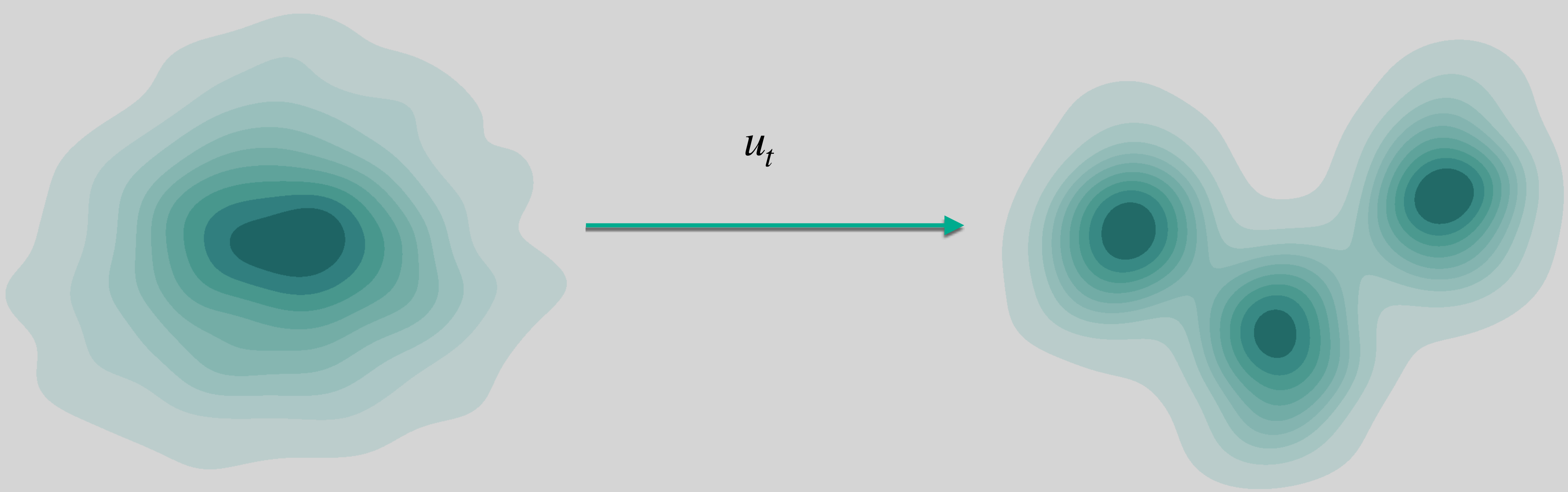
π_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Marginal Velocity Field



Source Distribution

π_0

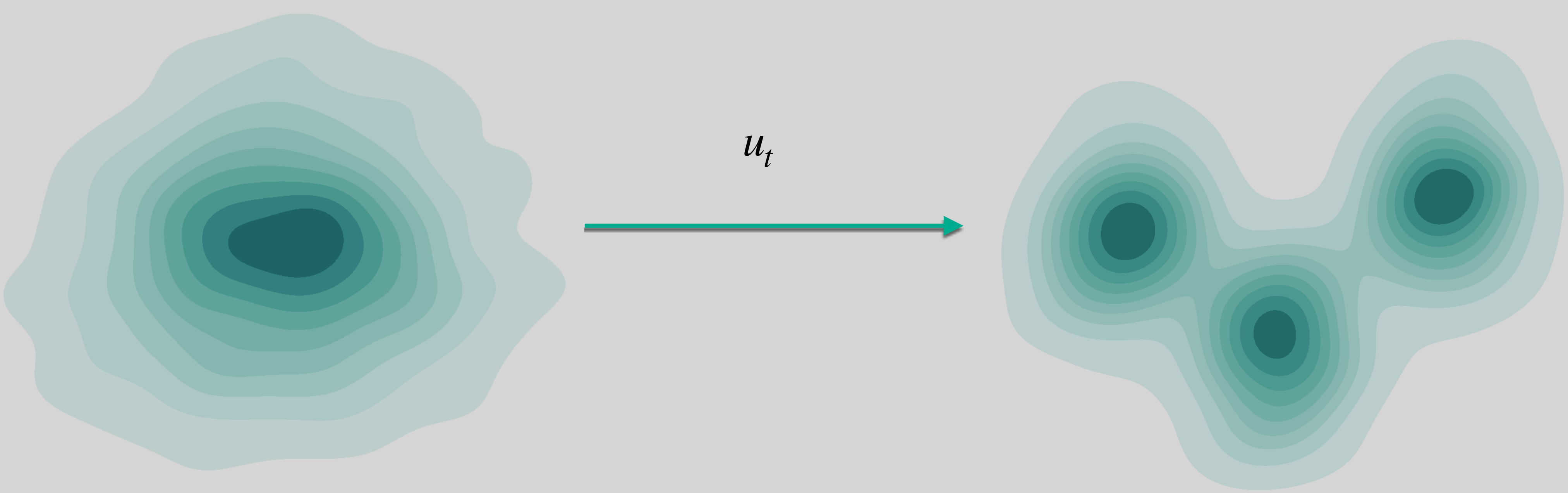
Target Distribution

π_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Marginal Velocity Field : Follows marginal probability path



Source Distribution

π_0

Target Distribution

π_1

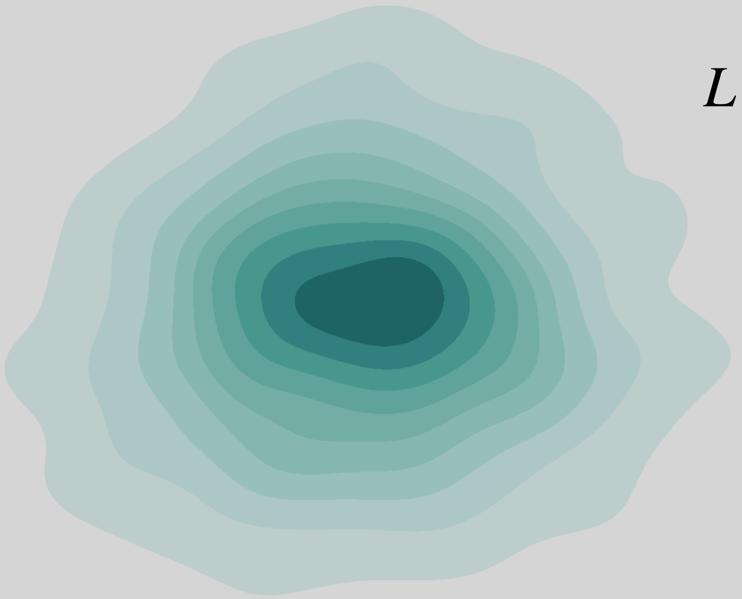
Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

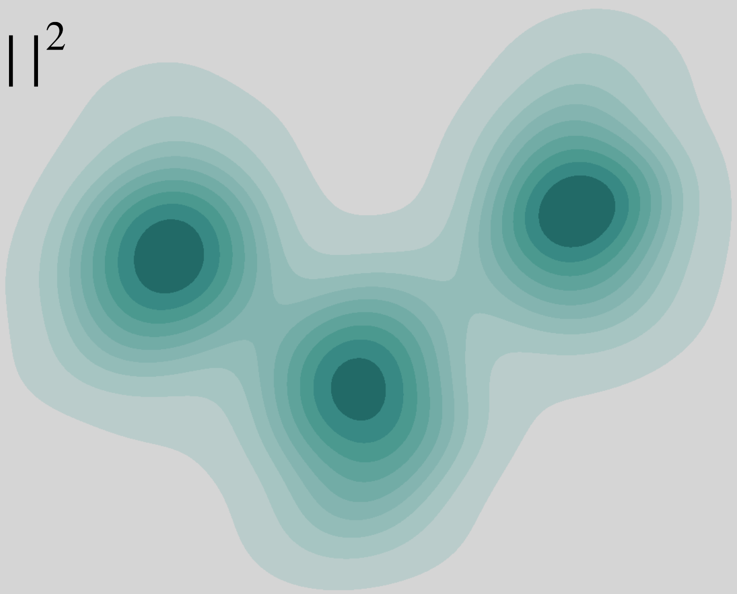
Marginal Velocity Field : Minimize

$$L = || u_t^{\theta}(X_t) - u_t^{target}(X_t) ||^2$$



Source Distribution

π_0



Target Distribution

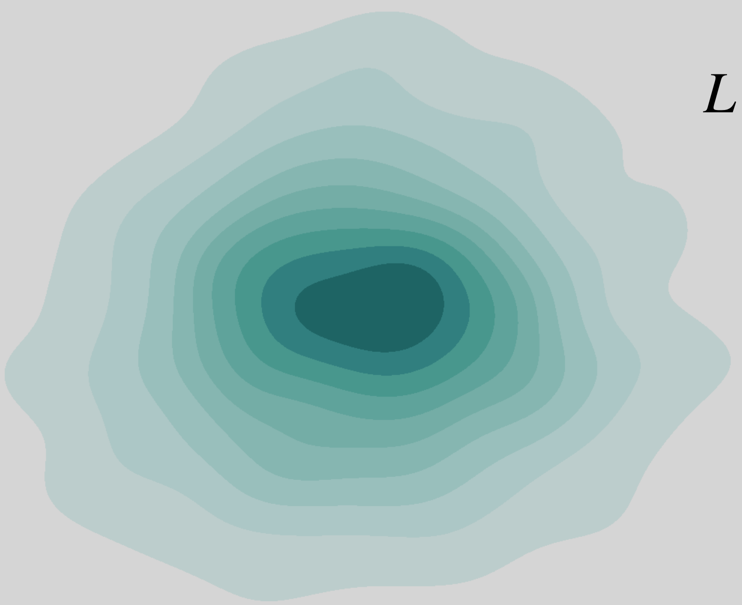
π_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

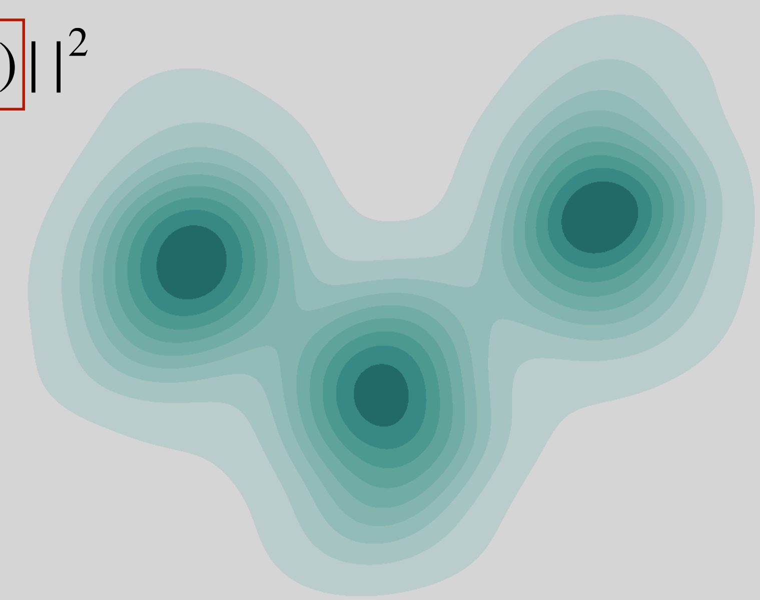
Marginal Velocity Field

$$L = ||u_t^\theta(X_t) - \boxed{u_t^{target}(X_t)}||^2$$



Source Distribution

π_0



Target Distribution

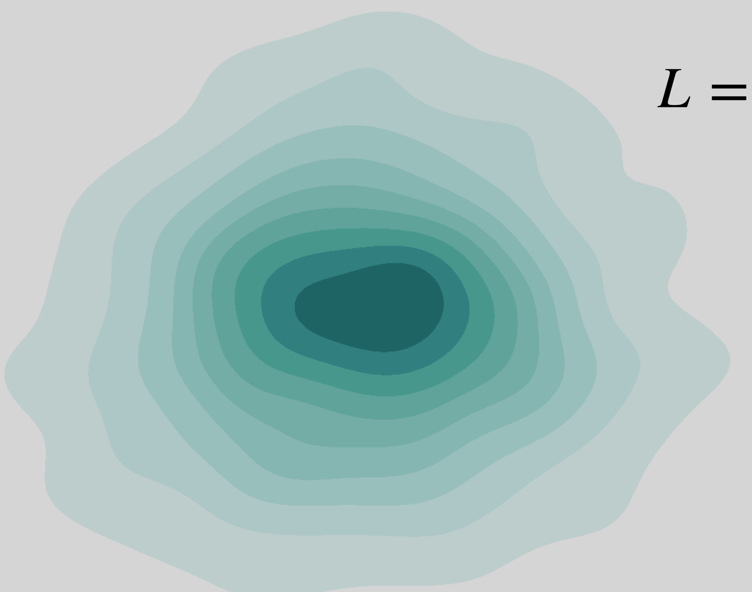
π_1

Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

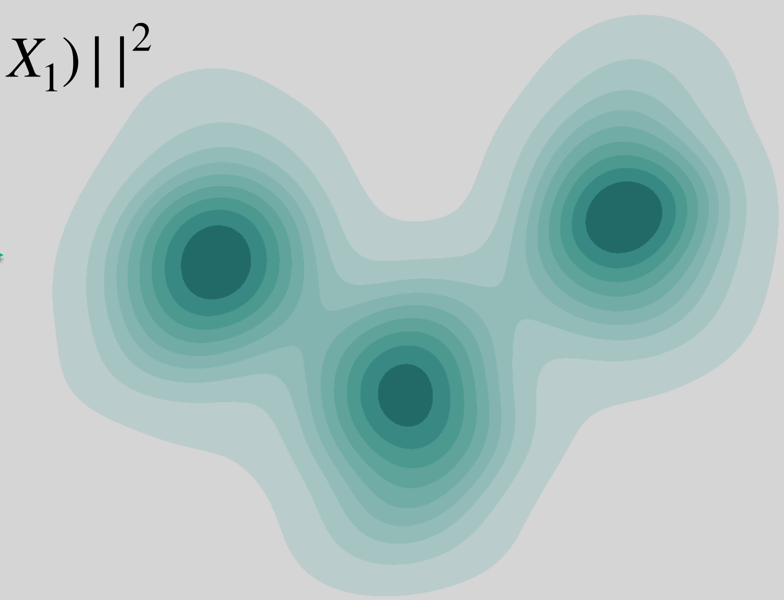
Conditional Velocity Field

$$L = ||u_t^{\theta}(X_t) - u_t^{target}(X_t | X_1)||^2$$



Source Distribution

π_0



Target Distribution

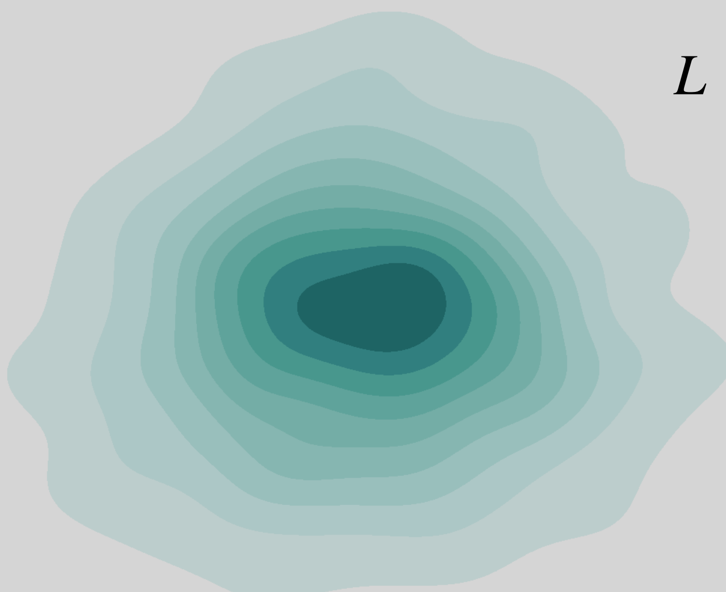
π_1

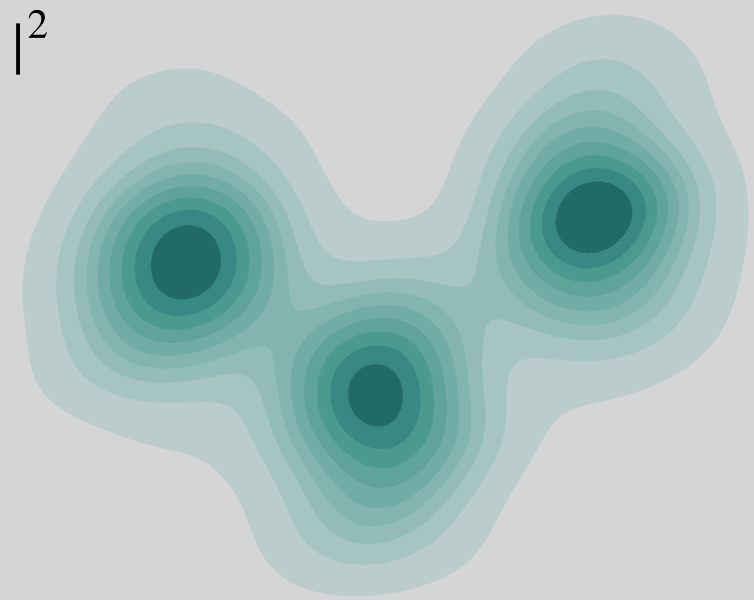
Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Theory of Flow Matching 101…

Conditional Velocity Field for optimal transport path

$$L = ||u_t^{\theta}(X_t) - (X_1 - X_0)||^2$$





Source Distribution
π_0

Target Distribution
π_1

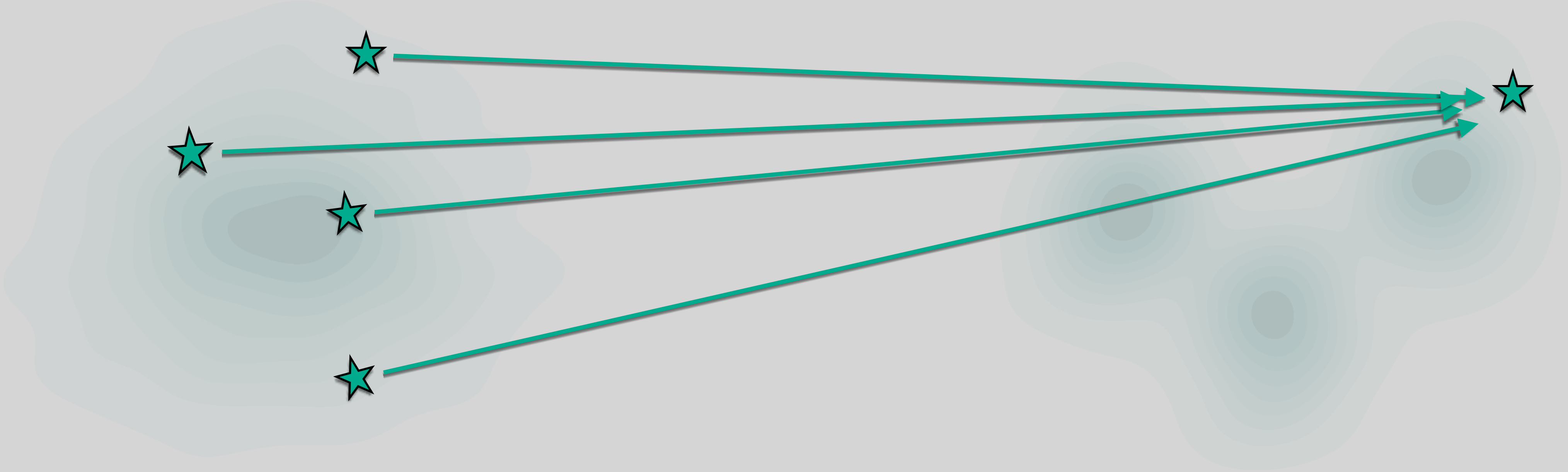
Lipman, Yaron, et al. "Flow Matching for Generative Modeling.." ICLR 2023.
Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 101: ReFlow

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 101: ReFlow

Recap: Random Coupling



Source Samples

X_0

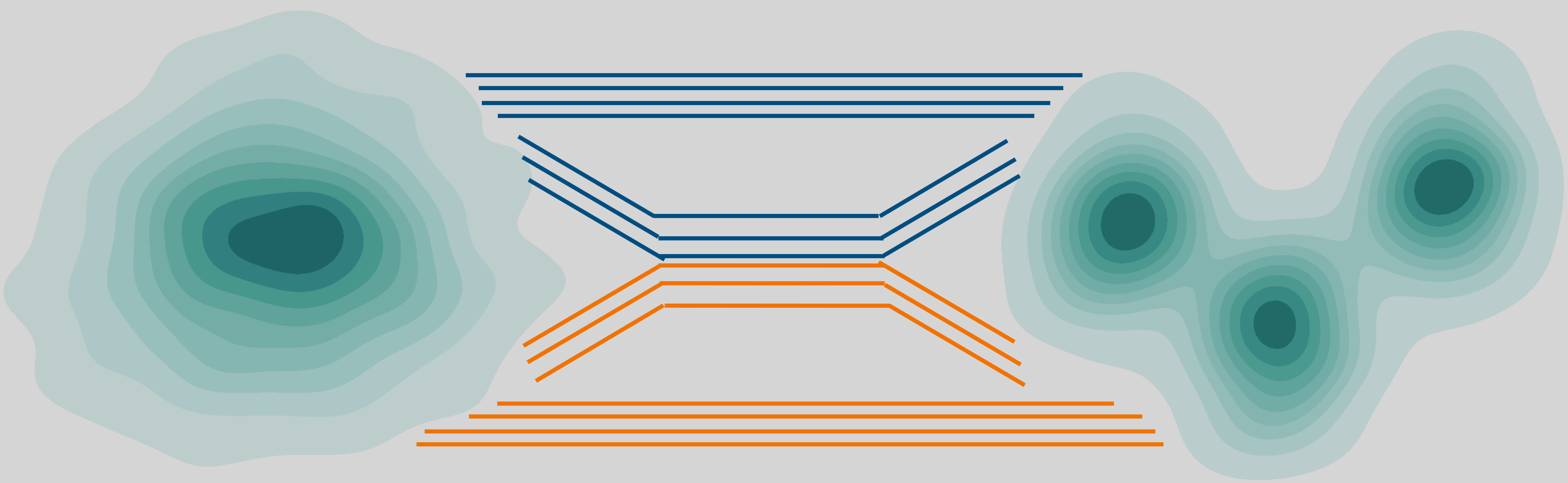
Target samples

X_1

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 101: ReFlow

Learnt Vector Fields



Source Distribution

π_0

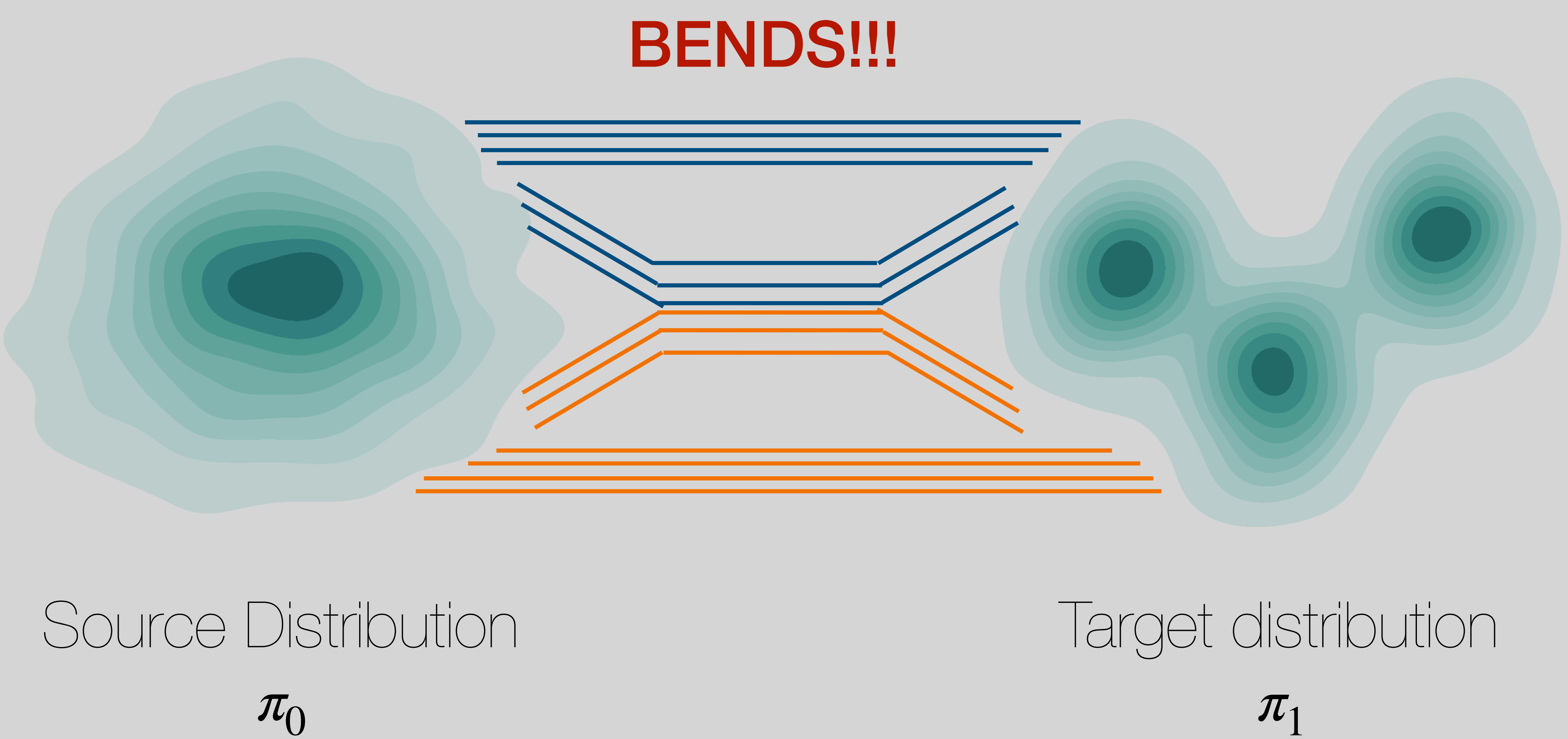
Target distribution

π_1

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 101: ReFlow

Learnt Vector Fields

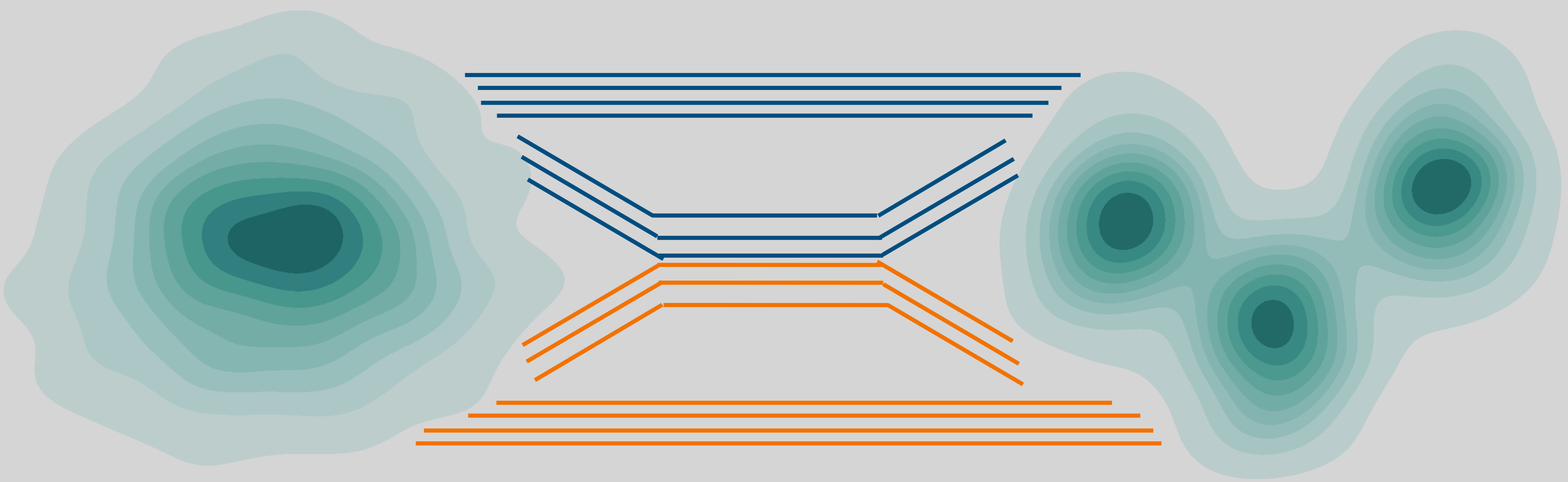


Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 101: ReFlow

Learnt Vector Fields

Requires more Sampling steps



Source Distribution

π_0

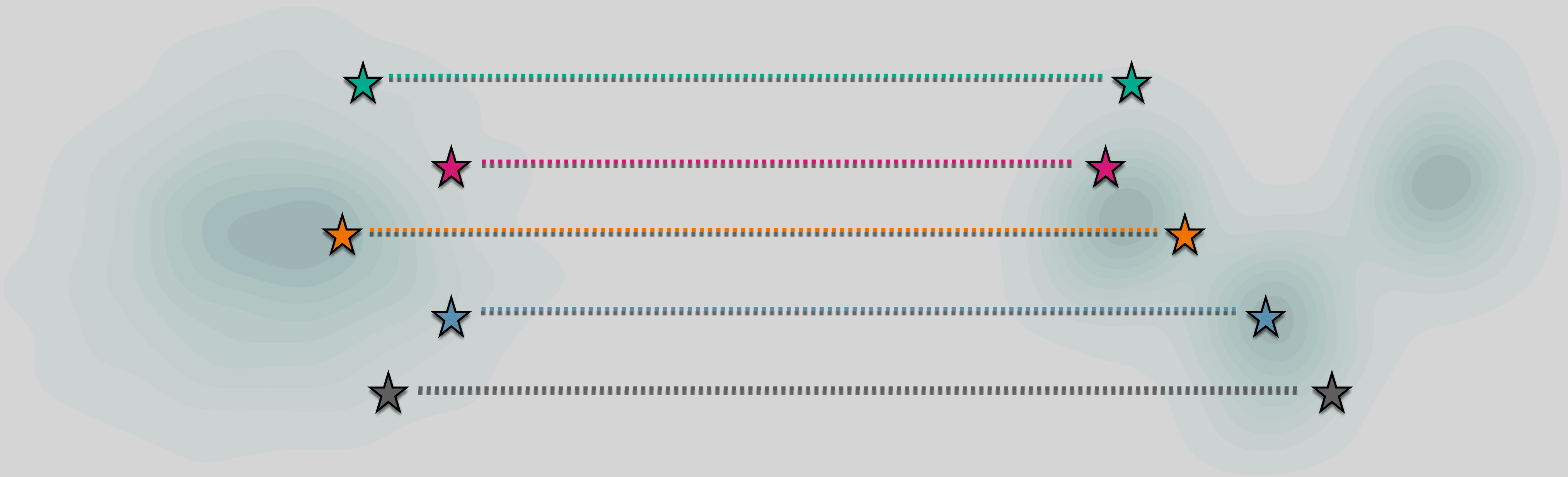
Target distribution

π_1

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 101: ReFlow

Distill with straight pairs



Source Samples

X_0

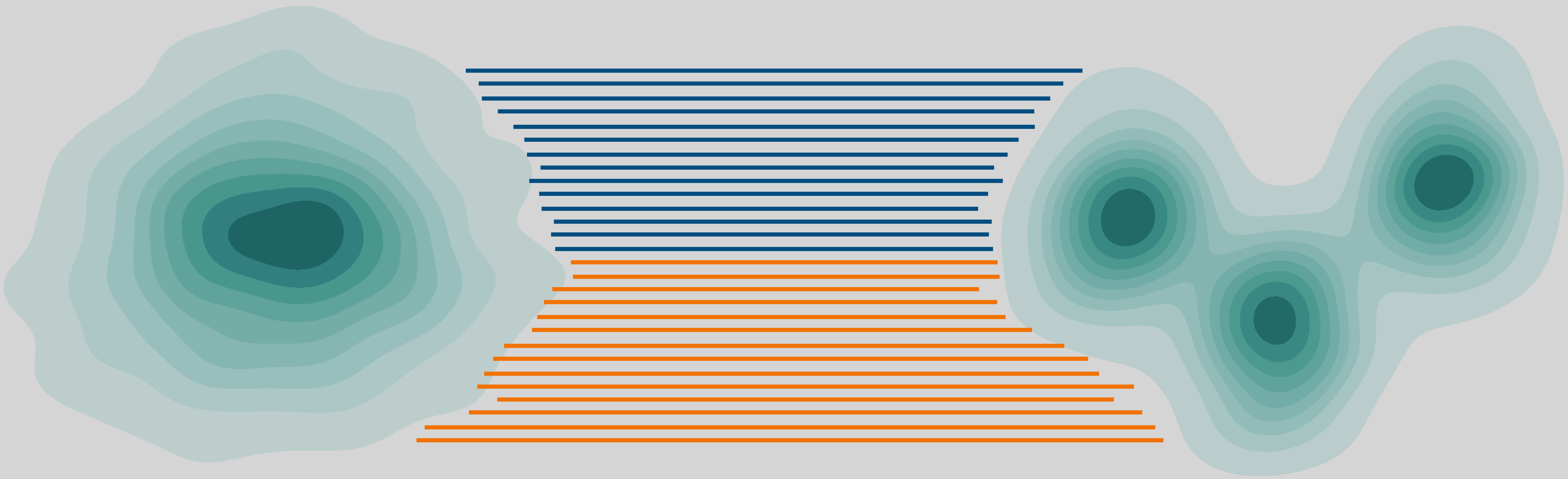
Target samples

X_1

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 101: ReFlow

Learnt Vector Fields



Source Distribution

π_0

Target distribution

π_1

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

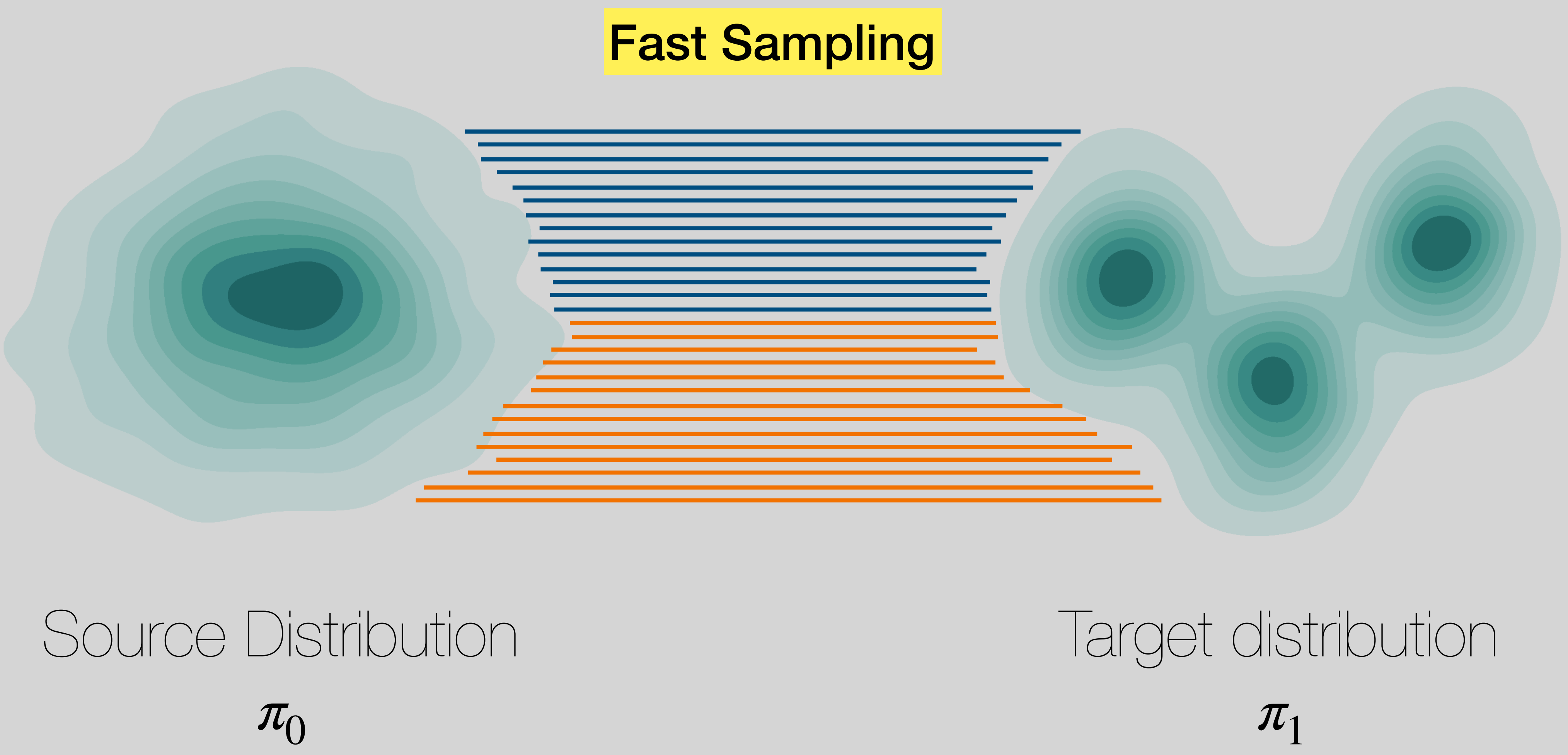
Update 101: ReFlow

Learnt Vector Fields

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 101: ReFlow

Learnt Vector Fields



Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 102: Better ODE Solvers

Recap: Sampling

Euler's Method

$$x_{next} = x_{old} + \frac{1}{num\_steps} \cdot u_\theta(t)$$

$x_{initial} \sim \mathcal{N}(0, I)$

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 102: Better ODE Solvers

Why not use better ones?

Heun Method

RK Method

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 103: Non-uniform timestep sampling

Recap: Randomly sample t in [0,1)

Source Samples

X_0

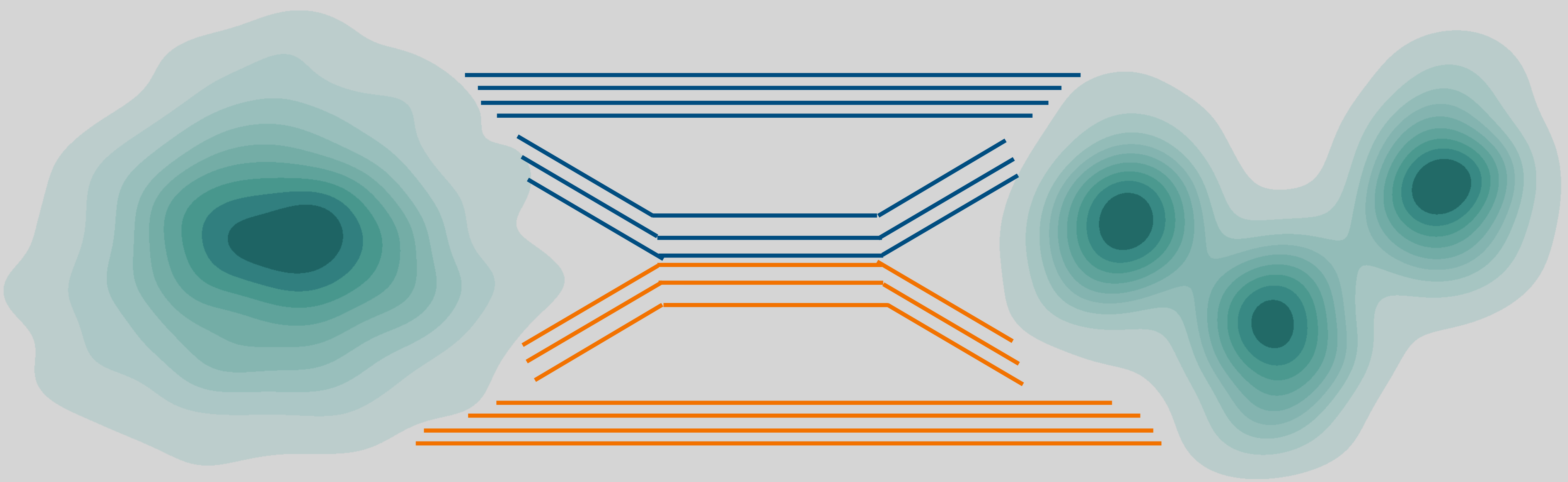
Target samples

X_1

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 103: Non-uniform timestep sampling

Recap: Learnt Vector Fields



Source Distribution

π_0

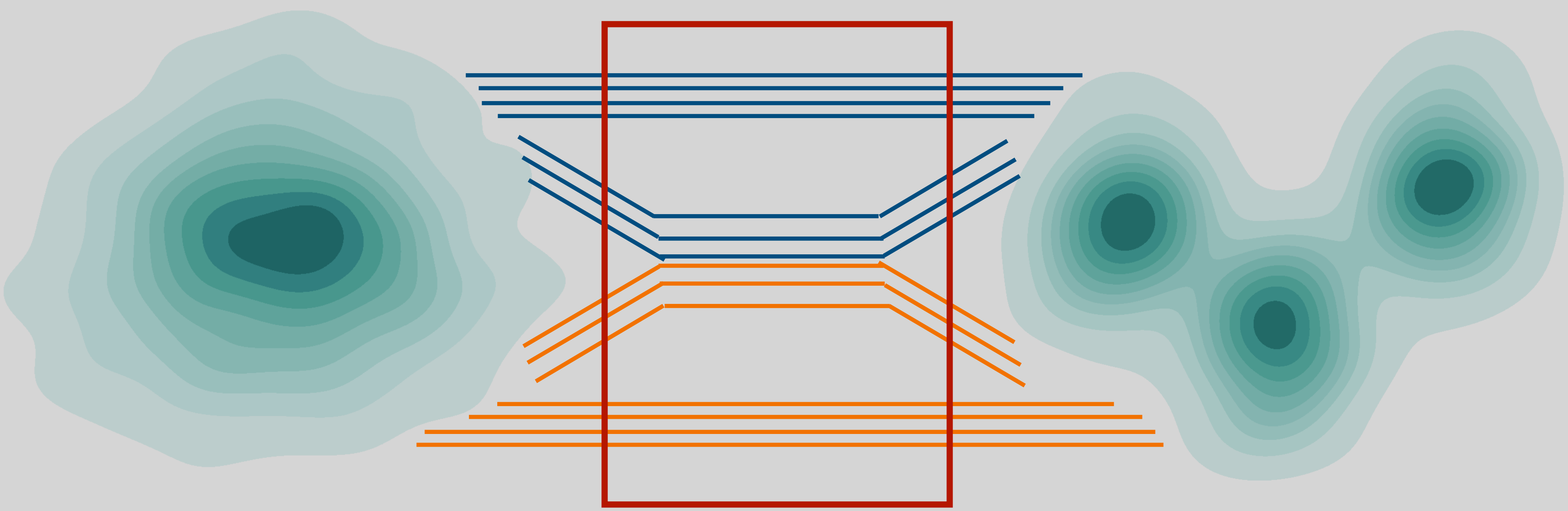
Target distribution

π_1

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 103: Non-uniform timestep sampling

Change in single path mostly in the middle



Source Distribution

π_0

Target distribution

π_1

Liu, Xingchao, et al. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow." ICLR 2023.

Update 103: Non-uniform timestep sampling

Logit-Normal Sampling

$$t = \frac{1}{1 + \exp(u)}$$

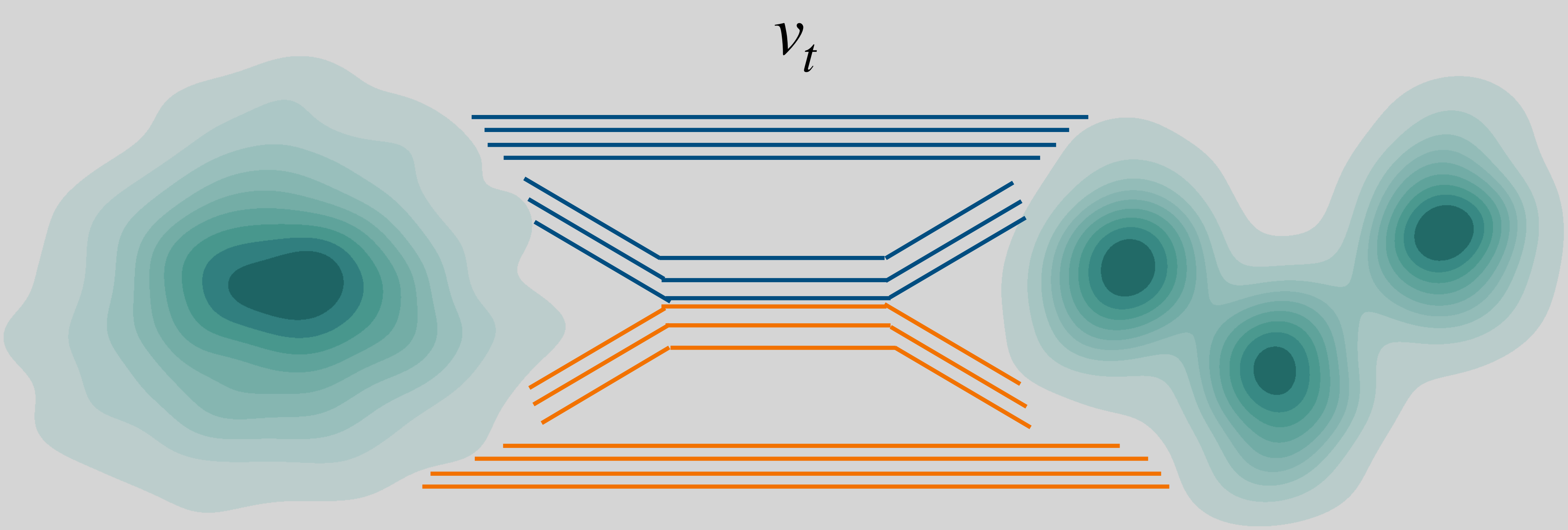
$$u \sim \mathcal{N}(0,1)$$

Esser, Patrick, et al. "Scaling Rectified Flow Transformers for High-Resolution Image Synthesis." ICML 2024.

Connection to Diffusion Models

Connection to Diffusion Models

Flow Matching: Learn a vector field



Source Distribution

π_0

Target distribution

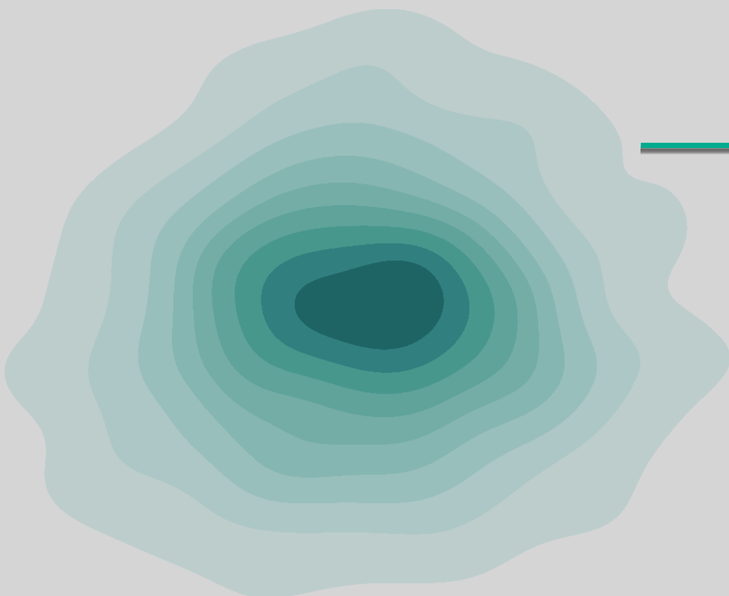
π_1

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.
Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

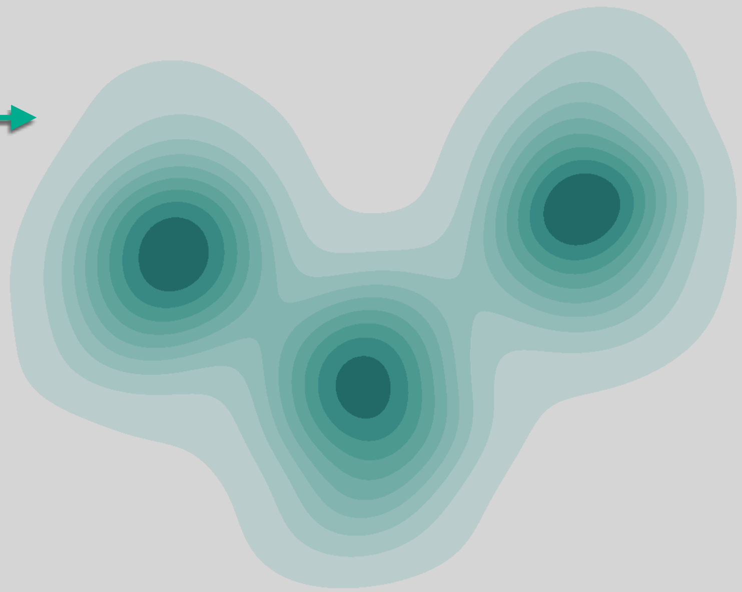
Flow Matching Sampling : Solving an ODE

$$dx = v_t \cdot dt$$



Source Distribution

π_0



Target distribution

π_1

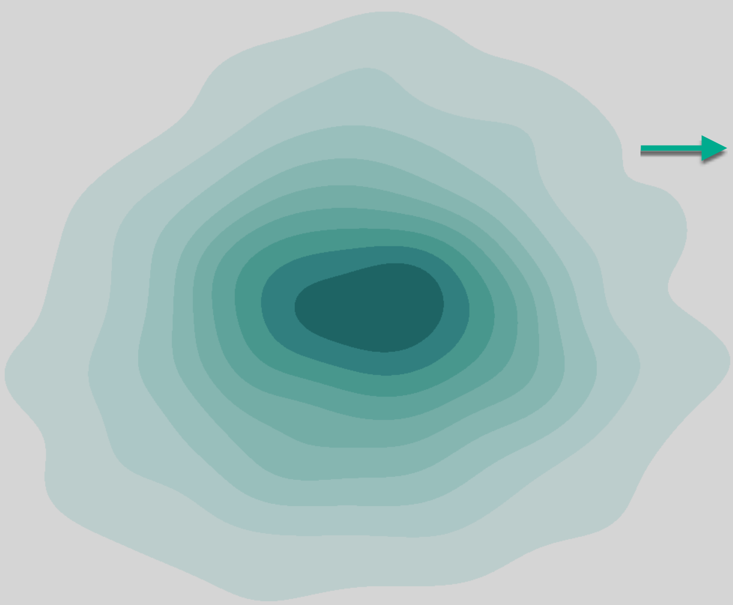
Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.
Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Flow Matching Sampling : Solving an ODE

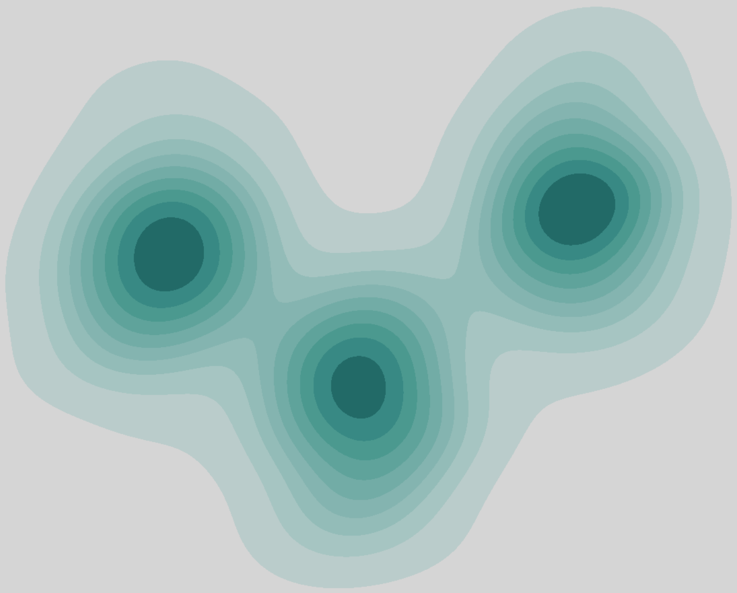
$$dx = v_t \cdot dt$$

$$x_{t+h} = x_t + h \cdot v_t$$



Source Distribution

π_0



Target distribution

π_1

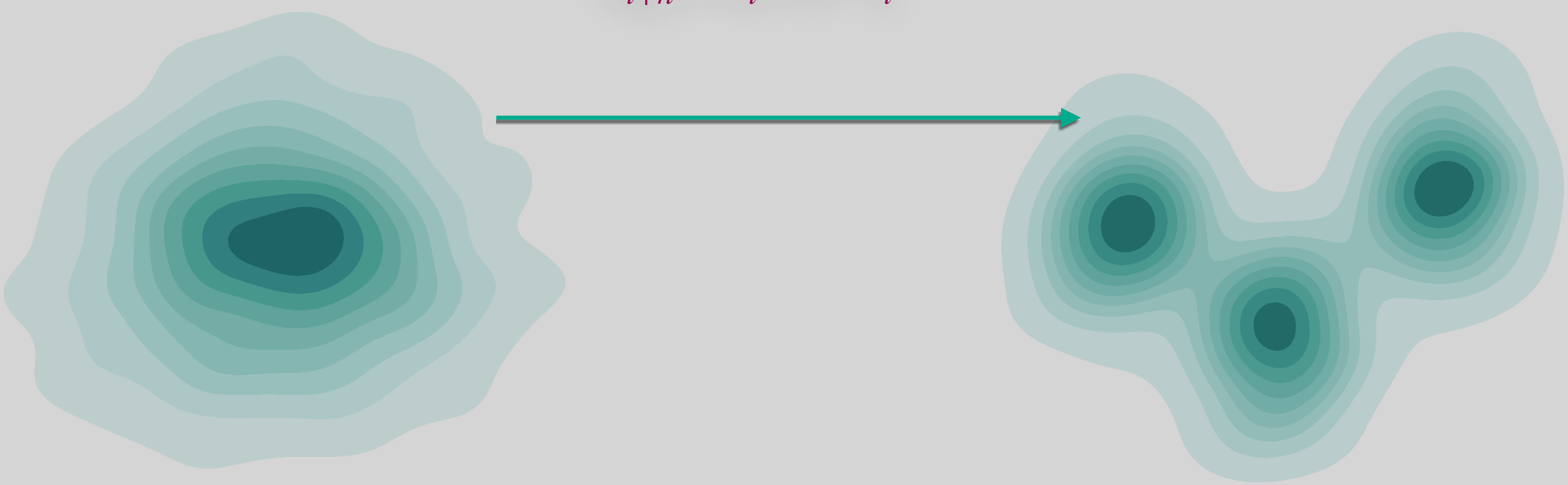
Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.
Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Flow Matching Sampling : Solving an ODE

$$dx = v_t \cdot dt$$

$$x_{t+h} = x_t + h \cdot v_t$$



Source Distribution

π_0

Target distribution

π_1

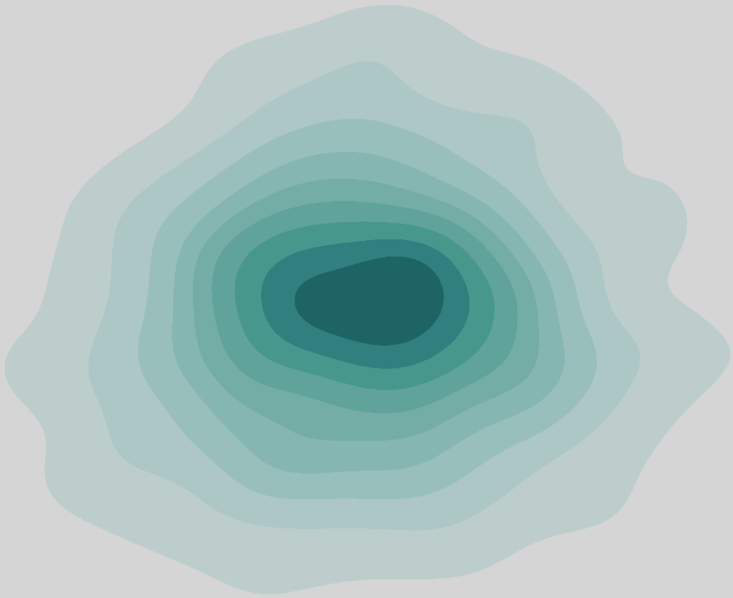
Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

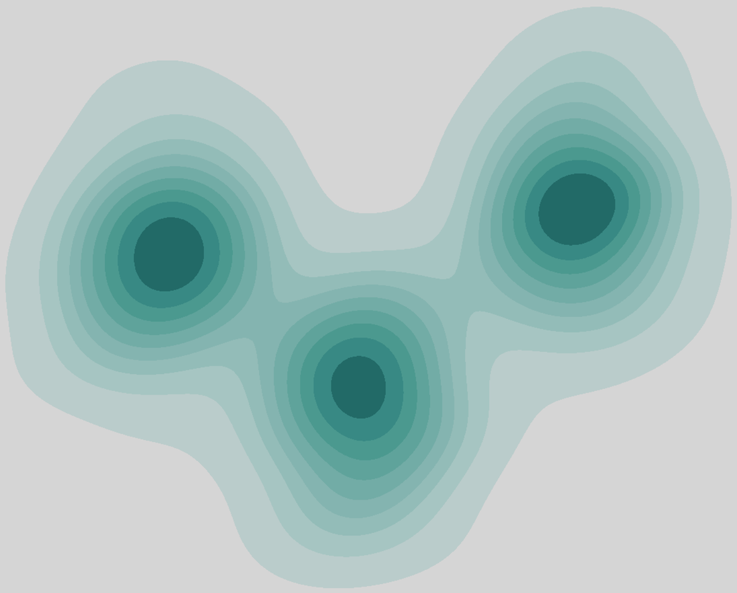
SDE and Diffusion

$$dx = [v_t + \lambda(w_t) \cdot s_t] \cdot dt + w_t \cdot dW$$



Source Distribution

π_0



Target distribution

π_1

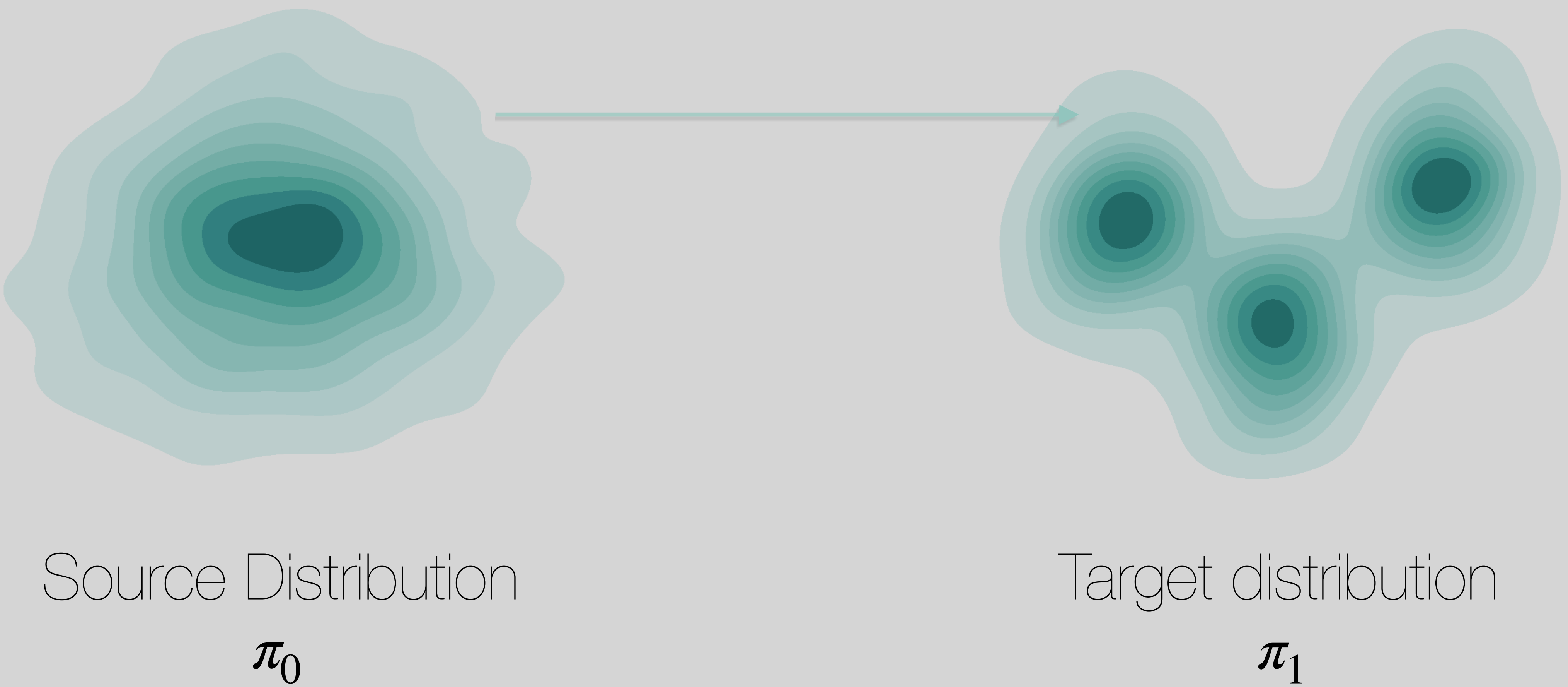
Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

SDE and Diffusion

$$dx = [v_t + \lambda(w_t) \cdot s_t] \cdot dt + w_t \cdot dW$$



Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

SDE and Diffusion

$$dx = [v_t + \lambda(w_t) \cdot s_t] \cdot dt + w_t \cdot dW$$

Noise

Source Distribution

π_0

Target distribution

π_1

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

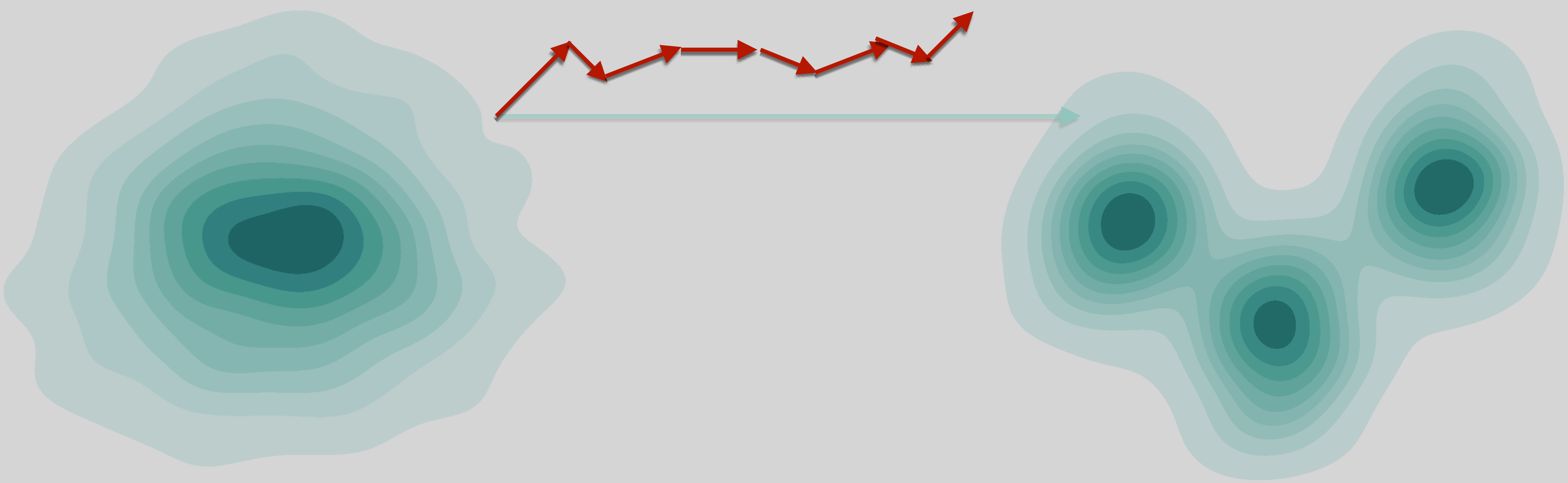
Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

SDE and Diffusion

$$dx = [v_t + \lambda(w_t) \cdot s_t] \cdot dt + w_t \cdot dW$$

Noise



Source Distribution

π_0

Target distribution

π_1

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.
Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

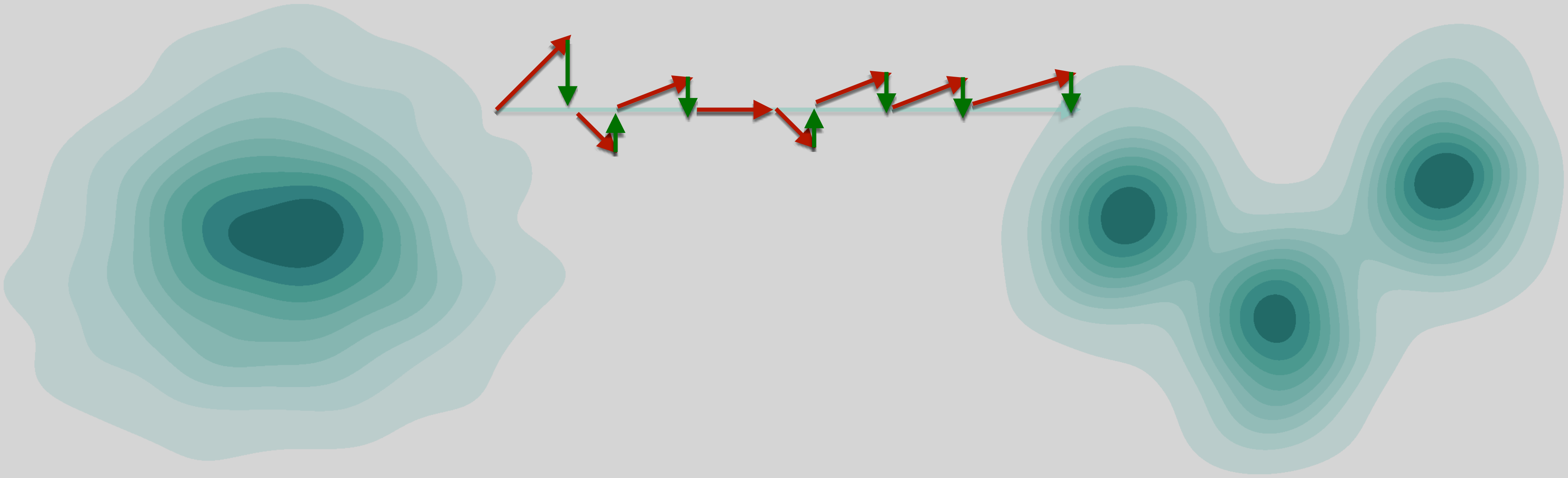
Connection to Diffusion Models

SDE and Diffusion

Correction

$$dx = [v_t + \lambda(w_t) \cdot s_t] \cdot dt + w_t \cdot dW$$

Noise



Source Distribution

π_0

Target distribution

π_1

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Diffusion Models

Predict Score

$$dx = [v_t + \lambda(w_t) \cdot s_t] \cdot dt + w_t \cdot dW$$

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Diffusion Models

Predict Score

$$dx = [v_t + \lambda(w_t) \cdot s_t] \cdot dt + w_t \cdot dW$$

Fun Fact: Using **Tweedie's formula**

$$v_t \leftrightarrow s_t$$

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Diffusion Models

$$x_{t+h} = x_t +$$

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.
Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Diffusion Models

$$x_{t+h} = x_t + h \cdot (s_t$$

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Diffusion Models

$$x_{t+h} = x_t + h \cdot (s_t + \frac{1}{\lambda(w_t)} \cdot v_t)$$

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Diffusion Models

DDIM Sampler

$$x_{t+h} = x_t + h \cdot (s_t + \frac{1}{\lambda(w_t)} \cdot v_t)$$

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Diffusion Models

$$x_{t+h} = x_t + h \cdot (s_t + \frac{1}{\lambda(w_t)} \cdot v_t) + w_t \cdot \epsilon$$

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.
Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Connection to Diffusion Models

Diffusion Models

DDPM Sampler

$$x_{t+h} = x_t + h \cdot (s_t + \frac{1}{\lambda(w_t)} \cdot v_t) + w_t \cdot \epsilon$$

Ma, Nanye, et al. "SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers." ECCV 2024.

Hu, Xixi, et al. "Flow to Diffusion: Langevin is a Guardrail." https://rectifiedflow.github.io/blog/2024/diffusion/

Thank you!

Source: Jack Black and Elmo