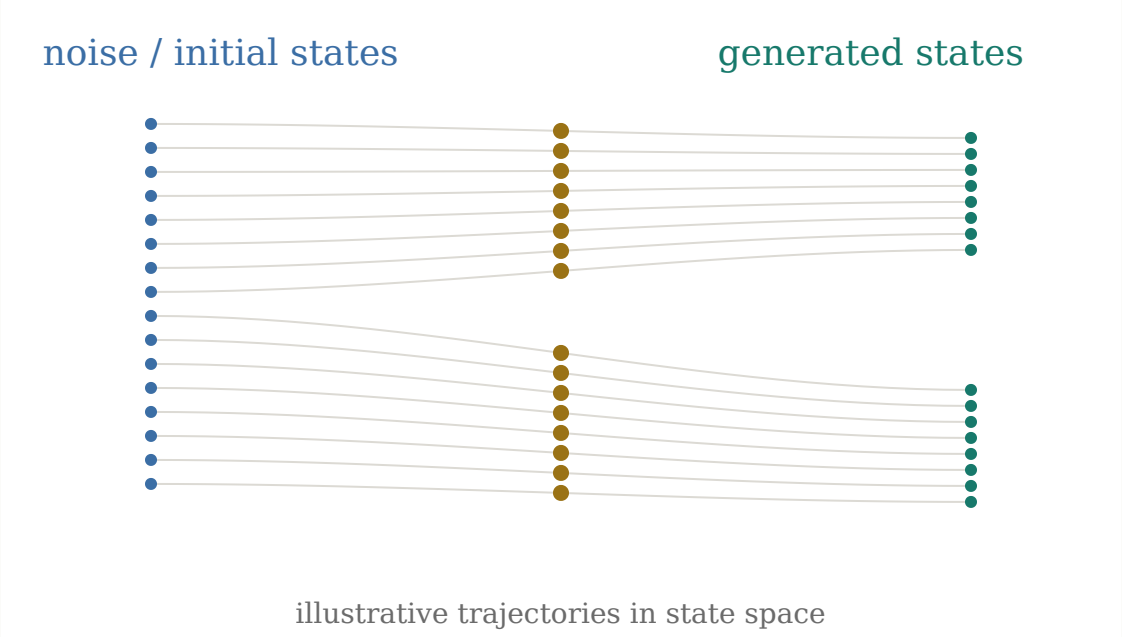
LECTURE 12 · FLOW MATCHING · DECEMBER 2

Flow Matching and VLA

From random vectors to robot actions

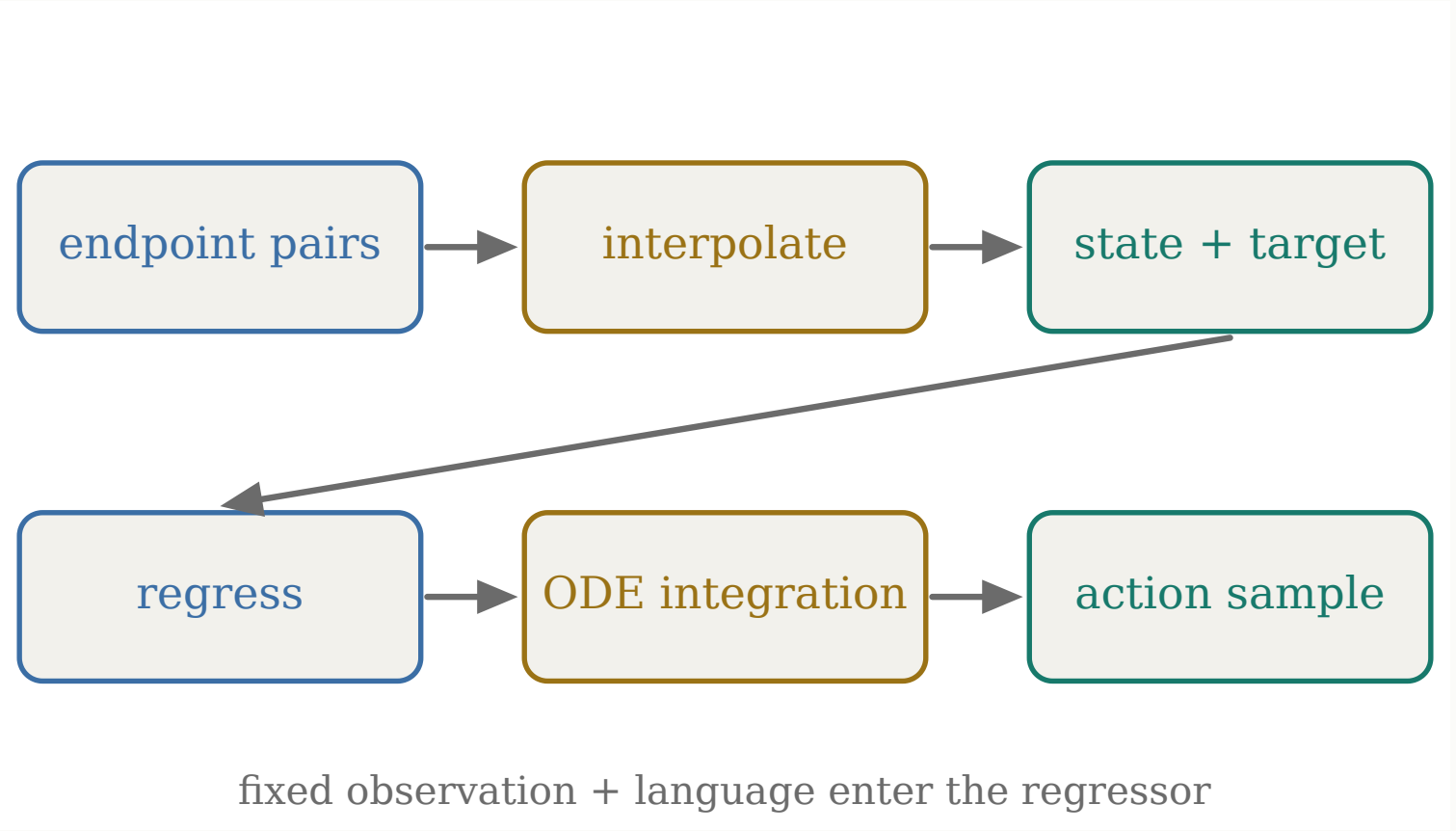


How can a learned rate turn noise into a possible action sequence?

CONCEPT MAP

Flow Matching and VLA

The mathematical route



fixed observation + language enter the regressor

Lipman et al. (2023), §§2–4 · original course examples

RECALL

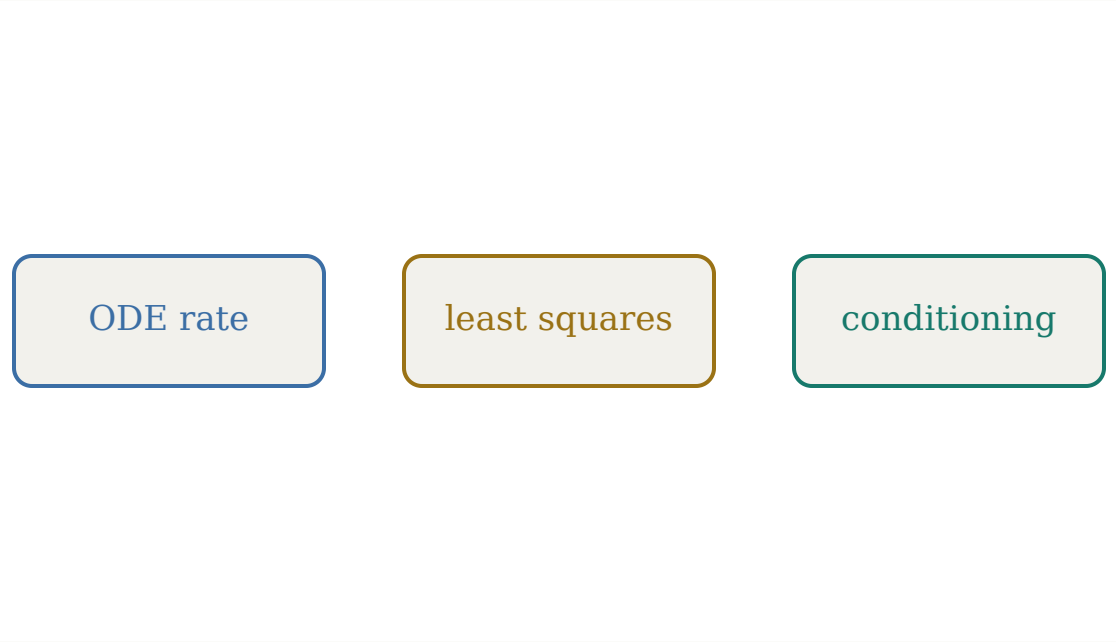
Flow Matching and VLA

Three results return

ODE: a rate determines a trajectory.

Squared error: conditional means minimize risk.

Conditioning: fix the observation when sampling.



Goodfellow et al. (2016), §§3.8–3.9.3 · original course examples

Recall: Lecture 7, p22: Euler · Lecture 5, p31: least-squares gradient · Lecture 8, pp35–36: conditioning

YOUR TURN

Flow Matching and VLA

Which clock is running?

k: physical robot step; $\tau \in [0,1]$: generator time. One action chunk is refined while the camera observation stays fixed.

A. Every flow step moves the robot.

B. Flow steps refine a candidate chunk.

C. Flow time is the camera frame index.

Black et al., π_0 (2024/2026), §IV; App. A-D · original course examples

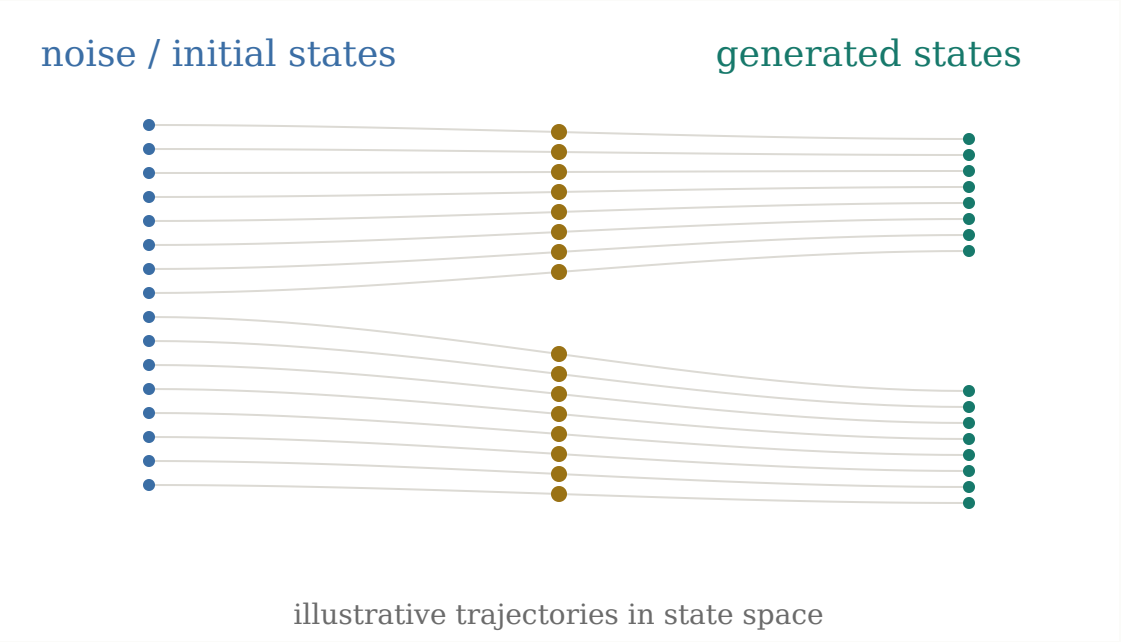
PROBLEM

Probability Flows

The target is a distribution

$$X_0 \sim p_0, \qquad X_1 \sim p_{\text{data}}$$

where $X_0, X_1 \in \mathbb{R}^d$: random vectors; p_0: easy noise law; p_{data}: data law.



illustrative trajectories in state space

Lipman et al. (2023), §§2–4 · original course examples

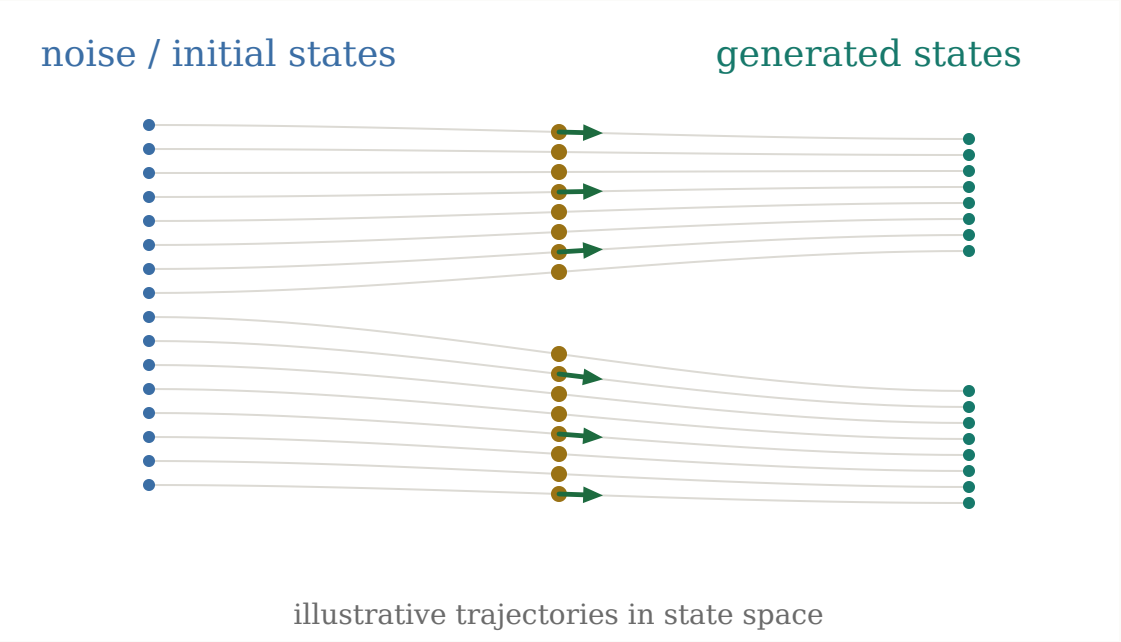
DEFINITION

Probability Flows

A learned velocity field

$$\frac{dx_\tau}{d\tau} = v_\theta(x_\tau, \tau), \quad x_0 \sim p_0$$

where $x_\tau, v_\theta \in \mathbb{R}^d$; θ: fixed learned parameters during sampling.



illustrative trajectories in state space

Lipman et al. (2023), §§2–4 · original course examples

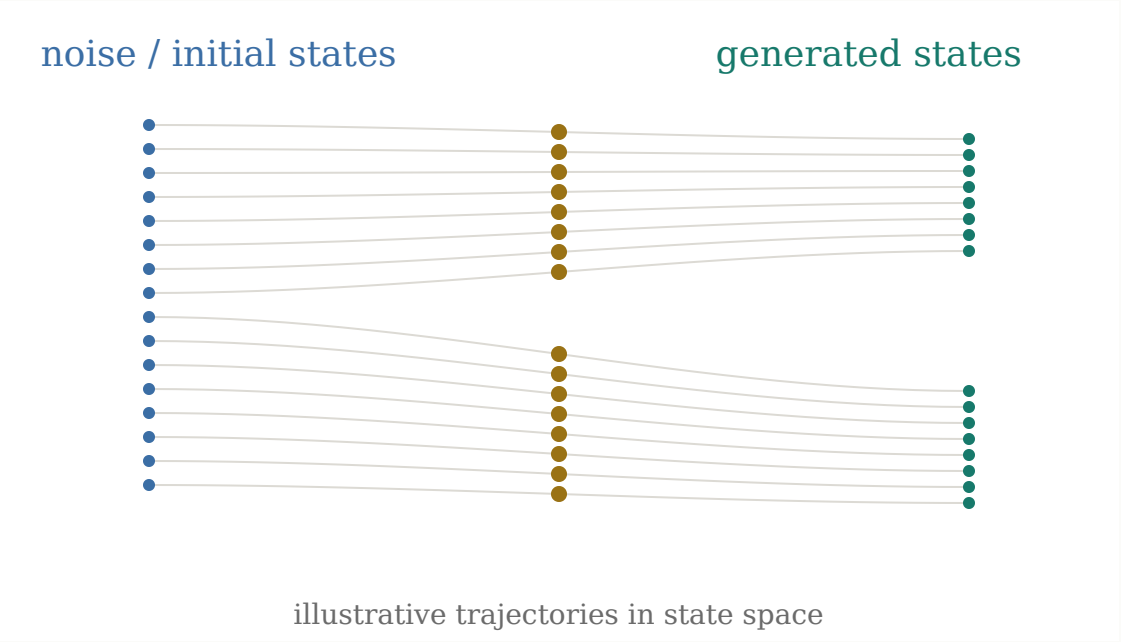
DEFINITION

Probability Flows

The flow map collects trajectories

$$\phi_\tau(x_0) = x_\tau, \qquad \phi_0(x_0) = x_0$$

where $\phi_\tau : \mathbb{R}^d \to \mathbb{R}^d$: map from an initial point to its state at time τ.



illustrative trajectories in state space

Lipman et al. (2023), §§2–4 · original course examples

WORKED EXAMPLE

Probability Flows

Worked example: translation

$$v(x, \tau) = 2, \quad \phi_\tau(x_0) = x_0 + 2\tau$$

$$x_\tau - x_0 = \int_0^\tau 2 \, ds = 2\tau$$

$$\dot{x}_\tau = 2, \quad \phi_0(x_0) = x_0; \quad X_\tau \sim \mathcal{N}(2\tau, 1)$$

where $x_0, x_\tau \in \mathbb{R}$: positions; velocity 2 is constant per unit generator time.

Lipman et al. (2023), §§2–4 · original course examples

YOUR TURN

Probability Flows

Transfer: a different velocity

$X_0 \sim \mathcal{N}(1, 4)$ and $dx_\tau/d\tau = -3$. Find the law at $\tau = 1$.

A Gaussian mean -2, variance 4

B Gaussian mean -2, variance 16

C Gaussian mean 4, variance 4

Goodfellow et al. (2016), §§3.8–3.9.3 · original course examples

DEFINITION

Probability Flows

Expansion changes density

$$x_\tau = e^{a\tau} x_0, \qquad p_\tau(x) = e^{-a\tau} p_0(e^{-a\tau} x)$$

$$\dot{x} = ax \;\Rightarrow\; x = e^{a\tau} x_0, \quad x_0 = e^{-a\tau} x$$

$$p_\tau(x)\, dx = p_0(e^{-a\tau} x) e^{-a\tau}\, dx$$

where $a \in \mathbb{R}$: fixed rate; p_τ: scalar density; x: evaluation point.

Lipman et al. (2023), §§2–4 · original course examples

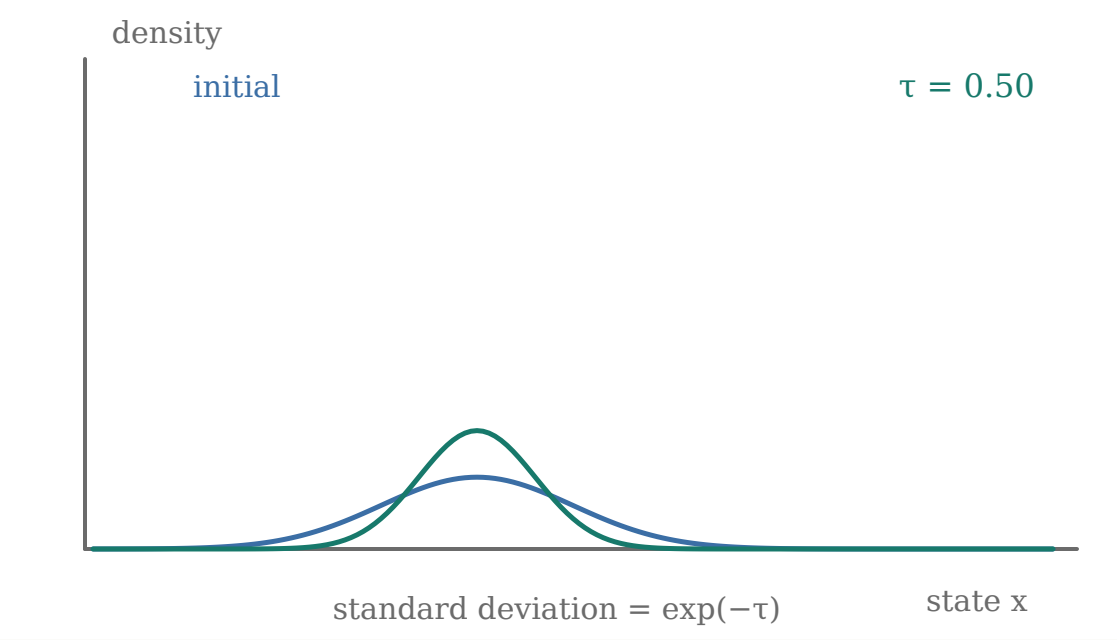
WORKED EXAMPLE

Probability Flows

Compression preserves total probability

$$X_\tau = e^{-\tau} X_0, \quad X_0 \sim \mathcal{N}(0, 1)$$

where X_τ: scalar random state; $\mathrm{Var}(X_\tau) = e^{-2\tau}$.



Lipman et al. (2023), §§2–4 · original course examples

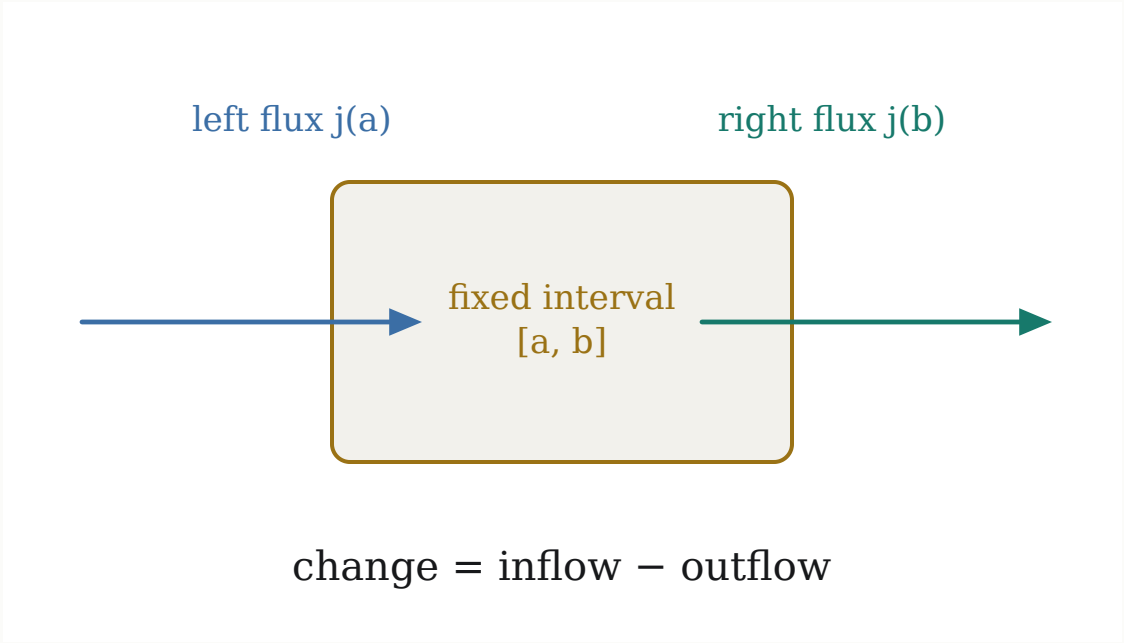
DERIVATION

Probability Flows

Probability enters and leaves an interval

$$\frac{d}{d\tau}\int_a^b p_\tau(x)\,dx = j_\tau(a) - j_\tau(b)$$

where $[a,b]$: fixed interval; $j_\tau(x) = p_\tau(x)v(x,\tau)$: probability flux.



Lipman et al. (2023), Appendix B · original course examples

DERIVATION

Probability Flows

Local probability conservation

$$\partial_\tau p_\tau(x) = -\partial_x[p_\tau(x)v(x,\tau)]$$

$$j(a) - j(b) = -\int_a^b \partial_x j(x)\,dx$$

$$\int_a^b [\partial_\tau p + \partial_x(pv)]\,dx = 0$$

$$\partial_\tau p = -\partial_x(pv), \qquad \partial_\tau p + \nabla\cdot(pv) = 0$$

where ∂_τ: time derivative at fixed x; ∂_x: spatial derivative.

Lipman et al. (2023), Appendix B · original course examples

YOUR TURN

Probability Flows

Does constant velocity freeze density?

$p_\tau(x) = p_0(x - 2\tau)$, $v = 2$. At a fixed x, does density generally stay constant?

A Yes, because velocity is constant.

B No, because the density profile moves.

C No, because total probability grows.

Lipman et al. (2023), Appendix B · original course examples

WORKED EXAMPLE

Probability Flows

Integrated example: a moving Gaussian

$$X_\tau = 2\tau + (1+\tau)X_0, \quad X_0 \sim \mathcal{N}(0,1)$$

$$\dot{x}_\tau = 2 + x_0, \qquad x_0 = \frac{x_\tau - 2\tau}{1+\tau}$$

$$v(x,\tau) = 2 + \frac{x - 2\tau}{1+\tau}$$

$$X_1 = 2 + 2X_0 \sim \mathcal{N}(2,4)$$

where $\mu_\tau = 2\tau$; $s_\tau = 1 + \tau > 0$: standard deviation.

Lipman et al. (2023), §§2–4 · original course examples

YOUR TURN

Probability Flows

Transfer: contracting toward a target mean

$X_\tau = 3\tau + (1 - \tau/2)X_0$, $X_0 \sim \mathcal{N}(0, 4)$. Find mean and variance at $\tau = 1$.

A Mean 3, variance 1

B Mean 3, variance 2

C Mean 0, variance 1

Goodfellow et al. (2016), §§3.8–3.9.3 · original course examples

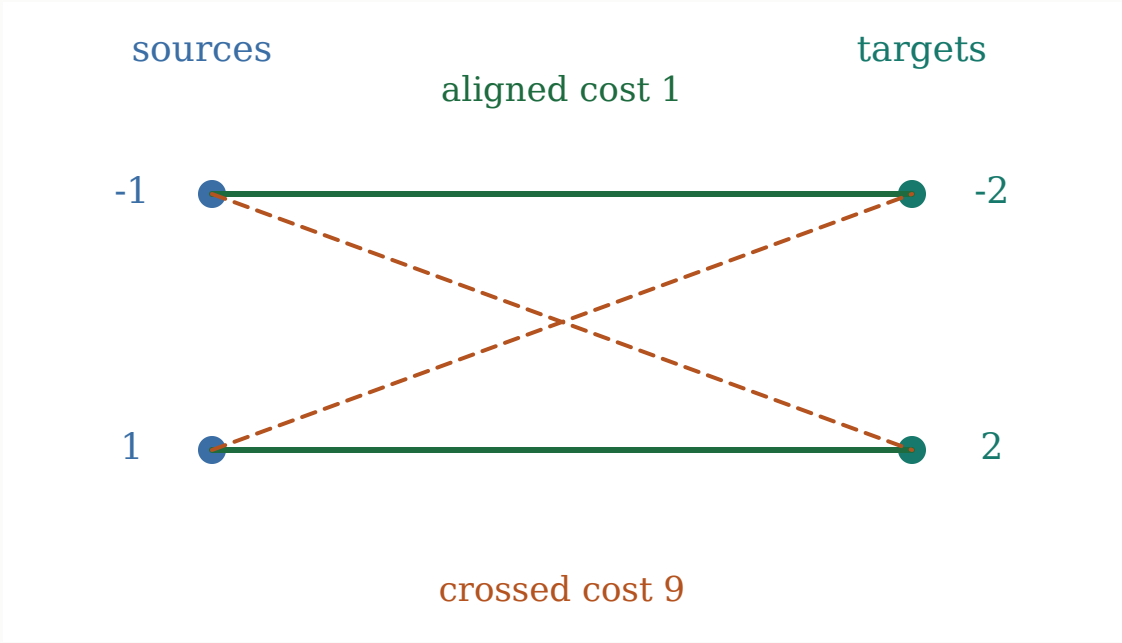
PROBLEM

Conditional Paths

A training pair supplies both endpoints

$$(X_0, X_1) \sim \pi, \qquad \pi_0 = p_0,\ \pi_1 = p_{\text{data}}$$

where π: joint pairing law; π_0, π_1: its two marginals, not model names.



Tong et al. (2024), §§2.2–3.2 · original course examples

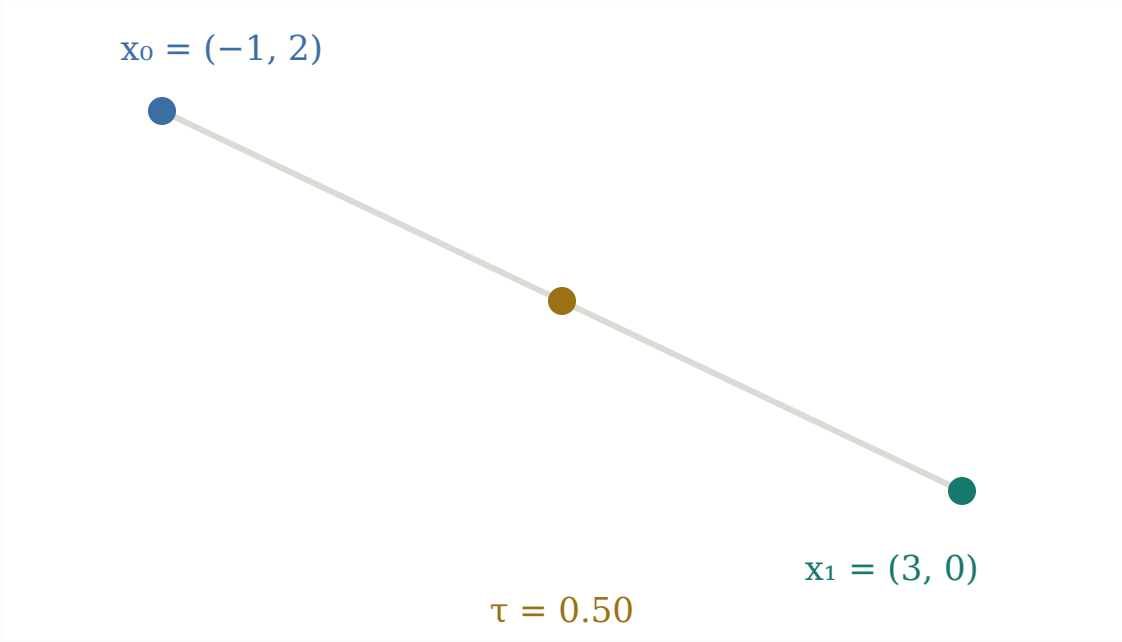
DEFINITION

Conditional Paths

Straight interpolation defines a path

$$X_\tau = (1-\tau)X_0 + \tau X_1, \quad 0 \le \tau \le 1$$

where $X_0, X_1 \in \mathbb{R}^d$: paired endpoints; τ: scalar interpolation weight.



Tong et al. (2024), §§2.2–3.2 · original course examples

DERIVATION

Conditional Paths

The target velocity is an endpoint difference

$$U = \frac{dX_\tau}{d\tau} = -X_0 + X_1$$

$$\frac{d}{d\tau}[(1-\tau)x_0] = -x_0, \qquad \frac{d}{d\tau}[\tau x_1] = x_1$$

$$u = x_1 - x_0, \qquad \Delta x = \Delta\tau\, u$$

where $U \in \mathbb{R}^d$: target rate for a fixed pair; endpoints stay fixed in this derivative.

Tong et al. (2024), §§2.2–3.2 · original course examples

WORKED EXAMPLE

Conditional Paths

Worked example: a 2D pair

$$x_0 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \quad x_1 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad \tau = \tfrac{1}{4}$$

$$u = (3 - (-1),\, 0 - 2)^T = (4, -2)^T$$

$$x_{1/4} = (-1, 2)^T + \tfrac{1}{4}(4, -2)^T = (0, 3/2)^T$$

$$x_{1/4} + \tfrac{3}{4} u = (3, 0)^T = x_1$$

where x_0: noise draw; x_1: data vector; x_τ: training input; u: velocity target.

Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Conditional Paths

Transfer: another pair

$x_0 = (2, -1)^T$, $x_1 = (-2, 3)^T$, $\tau = 1/4$. Choose (x_τ, u).

A $((1,\ 0),\ (-4,\ 4))$

B $((-1,\ 2),\ (-4,\ 4))$

C $((1,\ 0),\ (-1,\ 1))$

Tong et al. (2024), §§2.2–3.2 · original course examples

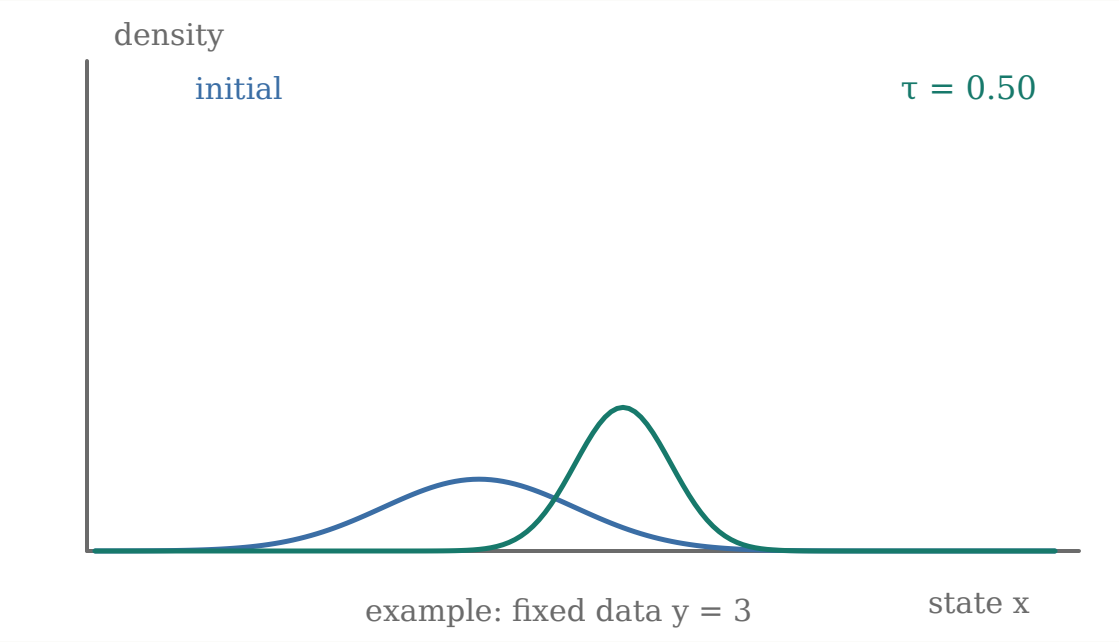
DEFINITION

Conditional Paths

Fixing data leaves a Gaussian noise cloud

$$X_\tau \mid X_1 = y \sim \mathcal{N}(\tau y, (1-\tau)^2 I)$$

where $X_0 \sim \mathcal{N}(0, I)$ independent of X_1; $y \in \mathbb{R}^d$ fixed; $\tau < 1$.



Lipman et al. (2023), §§2–4 · original course examples

YOUR TURN

Conditional Paths

Is the noise coefficient a variance?

$X_0 \sim \mathcal{N}(0,1)$, fixed $y = 4$, $X_\tau = (1-\tau)X_0 + \tau y$. Use $\tau = 1/2$.

A Mean 2, variance 1/2

B Mean 2, variance 1/4

C Mean 4, variance 1/4

Lipman et al. (2023), §§2–4 · original course examples

YOUR TURN

Conditional Paths

Can we use the endpoint during sampling?

Training stores (x_0, x_1, τ) and constructs x_τ. A new generation call has only noise and an observation.

A. Give the network the unknown x_1.

B. Use $x_1 - x_0$ directly at every step.

C. Predict velocity from x_τ, τ and observation.

Tong et al. (2024), §§2.2–3.2 · original course examples

DERIVATION

Conditional Paths

A conditional field can eliminate the noise

$$u(x, \tau \mid y) = \frac{y - x}{1 - \tau}, \quad \tau < 1$$

$$x_0 = \frac{x - \tau y}{1 - \tau}$$

$$y - x_0 = \frac{(1 - \tau)y - x + \tau y}{1 - \tau} = \frac{y - x}{1 - \tau}$$

where $x = (1 - \tau)x_0 + \tau y$; $u = y - x_0$; fixed endpoint y.

Lipman et al. (2023), §§2–4 · original course examples

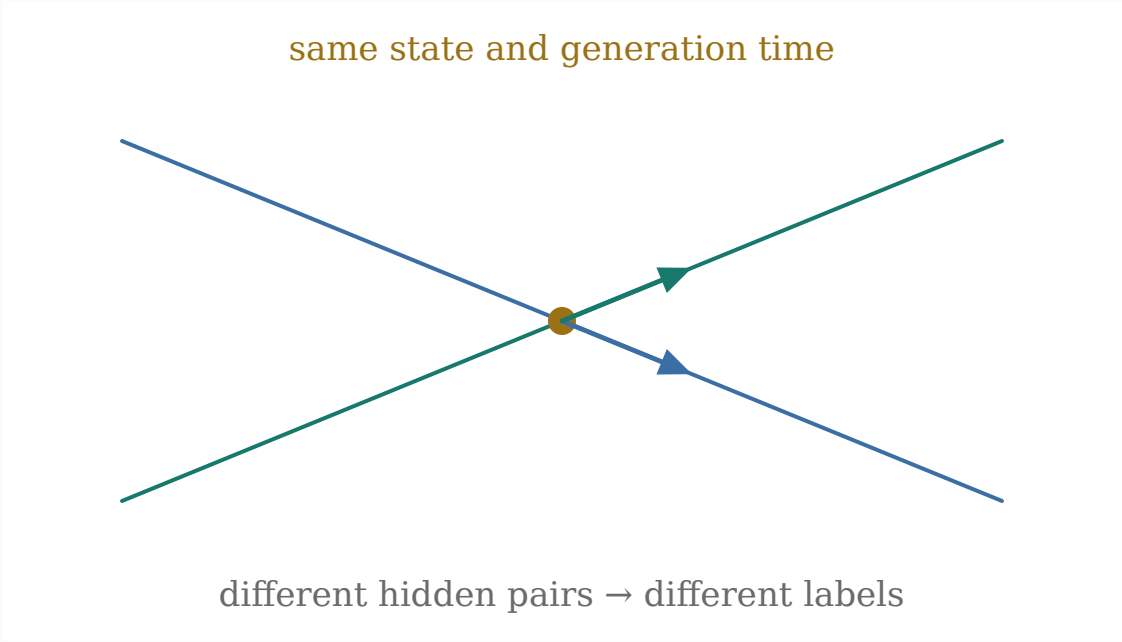
PROBLEM

Conditional Paths

One input can have several target velocities

$$U = X_1 - X_0, \qquad (X_\tau, \tau) = (x, \tau)$$

where U: random target under endpoint pairing; x: shared current location.



Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Conditional Paths

Which average does the field need?

At a fixed (x, τ), velocity targets are $+2$ with probability $3/4$ and -2 with probability $1/4$.

A. The unconditional endpoint average.

B. The target mean given this x and τ.

C. Always the most frequent target exactly.

Tong et al. (2024), §§2.2–3.2 · original course examples

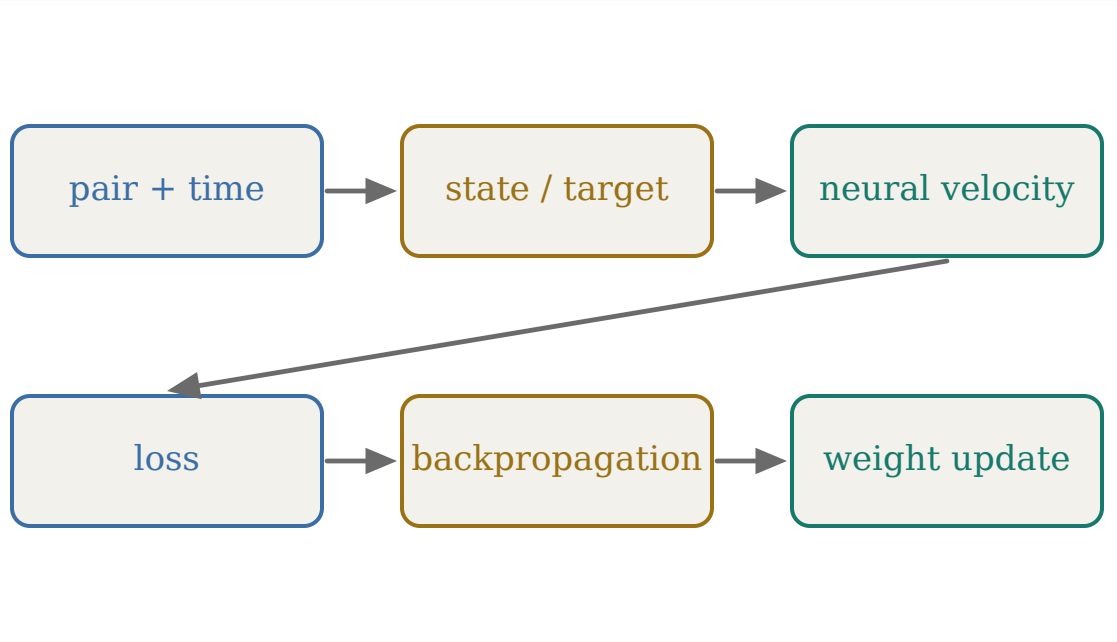
DEFINITION

Learning the Velocity

Squared error compares rates

$$\mathcal{L}(\theta) = \mathbb{E}\|v_\theta(X_\tau, \tau) - U\|^2$$

where $U = X_1 - X_0$; expectation over endpoint pairs and sampled time; θ: trainable weights.



Tong et al. (2024), §§2.2–3.2 · original course examples

DERIVATION

Learning the Velocity

Derivation: the best scalar prediction

$$R(a) = \mathbb{E}[(a - U)^2 \mid x, \tau]$$

$$R(a) = a^2 - 2a\mathbb{E}[U \mid x, \tau] + \mathbb{E}[U^2 \mid x, \tau]$$

$$R'(a) = 2a - 2\mathbb{E}[U \mid x, \tau]$$

$$a^* = \mathbb{E}[U \mid x, \tau], \qquad R''(a) = 2 > 0$$

where $a \in \mathbb{R}$: candidate prediction; conditional target has a finite second moment.

Tong et al. (2024), §§2.2–3.2 · original course examples

RESULT

Learning the Velocity

The vector result is coordinatewise

$$v^*(x,\tau) = \mathbb{E}[U \mid X_\tau = x, \tau]$$

$$\|a - U\|^2 = \sum_{j=1}^{d} (a_j - U_j)^2$$

$$\nabla_a R = 2(a - \mathbb{E}[U \mid x, \tau]) = 0$$

where $v^*, U \in \mathbb{R}^d$; squared norm sums all coordinate risks.

Tong et al. (2024), §§2.2–3.2 · original course examples

WORKED EXAMPLE

Learning the Velocity

Worked example: conflicting targets

$$R(a) = \tfrac{3}{4}(a-2)^2 + \tfrac{1}{4}(a+2)^2$$

$$R(a) = \tfrac{3}{4}(a^2 - 4a + 4) + \tfrac{1}{4}(a^2 + 4a + 4)$$

$$R(a) = a^2 - 2a + 4 = (a-1)^2 + 3$$

$$R(1) = 3 < 4 = R(2)$$

where a: predicted scalar rate; targets $2, -2$ have conditional weights $3/4, 1/4$.

Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Learning the Velocity

Transfer: changed conditional weights

$P(U=-1 \mid x,\tau)=1/4$, $P(U=3 \mid x,\tau)=3/4$. Minimize conditional squared error.

A Predict 1

B Predict 2

C Predict 3

Tong et al. (2024), §§2.2–3.2 · original course examples

DERIVATION

Learning the Velocity

Conditional supervision learns the marginal field

$$\mathbb{E}\|v - U\|^2 = \mathbb{E}\|v - m\|^2 + \mathbb{E}\|U - m\|^2$$

$$v - U = (v - m) + (m - U)$$

$$\mathbb{E}[(v - m)^T (m - U) \mid X_\tau, \tau] = (v - m)^T 0 = 0$$

$$\mathcal{L}_{\mathrm{CFM}} = \mathcal{L}_{\mathrm{FM}} + C, \qquad \nabla_\theta C = 0$$

where $m = \mathbb{E}[U \mid X_\tau, \tau]$; $v = v_\theta(X_\tau, \tau)$; finite second moments.

Tong et al. (2024), §§2.2–3.2 · original course examples

ALGORITHM

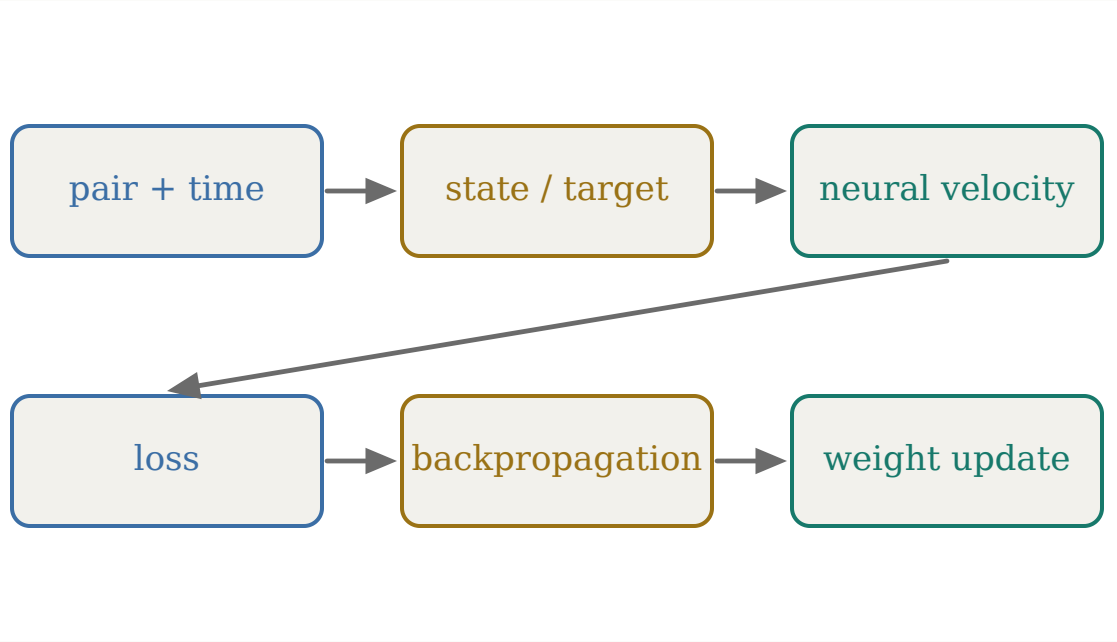
Learning the Velocity

Training is ordinary supervised regression

Sample endpoints and a generation time.

Construct the state and velocity target.

Predict velocity, compute squared error, backpropagate.



Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Learning the Velocity

Shapes and loss reduction

$B = 2$ samples, $d = 3$ coordinates. $V, U \in \mathbb{R}^{2\times 3}$. Use $L = \frac{1}{Bd}\sum_{i,j}(V_{ij} - U_{ij})^2$.

A Gradient is 2(V−U)

B Gradient is (V−U)/3

C Gradient has shape 3×2

Goodfellow et al. (2016), §§3.8–3.9.3 · original course examples

WORKED EXAMPLE

Learning the Velocity

Worked example: a scalar linear field

$$v_w(x,\tau) = wx, \quad (x,u) = (1,2),(2,1)$$

$$L'(w) = (w-2) + 2(2w-1) = 5w - 4$$

$$w_0 = 0, \quad \eta = .1, \quad w_1 = 0 - .1(-4) = .4$$

$$L(0) = 2.5, \qquad L(.4) = 1.3$$

where $w \in \mathbb{R}$: trainable weight; $L(w) = \frac{1}{2}[(w-2)^2 + (2w-1)^2]$.

Goodfellow et al. (2016), §§3.8–3.9.3 · original course examples

YOUR TURN

Learning the Velocity

Transfer: a negative input

$v_w(x) = wx$, one pair $(x, u) = (-2, 3)$, $L = \frac{1}{2}(wx - u)^2$, $w = 1$, $\eta = .1$.

A New w is 0

B New w is 2

C New w is 1.5

Goodfellow et al. (2016), §§3.8–3.9.3 · original course examples

SYNTHESIS

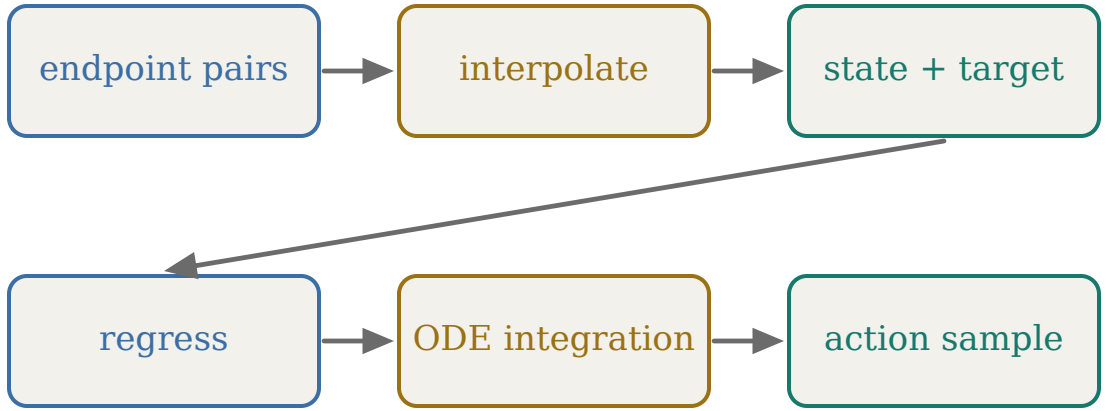
Learning the Velocity

The training chain is complete

Paired endpoints define targets.

Conditional least squares defines one field.

A new random start still needs integration.



fixed observation + language enter the regressor

Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Sampling the Learned Flow

What remains fixed while sampling?

$dx_\tau/d\tau = v_\theta(x_\tau, \tau, c)$; draw x_0 once. The condition c is one observation.

A. Optimize θ at every flow step.

B. Hold θ and c fixed; update x and τ.

C. Replace x with fresh noise every step.

Tong et al. (2024), §§2.2–3.2 · original course examples

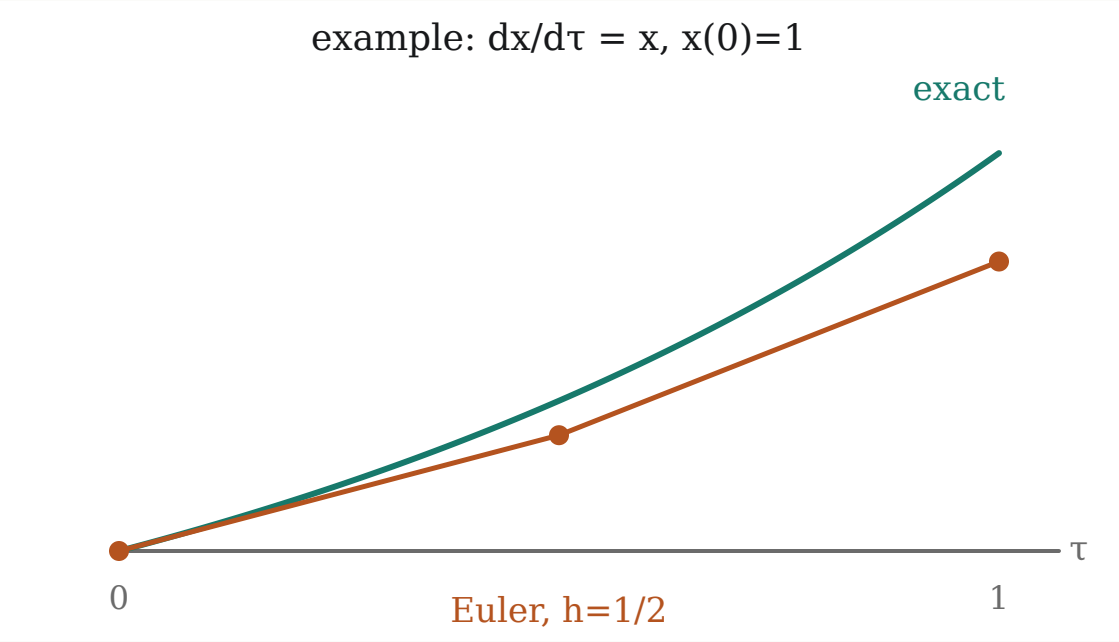
RECALL

Sampling the Learned Flow

Euler multiplies velocity by a step size

$$x_{j+1} = x_j + h\, v_\theta(x_j, \tau_j, c), \quad \tau_{j+1} = \tau_j + h$$

where j: numerical step index; $h > 0$: step size; $\tau_0 = 0$, $Nh = 1$.



Lipman et al. (2023), §§2–4 · original course examples

WORKED EXAMPLE

Sampling the Learned Flow

Worked example: two Euler steps

$$v(x,\tau) = x, \quad x_0 = 1, \quad h = \tfrac{1}{2}$$

$$x_1 = 1 + \tfrac{1}{2}(1) = 1.5$$

$$x_2 = 1.5 + \tfrac{1}{2}(1.5) = 2.25$$

$$x(1) = e \approx 2.71828$$

where x_j: numerical state; exact solution $x(\tau) = e^{\tau}$.

Lipman et al. (2023), §§2–4 · original course examples

YOUR TURN

Sampling the Learned Flow

Transfer: a time-dependent rate

$v(x, \tau) = 2\tau$, $x_0 = 1$, $h = 1/2$. Use two forward Euler steps from $\tau = 0$ to 1.

A 1.5

B 2

C 2.5

Lipman et al. (2023), §§2–4 · original course examples

DERIVATION

Sampling the Learned Flow

A midpoint estimate improves the local rate

$$k_1 = v(x_j, \tau_j), \quad k_2 = v(x_j + \tfrac{h}{2} k_1, \tau_j + \tfrac{h}{2})$$

$$k_1 = v(x_j, \tau_j), \qquad x_{\text{trial}} = x_j + \tfrac{h}{2} k_1$$

$$k_2 = v(x_{\text{trial}}, \tau_j + \tfrac{h}{2})$$

$$x_{j+1} = x_j + h k_2$$

where k_1, k_2: rates; $x_j + (h/2)k_1$: trial midpoint state; condition c held fixed.

Lipman et al. (2023), §§2–4 · original course examples

WORKED EXAMPLE

Sampling the Learned Flow

Worked example: midpoint on the same field

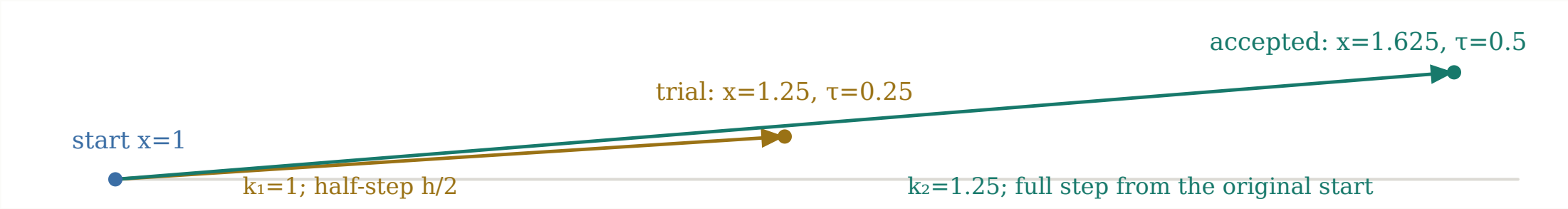
$$v(x,\tau) = x, \quad x_0 = 1, \quad h = \tfrac{1}{2}$$

$$k_1 = 1, \quad x_{\text{trial}} = 1 + .25(1) = 1.25, \quad k_2 = 1.25$$

$$x_1 = 1 + .5(1.25) = 1.625$$

$$x_2 = (1 + h + h^2/2)^2 = 1.625^2 = 2.640625$$

where k_1, k_2: rates; each accepted step uses $x_{j+1} = x_j + hk_2$.



Lipman et al. (2023), §§2–4 · original course examples

YOUR TURN

Sampling the Learned Flow

Transfer: a fixed evaluation budget

Budget: 8 calls to v_θ. Euler uses 1 call/step; explicit midpoint uses 2. Both integrate $[0,1]$.

A Euler h=1/8; midpoint h=1/4

B Both h=1/8

C Midpoint always gives the smaller error

Lipman et al. (2023), §§2–4 · original course examples

DIAGNOSTIC

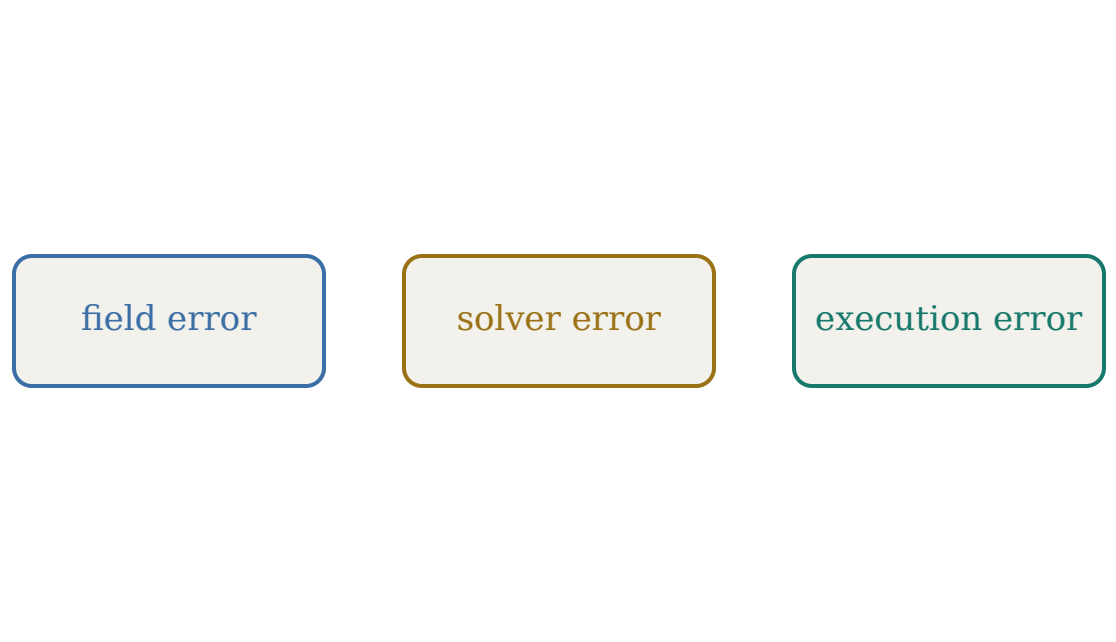
Sampling the Learned Flow

Different errors need different repairs

Field error: the learned velocity differs from the target.

Solver error: finite steps approximate that learned ODE.

Execution error: robot dynamics differ from assumptions.



Lipman et al. (2023), §§2–4 · original course examples

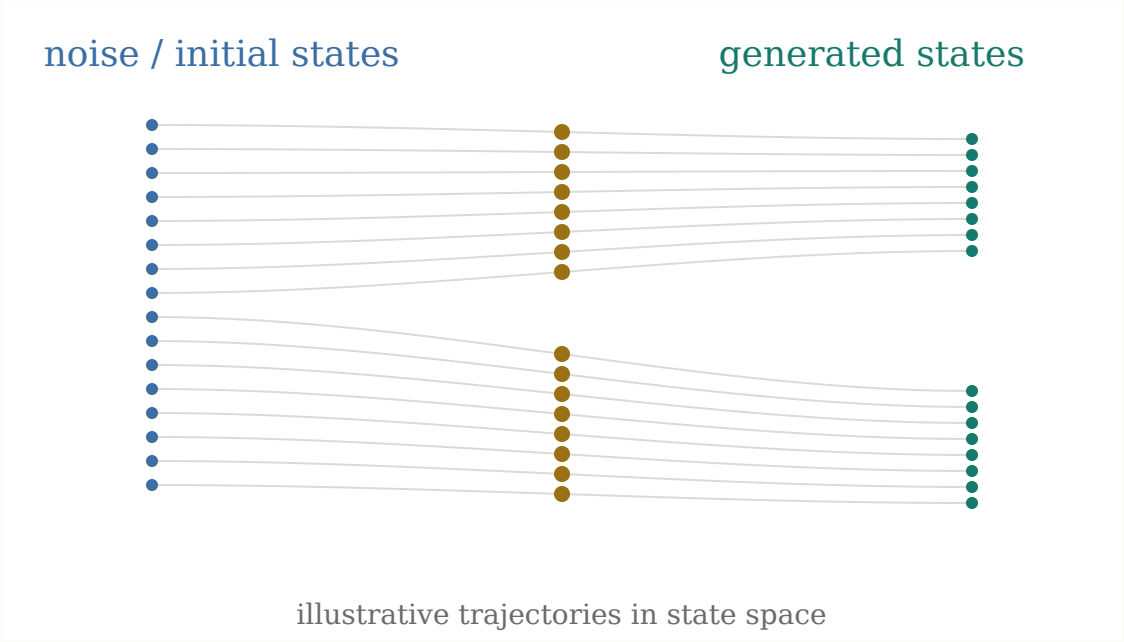
INTERPRETATION

Sampling the Learned Flow

Random starts preserve sample diversity

$$X_0 \sim p_0, \qquad X_1 = \phi_1(X_0; c)$$

where c: fixed observation; ϕ_1: deterministic learned flow map for that condition.



illustrative trajectories in state space

Lipman et al. (2023), §§2–4 · original course examples

YOUR TURN

Sampling the Learned Flow

Reversing the generation clock

$s = 1 - \tau$; $y_s = x_{1-s}$; $dx_\tau/d\tau = v(x_\tau, \tau)$. Find dy_s/ds.

A $v(y_s, 1-s)$

B $-v(y_s, 1-s)$

C $-v(y_s, s)$ always

[openpi: Pi0.compute_loss / sample_actions](#) · original course examples

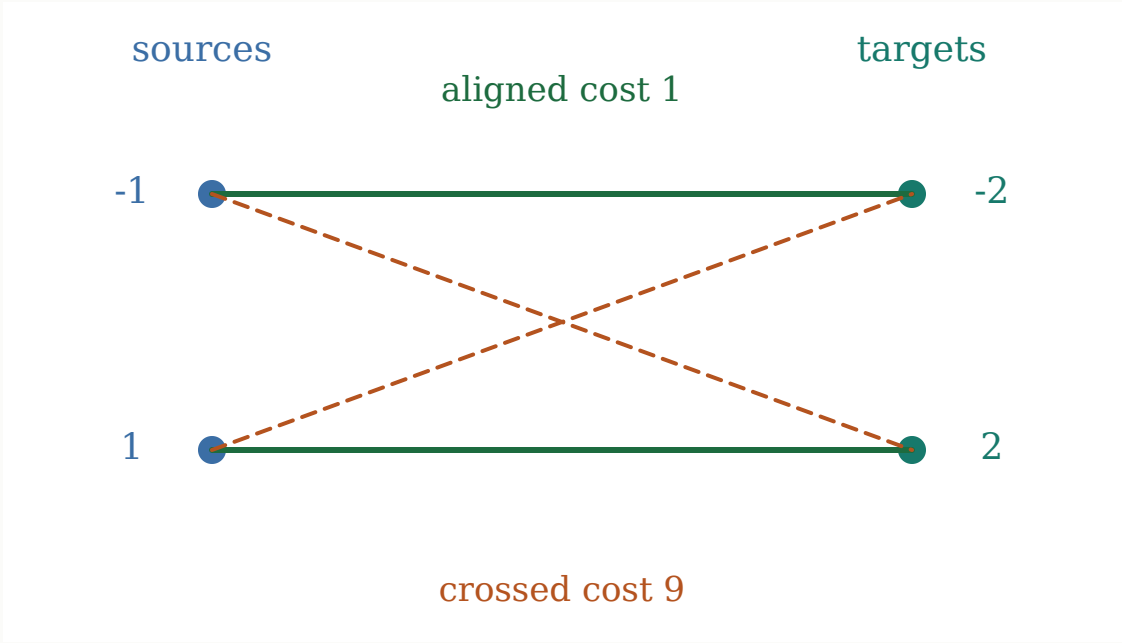
DEFINITION

Pairing and Optimal Transport

A coupling changes pairs, not endpoint laws

$$\pi(X_0 \in A) = p_0(A), \quad \pi(X_1 \in B) = p_1(B)$$

where π: joint endpoint law; A, B: measurable sets; p_0, p_1: fixed marginals.



Tong et al. (2024), §§2.2–3.2 · original course examples

WORKED EXAMPLE

Pairing and Optimal Transport

Worked example: two pairings

$$x_0 \in \{-1, 1\}, \quad x_1 \in \{-2, 2\}$$

$$C_{\text{aligned}} = \tfrac{1}{2}[(-2+1)^2 + (2-1)^2] = 1$$

$$C_{\text{crossed}} = \tfrac{1}{2}[(2+1)^2 + (-2-1)^2] = 9$$

where Each endpoint has mass $1/2$; pair cost is squared displacement $(x_1 - x_0)^2$.

Tong et al. (2024), §§2.2–3.2 · original course examples

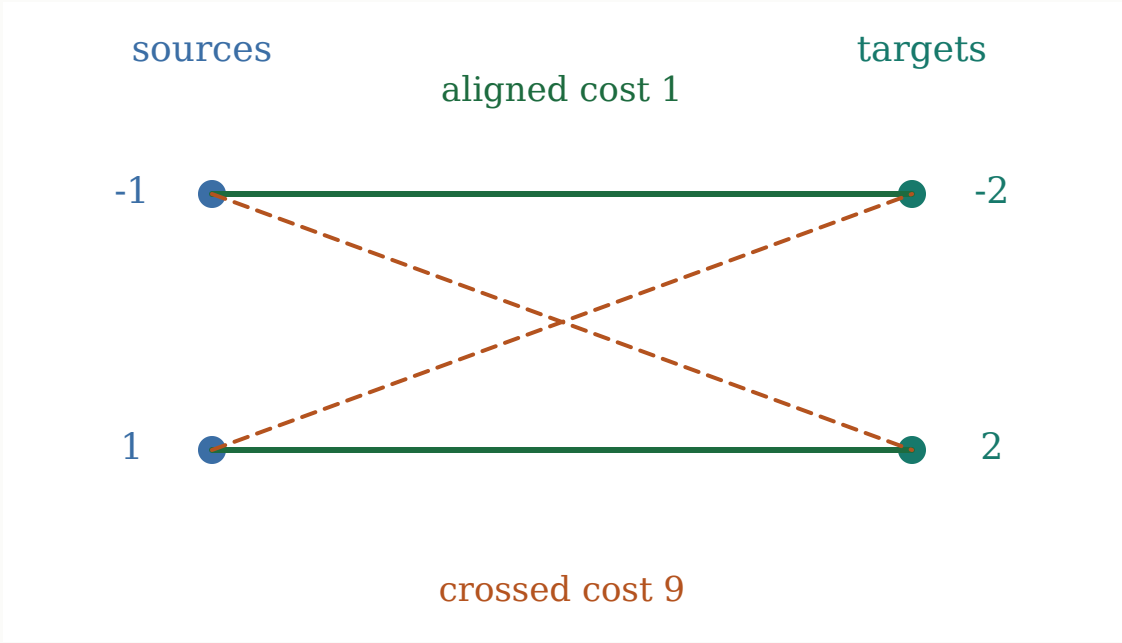
DEFINITION

Pairing and Optimal Transport

Optimal transport chooses the coupling

$$\pi^* \in \arg\min_{\pi \in \Pi(p_0, p_1)} \mathbb{E}_\pi \|X_1 - X_0\|^2$$

where $\Pi(p_0, p_1)$: couplings with the required marginals; cost uses Euclidean coordinates.



Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Pairing and Optimal Transport

Transfer: compute the assignment cost

Sources $0, 2$; targets $1, 4$; each has mass $1/2$. Compare aligned $(0 \to 1, 2 \to 4)$ and crossed pairs.

A. Aligned cost 2.5; crossed 8.5

B. Aligned cost 5; crossed 17

C. Both cost 2.5

Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Pairing and Optimal Transport

Are learned trajectories necessarily straight?

Training paths are straight. The learned field is $v^*(x, \tau) = \mathbb{E}[U \mid x, \tau]$.

A Yes, every generated path is one training segment.

B No; the conditional average can vary along a path.

C Every learned path must cross another path.

Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Pairing and Optimal Transport

What does a conditional path establish?

For each data point y, Gaussian noise contracts toward y along a simple conditional path. The data point may be sampled independently.

A. This alone proves globally optimal endpoint pairing.

B. A simple conditional path and global OT are different claims.

C. Every coupling has equal expected displacement cost.

Tong et al. (2024), §§2.2–3.2 · original course examples

YOUR TURN

Pairing and Optimal Transport

Re-pair examples with different conditions

Condition c says “move left”. Its data actions are negative. A batch also contains “move right” examples.

A. Re-pair all targets freely and retain old c.

B. Keep targets compatible with each condition.

C. Remove c from training and still claim conditioning.

Tong et al. (2024), §§2.2–3.2 · original course examples

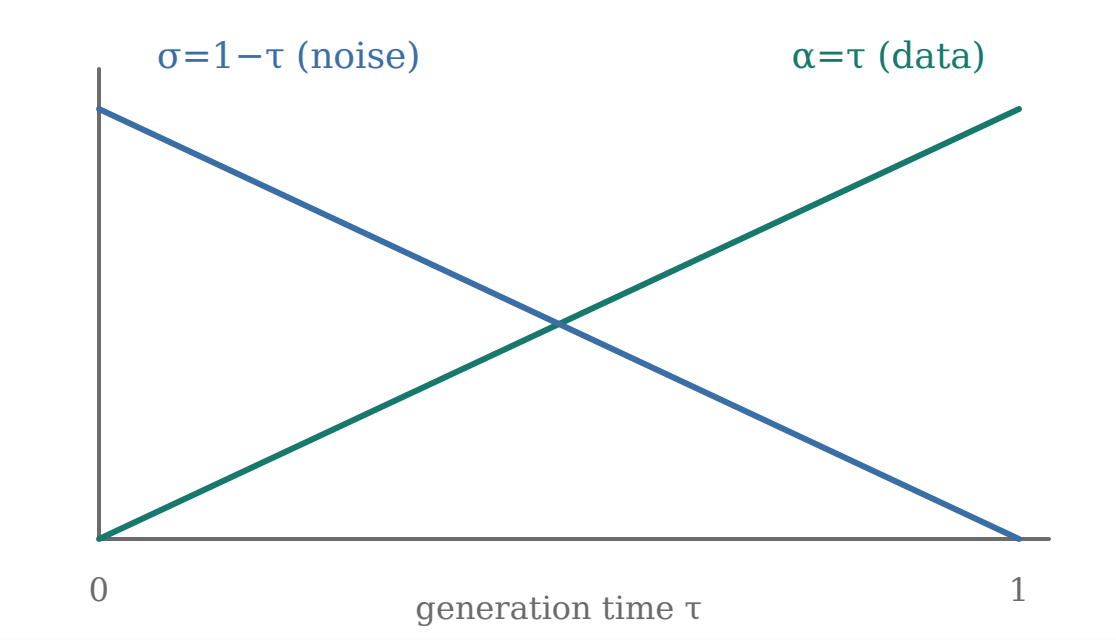
DEFINITION

The Bridge to Diffusion

A Gaussian path can use different schedules

$$X_\tau = \alpha(\tau)Y + \sigma(\tau)\varepsilon$$

where Y: data vector; $\varepsilon \sim \mathcal{N}(0, I)$ independent; α, σ: differentiable scalars.



Lipman et al. (2023), §4 and Appendix D · original course examples

DERIVATION

The Bridge to Diffusion

The score recovers average noise

$$s_\tau(x) = \nabla_x \log p_\tau(x)$$

$$\nabla_x \log p_\tau(x \mid y) = -\frac{x - \alpha y}{\sigma^2} = -\frac{\varepsilon}{\sigma}$$

$$s_\tau(x) = \mathbb{E}[\nabla_x \log p_\tau(x \mid Y) \mid x]$$

$$\mathbb{E}[\varepsilon \mid x] = -\sigma s_\tau(x)$$

where s_τ: score; $\sigma > 0$; Y: data, ε: noise. Interchange justified.

Lipman et al. (2023), §4 and Appendix D · original course examples

DERIVATION

The Bridge to Diffusion

Score to velocity

$$X_\tau = \alpha Y + \sigma \varepsilon$$

$$v = \dot{\alpha} m + \dot{\sigma} n$$

$$m = (x - \sigma n)/\alpha, \quad n = -\sigma s$$

$$v = (\dot{\alpha}/\alpha) x + (\dot{\alpha}\sigma^2/\alpha - \sigma\dot{\sigma}) s$$

where Y: data; independent $\varepsilon \sim \mathcal{N}(0, I)$; $m = \mathbb{E}[Y|x]$, $n = \mathbb{E}[\varepsilon|x]$; s: score; $\alpha \neq 0$, $\sigma > 0$. Dots: τ-rates.

Lipman et al. (2023), §4 and Appendix D · original course examples

DERIVATION

The Bridge to Diffusion

The straight-path bridge

$$v(x,\tau) = \frac{x}{\tau} + \frac{1-\tau}{\tau} s_\tau(x), \quad 0 < \tau < 1$$

$$\alpha = \tau, \quad \dot{\alpha} = 1, \quad \sigma = 1 - \tau, \quad \dot{\sigma} = -1$$

$$\frac{(1-\tau)^2}{\tau} + (1-\tau) = (1-\tau)\left[\frac{1-\tau}{\tau} + 1\right] = \frac{1-\tau}{\tau}$$

where $\alpha = \tau$, $\dot{\alpha} = 1$, $\sigma = 1 - \tau$, $\dot{\sigma} = -1$.

Lipman et al. (2023), §4 and Appendix D · original course examples

YOUR TURN

The Bridge to Diffusion

Worked bridge: a known Gaussian path

Fixed data $Y = 2$, $\tau = 1/2$, $x = 0$. Then $p_\tau = \mathcal{N}(1, 1/4)$. Find score and velocity.

A s=4, v=4

B s=2, v=2

C s=−4, v=−4

Lipman et al. (2023), §4 and Appendix D · original course examples

YOUR TURN

The Bridge to Diffusion

Transfer: reverse the clock consistently

At course time $\tau = 1/2$, velocity is $+4$. Code time is $s = 1 - \tau$ and a sampling step has $\Delta s = -.1$.

A Code velocity -4, state increment $+.4$

B Code velocity $+4$, state increment $-.4$

C Code velocity -4, state increment $-.4$

openpi: Pi0.compute_loss / sample_actions · original course examples

DEFINITION

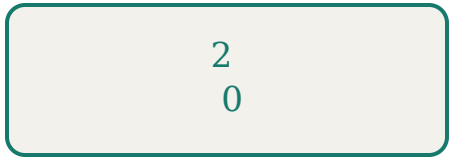
Vision-Language-Action Models

An action chunk is one generated vector

$$A_k = \mathrm{vec}(a_k, \ldots, a_{k+H-1}) \in \mathbb{R}^{Hd_a}$$

where k: robot step; H: chunk length; $a_k \in \mathbb{R}^{d_a}$: one action.

one candidate = several future actions



internal refinement ≠ robot execution

Black et al., π0 (2024/2026), §IV; App. A-D · original course examples

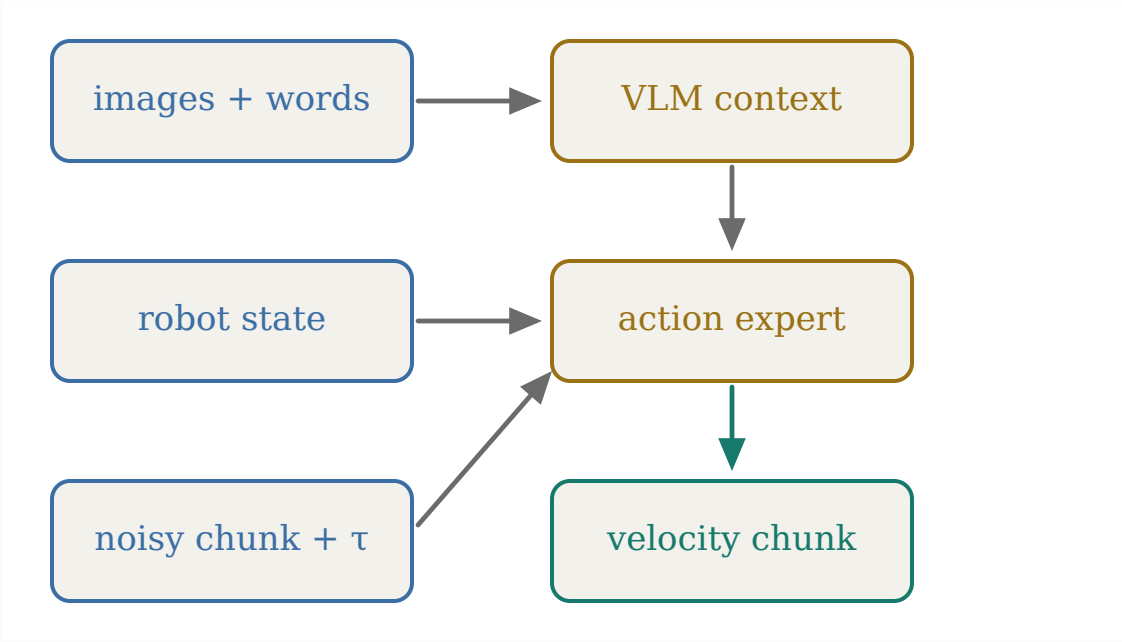
DEFINITION

Vision-Language-Action Models

The condition describes the task now

$$c_k = (I_k^{1:C}, \ell_k, q_k), \qquad v_\theta(A_k^\tau, \tau, c_k)$$

where $I^{1:C}$: camera images; ℓ: instruction; q: robot state; A^τ: noisy candidate.



Black et al., π₀ (2024/2026), §IV; App. A-D · original course examples

MODEL INSTANCE

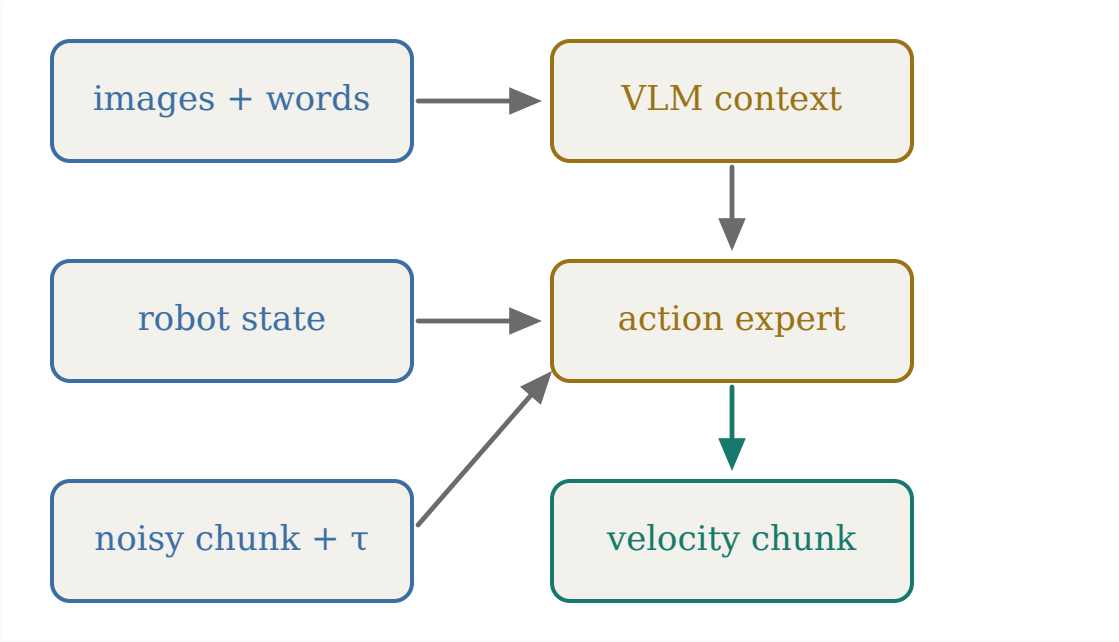
Vision-Language-Action Models

π₀ adds a continuous action expert

VLM representations supply image-language context.

A smaller action expert processes noisy actions.

Attention connects context to action refinement.



Black et al., π₀ (2024/2026), §IV; App. A-D · original course examples

CONVENTION BRIDGE

Vision-Language-Action Models

Official code uses the opposite clock

$$Z_s = (1-s)A + s\varepsilon, \quad U_s = \varepsilon - A, \quad s: 1 \to 0$$

$$Z_s = X_{1-s}, \qquad \frac{dZ_s}{ds} = -\frac{dX_\tau}{d\tau}$$

$$\Delta s = -h, \quad U_s = -U_\tau \quad \Rightarrow \quad \Delta s\, U_s = h U_\tau$$

where A: data action chunk; ε: noise; $s = 1 - \tau$; step $\Delta s < 0$.

openpi: Pi0.compute_loss / sample_actions · original course examples

YOUR TURN

Vision-Language-Action Models

A code step toward the data

Action target $A = 2$, noise $\varepsilon = -1$. Code clock starts $s = 1$, $z = -1$; use exact target rate and $\Delta s = -1/4$.

A u=−3, new z=−.25

B u=+3, new z=−1.75

C u=−3, new z=−1.75

openpi: Pi0.compute_loss / sample_actions · original course examples

YOUR TURN

Vision-Language-Action Models

Generation, execution, and replanning

Generate $H = 6$ actions. Execute the first $r = 2$, observe again, then generate a new chunk.

A. Six internal steps imply six robot actions.

B. Physical time advances by two executed actions.

C. The remaining four must always be executed.

Black et al., π_0 (2024/2026), §IV; App. A-D · original course examples

MODEL INSTANCE

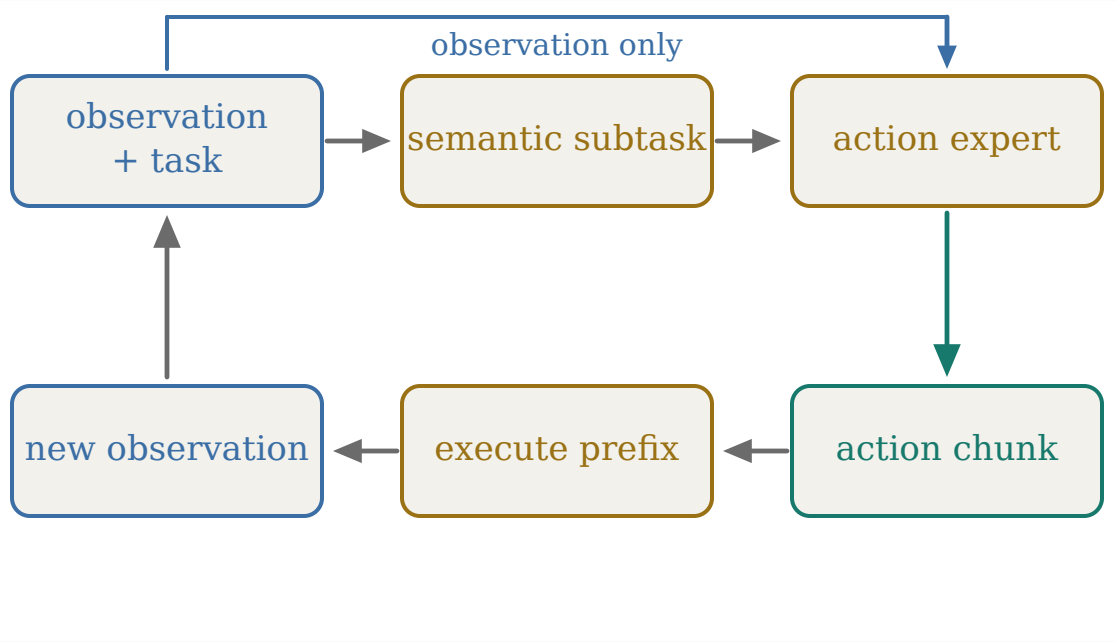
Vision-Language-Action Models

$\pi_{0.5}$ connects semantic and action prediction

A high-level prediction identifies a subtask.

The action expert conditions on that subtask.

Training combines robot and non-robot sources.



Physical Intelligence, $\pi_{0.5}$ (2025), §§IV-A–D · original course examples

OBJECTIVE

Vision-Language-Action Models

Discrete and continuous losses

$$\mathcal{L} = \mathcal{L}_{\text{token}} + \lambda \mathcal{L}_{\text{flow}}$$

$$\mathcal{L}_{\text{token}} = -\log p_\theta(y \mid \text{context})$$

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\|v_\theta(Z_s, s, c) - U_s\|^2$$

where $\mathcal{L}_{token}$: next-token cross-entropy; $\mathcal{L}_{flow}$: velocity regression; $\lambda > 0$: weight.

Physical Intelligence, $\pi_{0.5}$ (2025), §§IV-A–D · original course examples

YOUR TURN

Vision-Language-Action Models

What follows from a richer training mixture?

A VLA receives additional image-language and robot examples. Its low-level action model still uses flow matching.

A. The ODE becomes physical robot dynamics.

B. Representation learning changes; the flow objective can remain.

C. Generalization is guaranteed for every new home.

Physical Intelligence, $\pi_{0.5}$ (2025), §§IV-A–D · original course examples

WORKED EXAMPLE

From Data to an Executed Action

Integrated example: training a chunk

$$A = (2, 0)^T, \quad \varepsilon = (-2, 2)^T, \quad \tau = \tfrac{1}{4}$$

$$A^{1/4} = \tfrac{3}{4}(-2, 2)^T + \tfrac{1}{4}(2, 0)^T = (-1, 1.5)^T$$

$$U = (4, -2)^T, \quad V = (3, -1)^T, \quad e = V - U = (-1, 1)^T$$

$$L = \tfrac{1}{2}(e_1^2 + e_2^2) = 1, \qquad \nabla_V L = e$$

where $H = 2$, $d_a = 1$; fixed condition c; $A^\tau = (1 - \tau)\varepsilon + \tau A$.

Tong et al. (2024), §§2.2–3.2 · original course examples

WORKED EXAMPLE

From Data to an Executed Action

Generate a chunk; execute a prefix

$$\varepsilon = (-2, 2)^T, \quad v = (4, -2)^T, \quad h = \tfrac{1}{4}$$

$$x_j = (-2, 2)^T + \tfrac{j}{4}(4, -2)^T$$

$$x_1 = (-1, 1.5)^T, \quad x_2 = (0, 1)^T, \quad x_3 = (1, .5)^T, \quad x_4 = (2, 0)^T$$

$$H = 2, \quad N = 4, \quad H_{\text{exec}} = 1: \quad a_k = 2$$

where Toy field v; $N = 4$ solver steps; $H = 2$ actions; execute first action.

Black et al., π_0 (2024/2026), §IV; App. A-D · original course examples

YOUR TURN

From Data to an Executed Action

Transfer: the observation changes mid-execution

After one executed action, a new image shows an obstacle. Old condition c described a clear route.

A More ODE steps under old c incorporate the new image.

B Update the condition and replan under the policy.

C Low training MSE guarantees the old chunk is safe.

Black et al., π_0 (2024/2026), §IV; App. A-D · original course examples

SYNTHESIS

From Data to an Executed Action

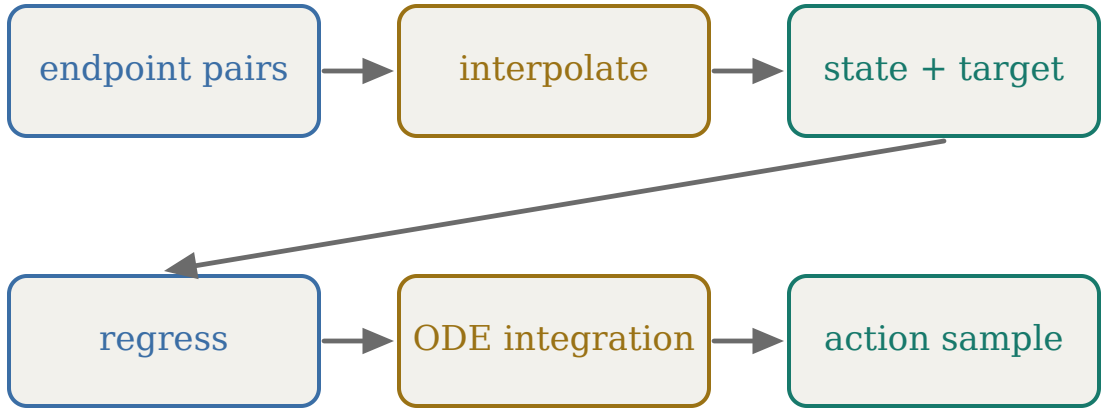
The course ingredients meet in one policy

Linear algebra represents chunks and attention.

Calculus trains a velocity predictor.

Probability defines the target law.

ODE integration generates a candidate action sequence.



fixed observation + language enter the regressor

Tong et al. (2024), §§2.2–3.2 · original course examples

References

Probability flows and regression

1. Lipman et al. (2023), §§2–4
2. Tong et al. (2024), §§2.2–3.2
3. Goodfellow et al. (2016), §§3.8–3.9.3

References

Robot models and implementation conventions

1. Black et al., π_0 (2024/2026), §IV; App. A-D
2. Physical Intelligence, $\pi_{0.5}$ (2025), §§IV-A–D
3. openpi: Pi0.compute_loss / sample_actions

Original diagrams and examples · CC BY-NC 4.0