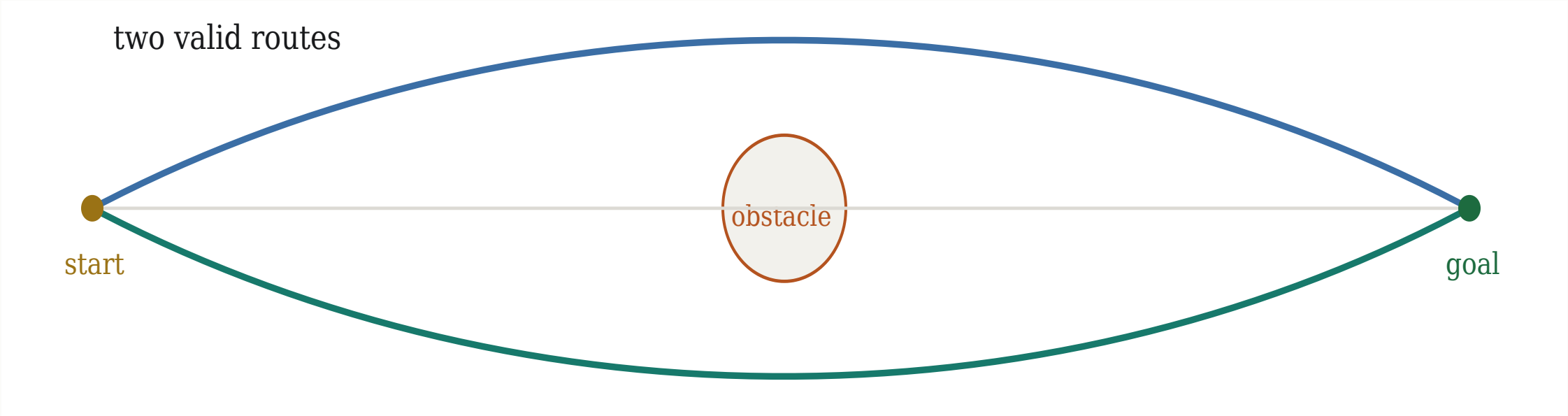
Diffusion and Diffusion Policy

From noisy vectors to robot action sequences



> One observation. Several successful action sequences.

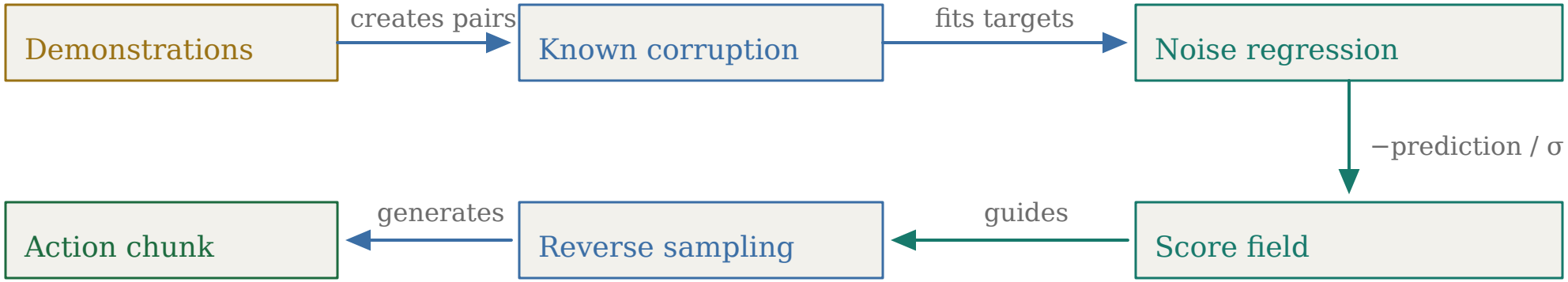
`input / noise` · `demonstrations` · `generated actions`

Diffusion Policy §4.1 · Chi et al. (2023/2024)

CONCEPT MAP

Diffusion and Diffusion Policy

A route from demonstrations to actions



DDPM §3; Diffusion Policy §2.3 · Ho, Jain & Abbeel (2020) · Chi et al. (2023/2024)

PREREQUISITE RECALL

Diffusion and Diffusion Policy

Four familiar tools meet one new task

Recall: Lecture 8 pp.12–18; Lecture 5 pp.11,29; Lecture 6 pp.30,62.

$$\mathbb{E}\|Y - g(X)\|^2$$

X, Y random input and target

g prediction rule

Gaussian noise

Input gradient

Least squares

Iterative update

DL §3.8; DDPM §3.2 · Ho, Jain & Abbeel (2020) · Goodfellow et al. (2016)

MOTIVATION

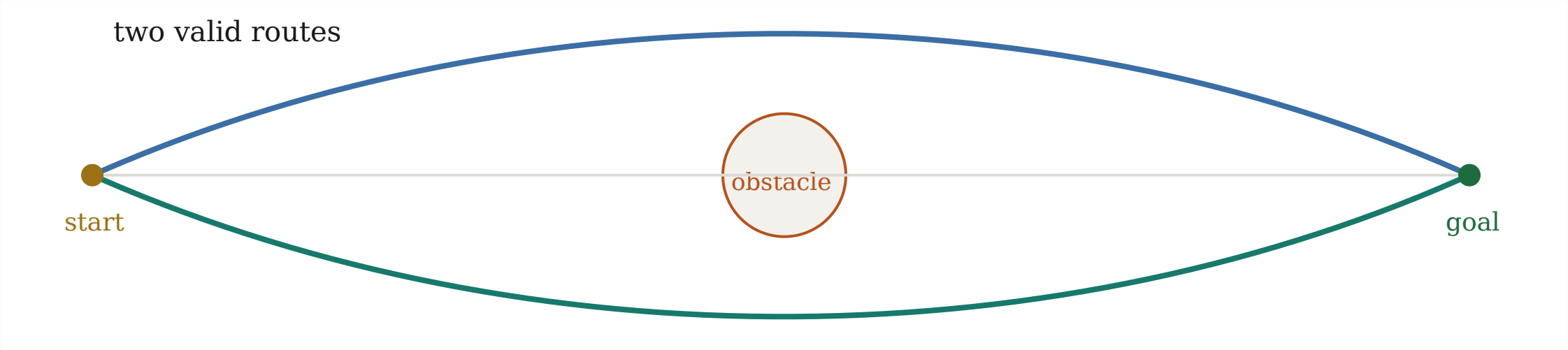
Multimodal actions

Two valid actions can have an invalid mean

$$P(A = -2 \mid o) = P(A = 2 \mid o) = \tfrac{1}{2}$$

A scalar action

o fixed observation



Diffusion Policy §4.1 · Chi et al. (2023/2024)

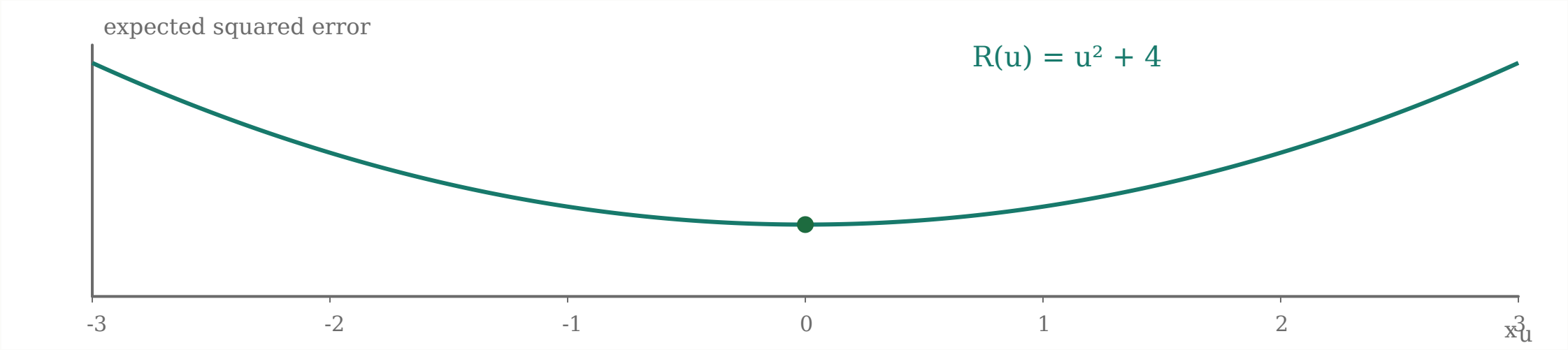
GENERAL RULE

Multimodal actions

Squared error selects a conditional mean

$$\mathbb{E}[(A-u)^2 \mid o] = \mathrm{Var}(A \mid o) + (u - \mathbb{E}[A \mid o])^2$$

- u chosen scalar prediction
- A, o action and observation



DL §3.8; course derivation · Goodfellow et al. (2016)

Multimodal actions

Squared error selects a conditional mean

Expand around the conditional mean

Fix o. Let m=E[A|o], u the chosen scalar prediction, and assume a finite second moment.

1. $A - u = (A - m) + (m - u)$.
2. $(A - u)^2 = (A - m)^2 + 2(A - m)(m - u) + (m - u)^2$.
3. $\mathbb{E}[A - m \mid o] = 0$, so the averaged cross term is zero.

DL §3.8; course derivation · Goodfellow et al. (2016)

Multimodal actions

Squared error selects a conditional mean

Only one term depends on the prediction

m is fixed when we change u.

1. $\mathbb{E}[(A-u)^2 \mid o] = \mathrm{Var}(A \mid o) + (m-u)^2$.
2. The variance is independent of u; the remaining square is minimized at zero.
3. Thus $u^* = m = \mathbb{E}[A \mid o]$.

DL §3.8; course derivation · Goodfellow et al. (2016)

WORKED EXAMPLE

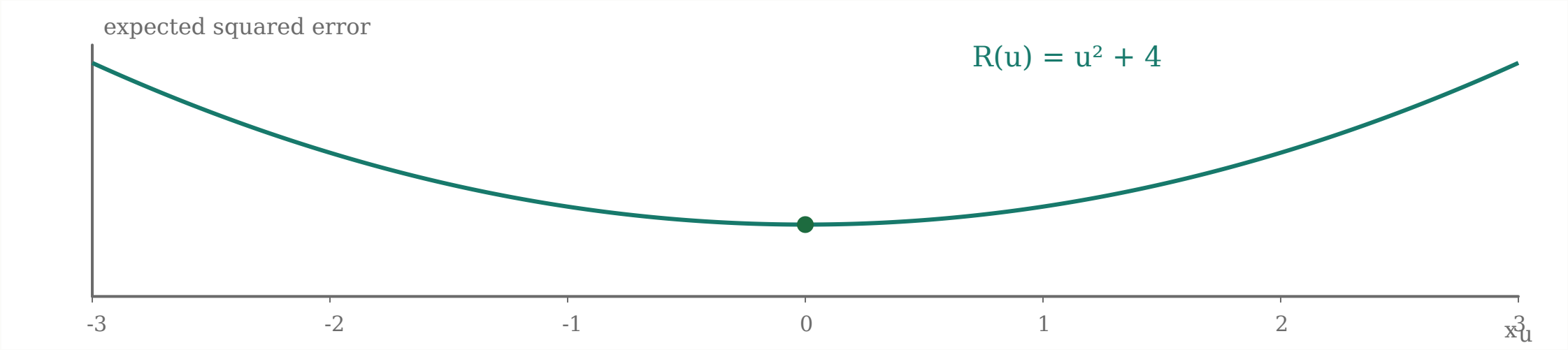
Multimodal actions

Worked example: the average misses both modes

WORKED EXAMPLE

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

- R conditional squared-error risk
- u action prediction



Diffusion Policy §4.1; course example · Chi et al. (2023/2024)

YOUR TURN

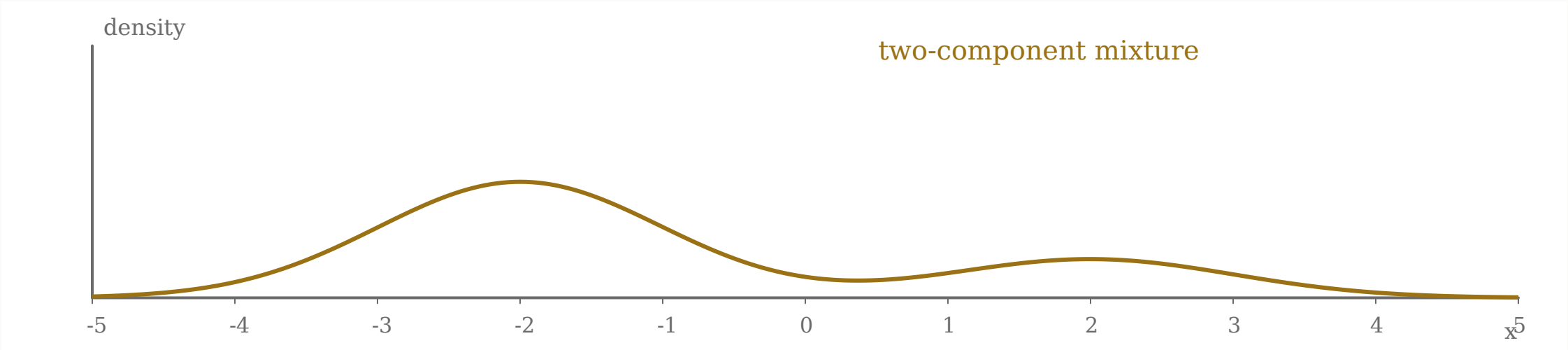
Multimodal actions

Transfer: one route is more common

Find the squared-error minimizer u; compare it with the mode.

$$P(A = -2 \mid o) = \tfrac{3}{4}, \quad P(A = 2 \mid o) = \tfrac{1}{4}$$

- A action
- u fixed prediction to optimize



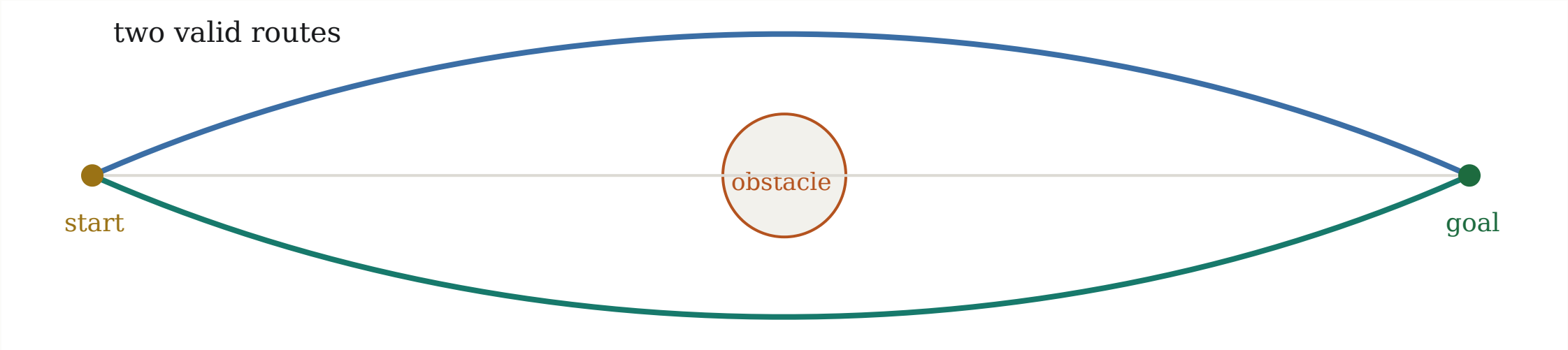
DL §3.8; course exercise · Goodfellow et al. (2016)

YOUR TURN

Multimodal actions

A distribution and its mean answer different questions

Fixed observation; equally likely actions -2 and 2.
Compare: conditional mean vs a random draw.

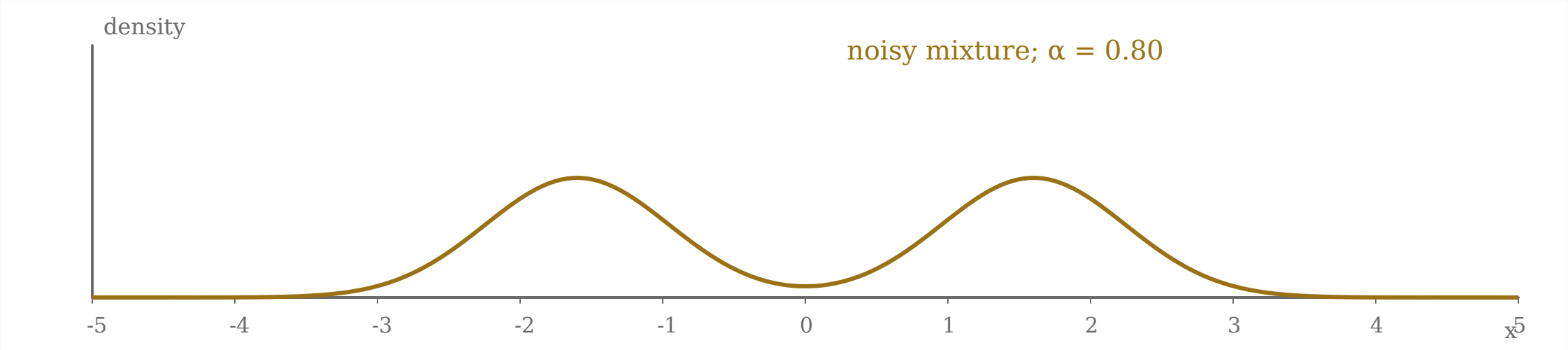


Diffusion Policy §4.1; course exercise · Chi et al. (2023/2024)

MOTIVATION

Forward diffusion

Noise creates a supervised learning task



DDPM §2, Algorithms 1–2 · Ho, Jain & Abbeel (2020)

OBJECTS

Forward diffusion

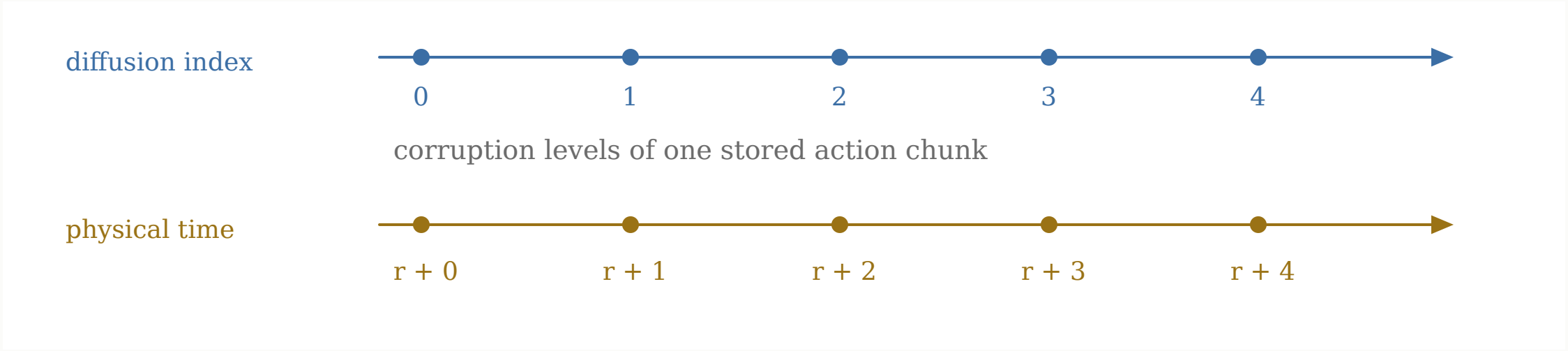
Two clocks and three kinds of vectors

$$x_0, x_k, \epsilon \in \mathbb{R}^d, \quad k \in \{1, \dots, K\}$$

x_0, x_k clean and noisy vectors

k, K noise index and final level

ϵ standard Gaussian noise



DDPM §2; Diffusion Policy §2.3 · Ho, Jain & Abbeel (2020) · Chi et al. (2023/2024)

DEFINITION

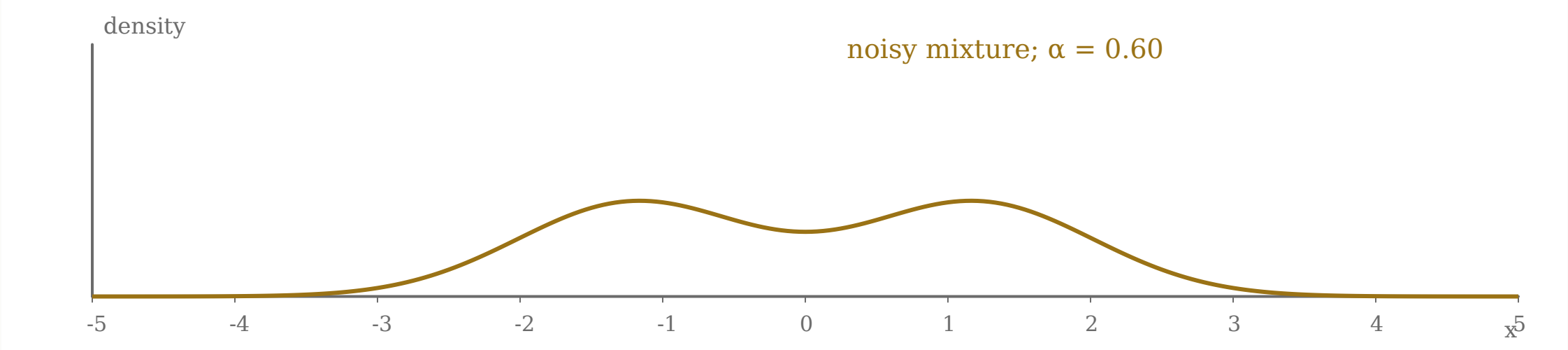
Forward diffusion

One step shrinks signal and adds fresh noise

$$x_k = \sqrt{1-\beta_k}\, x_{k-1} + \sqrt{\beta_k}\, z_k$$

$0 < \beta_k < 1$ chosen noise variance

$z_k \sim \mathcal{N}(0, I)$ fresh independent noise



DDPM §2, Eq.2 · Ho, Jain & Abbeel (2020)

YOUR TURN

Forward diffusion

Check the conditional mean and covariance

Find the conditional mean and covariance of x_k.

$$x_k = \sqrt{1-\beta_k}\, x_{k-1} + \sqrt{\beta_k}\, z_k$$

x_{k-1} fixed when conditioned on

z_k independent N(0,I) vector

DDPM Eq.2; DL §3.8 · Ho, Jain & Abbeel (2020) · Goodfellow et al. (2016)

YOUR TURN

Forward diffusion

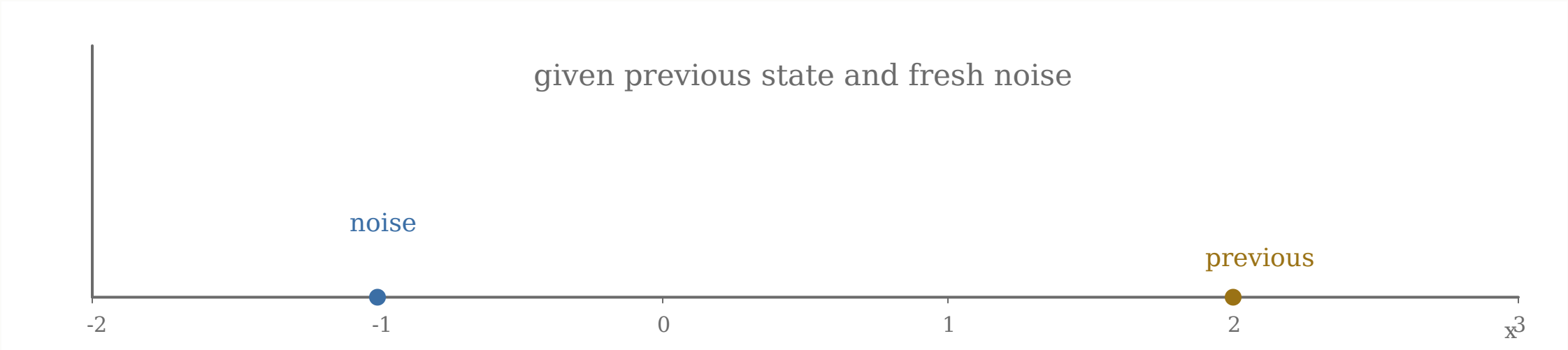
Transfer: variance is not the noise multiplier

Compute the realized x_k; show both coefficients.

$$\beta_k = 0.36, \quad x_{k-1} = 2, \quad z_k = -1$$

β_k variance

x_{k-1}, z_k given scalar values



DDPM Eq.2; course exercise · Ho, Jain & Abbeel (2020)

DERIVATION

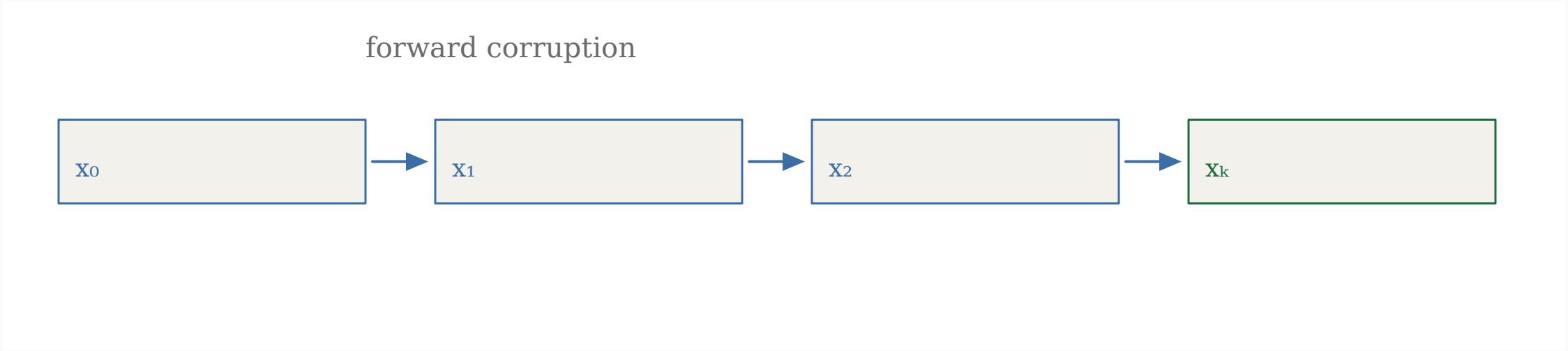
Forward diffusion

Two steps expose the cumulative signal

$$x_2 = \sqrt{a_2 a_1}\, x_0 + \sqrt{a_2 \beta_1}\, z_1 + \sqrt{\beta_2}\, z_2$$

$a_j = 1 - \beta_j$ one-step retention

z_1, z_2 independent N(0,I) noise



DDPM §2, Eq.4; course expansion · Ho, Jain & Abbeel (2020)

Forward diffusion

Two steps expose the cumulative signal

Substitute the first transition into the second

a_j=1−beta_j. The independent z1 and z2 are standard Gaussian vectors.

1. $x_1 = \sqrt{a_1} x_0 + \sqrt{\beta_1} z_1.$
2. $x_2 = \sqrt{a_2} x_1 + \sqrt{\beta_2} z_2.$
3. $x_2 = \sqrt{a_2 a_1} x_0 + \sqrt{a_2 \beta_1} z_1 + \sqrt{\beta_2} z_2.$

DDPM §2, Eq.4; course expansion · Ho, Jain & Abbeel (2020)

DERIVATION

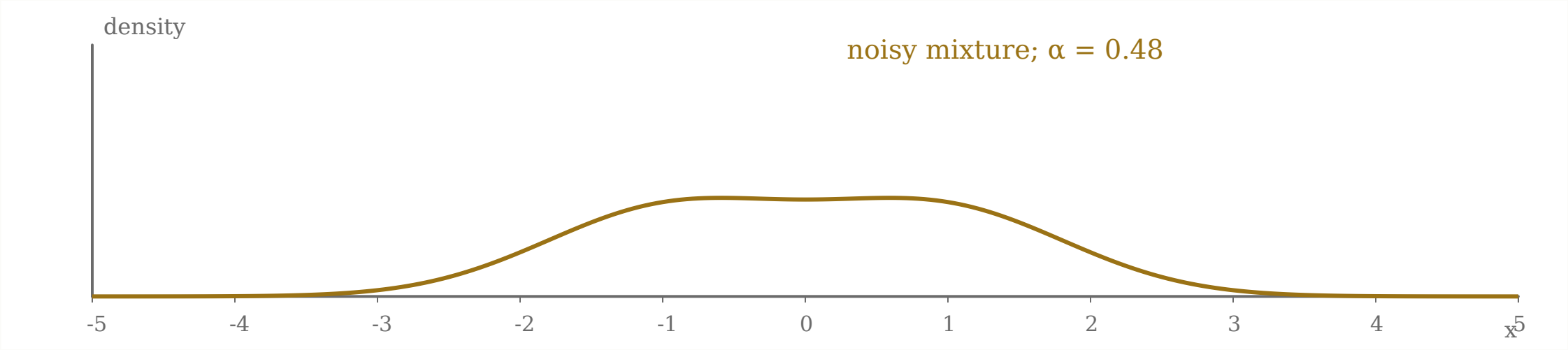
Forward diffusion

Independent Gaussian variances combine

$$a_2\beta_1 + \beta_2 = (1-\beta_2)\beta_1 + \beta_2 = 1 - a_1a_2$$

$a_j = 1 - \beta_j$ retained variance fraction

z_1, z_2 independent standard Gaussians



DDPM Eq.4; course derivation · Ho, Jain & Abbeel (2020)

Forward diffusion

Independent Gaussian variances combine

Combine independent centered noise terms

The two noise coefficients are sqrt(a2 beta1) and sqrt(beta2).

1. $\mathrm{Cov}(\sqrt{a_2\beta_1}z_1 + \sqrt{\beta_2}z_2) = (a_2\beta_1 + \beta_2)I.$
2. $a_2\beta_1 + \beta_2 = (1-\beta_2)\beta_1 + \beta_2 = 1 - a_1a_2.$
3. The sum is Gaussian. Replace it in distribution by a standard Gaussian scaled by sqrt(1−a1 a2).

DDPM Eq.4; course derivation · Ho, Jain & Abbeel (2020)

RESULT

Forward diffusion

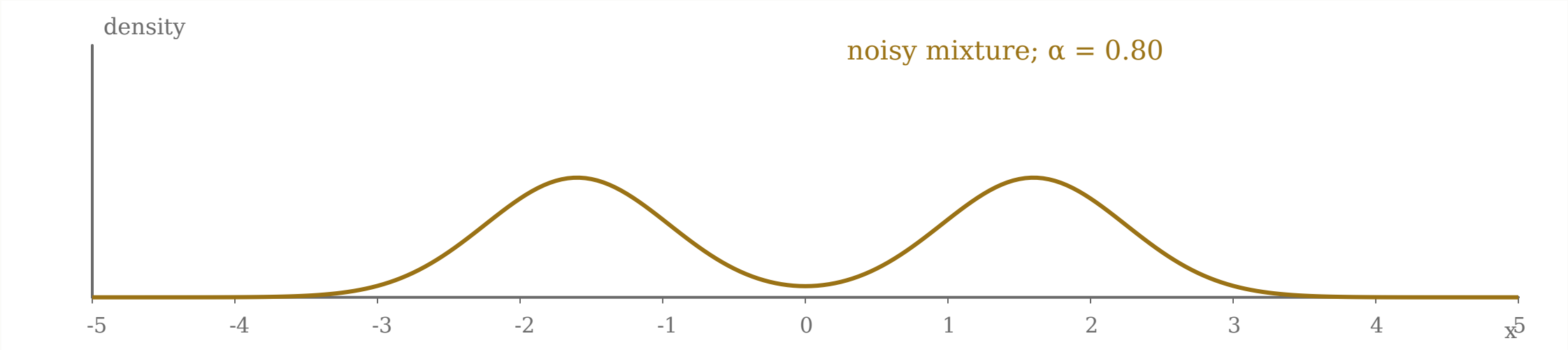
Any noise level can be sampled directly

$$x_k = \alpha_k x_0 + \sigma_k \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

$\alpha_k = \sqrt{\bar{a}_k}$ signal coefficient

$\sigma_k = \sqrt{1 - \bar{a}_k}$ noise standard deviation

$\bar{a}_k = \prod_{j=1}^{k} a_j$ cumulative retention



DDPM Eq.4; notation bridge · Ho, Jain & Abbeel (2020)

INTERPRETATION

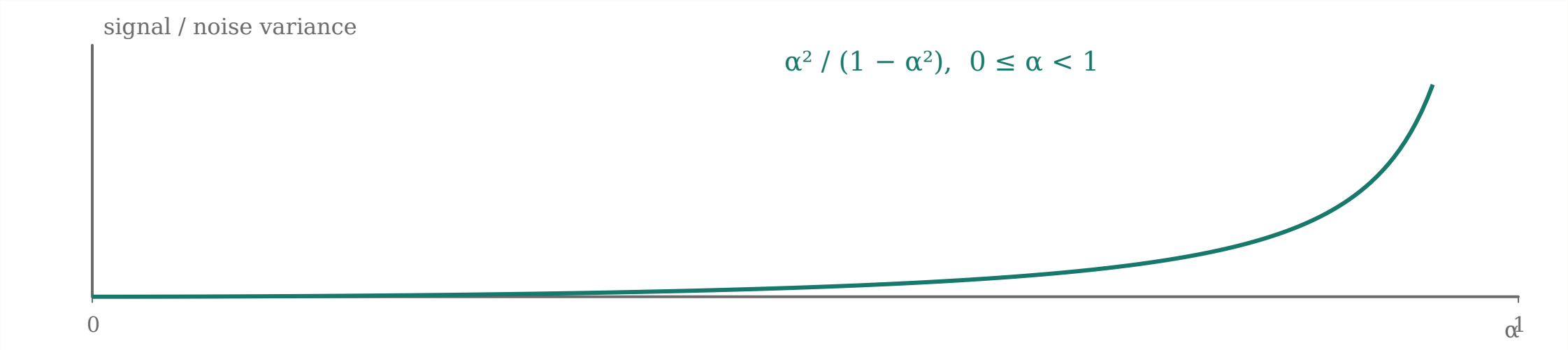
Forward diffusion

A schedule controls signal-to-noise ratio

$$\mathrm{SNR}_k = \frac{\alpha_k^2}{\sigma_k^2} = \frac{\bar{a}_k}{1 - \bar{a}_k}$$

$\bar{a}_k$ cumulative signal retention

SNR_k coefficient ratio, for unit-scale signal



DDPM §4; Diffusion Policy §3.3 · Ho, Jain & Abbeel (2020) · Chi et al. (2023/2024)

WORKED EXAMPLE

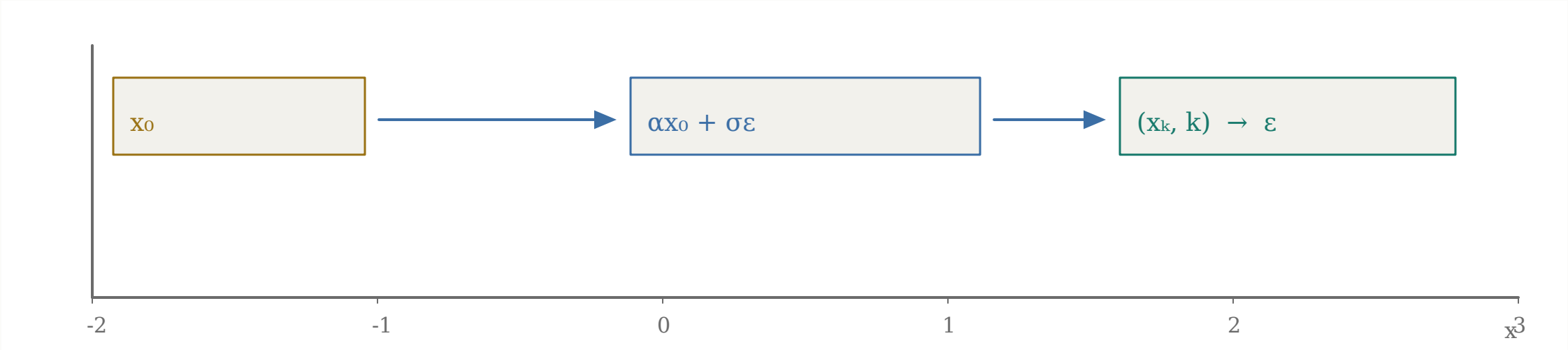
Forward diffusion

Worked example: one noisy training pair

WORKED EXAMPLE

$$x_0 = 2,\ \alpha_k = 0.8,\ \sigma_k = 0.6,\ \epsilon = -0.5$$

- x_0 clean scalar
- ϵ known training noise
- α_k, σ_k fixed coefficients



DDPM Algorithm 1; course example · Ho, Jain & Abbeel (2020)

YOUR TURN

Forward diffusion

Transfer: a two-component action

Compute x, then recover the sampled noise.

$$x_0 = \begin{bmatrix} 2 \\ -1 \end{bmatrix},\ \epsilon = \begin{bmatrix} -0.5 \\ 1 \end{bmatrix},\ \alpha = 0.8,\ \sigma = 0.6$$

$x_0, \epsilon \in \mathbb{R}^2$ given vectors

$x = \alpha x_0 + \sigma \epsilon$ noisy vector to find

clean vector

noise vector

noisy vector

same shape; scalar scaling and vector addition

DDPM Eq.4; course exercise · Ho, Jain & Abbeel (2020)

YOUR TURN

Forward diffusion

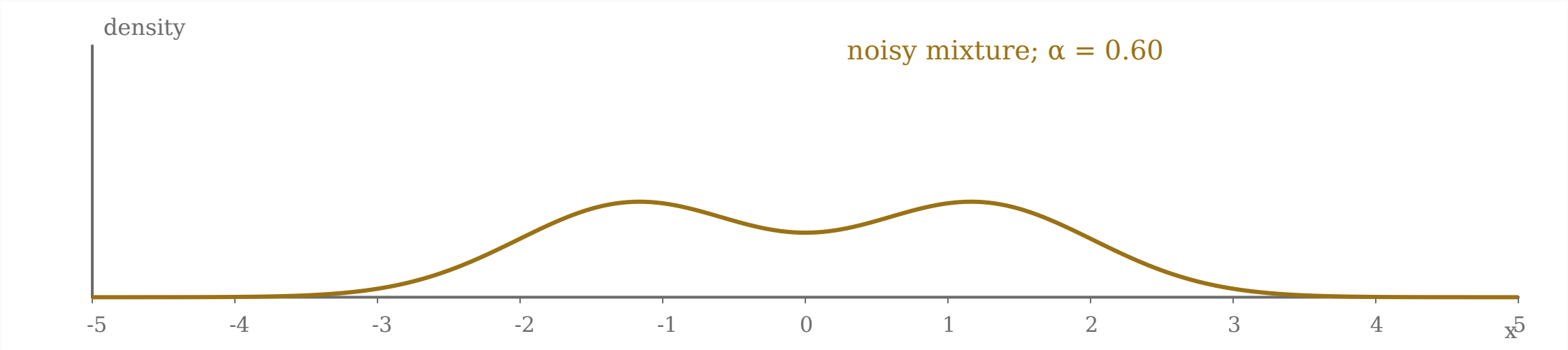
The marginal covariance also contains data spread

Find the mean and covariance of X_k.

$$X_k = \alpha_k X_0 + \sigma_k \epsilon, \quad X_0 \perp \epsilon$$

μ_0, Σ_0 clean-data mean and covariance

$\epsilon \sim \mathcal{N}(0, I)$ independent noise



DDPM Eq.4; DL §3.8 · Ho, Jain & Abbeel (2020) · Goodfellow et al. (2016)

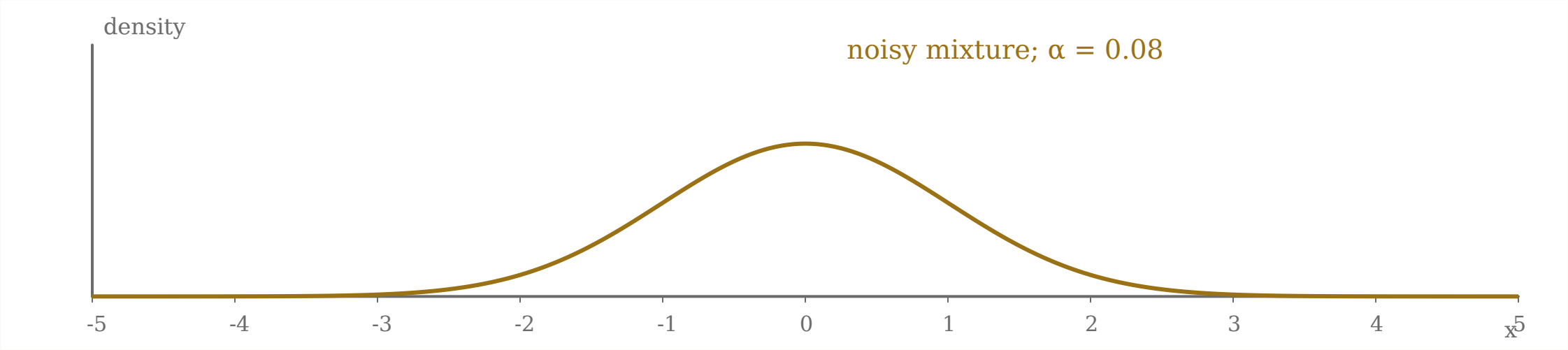
ASSUMPTION

Forward diffusion

The terminal Gaussian is a schedule approximation

$$\alpha_K \approx 0, \quad \sigma_K \approx 1 \quad \Longrightarrow \quad X_K \approx \mathcal{N}(0, I)$$

- K last diffusion index
- X_K marginal noisy-data vector



DDPM §4, terminal prior · Ho, Jain & Abbeel (2020)

YOUR TURN

Forward diffusion

Does this terminal schedule match the prior?

Can the terminal law equal N(0,1)? Check its moments.

$$\alpha_K = 0.8,\ \sigma_K = 0.6, \quad \mathbb{E}[X_0] = 5,\ \mathrm{Var}(X_0) = 1$$

X_0, X_K scalar clean and noisy data

$N(0,1)$ proposed generation prior

DDPM §4; course exercise · Ho, Jain & Abbeel (2020)

DEFINITION

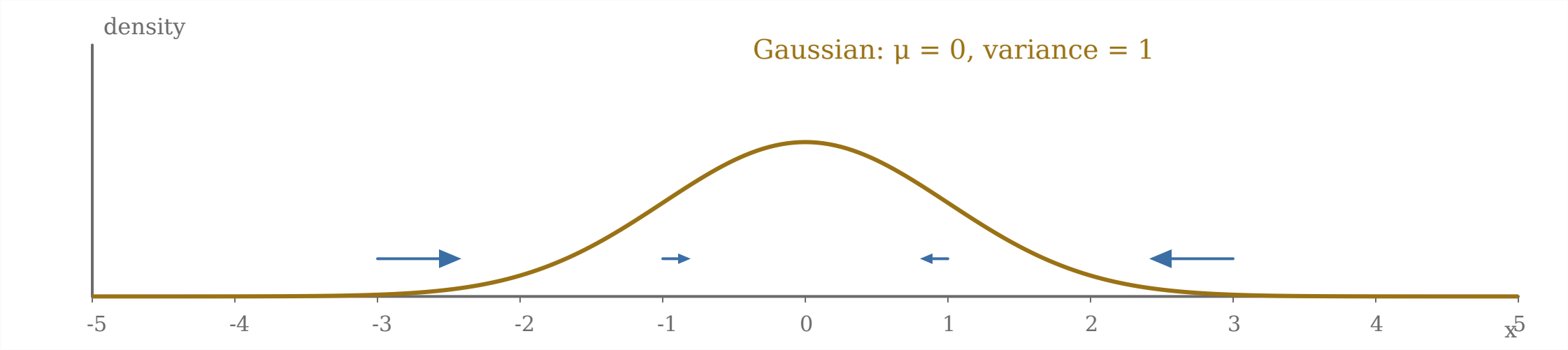
The score

A density supplies a local direction

$$s_k(x) = \nabla_x \log p_k(x)$$

p_k density at noise level k

$s_k(x) \in \mathbb{R}^d$ gradient with respect to x



Vincent (2011), §2; DDPM §3.2 · Ho, Jain & Abbeel (2020) · Vincent (2011)

STARTING RELATION

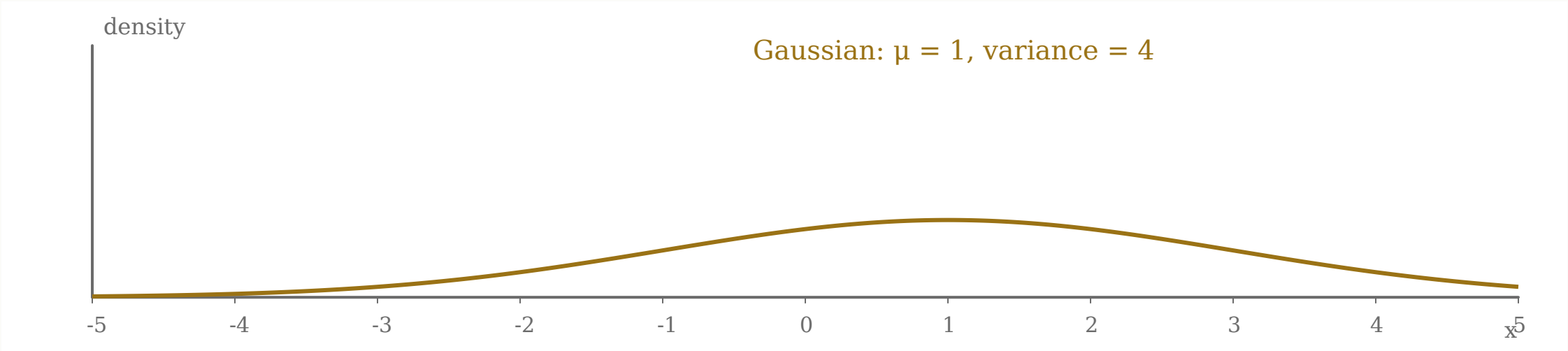
The score

Start with a Gaussian log density

$$\log p(x) = C - \frac{(x-\mu)^2}{2v}$$

$\mu \in \mathbb{R},\ v > 0$ fixed mean and variance

$C = -\frac{1}{2}\log(2\pi v)$ constant with respect to x



DL §3.9.3; course derivation · Goodfellow et al. (2016)

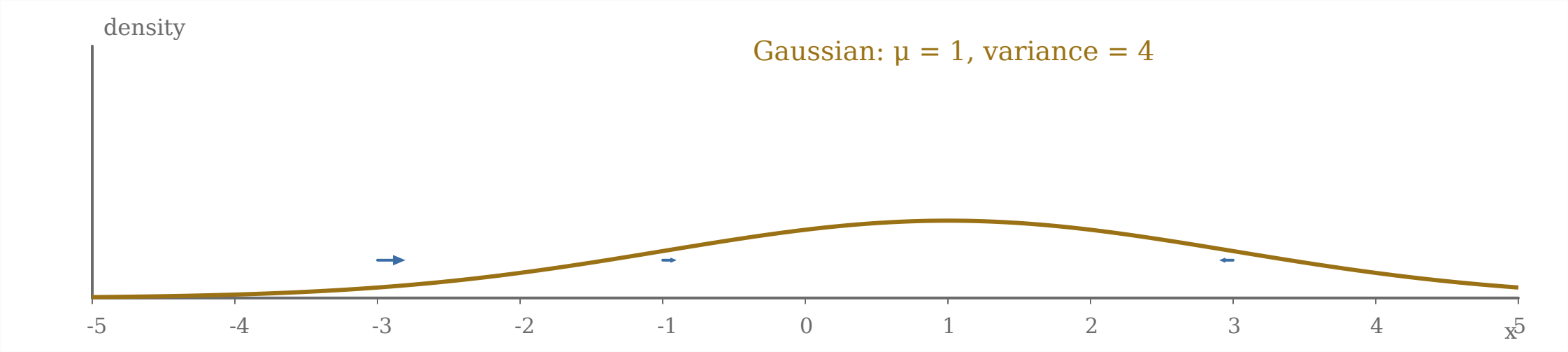
DERIVATION

The score

Differentiate the quadratic one step at a time

$$s(x) = -\frac{1}{2v}\,2(x-\mu) = -\frac{x-\mu}{v}$$

- μ, v fixed mean and variance
- $s(x)$ scalar score



DL §3.9.3; course derivation · Goodfellow et al. (2016)

The score

Differentiate the quadratic one step at a time

Differentiate the scalar Gaussian log density

mu and positive variance v are fixed; C is constant in x.

1. $\log p(x) = C - (x-\mu)^2/(2v)$.
2. $\partial_x C = 0$, and $\partial_x (x-\mu)^2 = 2(x-\mu)$.
3. $s(x) = -[2(x-\mu)]/(2v) = -(x-\mu)/v$.

DL §3.9.3; course derivation · Goodfellow et al. (2016)

GENERAL RULE

The score

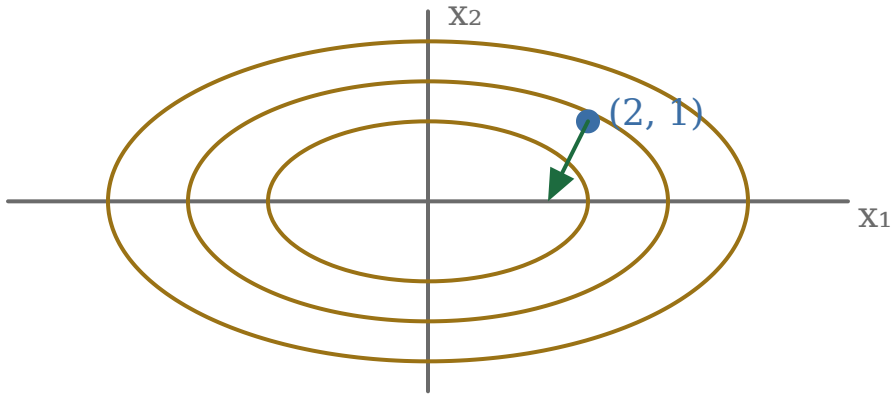
Covariance shapes the vector direction

$$s(x) = -\Sigma^{-1}(x - \mu)$$

$x, \mu \in \mathbb{R}^d$ state and mean

$\Sigma \succ 0$ fixed covariance

$\Sigma = \mathrm{diag}(4, 1)$



DL §3.9.3; Lecture 5 quadratic differential · Goodfellow et al. (2016)

The score

Covariance shapes the vector direction

Differentiate a symmetric quadratic form

r=x−mu is a column vector. Positive-definite covariance Sigma is fixed.

1. $\log p(x) = C - \frac{1}{2} r^T \Sigma^{-1} r.$
2. $d(r^T \Sigma^{-1} r) = dr^T \Sigma^{-1} r + r^T \Sigma^{-1} dr = 2 r^T \Sigma^{-1} dx.$
3. $d \log p(x) = (-\Sigma^{-1} r)^T dx$, so $s(x) = -\Sigma^{-1}(x - \mu).$

DL §3.9.3; Lecture 5 quadratic differential · Goodfellow et al. (2016)

YOUR TURN

The score

Compute an anisotropic score

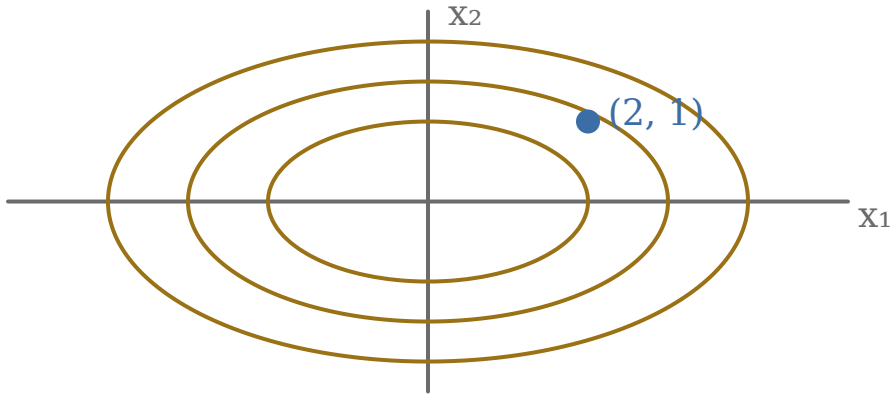
Compute s(x). Is its direction parallel to −x?

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}, x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Σ covariance, not standard deviations

$s(x) = -\Sigma^{-1}(x - \mu)$ score to find

Σ = diag(4, 1)



DL §3.9.3; course exercise · Goodfellow et al. (2016)

DERIVATION

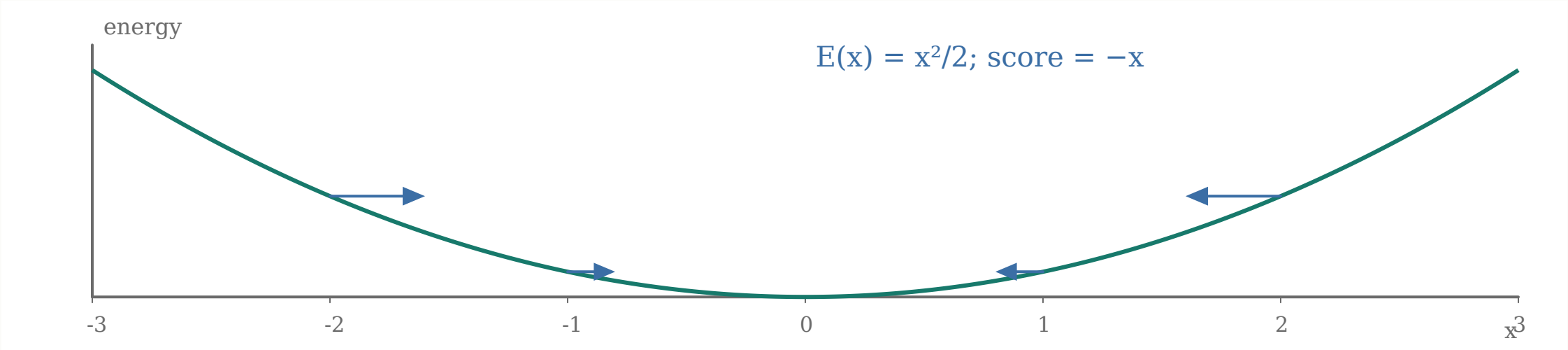
The score

An unknown normalizing constant disappears

$$p(x) = \frac{e^{-E(x)}}{Z} \quad \Longrightarrow \quad \nabla_x \log p(x) = -\nabla_x E(x)$$

$E(x)$ differentiable energy

Z finite positive constant independent of x



Vincent (2011), §2 · Vincent (2011)

DERIVATION

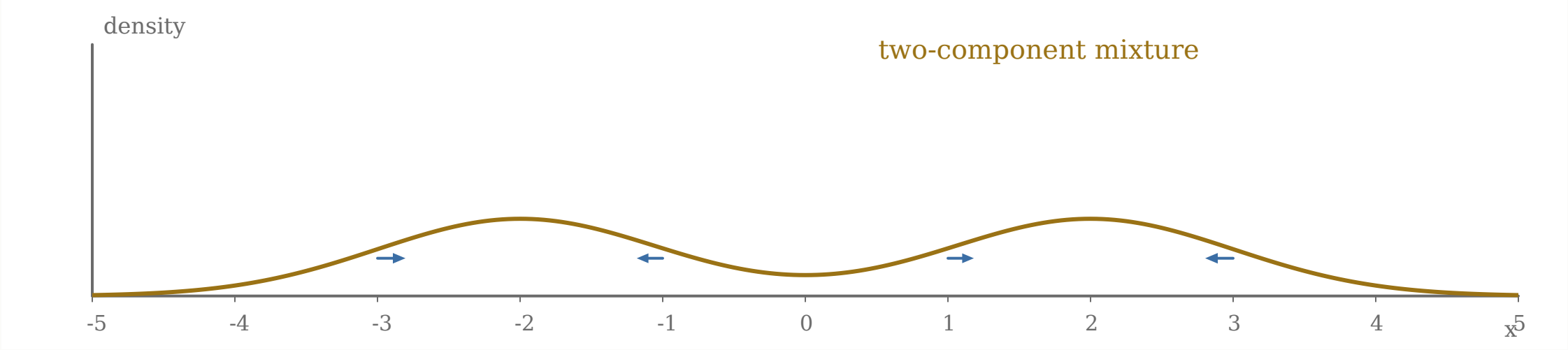
The score

A mixture score averages local component scores

$$s(x) = \sum_j r_j(x) s_j(x), \quad r_j(x) = \frac{w_j p_j(x)}{\sum_i w_i p_i(x)}$$

w_j, p_j weight and component density

$r_j(x)$ responsibility at x



DL §3.9.6; course derivation · Goodfellow et al. (2016)

The score

A mixture score averages local component scores

Differentiate the sum before taking responsibilities

The weights w_j are fixed and sum to one; component densities are differentiable.

1. $p(x) = \sum_j w_j p_j(x)$, so $\nabla p(x) = \sum_j w_j \nabla p_j(x)$.
2. $s(x) = \nabla p(x)/p(x) = \sum_j [w_j p_j(x)/p(x)] \, \nabla \log p_j(x)$.
3. Set $r_j(x) = w_j p_j(x)/p(x)$ to obtain $s(x) = \sum_j r_j(x) s_j(x)$.

DL §3.9.6; course derivation · Goodfellow et al. (2016)

YOUR TURN

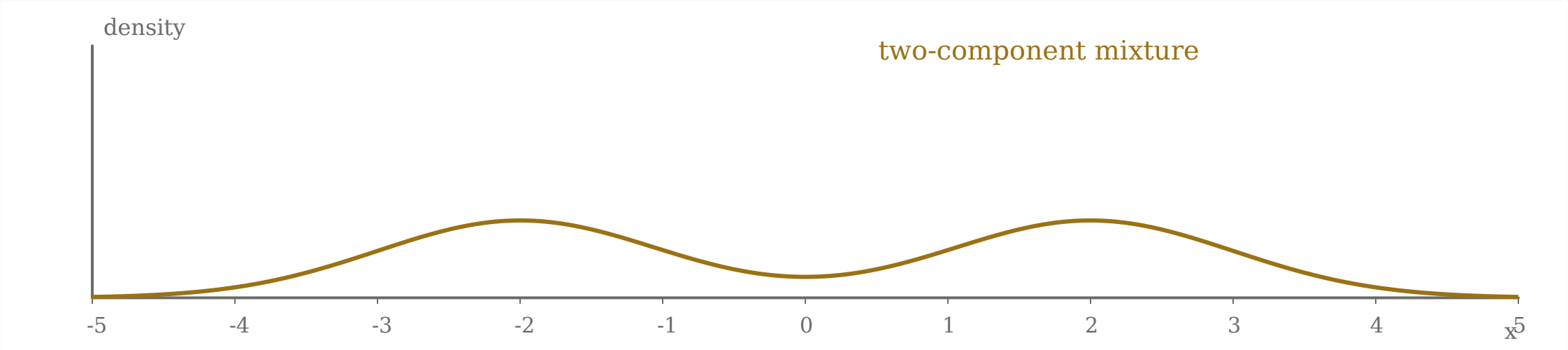
The score

What does a stationary point tell us?

Find s(0). Is x=0 a mode? Explain.

$$p(x) = \tfrac{1}{2}\mathcal{N}(x; -2, 1) + \tfrac{1}{2}\mathcal{N}(x; 2, 1)$$

- x scalar state
- $s(x)$ gradient of log mixture density



DL §3.9.6; course exercise · Goodfellow et al. (2016)

YOUR TURN

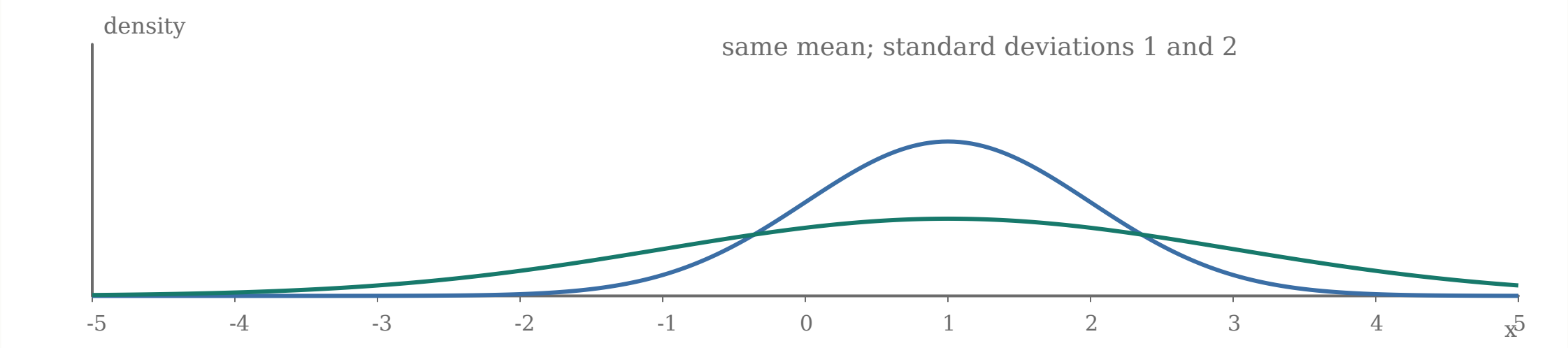
The score

Transfer: doubling standard deviation

Compare standard deviations 1 and 2. Compute both scores.

$$\mu = 1, \quad x = 3, \quad s(x) = -(x - \mu)/v$$

- v variance
- $\sqrt{v}$ standard deviation



DL §3.9.3; course exercise · Goodfellow et al. (2016)

SETUP

Learning to denoise

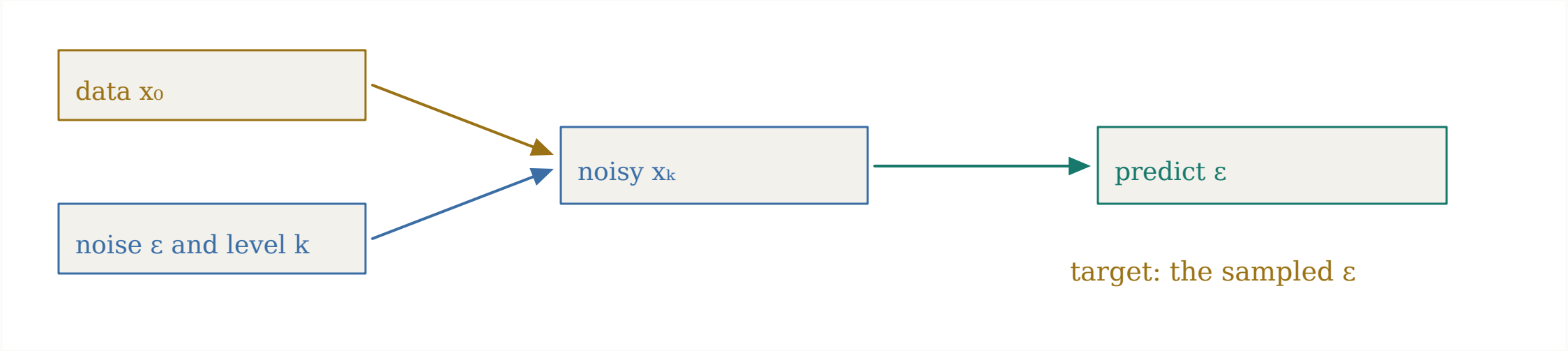
Known corruption gives a target for every example

$$(x_0, k, \epsilon) \longmapsto \big((x_k, k), \epsilon\big)$$

$x_k = \alpha_k x_0 + \sigma_k \epsilon$ network input

ϵ supervised target

k sampled noise level



DDPM Algorithm 1 · Ho, Jain & Abbeel (2020)

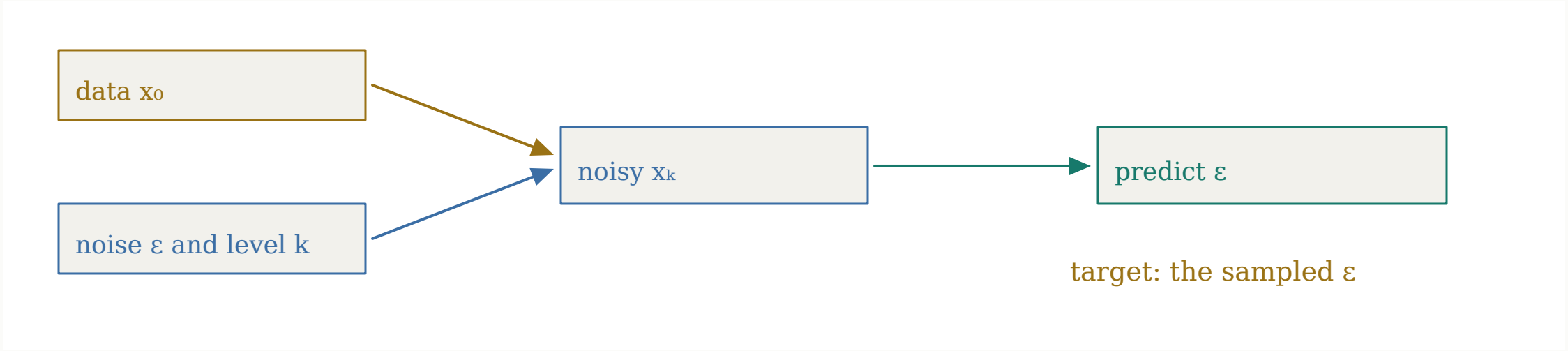
DEFINITION

Learning to denoise

The simple noise-prediction objective

$$L(\theta) = \mathbb{E}\|\epsilon_\theta(\alpha_k x_0 + \sigma_k \epsilon, k) - \epsilon\|^2$$

- θ trainable network parameters
- $\epsilon_\theta \in \mathbb{R}^d$ predicted noise
- $\mathbb{E}$ average over data, uniform k, and noise



DDPM §3.4, Eq.14 · Ho, Jain & Abbeel (2020)

DERIVATION

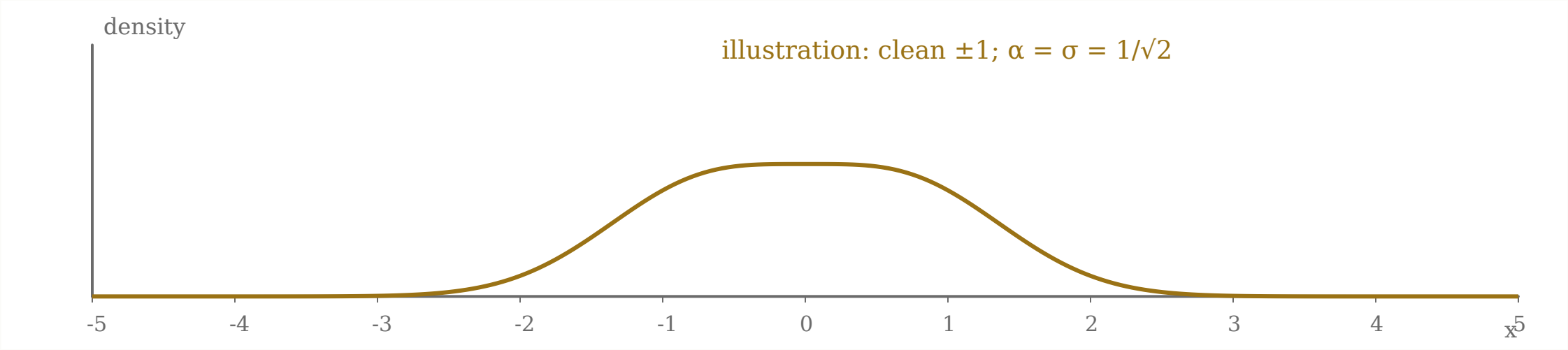
Learning to denoise

The optimal predictor is a conditional mean

$$\mathbb{E}[\|\epsilon - g\|^2 \mid x_k, k] = \mathbb{E}[\|\epsilon - m\|^2 \mid x_k, k] + \|g - m\|^2$$

$m = \mathbb{E}[\epsilon \mid x_k, k]$ conditional mean

$g \in \mathbb{R}^d$ chosen prediction



DDPM Eq.14; course least-squares derivation · Ho, Jain & Abbeel (2020)

Learning to denoise

The optimal predictor is a conditional mean

Separate conditional mean from unexplained noise

Condition on x_k and k. Let m=E[epsilon|x_k,k], with finite conditional second moment.

1. $\epsilon - g = (\epsilon - m) + (m - g).$
2. $\mathbb{E}[(\epsilon - m)^T (m - g) \mid x_k, k] = 0.$
3. $\mathbb{E}[\|\epsilon - g\|^2 \mid x_k, k] = \mathbb{E}[\|\epsilon - m\|^2 \mid x_k, k] + \|g - m\|^2.$
4. Only the last term depends on g, so $g^* = m.$

DDPM Eq.14; course least-squares derivation · Ho, Jain & Abbeel (2020)

DERIVATION

Learning to denoise

One clean example has an accessible conditional score

$$\nabla_x \log q_k(x \mid x_0) = -\frac{x - \alpha_k x_0}{\sigma_k^2} = -\frac{\epsilon}{\sigma_k}$$

$x = \alpha_k x_0 + \sigma_k \epsilon$ corrupted realization

x_0, k fixed; sigma_k > 0

Vincent (2011), §4; DDPM §3.2 · Ho, Jain & Abbeel (2020) · Vincent (2011)

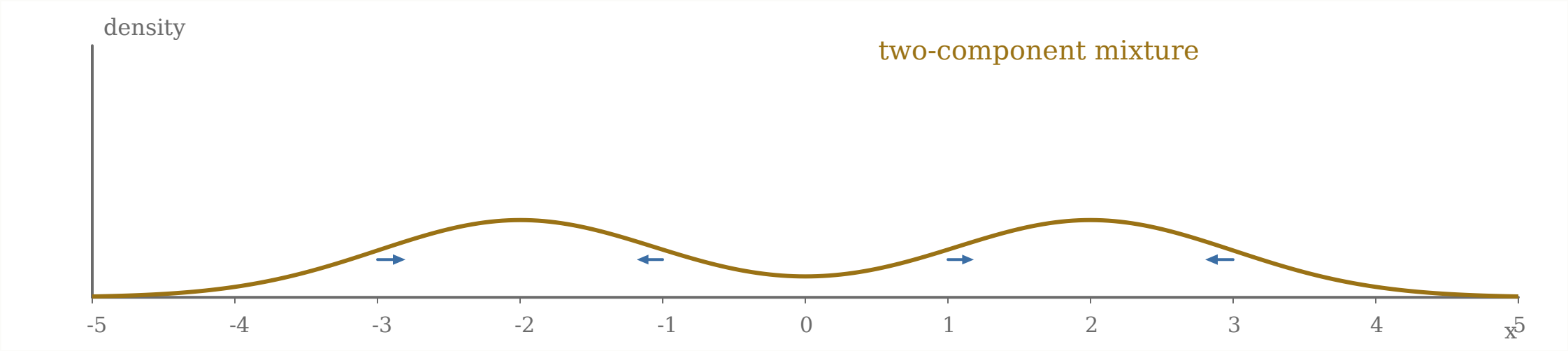
DERIVATION

Learning to denoise

Average conditional scores to obtain the score

$$\nabla_x \log p_k(x) = \int p(x_0 \mid x) \nabla_x \log q_k(x \mid x_0)\, dx_0$$

p_k	noisy marginal density; k fixed
$p(x_0 \mid x)$	posterior over clean data
Assumption	differentiate under the integral



Vincent (2011), §4 · Vincent (2011)

Learning to denoise

Average conditional scores to obtain the score

Differentiate the noisy marginal density

q_k(x|x0) is the Gaussian corruption density; p0 is the clean-data density; k is fixed.

1. $p_k(x) = \int p_0(x_0) q_k(x \mid x_0)\, dx_0.$
2. $\nabla_x p_k(x) = \int p_0(x_0) \nabla_x q_k(x \mid x_0)\, dx_0.$
3. The second equality assumes the derivative can pass under the integral.

Vincent (2011), §4 · Vincent (2011)

Learning to denoise

Average conditional scores to obtain the score

Divide by the marginal and apply Bayes rule

Assume p_k(x)>0. The posterior is p(x0|x)=p0(x0)q_k(x|x0)/p_k(x).

1. $\nabla_x \log p_k(x) = \int \frac{p_0(x_0) q_k(x|x_0)}{p_k(x)} \frac{\nabla_x q_k(x|x_0)}{q_k(x|x_0)} \, dx_0.$
2. $s_k(x) = \mathbb{E}[\nabla_x \log q_k(x \mid X_0) \mid X_k = x].$
3. For discrete clean support, replace the integrals by sums.

Vincent (2011), §4 · Vincent (2011)

YOUR TURN

Learning to denoise

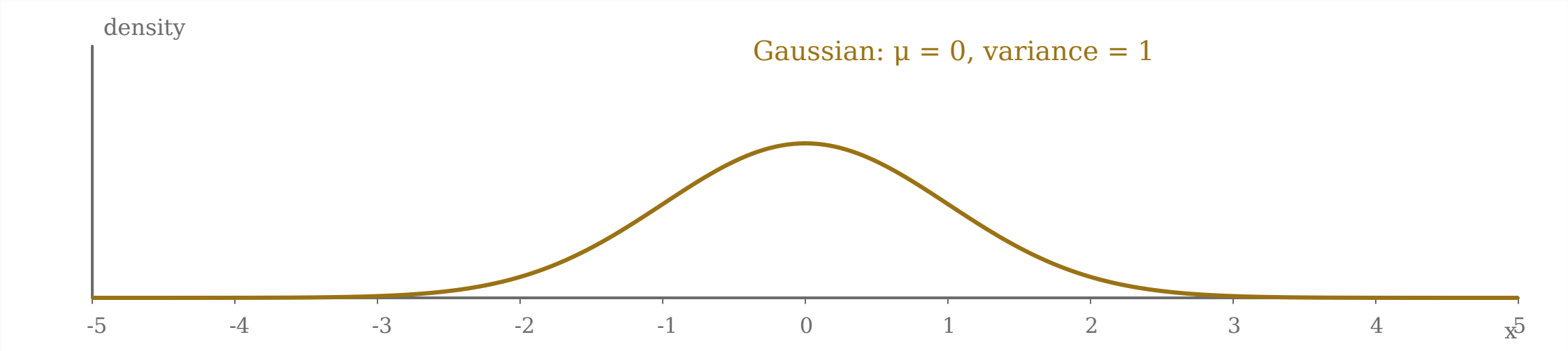
Noise prediction gives a score estimate

Explain the minus sign and the factor 1/σ_k.

$$s_k(x) = -\frac{\mathbb{E}[\epsilon \mid x_k = x, k]}{\sigma_k} \approx -\frac{\epsilon_\theta(x, k)}{\sigma_k}$$

s_k marginal noisy-data score

$\sigma_k > 0$ noise standard deviation



DDPM §3.2; Vincent (2011) §4 · Ho, Jain & Abbeel (2020) · Vincent (2011)

WORKED EXAMPLE

Learning to denoise

Worked example: noise prediction becomes a score

WORKED EXAMPLE

$$\sigma_k = 0.6, \quad \epsilon_\theta(x, k) = \begin{bmatrix} -0.3 \\ 0.6 \end{bmatrix}$$

ϵ_θ network prediction

$s_\theta = -\epsilon_\theta / \sigma_k$ score estimate

noise prediction

divide by $-\sigma$

score estimate

a density direction; the sampler still needs an update rule

DDPM §3.2; course example · Ho, Jain & Abbeel (2020)

YOUR TURN

Learning to denoise

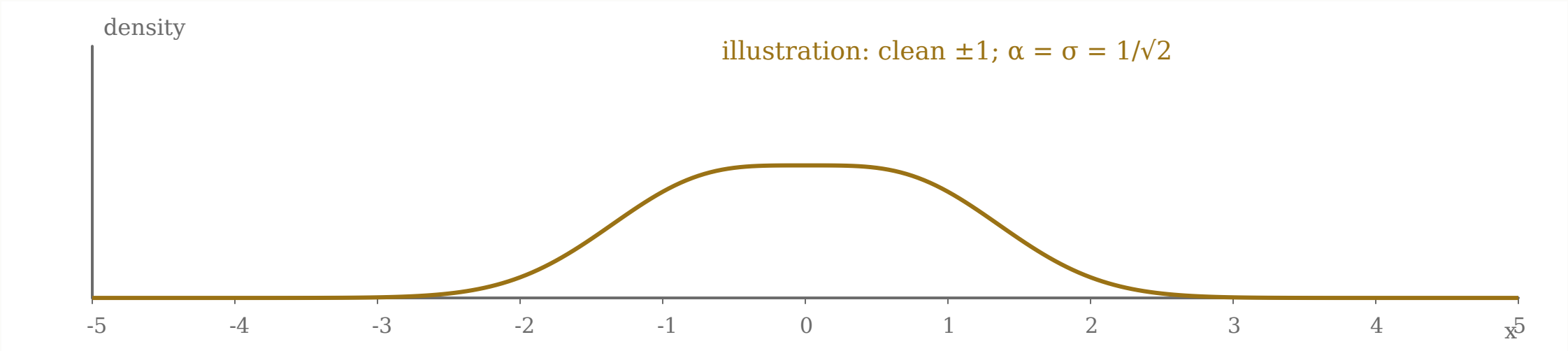
What can a zero noise prediction mean?

Find the noise for each clean hypothesis, then its mean.

$$X_0 \in \{-1, 1\}, \quad \alpha = \sigma = 1/\sqrt{2}, \quad X_k = 0$$

X_0 two equally likely clean values

ϵ noise conditioned on X_k=0



DDPM Eq.14; course exercise · Ho, Jain & Abbeel (2020)

RESULT

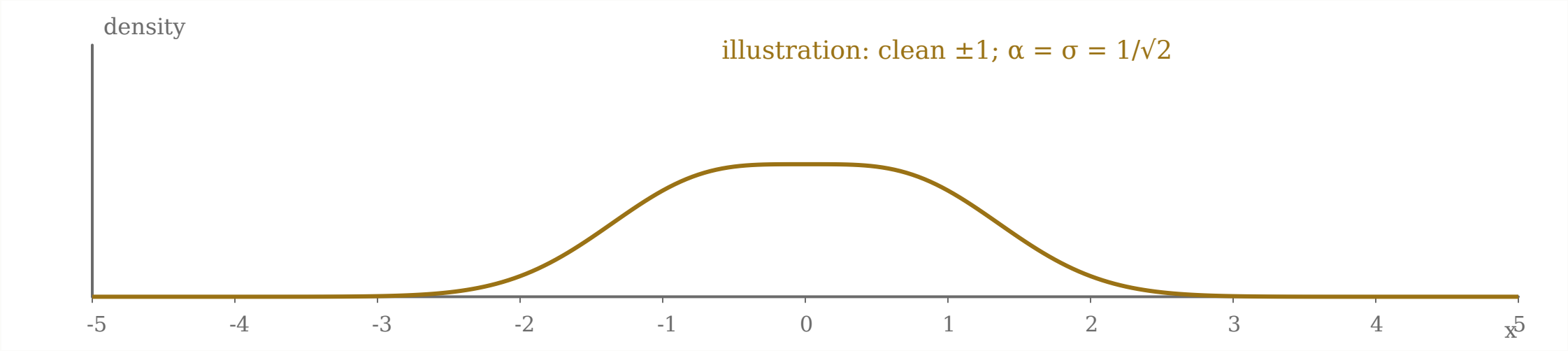
Learning to denoise

A denoising identity gives the posterior mean

$$\mathbb{E}[X_0 \mid x_k = x] = \frac{x + \sigma_k^2 s_k(x)}{\alpha_k}$$

$\alpha_k > 0, \sigma_k > 0$ fixed coefficients

s_k true marginal score



Efron (2011), Eq.1.4; scaled Tweedie identity · Efron (2011)

Learning to denoise

A denoising identity gives the posterior mean

Invert the known corruption algebraically

X_k=alpha X0+sigma epsilon, with positive alpha and sigma. Condition on X_k=x.

1. $X_0 = (x - \sigma\epsilon)/\alpha.$
2. $\mathbb{E}[X_0 \mid X_k = x] = (x - \sigma\mathbb{E}[\epsilon \mid X_k = x])/\alpha.$
3. The observed x is fixed inside this conditional expectation.

Efron (2011), Eq.1.4; scaled Tweedie identity · Efron (2011)

Learning to denoise

A denoising identity gives the posterior mean

Substitute the noise-score relation

The exact marginal score obeys E[epsilon|X_k=x]=−sigma s_k(x).

1. $\mathbb{E}[X_0 \mid x] = (x - \sigma[-\sigma s_k(x)])/\alpha.$
2. $\mathbb{E}[X_0 \mid x] = (x + \sigma^2 s_k(x))/\alpha.$
3. This is a posterior mean, not a random sample or a guaranteed mode.

Efron (2011), Eq.1.4; scaled Tweedie identity · Efron (2011)

WORKED EXAMPLE

Learning to denoise

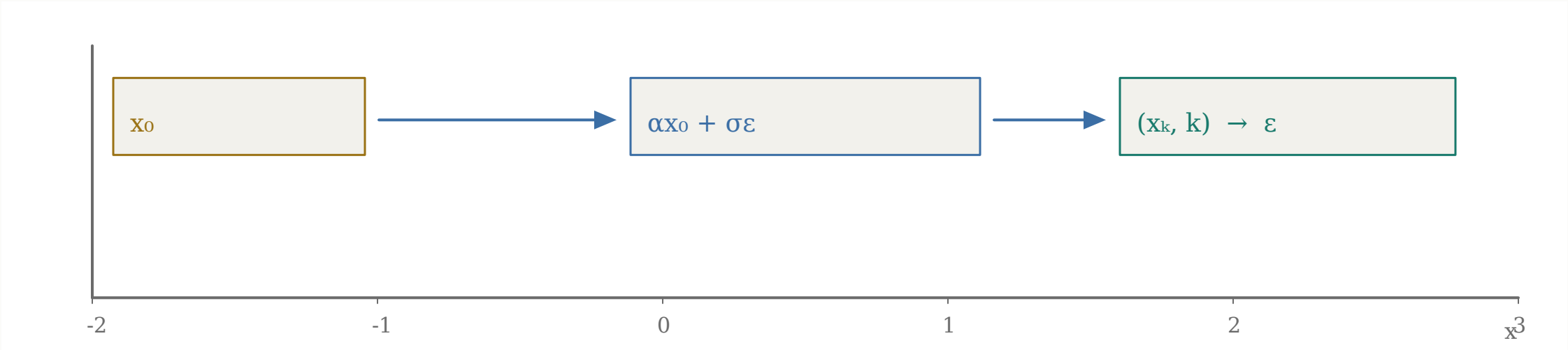
Worked example: two equivalent denoising routes

WORKED EXAMPLE

$$x = 1.3, \quad \alpha = 0.8, \quad \sigma = 0.6, \quad \epsilon_\theta = -0.5$$

$\hat{x}_0$ network-based clean estimate

$s_\theta = -\epsilon_\theta / \sigma$ score estimate



DDPM Eq.11; Efron Eq.1.4; course example · Ho, Jain & Abbeel (2020) · Efron (2011)

YOUR TURN

Learning to denoise

How does predictor error reach the clean estimate?

Find $\Delta\hat{x}_0$. Compare $\alpha=0.8$ and $\alpha=0.1$, with $\sigma^2=1-\alpha^2$.

$$\hat{x}_0 = \frac{x - \sigma\hat{\epsilon}}{\alpha}, \quad \hat{\epsilon} = \epsilon^* + \delta$$

- ϵ^* optimal mean-noise predictor
- δ prediction error; alpha > 0

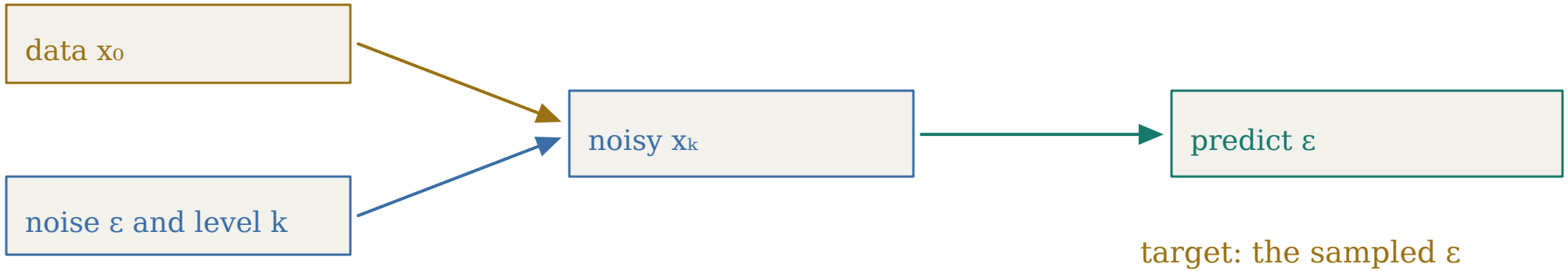
DDPM Eq.11; course exercise · Ho, Jain & Abbeel (2020)

ALGORITHM

Learning to denoise

Training updates weights, not the noisy input

Sample x_0, k, ε → form x_k → predict ε → average squared error → update θ.



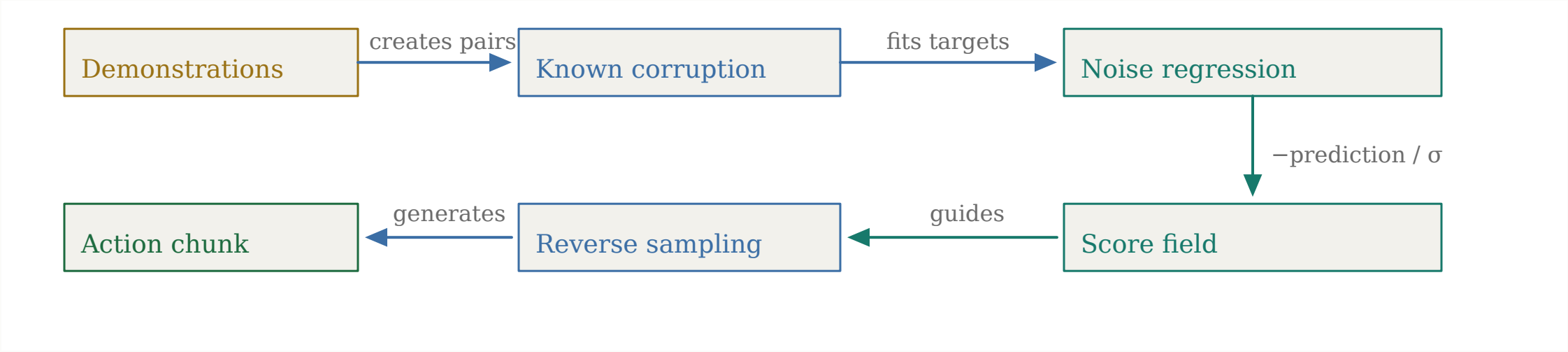
DDPM Algorithm 1 · Ho, Jain & Abbeel (2020)

YOUR TURN

Learning to denoise

Checkpoint: trace an input to a score

Data → known corruption → noise regression → score
Which quantity is sampled, observed, and learned?



DDPM Algorithms 1–2; Vincent (2011) §4 · Ho, Jain & Abbeel (2020) · Vincent (2011)

SETUP

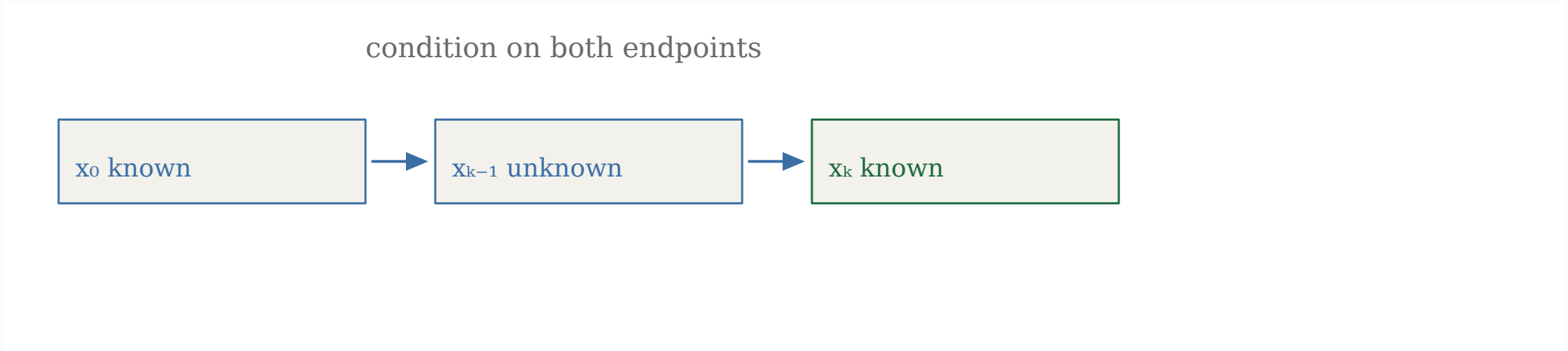
Reverse sampling

The reverse step needs a conditional distribution

$$q(x_{k-1} \mid x_k, x_0) \propto q(x_k \mid x_{k-1}) q(x_{k-1} \mid x_0)$$

x_0, x_k temporarily known endpoints

x_{k-1} unknown intermediate state; k > 1



DDPM §2, Eqs.6–7 · Ho, Jain & Abbeel (2020)

DERIVATION

Reverse sampling

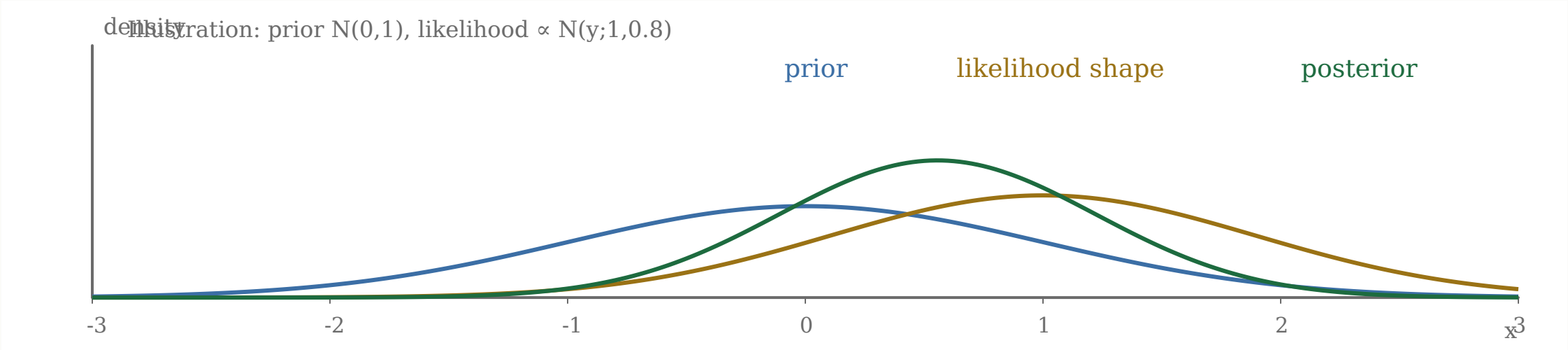
The Gaussian product adds two quadratic penalties

$$-2\log q(y \mid x_k, x_0) = \frac{\|y - \alpha_{k-1}x_0\|^2}{\sigma_{k-1}^2} + \frac{\|x_k - \sqrt{a_k}y\|^2}{\beta_k} + C$$

$y = x_{k-1}$ unknown vector; k > 1

$a_k = 1 - \beta_k$ one-step retention

C constant in y



DDPM Appendix A, posterior derivation · Ho, Jain & Abbeel (2020)

DERIVATION

Reverse sampling

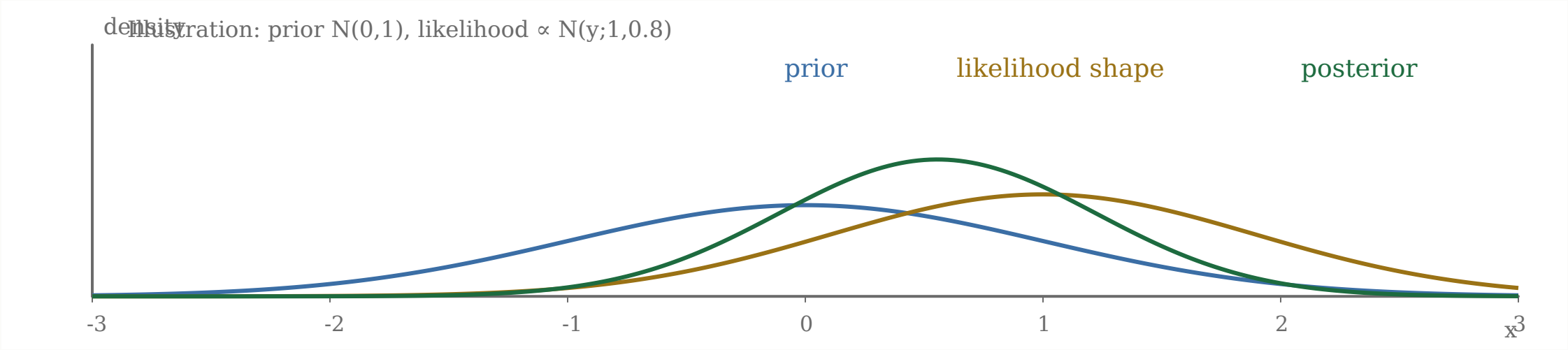
Complete the square to identify mean and variance

$$\Lambda = \frac{1}{\sigma_{k-1}^2} + \frac{a_k}{\beta_k}, \quad h = \frac{\alpha_{k-1} x_0}{\sigma_{k-1}^2} + \frac{\sqrt{a_k} x_k}{\beta_k}$$

ΛI posterior precision

h linear coefficient vector

$\tilde{\beta} = 1/\Lambda,\ \tilde{\mu} = h/\Lambda$ variance and mean



DDPM Eqs.6–7; Appendix A · Ho, Jain & Abbeel (2020)

Reverse sampling

Complete the square to identify mean and variance

Expand and collect terms in the unknown state

y=x_{k−1}; both x0 and x_k are fixed. Previous variance sigma² is positive.

1. $\Lambda = 1/\sigma_{k-1}^2 + a_k/\beta_k.$
2. $h = \alpha_{k-1} x_0/\sigma_{k-1}^2 + \sqrt{a_k} x_k/\beta_k.$
3. $-2\log q(y \mid x_k, x_0) = \Lambda \|y\|^2 - 2h^T y + C.$

DDPM Eqs.6–7; Appendix A · Ho, Jain & Abbeel (2020)

Reverse sampling

Complete the square to identify mean and variance

Complete the square and read the Gaussian

Lambda is a positive scalar precision; h is a vector.

1. $\Lambda\|y\|^2 - 2h^T y + C = \Lambda\|y - h/\Lambda\|^2 + C'.$
2. $\tilde{\mu} = h/\Lambda,\ \tilde{\beta} = 1/\Lambda.$
3. $q(y \mid x_k, x_0) = \mathcal{N}(y; \tilde{\mu}, \tilde{\beta} I).$

DDPM Eqs.6–7; Appendix A · Ho, Jain & Abbeel (2020)

RESULT

Reverse sampling

The posterior coefficients have a closed form

$$\tilde{\mu}_k = c_0 x_0 + c_k x_k, \quad \tilde{\beta}_k = \frac{\beta_k(1-\bar{a}_{k-1})}{1-\bar{a}_k}$$

$c_0 = \beta_k \alpha_{k-1}/(1-\bar{a}_k)$ clean coefficient

$c_k = \sqrt{a_k}(1-\bar{a}_{k-1})/(1-\bar{a}_k)$ noisy coefficient

condition on both endpoints



DDPM Eq.7 · Ho, Jain & Abbeel (2020)

Reverse sampling

The posterior coefficients have a closed form

Simplify the precision denominator

a_k=1−beta_k; bar-a_k=a_k bar-a_{k−1}.

1. $\Lambda = \frac{\beta_k + a_k(1-\bar{a}_{k-1})}{\beta_k(1-\bar{a}_{k-1})}$.
2. $\beta_k + a_k(1-\bar{a}_{k-1}) = 1-\bar{a}_k$.
3. $\tilde{\beta}_k = 1/\Lambda = \beta_k(1-\bar{a}_{k-1})/(1-\bar{a}_k)$.
4. $\tilde{\mu}_k = \tilde{\beta}_k h = c_0 x_0 + c_k x_k$.

DDPM Eq.7 · Ho, Jain & Abbeel (2020)

DERIVATION

Reverse sampling

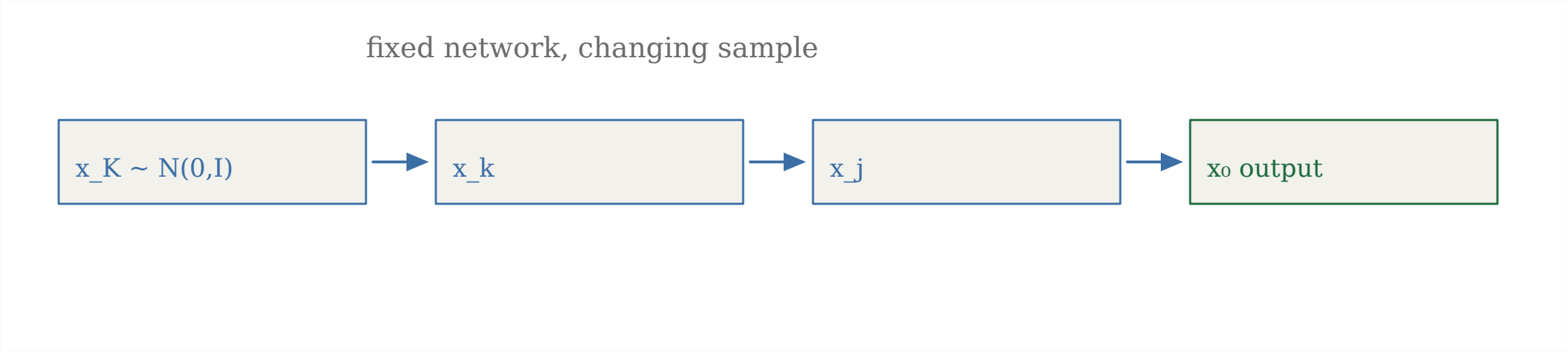
Replace clean data by the learned noise estimate

$$\mu_\theta(x_k, k) = \frac{1}{\sqrt{a_k}} \left(x_k - \frac{\beta_k}{\sigma_k} \epsilon_\theta(x_k, k) \right)$$

$a_k = 1 - \beta_k$ one-step retention

$\sigma_k = \sqrt{1 - \bar{a}_k}$ cumulative noise scale

μ_θ learned reverse mean



DDPM §3.2, Eq.11 · Ho, Jain & Abbeel (2020)

Reverse sampling

Replace clean data by the learned noise estimate

Insert the clean estimate into the known-clean mean

alpha_k=sqrt(a_k) alpha_{k−1}; sigma_k²=1−bar-a_k.

1. $\hat{x}_0 = (x_k - \sigma_k \hat{\epsilon})/\alpha_k.$
2. $\mu_\theta = \frac{\beta_k \alpha_{k-1}}{\sigma_k^2}\hat{x}_0 + \frac{\sqrt{a_k}(1-\bar{a}_{k-1})}{\sigma_k^2}x_k.$
3. The noise coefficient becomes $-\beta_k/(\sqrt{a_k}\sigma_k)$.

DDPM §3.2, Eq.11 · Ho, Jain & Abbeel (2020)

Reverse sampling

Replace clean data by the learned noise estimate

Simplify the remaining state coefficient

beta_k+a_k(1−bar-a_{k−1})=sigma_k².

1. Coefficient on x_k: $[\beta_k + a_k(1-\bar{a}_{k-1})]/(\sqrt{a_k}\sigma_k^2) = 1/\sqrt{a_k}$.
2. $\mu_\theta = (x_k - \beta_k\hat{\epsilon}/\sigma_k)/\sqrt{a_k}$.
3. With optimal mean-noise prediction this gives the reverse conditional mean; a fixed-variance Gaussian still approximates its full law.

DDPM §3.2, Eq.11 · Ho, Jain & Abbeel (2020)

WORKED EXAMPLE

Reverse sampling

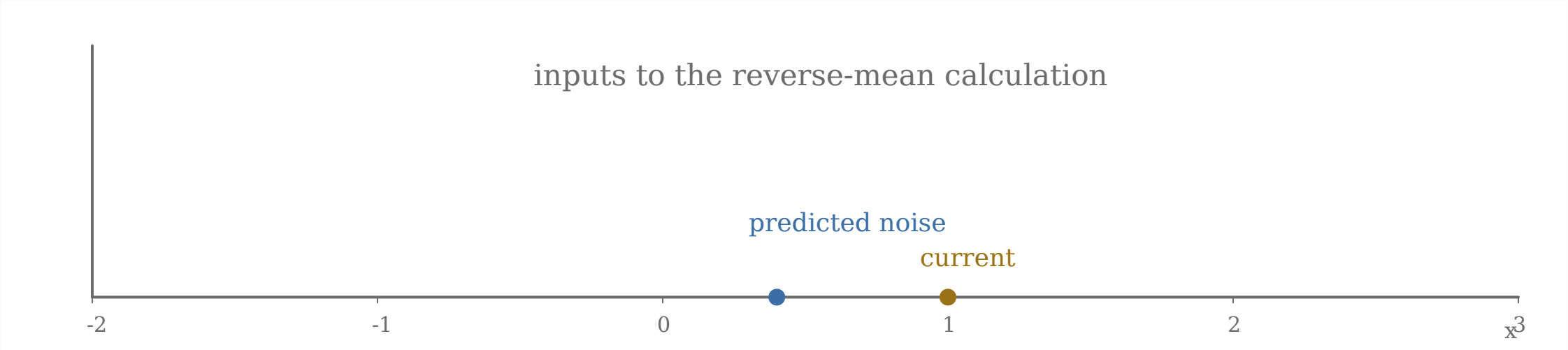
Worked example: calculate one reverse mean

WORKED EXAMPLE

$$\beta_k = 0.36, \quad a_k = 0.64, \quad \sigma_k = 0.8, \quad x_k = 1, \quad \hat{\epsilon} = 0.4$$

$\hat{\epsilon}$ network prediction

μ_θ reverse mean to compute



DDPM Eq.11; course example · Ho, Jain & Abbeel (2020)

YOUR TURN

Reverse sampling

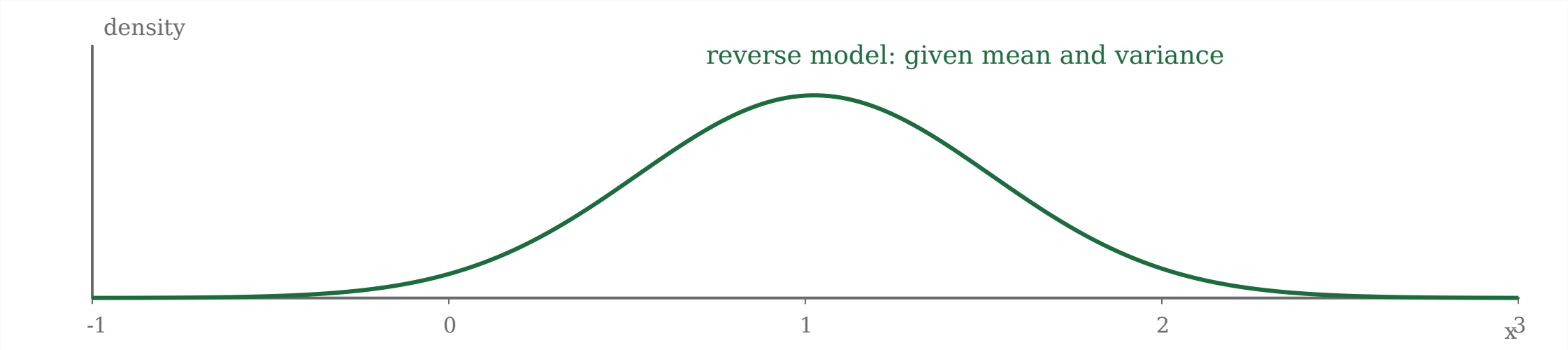
A reverse draw uses a standard deviation

Compute one draw and the variance across repeated draws.

$$x_{k-1} = \mu_\theta + \sqrt{\tilde{\beta}_k}\, z, \quad z \sim \mathcal{N}(0, I)$$

$\mu_\theta = 1.025$ reverse mean from the previous example

$\tilde{\beta}_k = 0.24609375,\ z = -1$ derived posterior variance; fresh draw



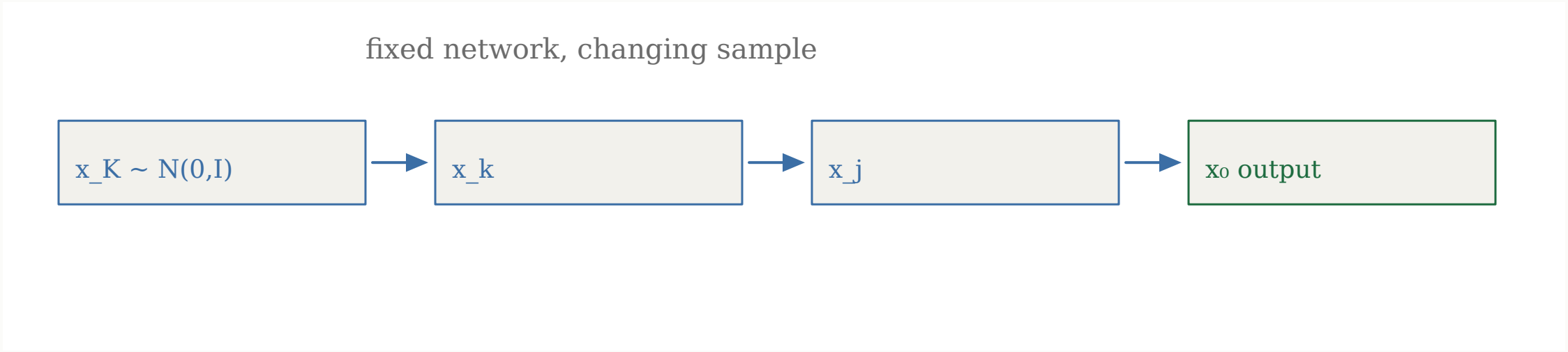
DDPM Algorithm 2; course exercise · Ho, Jain & Abbeel (2020)

ALGORITHM

Reverse sampling

DDPM generation runs from noise to data

Draw x_K ~ N(0,I). For k=K,…,1: predict noise; compute μθ; draw the next state. At k=1, return the continuous clean endpoint without fresh noise.



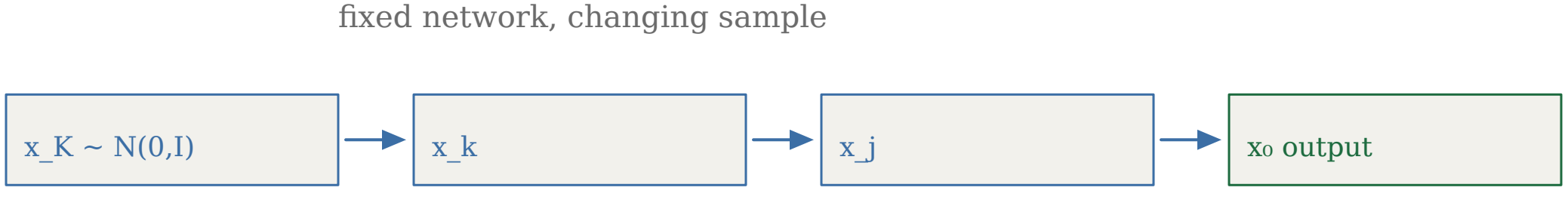
DDPM Algorithm 2; continuous-action adaptation · Ho, Jain & Abbeel (2020)

YOUR TURN

Reverse sampling

Does a mean predictor force identical samples?

Same observation and trained weights.
Different initial Gaussian draws.
Must the final samples be identical?

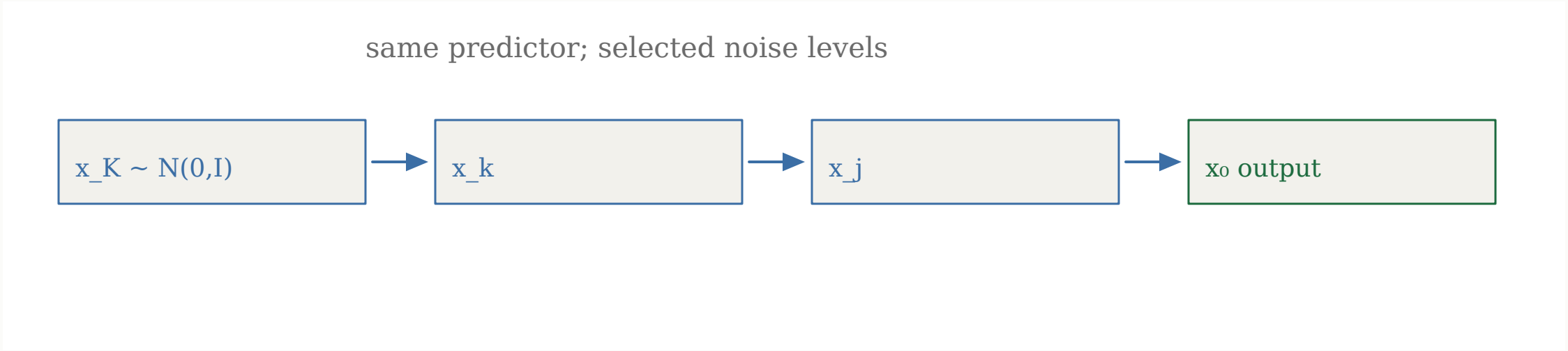


DDPM Algorithm 2; course exercise · Ho, Jain & Abbeel (2020)

TRANSITION

Reverse sampling

DDIM changes the path, using the same predictor



DDIM §§3–4 · Song, Meng & Ermon (2021)

GENERAL RULE

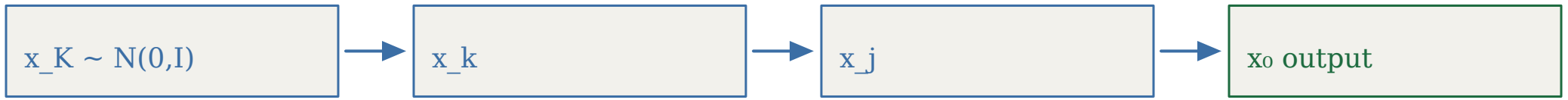
Reverse sampling

A deterministic DDIM step keeps the predicted direction

$$\hat{x}_0 = \frac{x_k - \sigma_k \hat{\epsilon}}{\alpha_k}, \qquad x_j = \alpha_j \hat{x}_0 + \sigma_j \hat{\epsilon}$$

- $j < k$ chosen less noisy level
- $\hat{\epsilon} = \epsilon_\theta(x_k, k)$ current prediction
- $\alpha_i^2 + \sigma_i^2 = 1$ schedule; alpha_k > 0

same predictor; selected noise levels



DDIM §4.1, Eq.12 with eta=0 · Song, Meng & Ermon (2021)

WORKED EXAMPLE

Reverse sampling

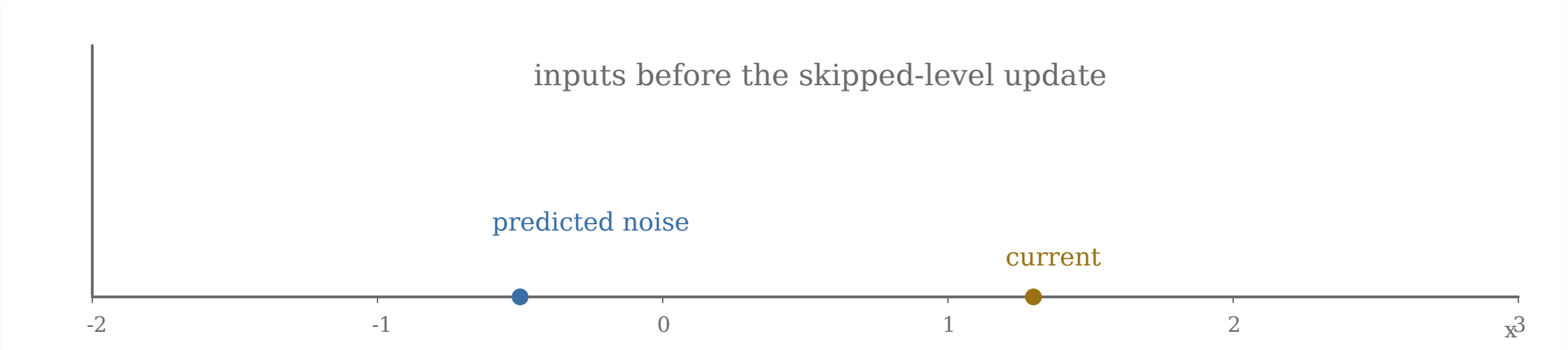
Worked example: skip to a cleaner level

WORKED EXAMPLE

$$x_k = 1.3,\ \hat{\epsilon} = -0.5,\ (\alpha_k, \sigma_k) = (0.8, 0.6),\ (\alpha_j, \sigma_j) = (\sqrt{0.91}, 0.3)$$

$j < k$ selected next level

$\hat{x}_0$ clean estimate



DDIM Eq.12; course example · Song, Meng & Ermon (2021)

YOUR TURN

Reverse sampling

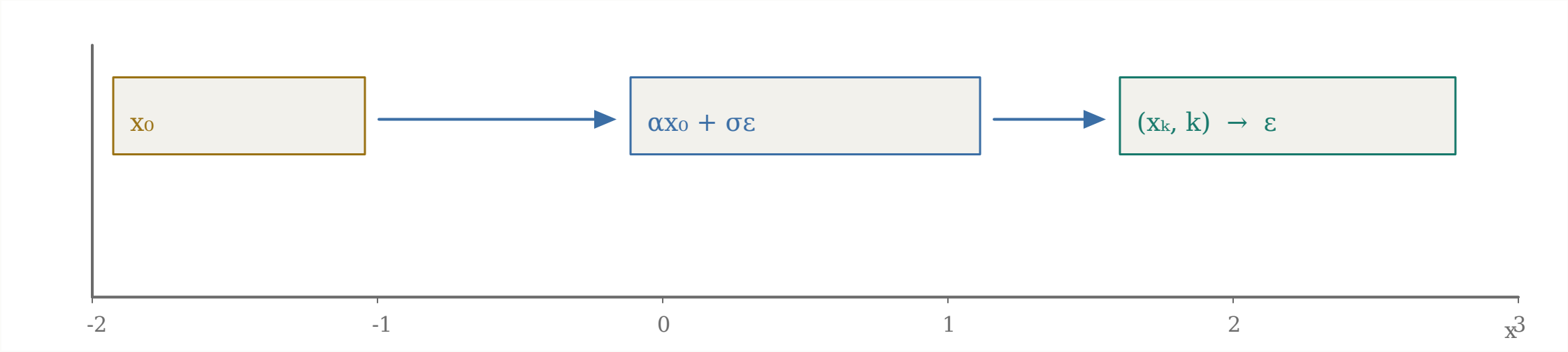
Transfer: check deterministic sampling at the endpoint

Compute the deterministic clean-endpoint output. Is the whole generated distribution deterministic?

$$x_k = 0.9, \quad \alpha_k = 0.8, \quad \sigma_k = 0.6, \quad \hat{\epsilon} = 0.5$$

$\alpha_0 = 1, \sigma_0 = 0$ clean endpoint

x_0^{out} deterministic output to find



DDIM Eq.12; course exercise · Song, Meng & Ermon (2021)

GENERAL RULE

Reverse sampling

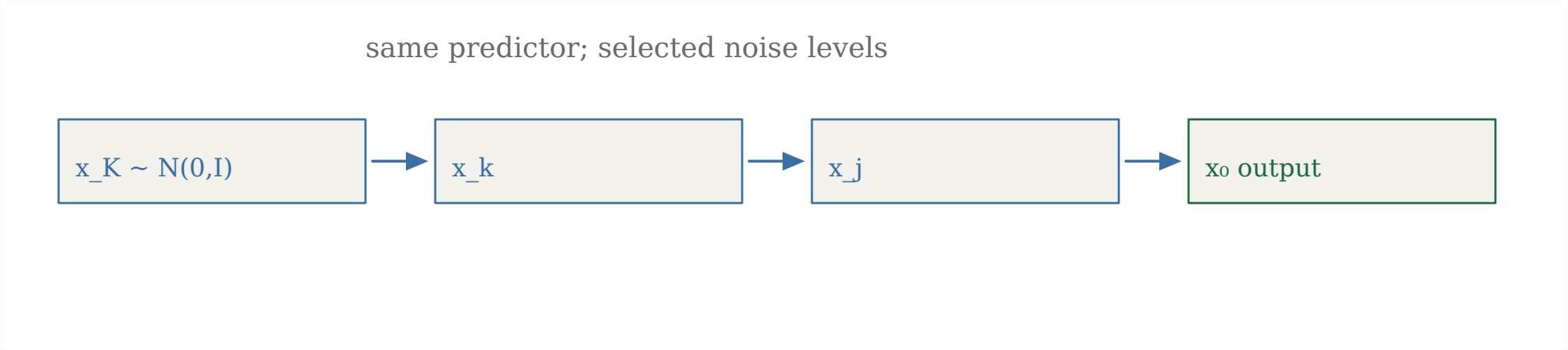
A generalized step allocates fresh randomness

$$x_j = \alpha_j \hat{x}_0 + \sqrt{\sigma_j^2 - \rho_{k,j}^2}\,\hat{\epsilon} + \rho_{k,j} z$$

$z \sim \mathcal{N}(0, I)$ fresh independent draw

$\rho_{k,j}$ chosen noise standard deviation

$\hat{x}_0, \hat{\epsilon}$ current clean/noise estimates



DDIM Eq.12 · Song, Meng & Ermon (2021)

PARAMETERIZATION

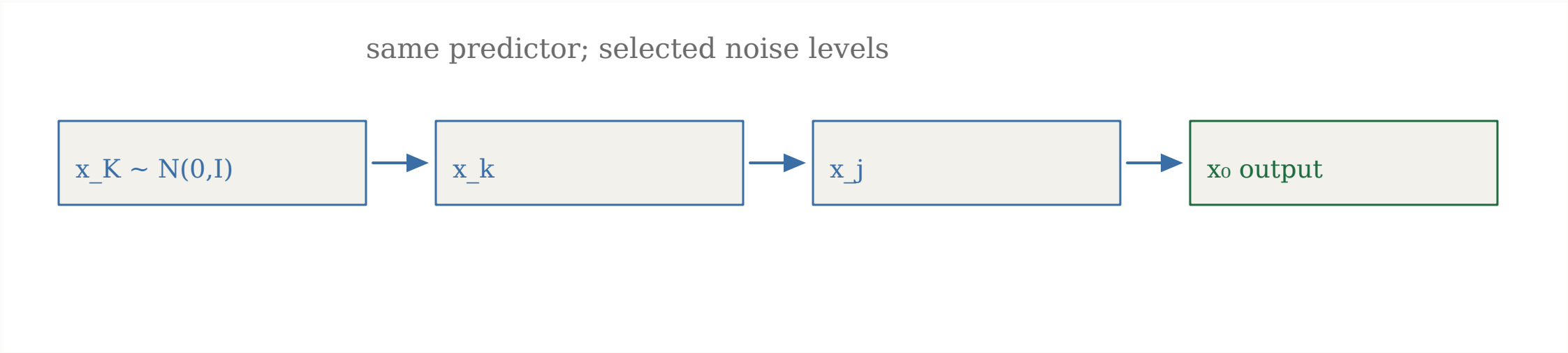
Reverse sampling

The eta parameter selects the stochasticity

$$\rho_{k,j} = \eta \sqrt{\frac{1-\alpha_j^2}{1-\alpha_k^2}\left(1-\frac{\alpha_k^2}{\alpha_j^2}\right)}$$

$j < k,\ 0 \le \eta \le 1$ selected levels and parameter

$\alpha_j > \alpha_k > 0$ cumulative signal coefficients



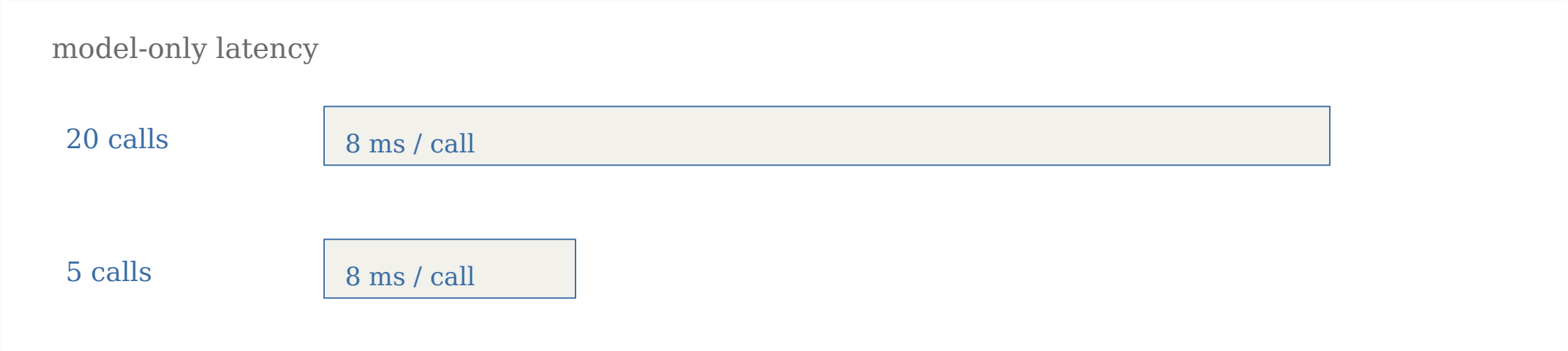
DDIM §4.1 and Appendix C.1 · Song, Meng & Ermon (2021)

YOUR TURN

Reverse sampling

A step budget trades computation for approximation

Toy measured cost: 8 ms per network evaluation.
Compare 20 evaluations with 5.
Which claims require an experiment?



Diffusion Policy §3.4; course exercise · Chi et al. (2023/2024)

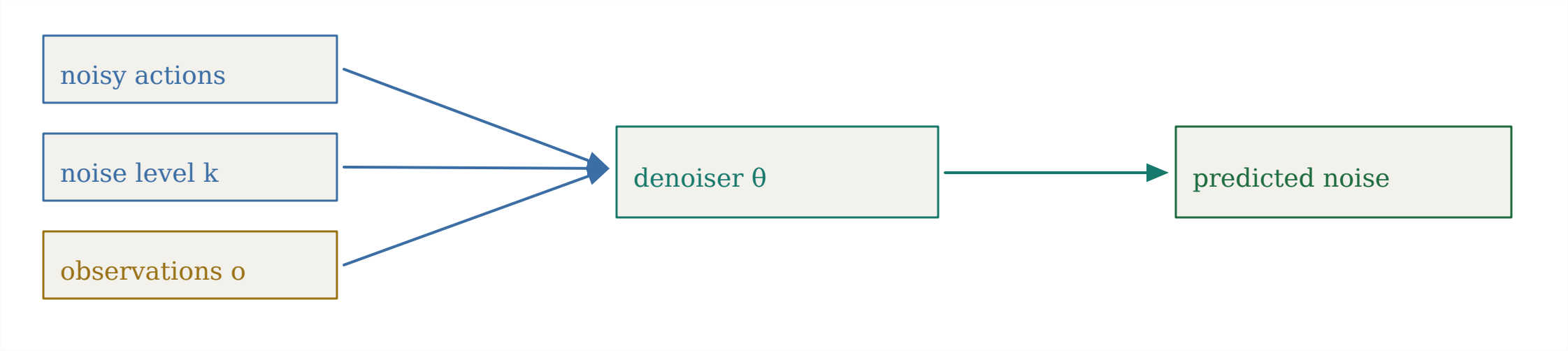
DEFINITION

Conditioning

The action distribution depends on observations

$$s_k(x \mid o) = \nabla_x \log p_k(x \mid o)$$

- o fixed observation context
- x noisy action vector
- s_k conditional score



Diffusion Policy §2.3 · Chi et al. (2023/2024)

ARCHITECTURE

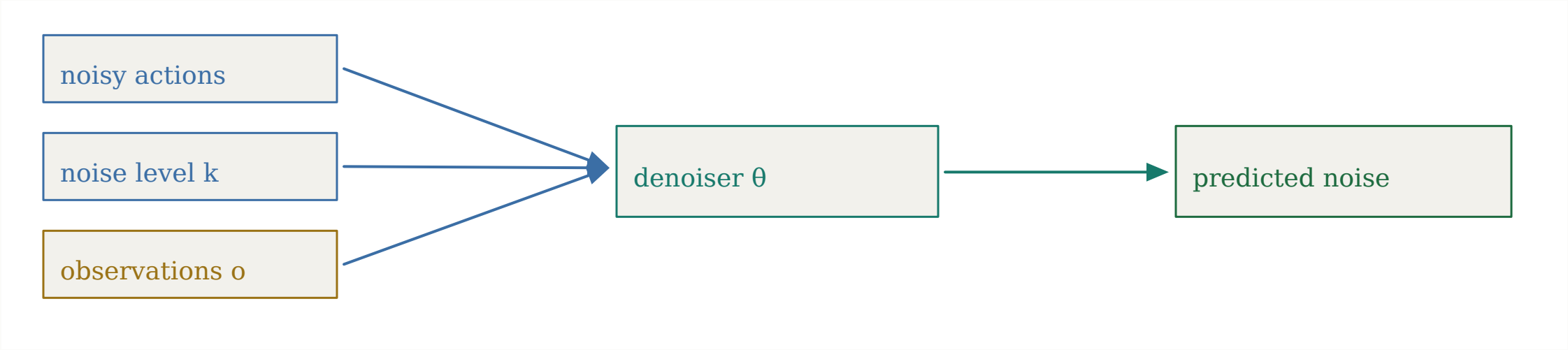
Conditioning

The denoiser has three inputs and one matching output

$$\epsilon_\theta : (x_k, k, o) \longmapsto \hat{\epsilon} \in \mathbb{R}^d$$

$x_k \in \mathbb{R}^d$ noisy action sequence

k, o noise level and encoded observations



Diffusion Policy §§2.3,3.1–3.2 · Chi et al. (2023/2024)

DEFINITION

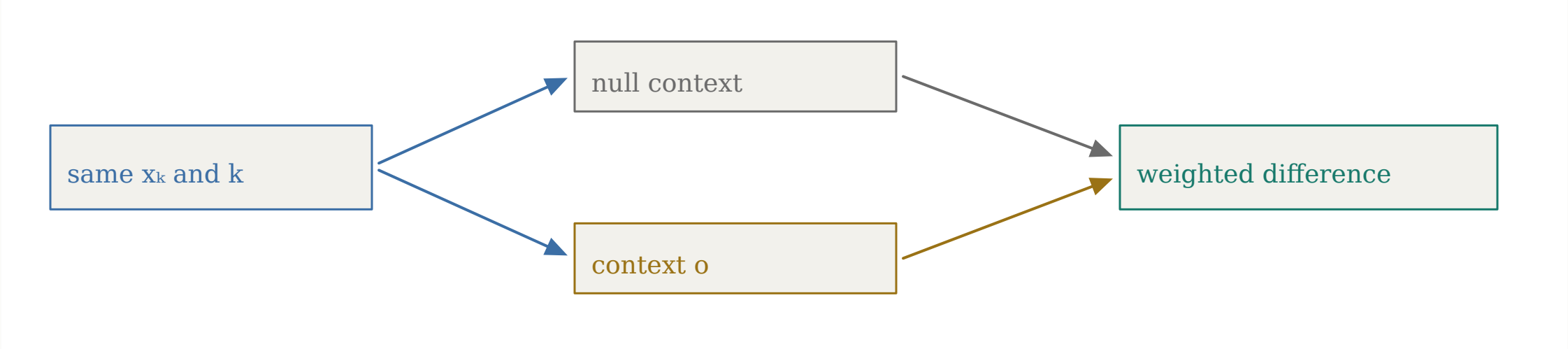
Conditioning

Classifier-free guidance combines two predictions

$$\hat{\epsilon}_g = \hat{\epsilon}_u + w(\hat{\epsilon}_c - \hat{\epsilon}_u)$$

$\hat{\epsilon}_c, \hat{\epsilon}_u$ conditional and null-context predictions

w guidance scale; same x and k



Classifier-Free Guidance §3.2 · Ho & Salimans (2022)

INTERPRETATION

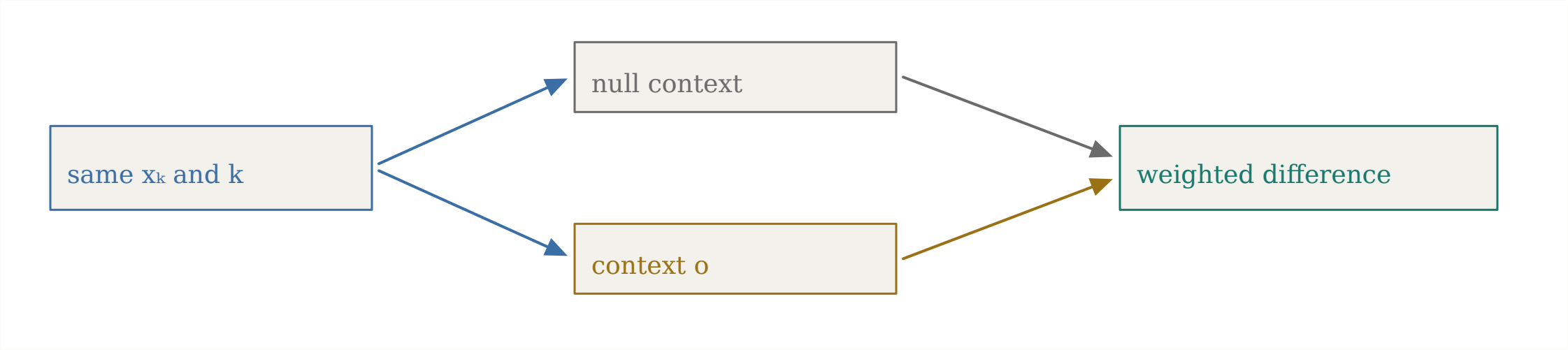
Conditioning

Guidance changes the fixed-level density direction

$$s_g = (1 - w)s_u + w s_c = \nabla_x \log \left[p_k(x)^{1-w} p_k(x \mid o)^w \right]$$

s_u, s_c exact scores at the same level

w guidance scale; normalizability required



Classifier-Free Guidance §3.2; convention bridge · Ho & Salimans (2022)

Conditioning

Guidance changes the fixed-level density direction

Convert the linear prediction rule into scores

Both predictions use the same positive noise scale sigma_k.

1. $\hat{\epsilon}_g = (1 - w)\hat{\epsilon}_u + w\hat{\epsilon}_c.$
2. $s_g = -\hat{\epsilon}_g/\sigma_k = (1 - w)s_u + ws_c.$
3. For exact scores, $s_g = \nabla_x[(1 - w)\log p_k(x) + w \log p_k(x \mid o)].$

Classifier-Free Guidance §3.2; convention bridge · Ho & Salimans (2022)

Conditioning

Guidance changes the fixed-level density direction

Combine logarithms with the necessary qualification

Assume the power product can be normalized.

1. $s_g = \nabla_x \log[p_k(x)^{1-w} p_k(x \mid o)^w]$.
2. This is a fixed-noise-level identity. It does not alone identify the final distribution of a finite guided sampler.
3. The practical scale trades conditioning strength against diversity and artifacts.

Classifier-Free Guidance §3.2; convention bridge · Ho & Salimans (2022)

YOUR TURN

Conditioning

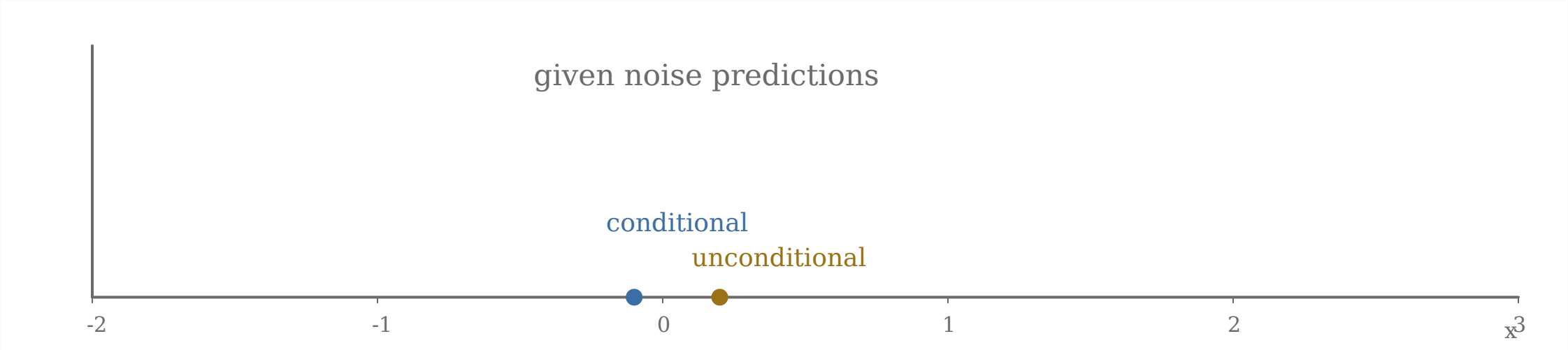
Transfer: calculate and interpret guidance

Compute guided noise and score for each w.

$$\hat{\epsilon}_u = 0.2, \quad \hat{\epsilon}_c = -0.1, \quad \sigma_k = 0.5$$

$w = 0, 1, 2$ scales to compare

$s_g = -\hat{\epsilon}_g / \sigma_k$ guided score



Classifier-Free Guidance §3.2; course exercise · Ho & Salimans (2022)

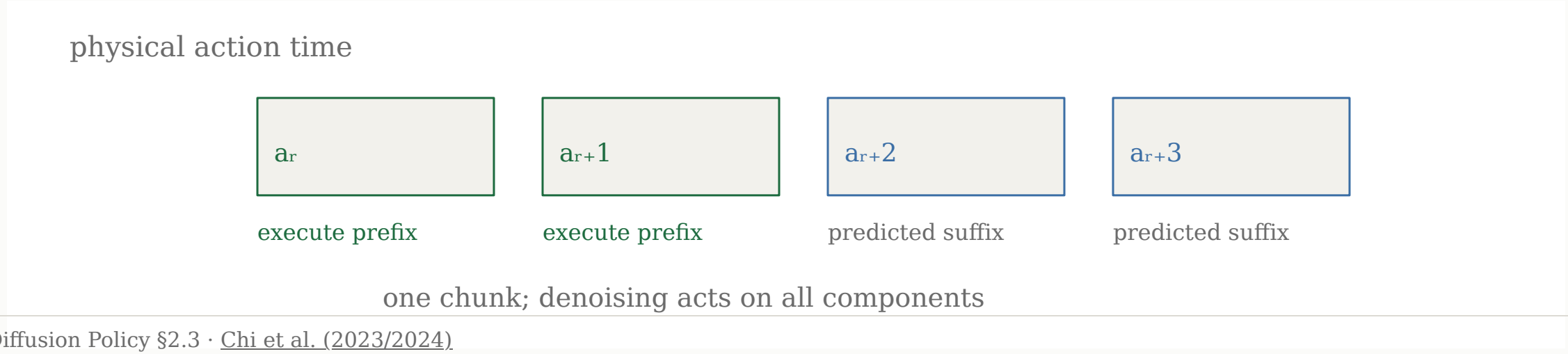
OBJECTS

Diffusion Policy

An action chunk is the vector being generated

$$A_r^{(0)} = \begin{bmatrix} a_r^T \\ \vdots \\ a_{r+H_p-1}^T \end{bmatrix} \in \mathbb{R}^{H_p \times d_a}$$

- r physical time
- H_p, d_a prediction horizon and action dimension
- (0) clean diffusion level



Diffusion Policy §2.3 · Chi et al. (2023/2024)

YOUR TURN

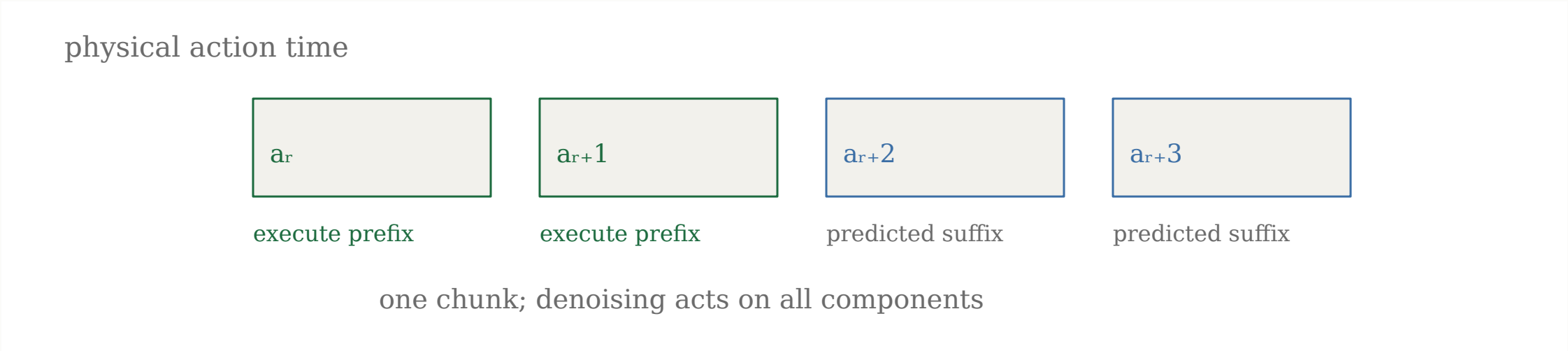
Diffusion Policy

Three horizons serve different purposes

Count observations, predicted scalars, and executed steps.

$$H_o = 2, \quad H_p = 4, \quad H_a = 2, \quad d_a = 2$$

- H_o observations supplied
- H_p actions predicted
- H_a actions executed before replanning



Diffusion Policy §2.3; course exercise · Chi et al. (2023/2024)

TRAINING OBJECTIVE

Diffusion Policy

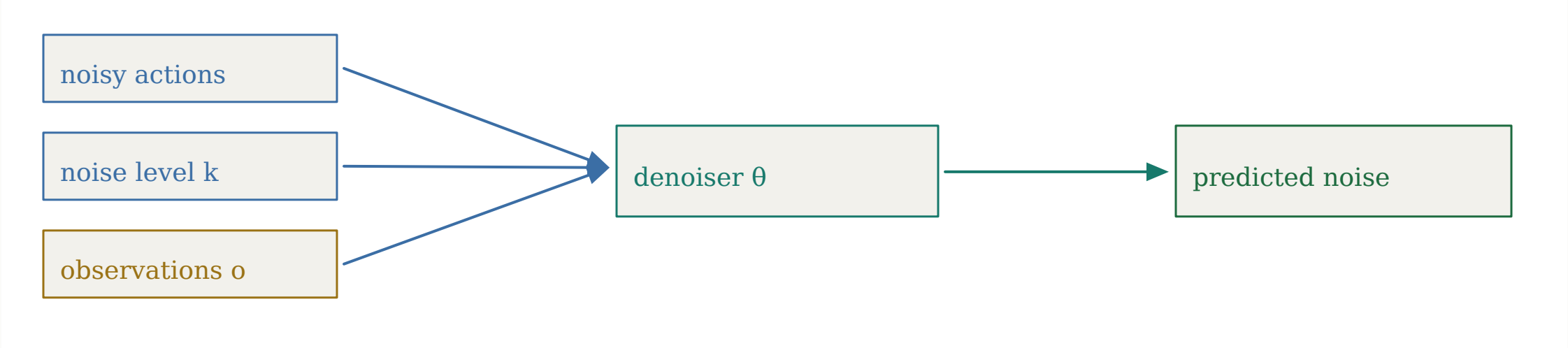
Training pairs observations with demonstrated chunks

$$L(\theta) = \mathbb{E}\|\epsilon_\theta(A_r^{(k)}, k, o_r) - \epsilon\|_F^2$$

$A_r^{(k)} = \alpha_k A_r^{(0)} + \sigma_k \epsilon$ noisy chunk

o_r observation history

ϵ same-shaped Gaussian noise



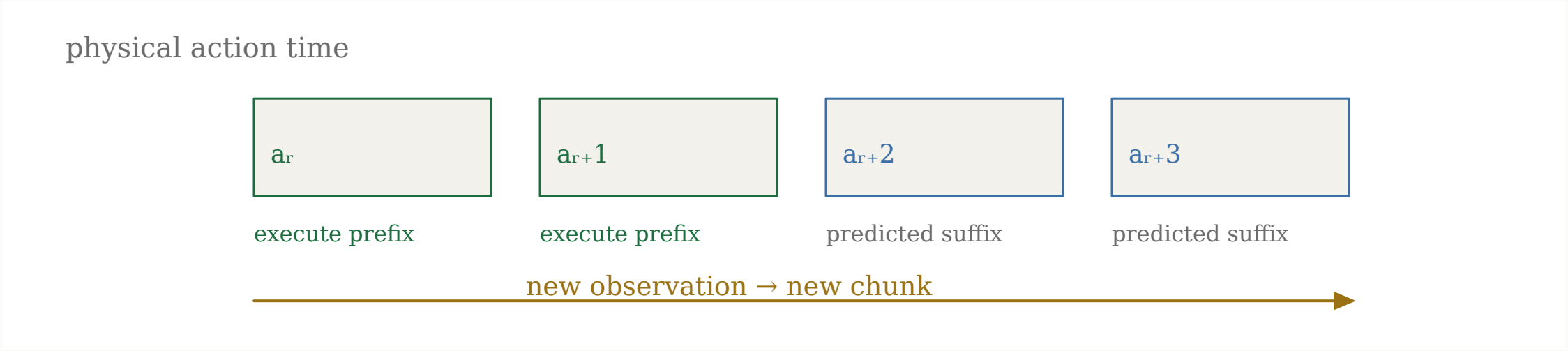
Diffusion Policy §2.3, Eq.5; DDPM Eq.14 · Ho, Jain & Abbeel (2020) · Chi et al. (2023/2024)

YOUR TURN

Diffusion Policy

A receding horizon closes the feedback loop

Observe → sample chunk → execute prefix → observe again
What happens to the unused suffix?



Diffusion Policy §2.3 · Chi et al. (2023/2024)

INTEGRATED WORKED EXAMPLE

Diffusion Policy

Integrated example: a normalized two-action chunk

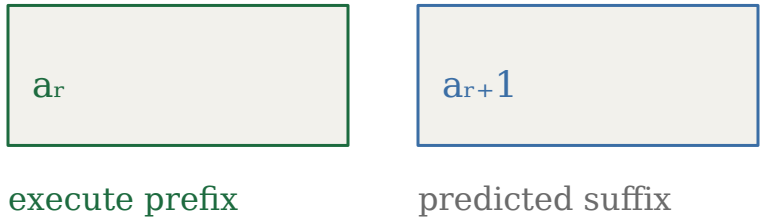
WORKED EXAMPLE

$$x_1 = \begin{bmatrix} 1.3 \\ -0.2 \end{bmatrix}, \quad \hat{\epsilon} = \begin{bmatrix} -0.5 \\ 1 \end{bmatrix}, \quad \alpha_1 = 0.8,\ \sigma_1 = 0.6$$

x_1 two noisy normalized scalar actions

$a = 0.4 + 0.1x$ conversion to target position in m

physical action time



one chunk; denoising acts on all components

Diffusion Policy §§2.3,A.1; course example · Chi et al. (2023/2024)

INTEGRATED WORKED EXAMPLE

Diffusion Policy

Integrated example: the final denoising step

WORKED EXAMPLE

$$\hat{x}_0 = \frac{x_1 - 0.6\hat{\epsilon}}{0.8} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

$x_1 = (1.3, -0.2)^T$ noisy chunk

$\hat{\epsilon} = (-0.5, 1)^T$ predicted noise



recover the clean normalized chunk before restoring units

DDPM Eq.11; DDIM Eq.12; course example · Ho, Jain & Abbeel (2020) · Song, Meng & Ermon (2021)

INTEGRATED WORKED EXAMPLE

Diffusion Policy

Integrated example: restore units and execute a prefix

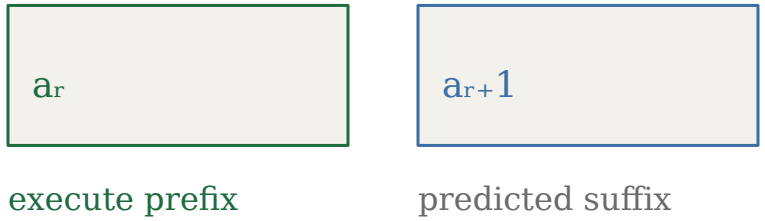
WORKED EXAMPLE

$$a = 0.4 + 0.1x, \quad x = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \quad \Rightarrow \quad A_r^{(0)} = \begin{bmatrix} 0.6 \\ 0.3 \end{bmatrix} \text{m}$$

$H_p = 2, H_a = 1$ predict two, execute one

a target position; not displacement

physical action time



one chunk; denoising acts on all components

Diffusion Policy §2.3; course example · Chi et al. (2023/2024)

YOUR TURN

Diffusion Policy

Transfer: change execution without changing the sampler

Targets: 0.6 m, then 0.3 m. Compare H_a=1 and H_a=2. An obstacle moves after the first action: which policy can refresh its second target?

Diffusion Policy §2.3; course exercise · Chi et al. (2023/2024)

COMPARISON

Diffusion Policy

ACT and diffusion share a chunked control interface

	ACT	Diffusion
Training	reconstruction + latent KL	noise prediction
Generation	standard latent z = 0	iterative, random initialization
Output	action chunk	action chunk

ACT §IV, Fig.3; Diffusion Policy §§2–3 · Chi et al. (2023/2024) · Zhao et al. (2023)

YOUR TURN

Diffusion Policy

A low denoising loss is not a deployment certificate

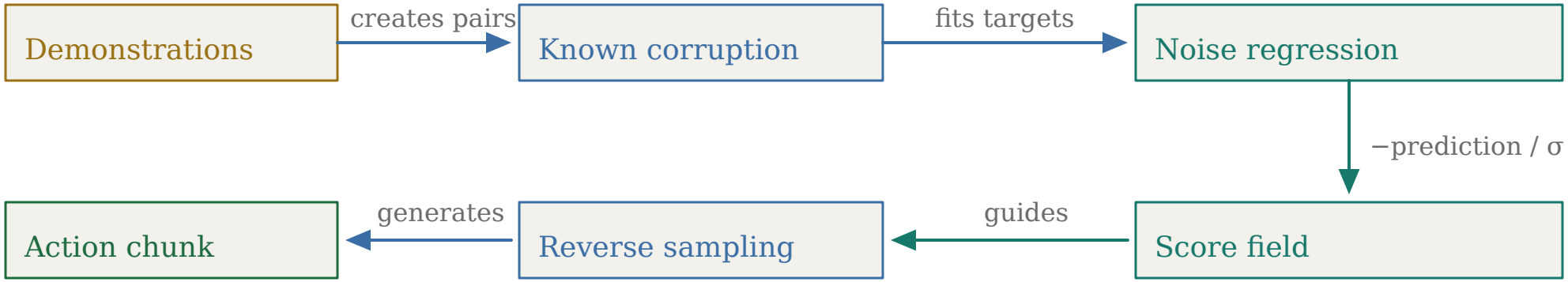
Training metric: noise-prediction error.
Deployment question: does the closed-loop task succeed?
Give two missing checks.

Diffusion Policy §§5,9; course exercise · Chi et al. (2023/2024)

SYNTHESIS

Diffusion and Diffusion Policy

The complete mathematical chain



DDPM §§2–3; Diffusion Policy §2.3 · Ho, Jain & Abbeel (2020) · Chi et al. (2023/2024)

References

Diffusion, denoising and conditional generation

Ho, Jain & Abbeel. Denoising Diffusion Probabilistic Models. NeurIPS, 2020.

Song, Meng & Ermon. Denoising Diffusion Implicit Models. ICLR, 2021.

Vincent. A Connection Between Score Matching and Denoising Autoencoders. Neural Computation, 2011.

Efron. Tweedie’s Formula and Selection Bias. JASA, 2011.

Original course figures and exercises. Sources cited for methods; no paper figures reproduced.

References

Robot policies and the comparison baseline

Chi et al. Diffusion Policy. RSS 2023; IJRR extension, 2024.

Zhao et al. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. RSS, 2023.

Ho & Salimans. Classifier-Free Diffusion Guidance. 2022.

Goodfellow, Bengio & Courville. Deep Learning. MIT Press, 2016. Ch.3.

Original course figures and exercises. Sources cited for methods; no paper figures reproduced.