Discretized Approximate Ancestral Sampling

Jona Ballé
NYU Tandon School of Engineering

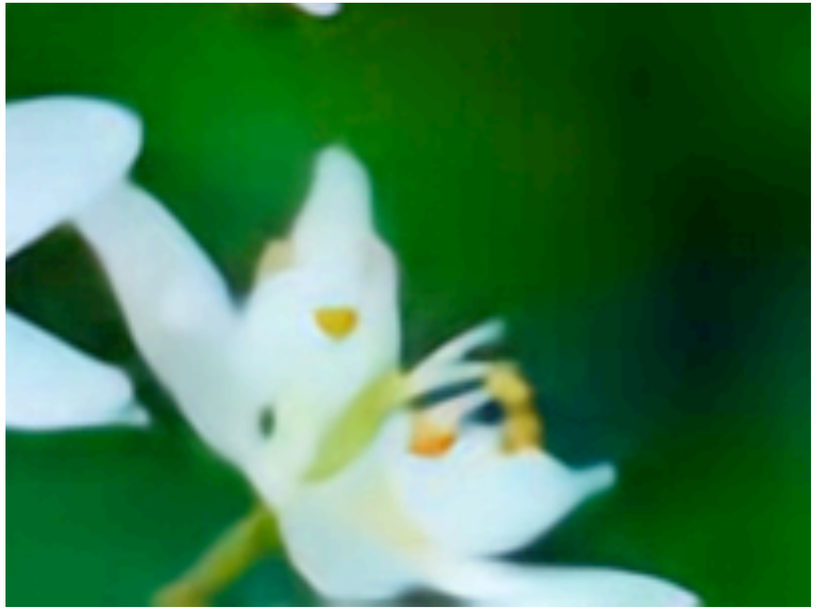
ISIT Learn to Compress & Compress to Learn Workshop
25 June 2025

Alfredo De la Fuente
Google

Saurabh Singh
Google DeepMind

original

NTC (2017)

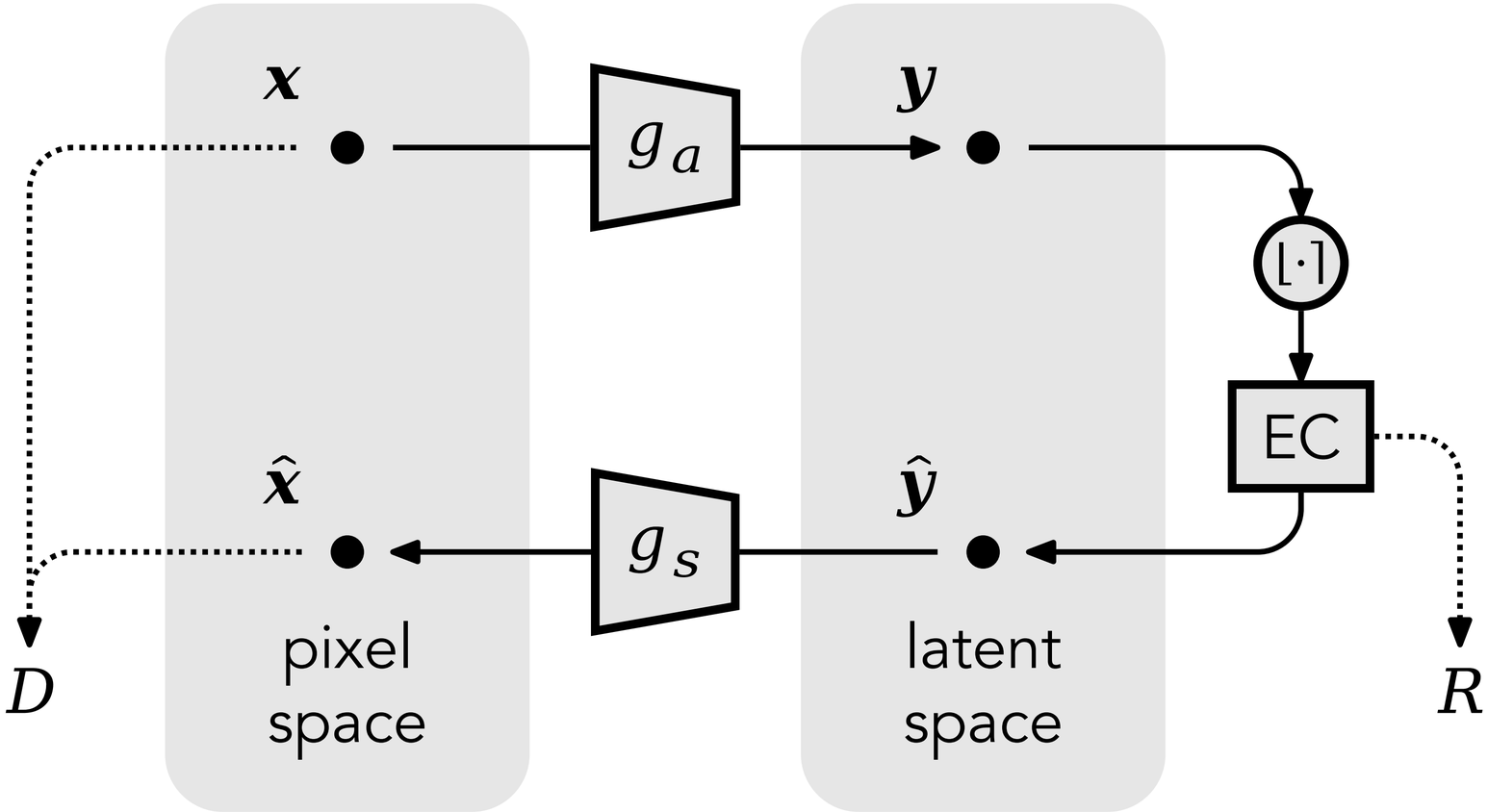


JPEG

JPEG 2000

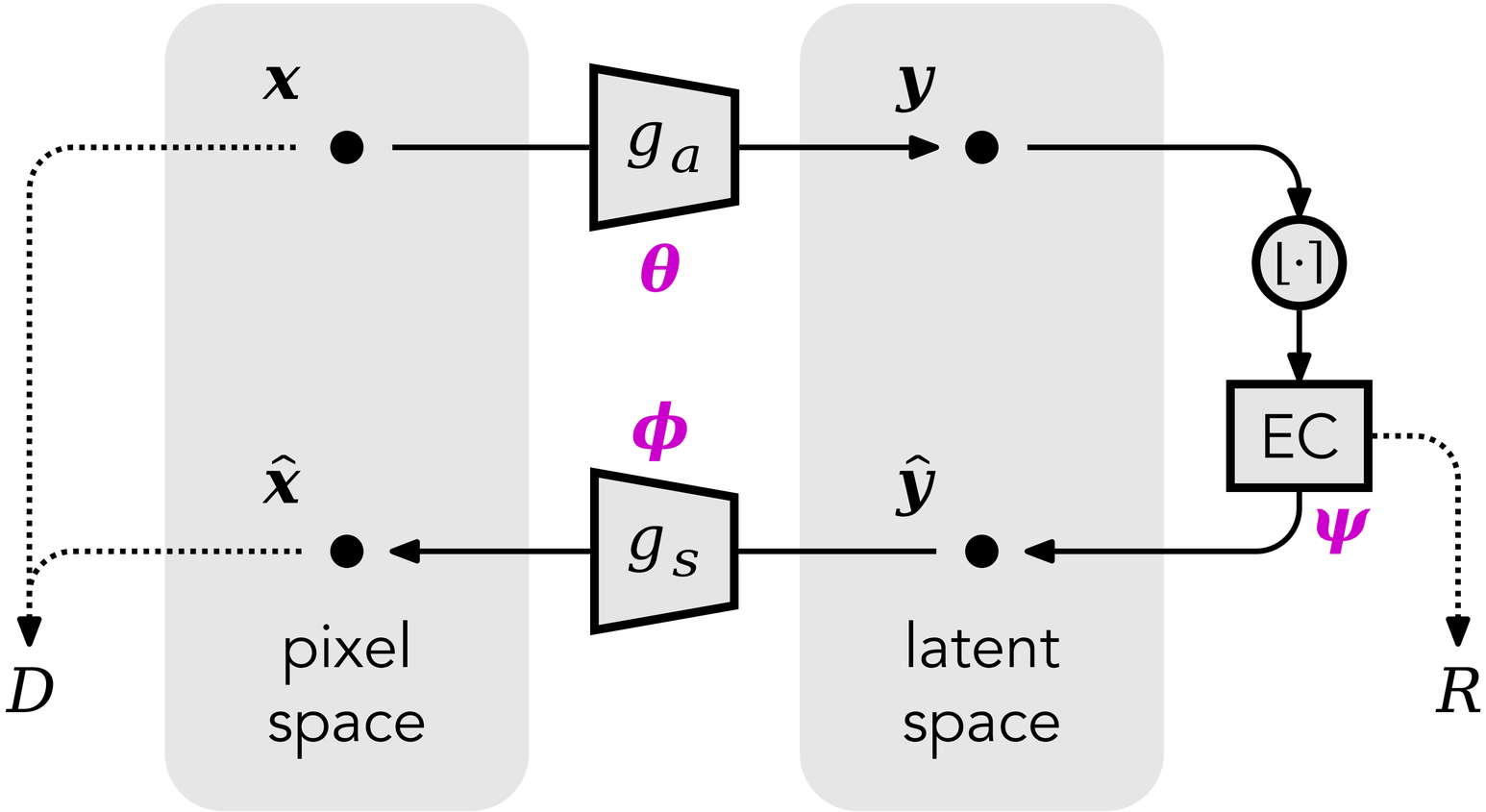
Ballé, Laparra & Simoncelli (Int. Conf. on Learning Representations, 2017)

Nonlinear transform coding (end-to-end optimized)



Ballé, Laparra & Simoncelli (Int. Conf. on Learning Representations, 2017)

Nonlinear transform coding (end-to-end optimized)

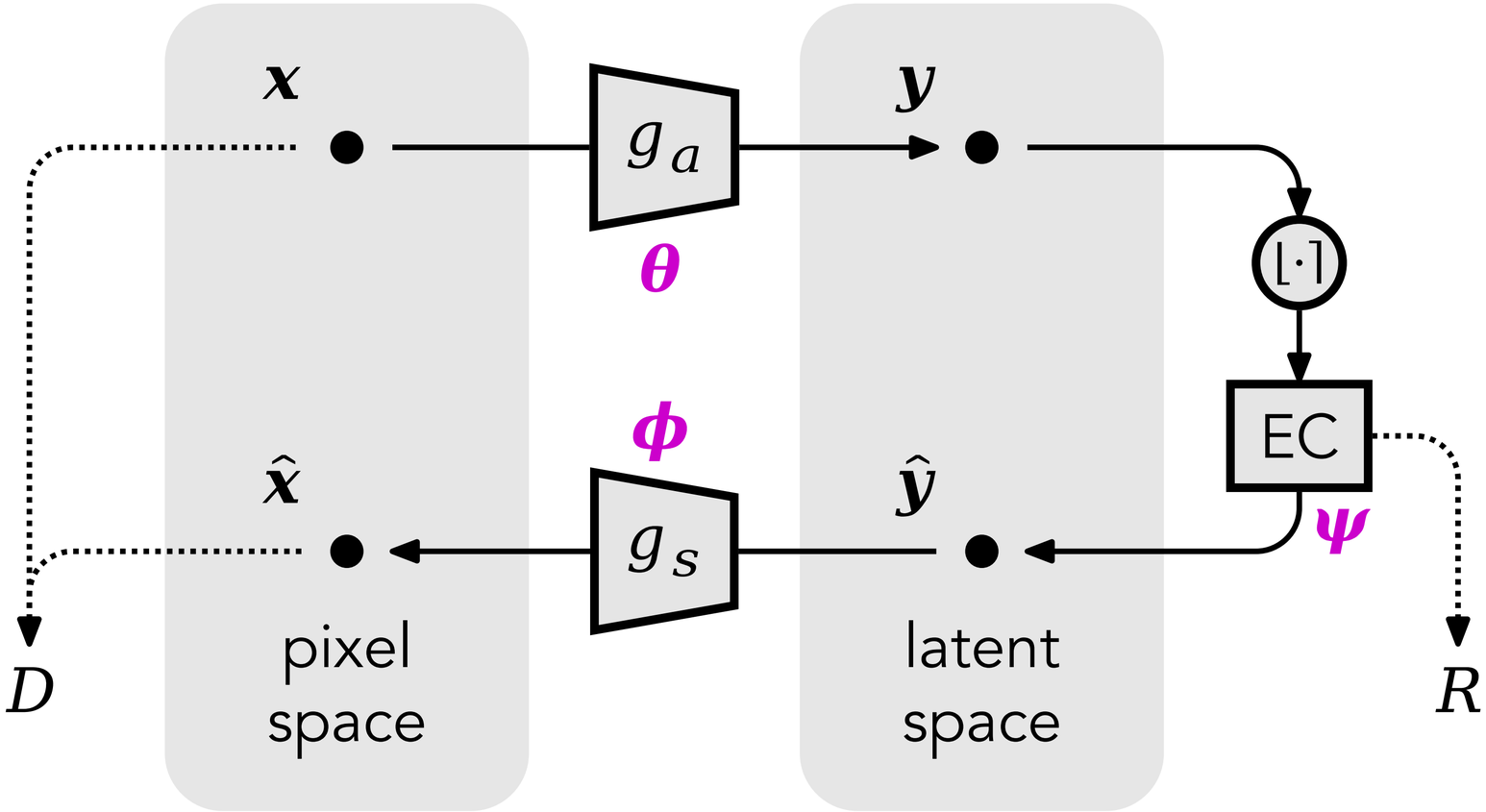


quantization (rounding)

entropy coding

Ballé, Laparra & Simoncelli (Int. Conf. on Learning Representations, 2017)

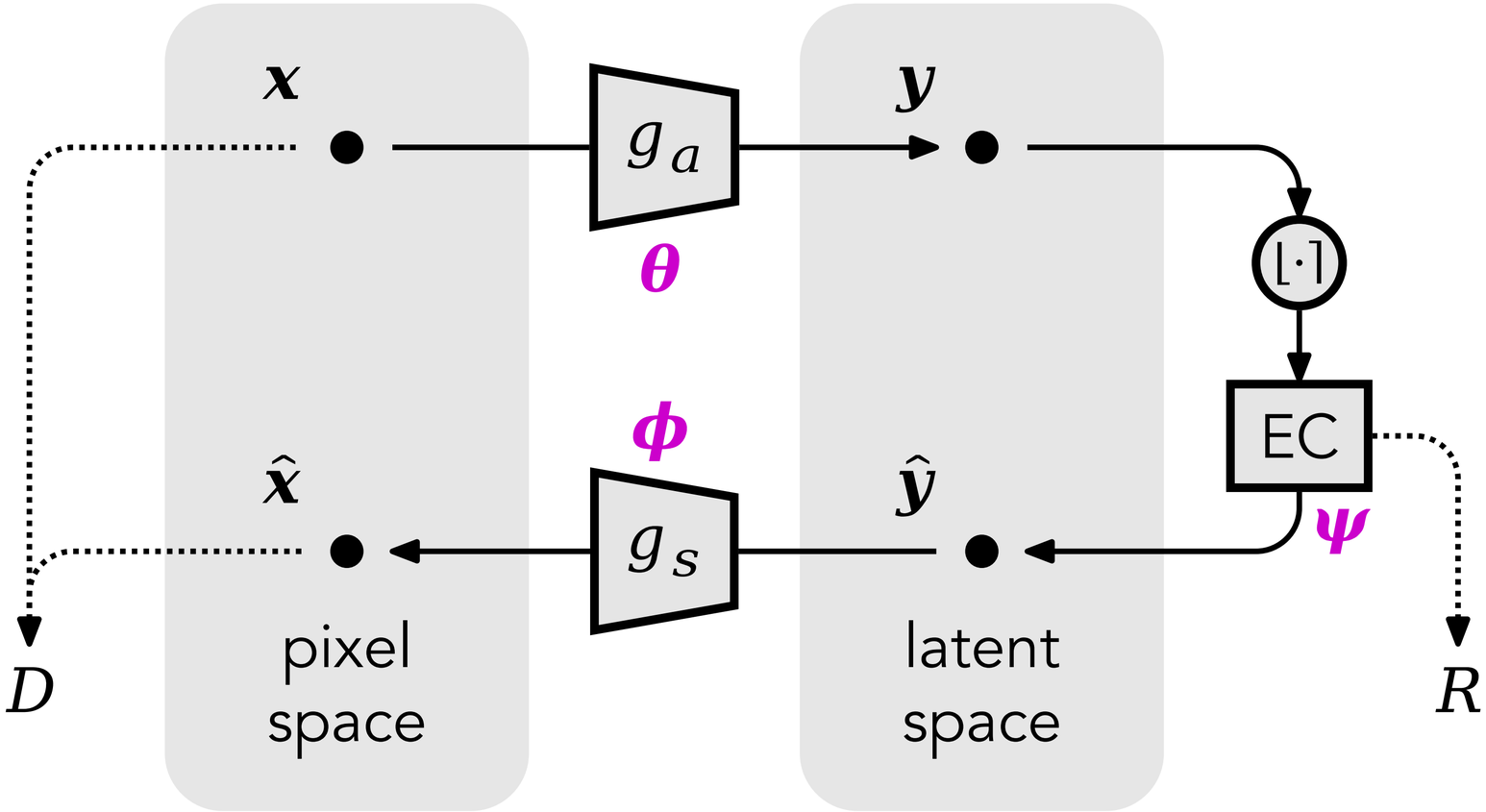
Nonlinear transform coding (end-to-end optimized)



$$L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \underbrace{\mathbb{E}_{\boldsymbol{x}}[-\log_2 p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}})]}_{R} + \lambda \underbrace{\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2]}_{D}$$

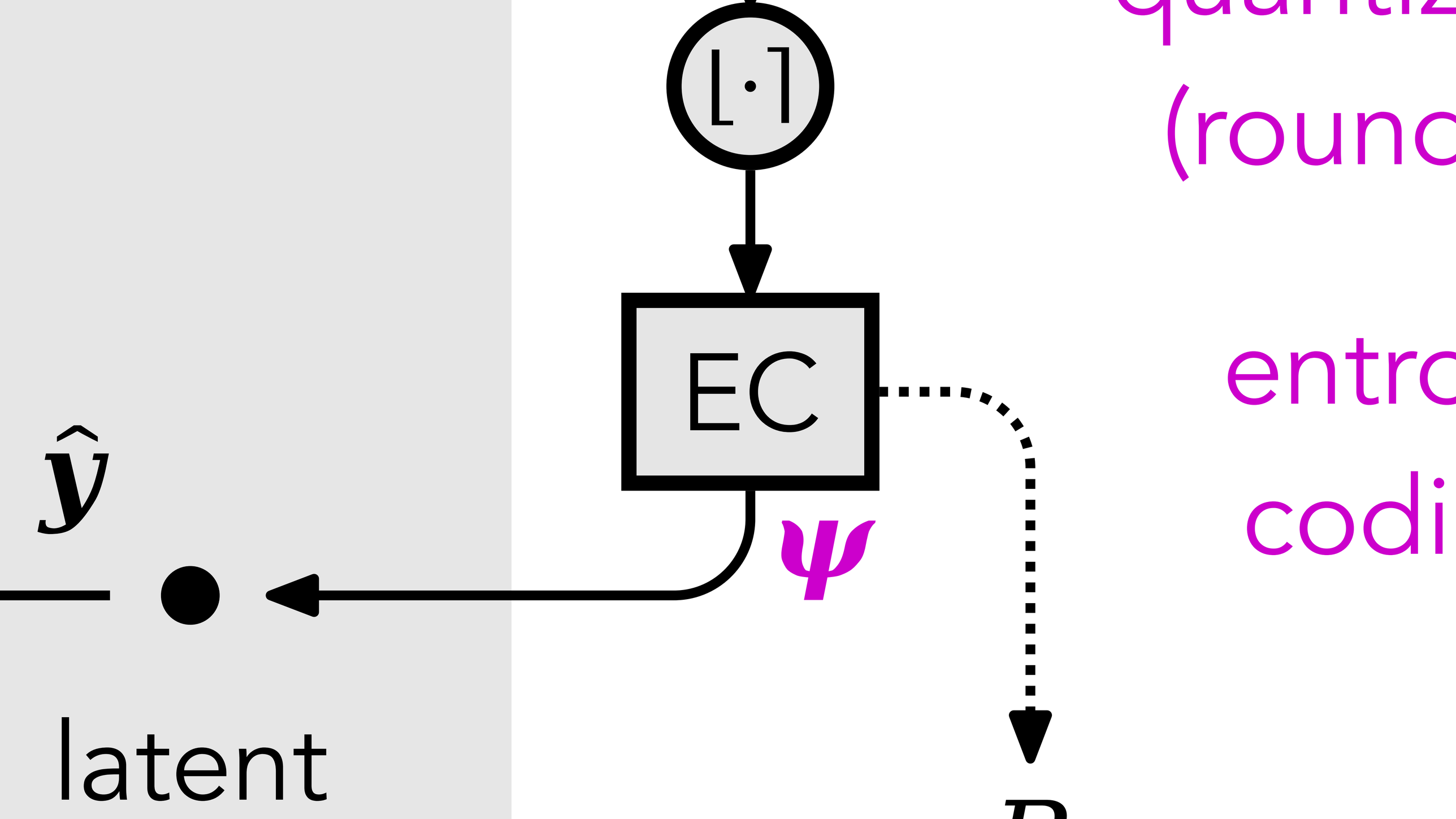
Ballé, Laparra & Simoncelli (Int. Conf. on Learning Representations, 2017)

Nonlinear transform coding (end-to-end optimized)



$$L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \underbrace{\mathbb{E}_{\boldsymbol{x}}[-\log_2 p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}})]}_{R} + \lambda \underbrace{\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2]}_{D}$$

Ballé, Laparra & Simoncelli (Int. Conf. on Learning Representations, 2017)


$\lfloor \cdot \rceil$
EC
$\hat{y}$
ψ
latent
quantiz
(round
entro
codi

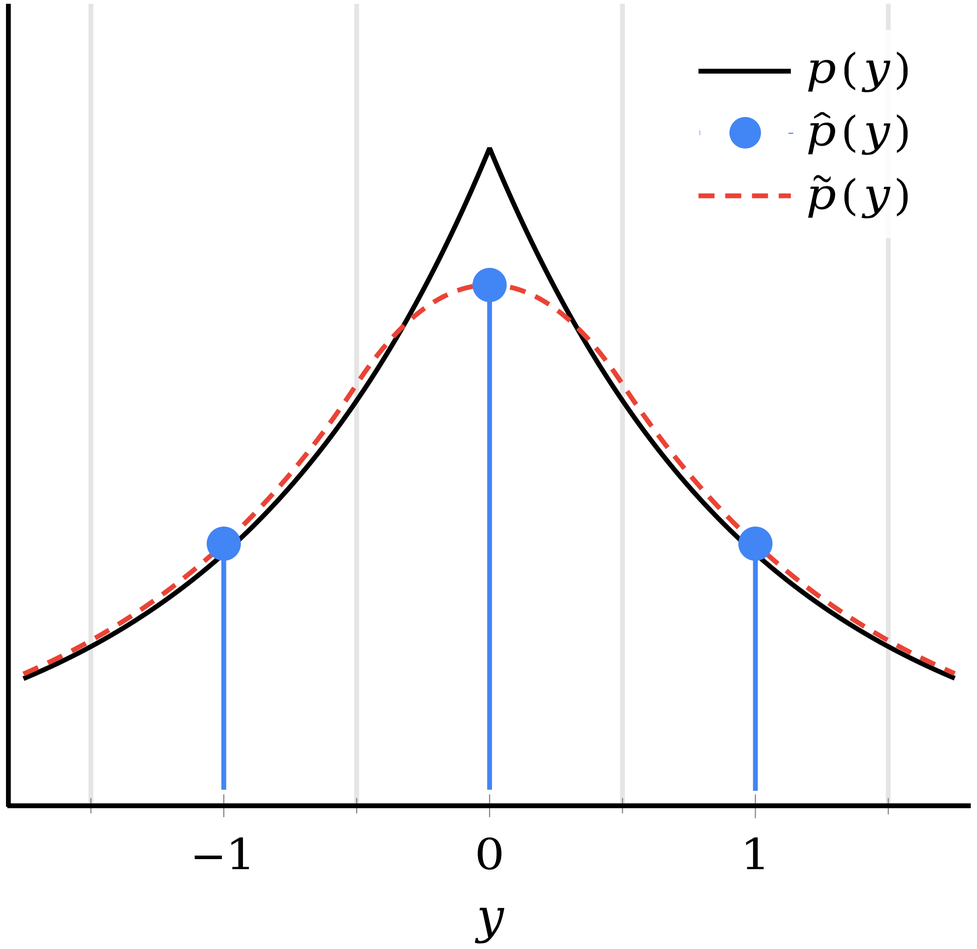
Quantization replaced by uniform noise for training

Quantized y described by probability mass function at integer locations.

"Noisy" y described by probability density function.

At integer locations, they coincide by construction.

Estimate PDF during training, then use PMF for compression.



Ballé, Laparra & Simoncelli (PCS, 2016)

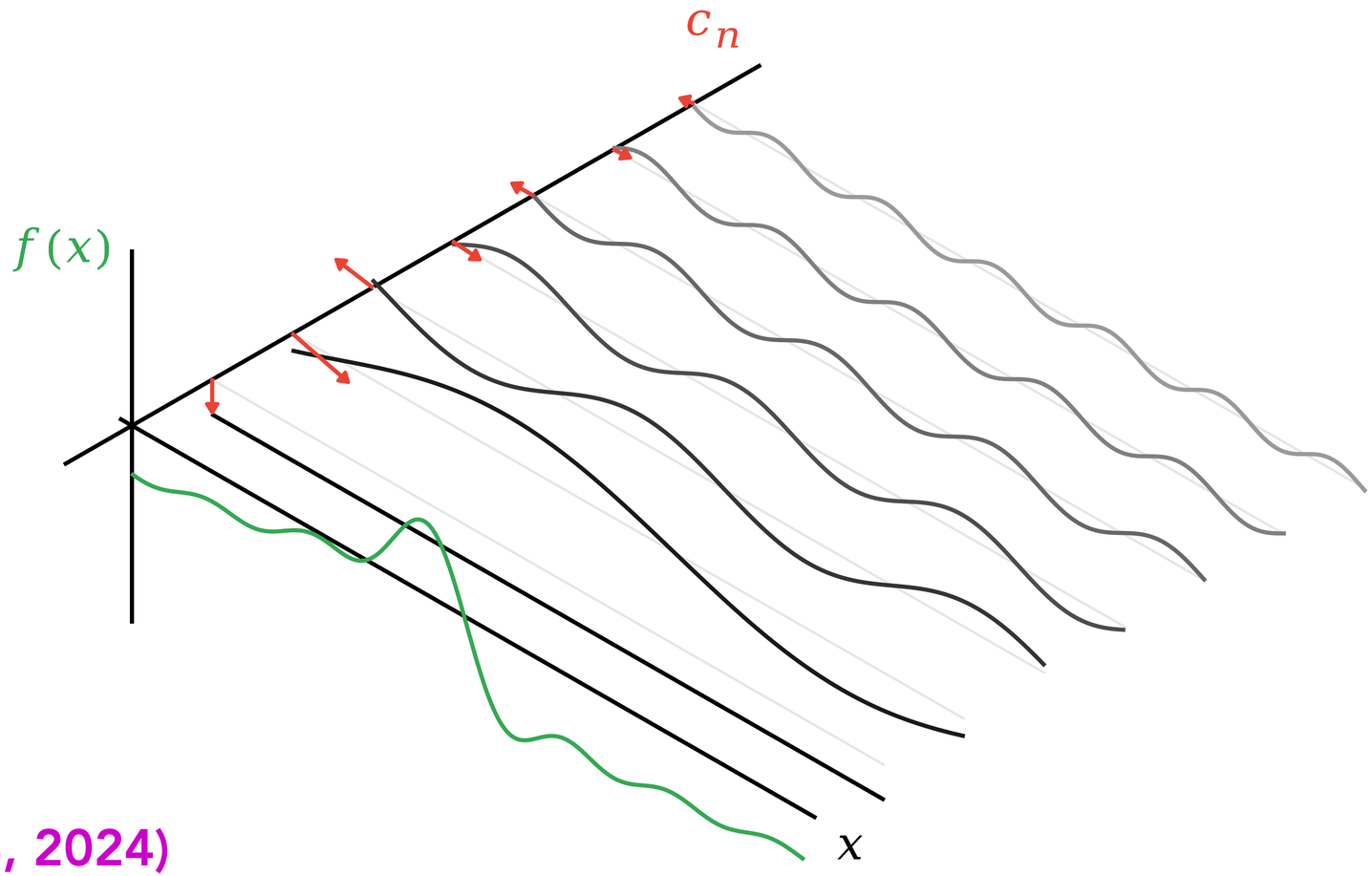
Fourier Basis Density Model

- Parametric densities are inflexible.
- Mixture models get stuck in local minima.
- The cumulative density model we proposed earlier is slow to train.
- **Idea:** represent a density by a truncated Fourier series.

De la Fuente, Singh & Ballé (PCS, 2024)

1. Let's be real!

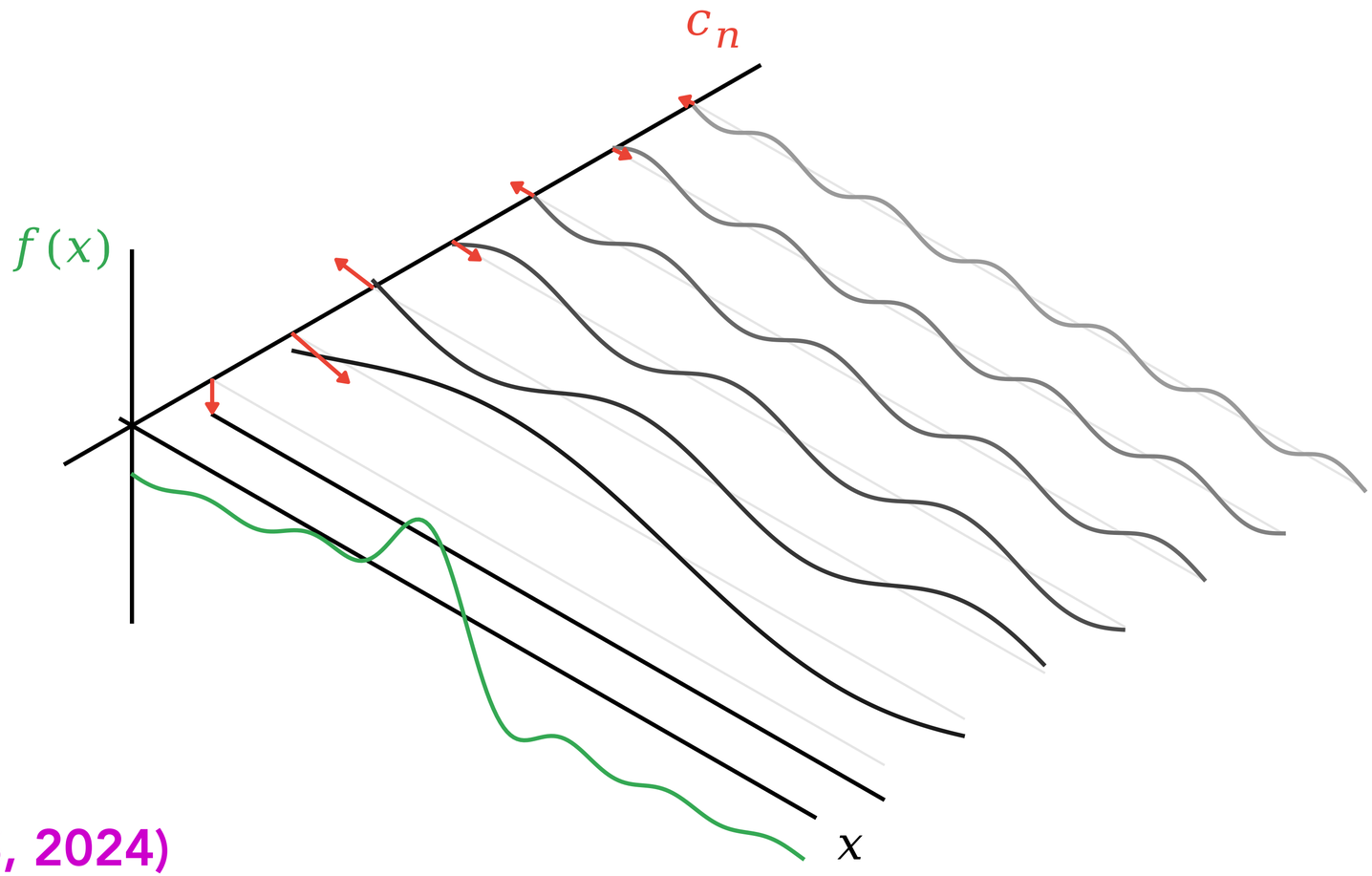
$$f\left(x;\{c_n\}_{-N}^{N}\right) = \sum_{n=-N}^{N} c_n \exp(in\pi x)$$



De la Fuente, Singh & Ballé (PCS, 2024)

1. Let's be real!

$$f\left(x;\{c_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$



De la Fuente, Singh & Ballé (PCS, 2024)

2. Don’t be negative!

$$f\left(x;\{c_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

De la Fuente, Singh & Ballé (PCS, 2024)

2. Don’t be negative!

$$f\left(x;\{c_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

Theorems: Herglotz and Wiener–Khinchin

De la Fuente, Singh & Ballé (PCS, 2024)

2. Don’t be negative!

$$f\left(x;\{c_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

Theorems: Herglotz and Wiener–Khinchin

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

De la Fuente, Singh & Ballé (PCS, 2024)

2. Don’t be negative!

$$f\left(x;\{c_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

Theorems: Herglotz and Wiener–Khinchin

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

$$\begin{bmatrix} d & e & f & g \\ c & d & e & f \\ b & c & d & e \\ a & b & c & d \end{bmatrix}$$

De la Fuente, Singh & Ballé (PCS, 2024)

2. Don’t be negative!

$$f\left(x;\{c_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

Theorems: Herglotz and Wiener–Khinchin

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

$$\begin{bmatrix} d & e & f & g \\ c & d & e & f \\ b & c & d & e \\ a & b & c & d \end{bmatrix} \quad (C = A^\top A)$$

De la Fuente, Singh & Ballé (PCS, 2024)

2. Don’t be negative!

$$f\left(x;\{a_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

Theorems: Herglotz and Wiener–Khinchin

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

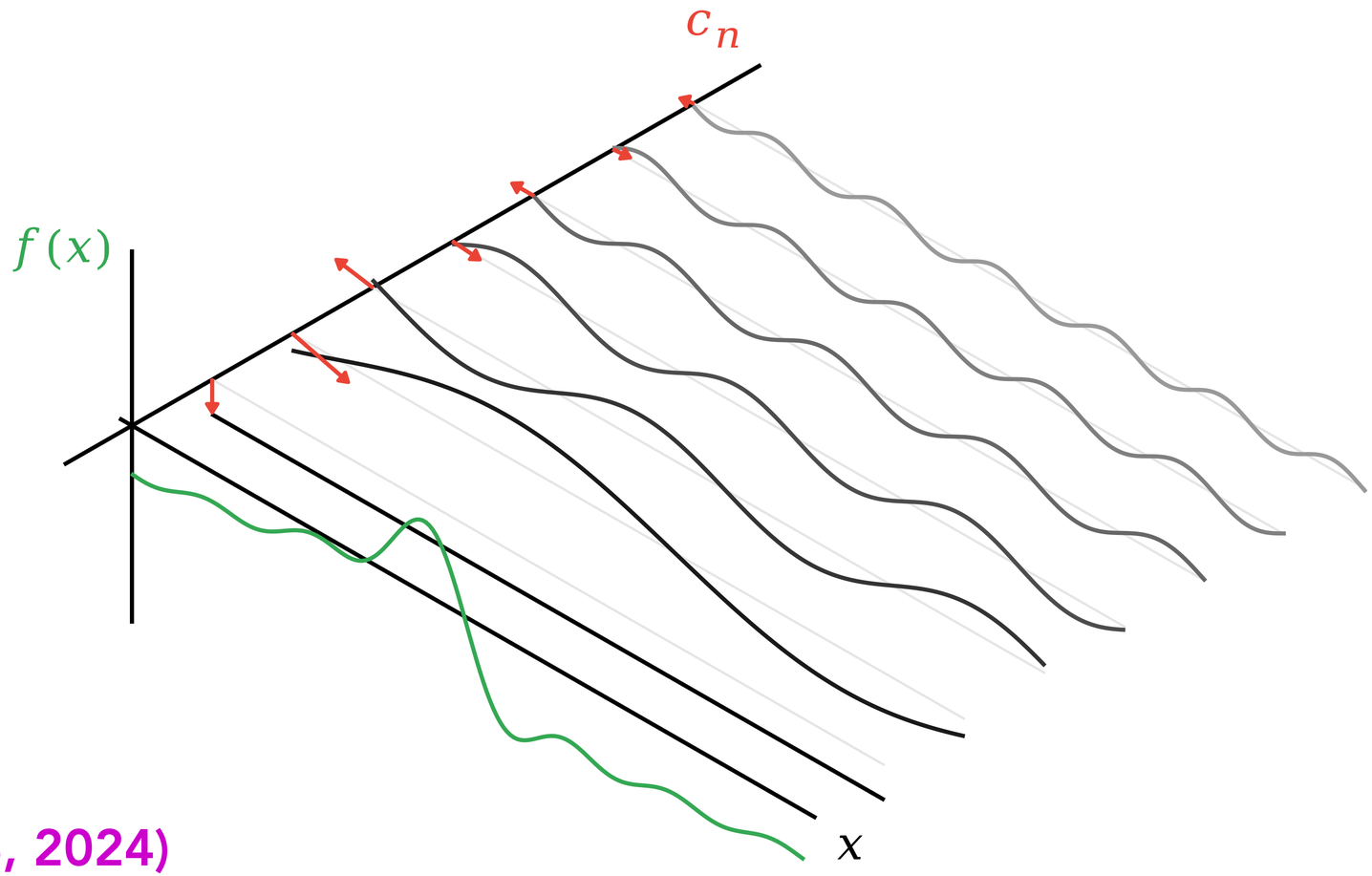
$$\begin{bmatrix} d & e & f & g \\ c & d & e & f \\ b & c & d & e \\ a & b & c & d \end{bmatrix} \quad (C = A^\top A)$$

De la Fuente, Singh & Ballé (PCS, 2024)

2. Don’t be negative!

$$f\left(x;\{a_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$



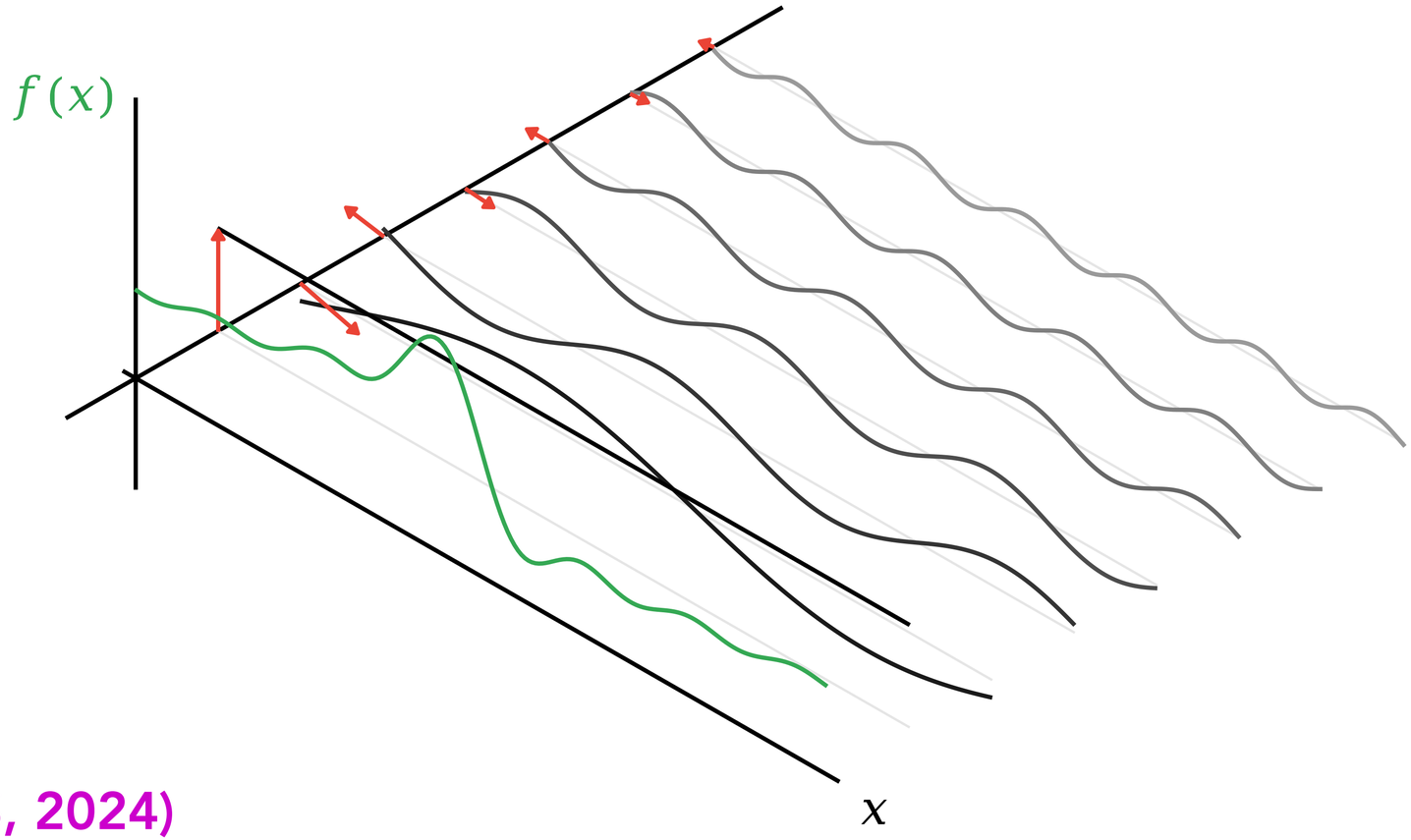
De la Fuente, Singh & Ballé (PCS, 2024)

2. Don’t be negative!

$$f\left(x;\{a_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

$$c_n = a_k * a_{-k}^*$$

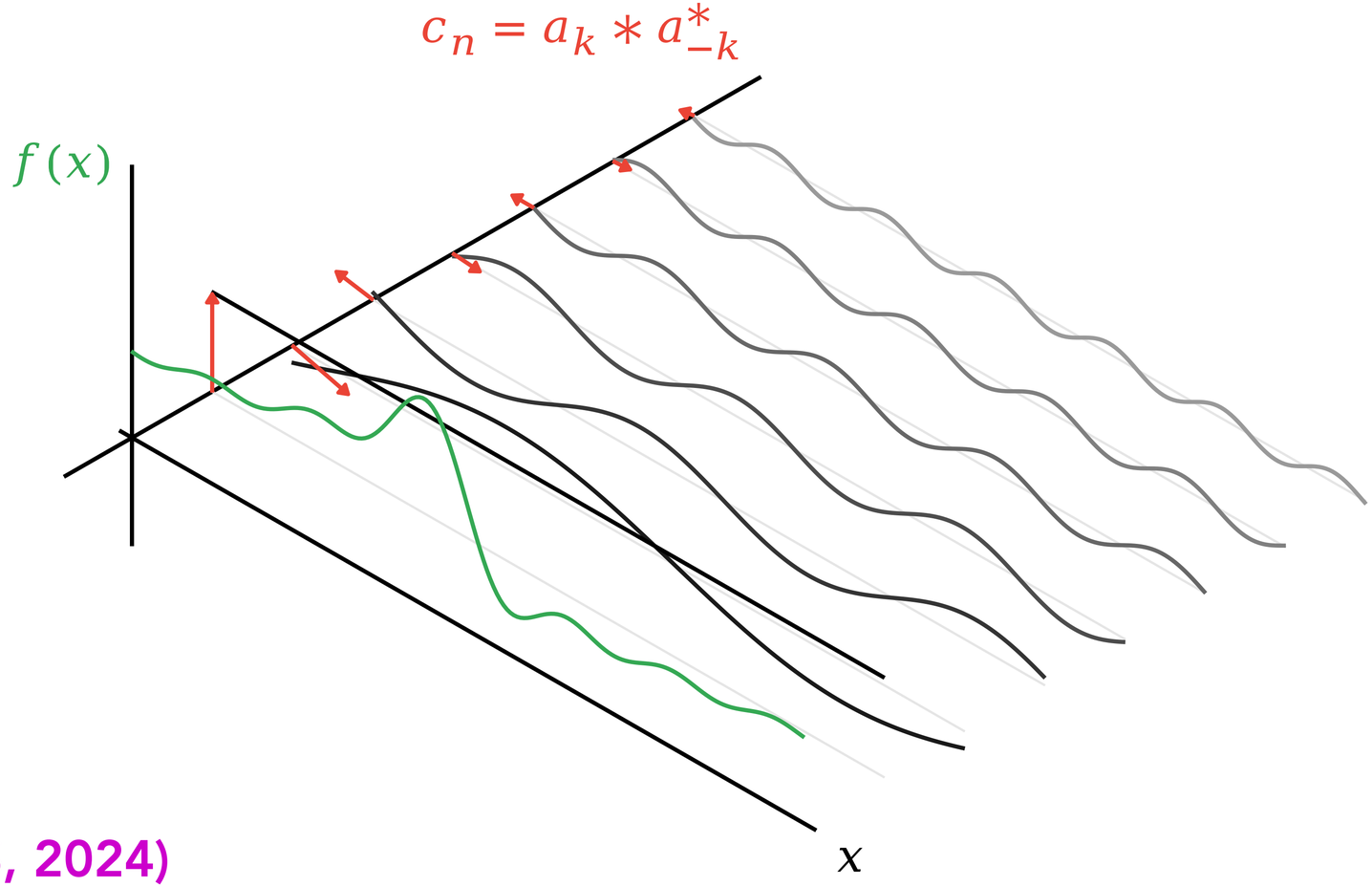


De la Fuente, Singh & Ballé (PCS, 2024)

3. Normalize it!

$$f\left(x;\{a_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$



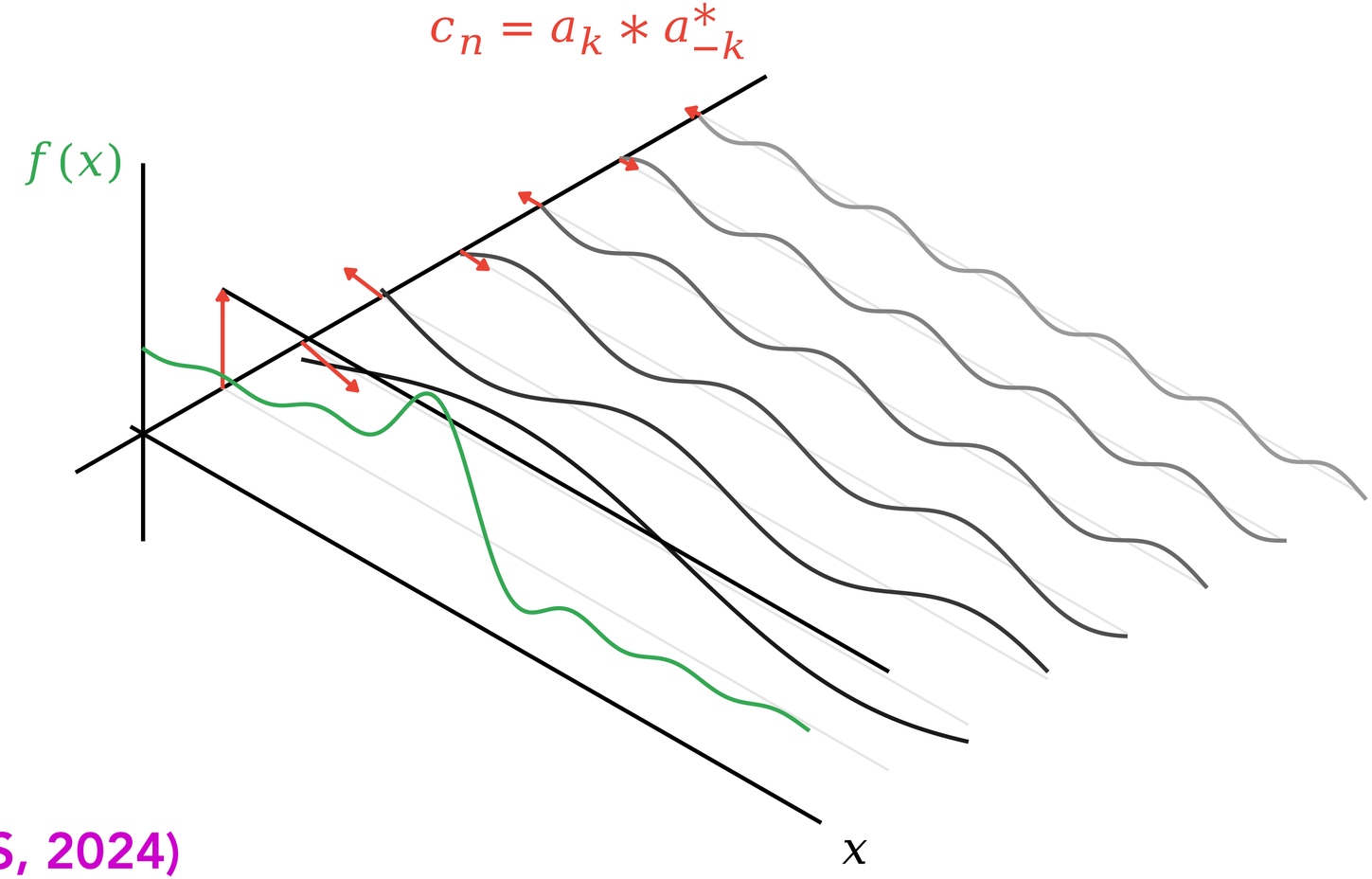
De la Fuente, Singh & Ballé (PCS, 2024)

3. Normalize it!

$$f\left(x;\{a_n\}_0^N\right) = c_0 + 2\sum_{n=1}^{N} \Re\{c_n \exp(in\pi x)\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

$$\int_{-1}^{1} f\left(x;\{a_n\}_0^N\right) = 2c_0$$



De la Fuente, Singh & Ballé (PCS, 2024)

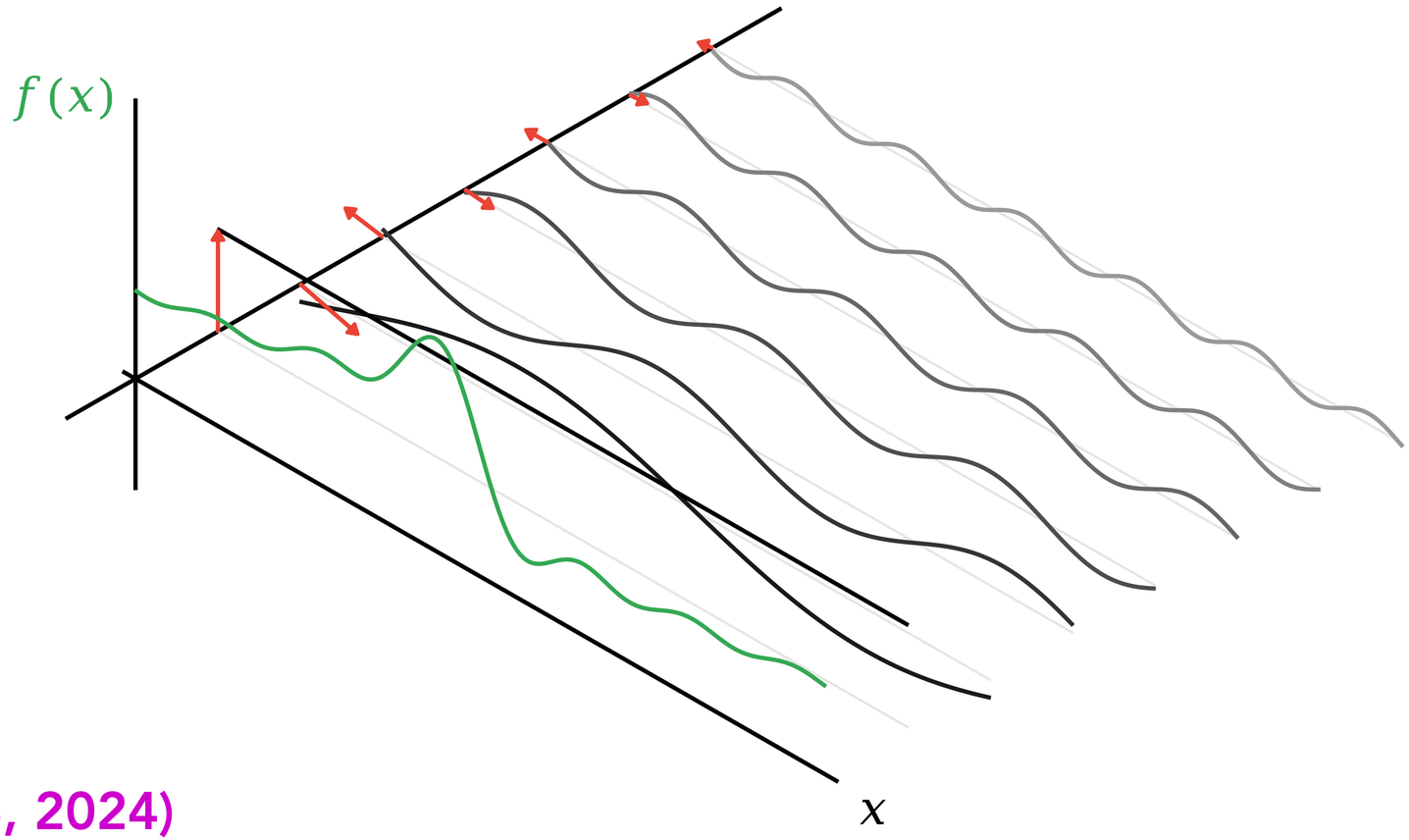
3. Normalize it!

$$p\left(x;\{a_n\}_0^N\right) = \frac{1}{2} + \sum_{n=1}^{N} \Re\left\{\frac{c_n}{c_0}\exp(in\pi x)\right\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

$$\int_{-1}^{1} f\left(x;\{a_n\}_0^N\right) = 2c_0$$

$$c_n = a_k * a_{-k}^*$$



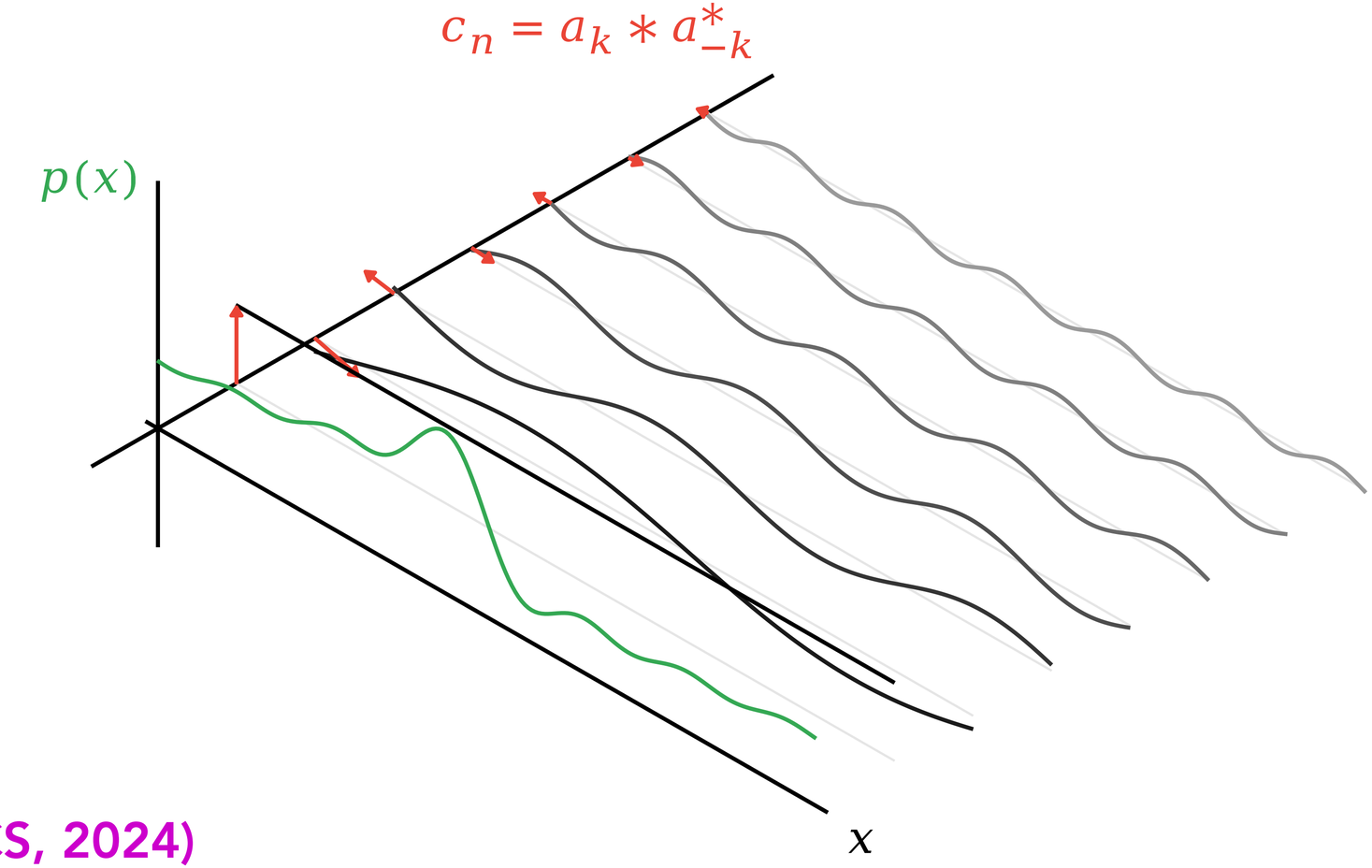
De la Fuente, Singh & Ballé (PCS, 2024)

3. Normalize it!

$$p\left(x;\{a_n\}_0^N\right) = \frac{1}{2} + \sum_{n=1}^{N} \Re\left\{\frac{c_n}{c_0}\exp(in\pi x)\right\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

$$\int_{-1}^{1} f\left(x;\{a_n\}_0^N\right) = 2c_0$$



De la Fuente, Singh & Ballé (PCS, 2024)

4. Expand support to $\mathbb{R}$

$$p\left(x; \{a_n\}_0^N\right) = \frac{1}{2} + \sum_{n=1}^{N} \Re\left\{\frac{c_n}{c_0} \exp(in\pi x)\right\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

De la Fuente, Singh & Ballé (PCS, 2024)

4. Expand support to $\mathbb{R}$

$$p\left(x;\{a_n\}_0^N\right) = \frac{1}{2} + \sum_{n=1}^{N} \Re\left\{\frac{c_n}{c_0}\exp(in\pi x)\right\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

Change of variables:

$$g : (-1, 1) \to \mathbb{R}$$

De la Fuente, Singh & Ballé (PCS, 2024)

4. Expand support to $\mathbb{R}$

$$p\left(x;\{a_n\}_0^N\right) = \frac{1}{2} + \sum_{n=1}^{N} \Re\left\{\frac{c_n}{c_0}\exp(in\pi x)\right\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

Change of variables:

$$g : (-1, 1) \to \mathbb{R}$$

$$g(x; s, t) = s \cdot \tanh^{-1}(x) + t$$

$$g^{-1}(x; s, t) = \tanh\left(\frac{x - t}{s}\right)$$

De la Fuente, Singh & Ballé (PCS, 2024)

4. Expand support to $\mathbb{R}$

$$p\left(x;\{a_n\}_0^N\right) = \frac{1}{2} + \sum_{n=1}^{N} \Re\left\{\frac{c_n}{c_0}\exp(in\pi x)\right\}$$

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^*$$

Change of variables:

$$g(x;s,t) = s \cdot \tanh^{-1}(x) + t$$

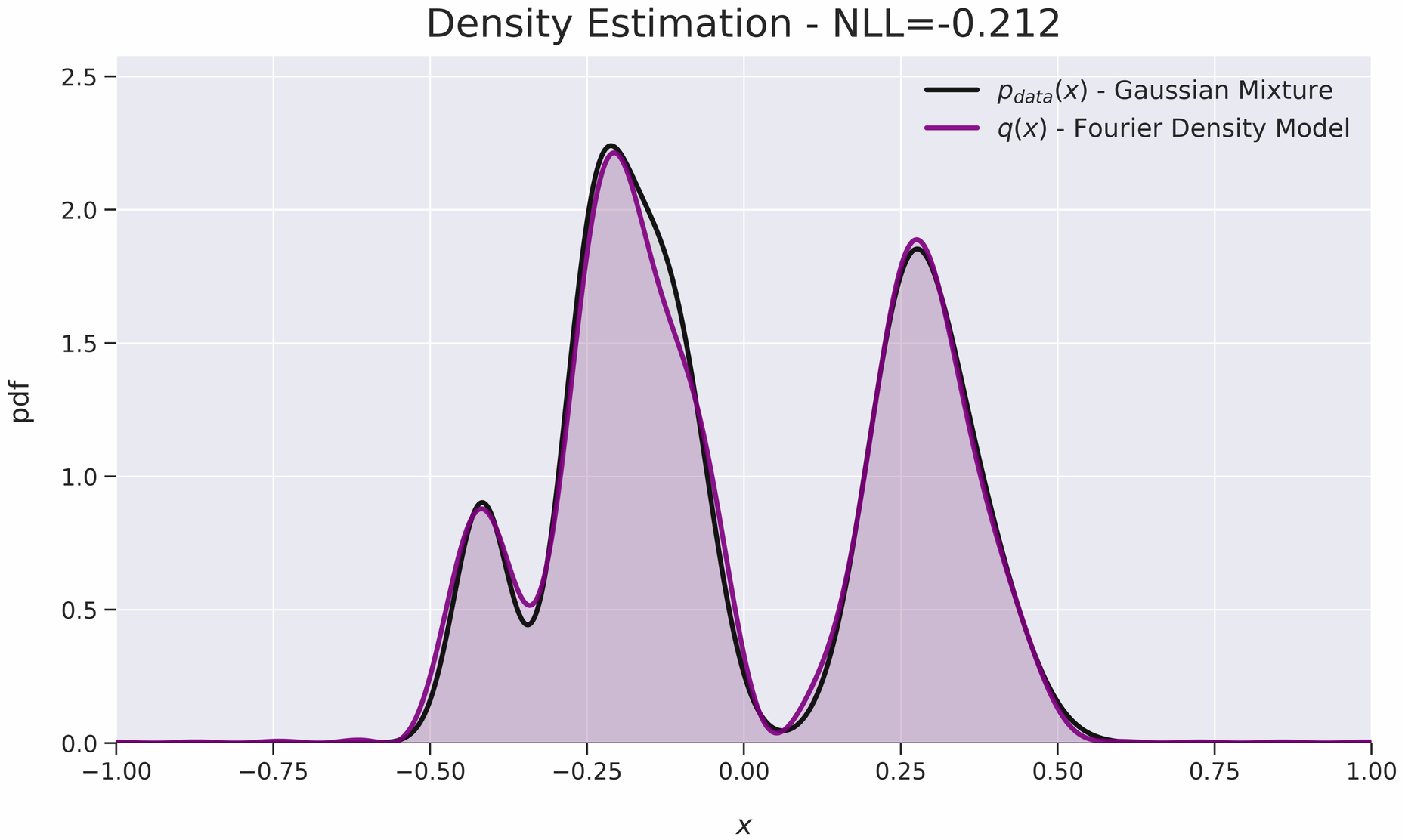
$$g : (-1,1) \to \mathbb{R}$$

$$g^{-1}(x;s,t) = \tanh\left(\frac{x-t}{s}\right)$$

$$q\left(x;\{a_n\}_0^N, s, t\right) = p\left(g^{-1}(x;s,t);\{a_n\}_0^N\right) \cdot (g^{-1})'(x;s,t)$$

De la Fuente, Singh & Ballé (PCS, 2024)

Density modeling: training example



How to draw samples from this density?

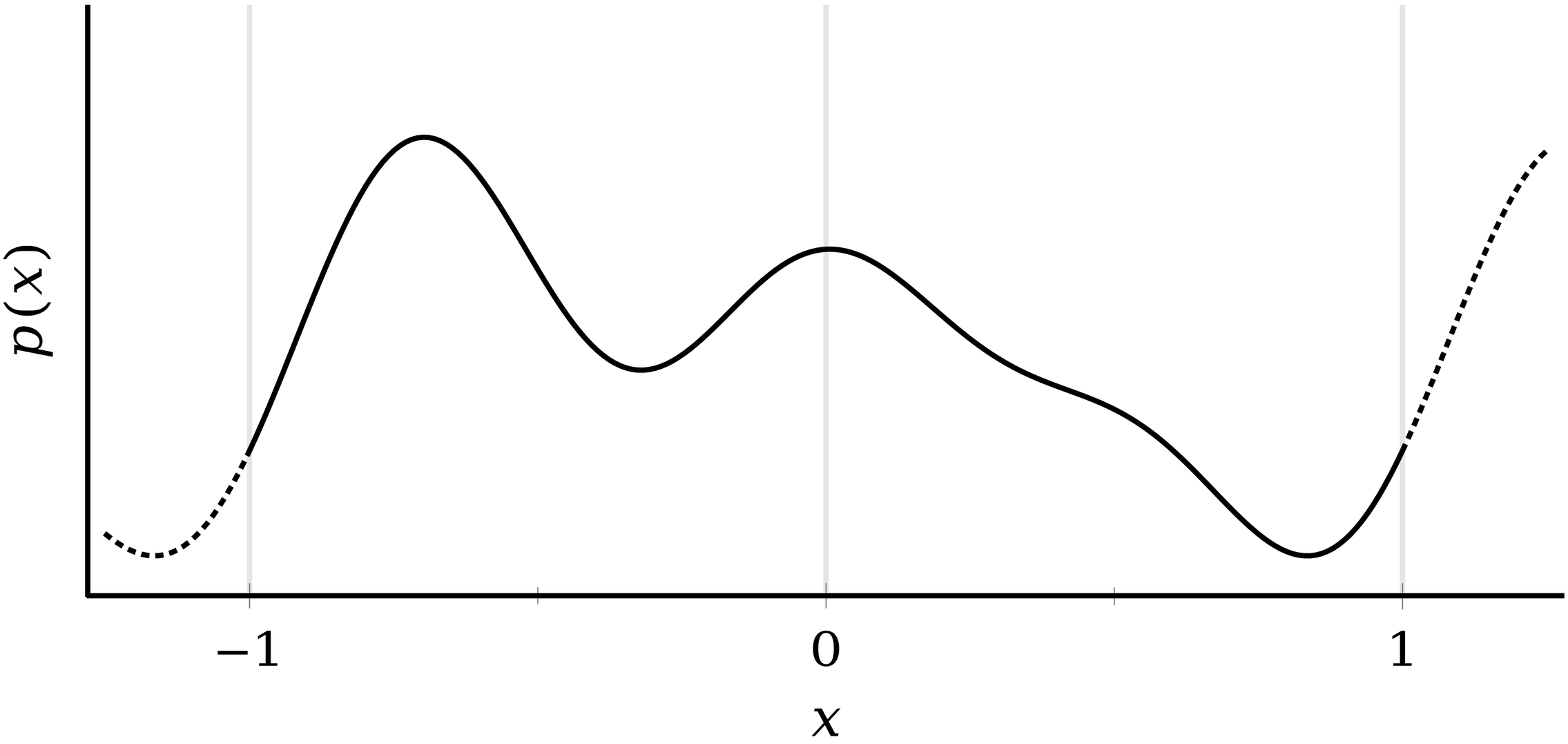
- **Inverse transform sampling:** Draw a sample U from a uniform distribution, then compute $P^{-1}(U)$, where P is the cumulative distribution function.
  - Not feasible since P^{-1} is not accessible in closed form.
- **Rejection sampling:** Define an envelope distribution $e(x)$ such that $M\ e(x) \geq p(x)$ everywhere. Then draw a sample X from $e(x)$, and reject samples based on the ratio $p(X)\ /\ (M\ e(X))$.
  - Difficult to determine $e(x)$ and M such that envelope is tight; if not tight, low acceptance rate makes it slow.

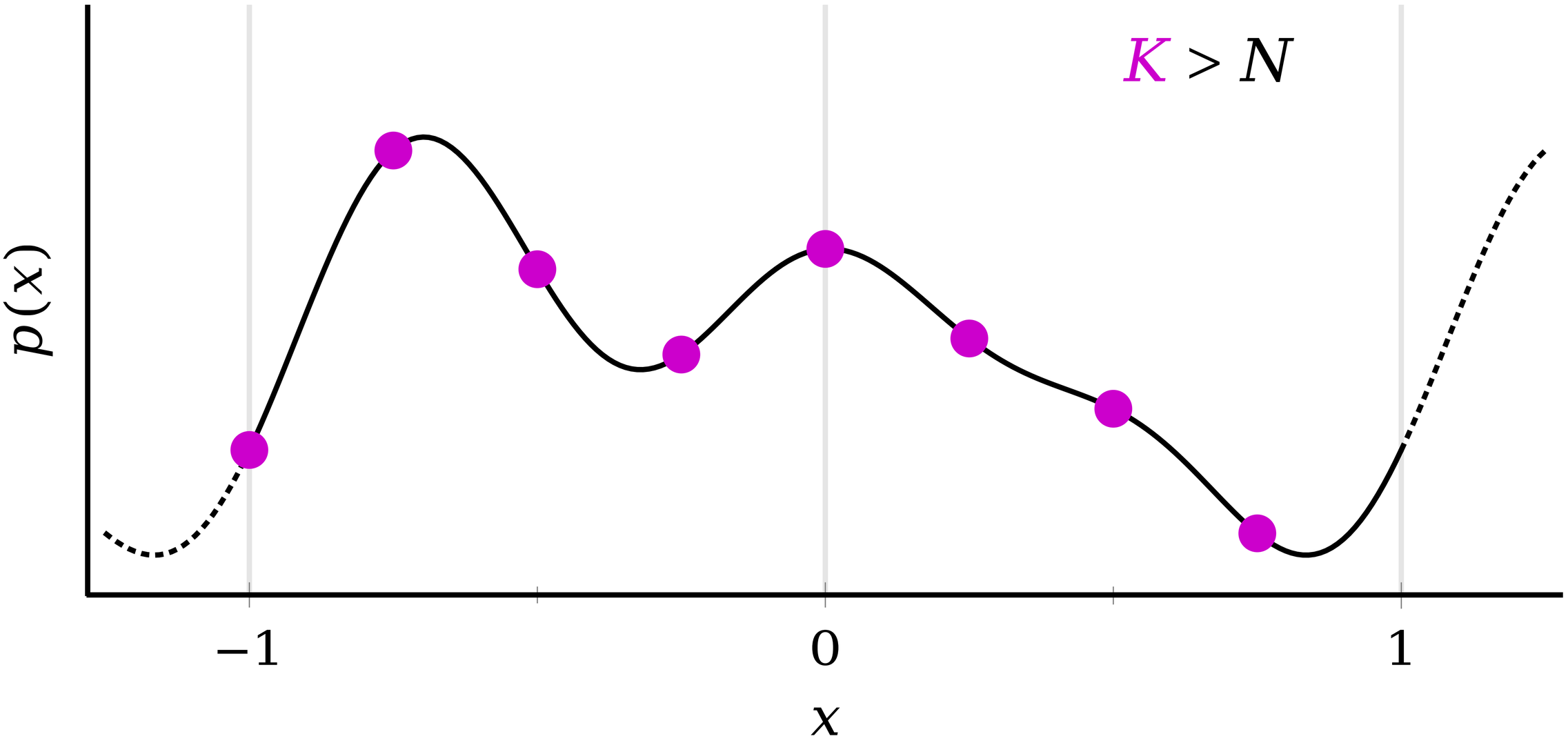
Speaking of sampling…

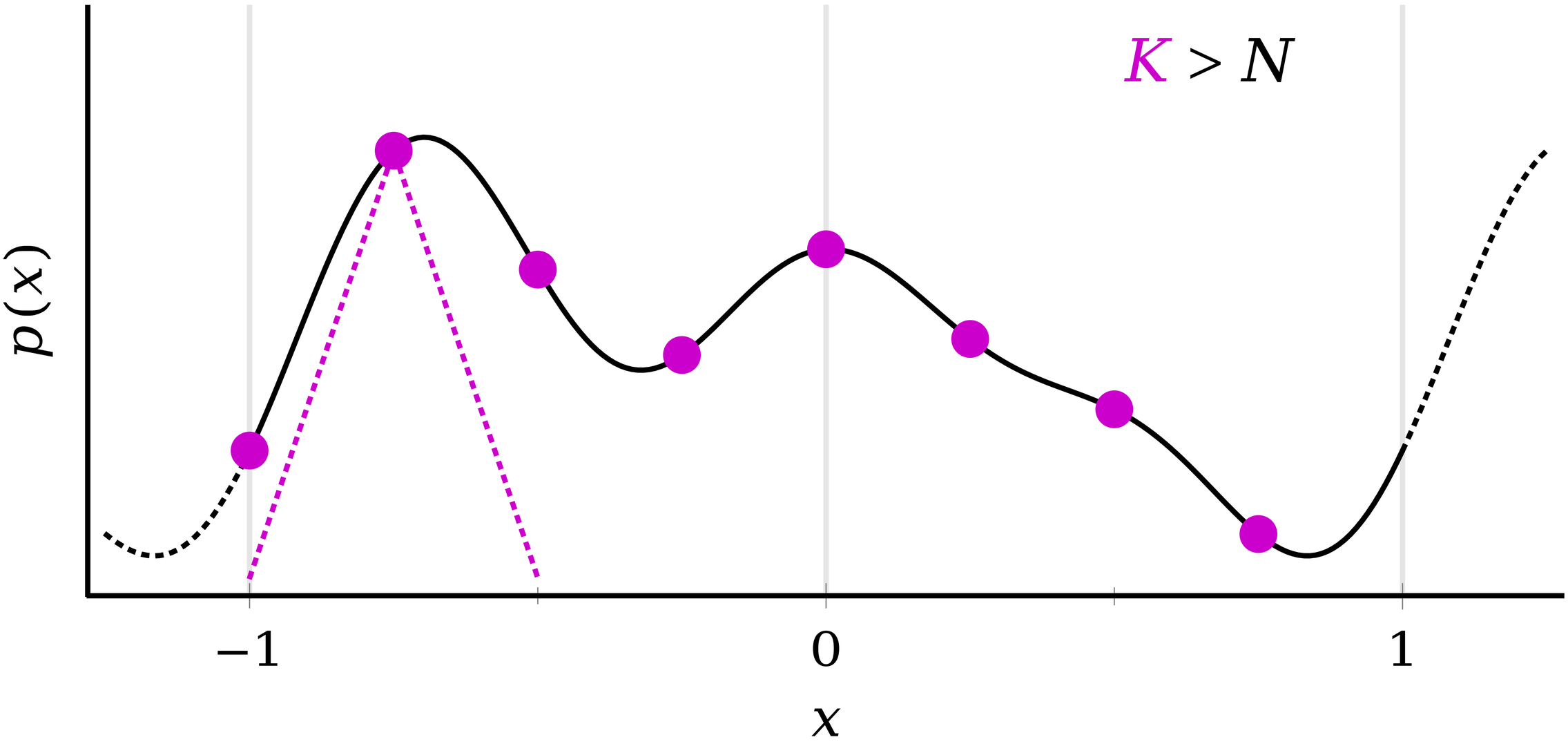
- In **Digital Signal Processing** (DSP), band-limited signals are sampled above the Nyquist rate, so they can be reconstructed with linear filtering.
- Here, the **density** is band-limited, instead of a signal.

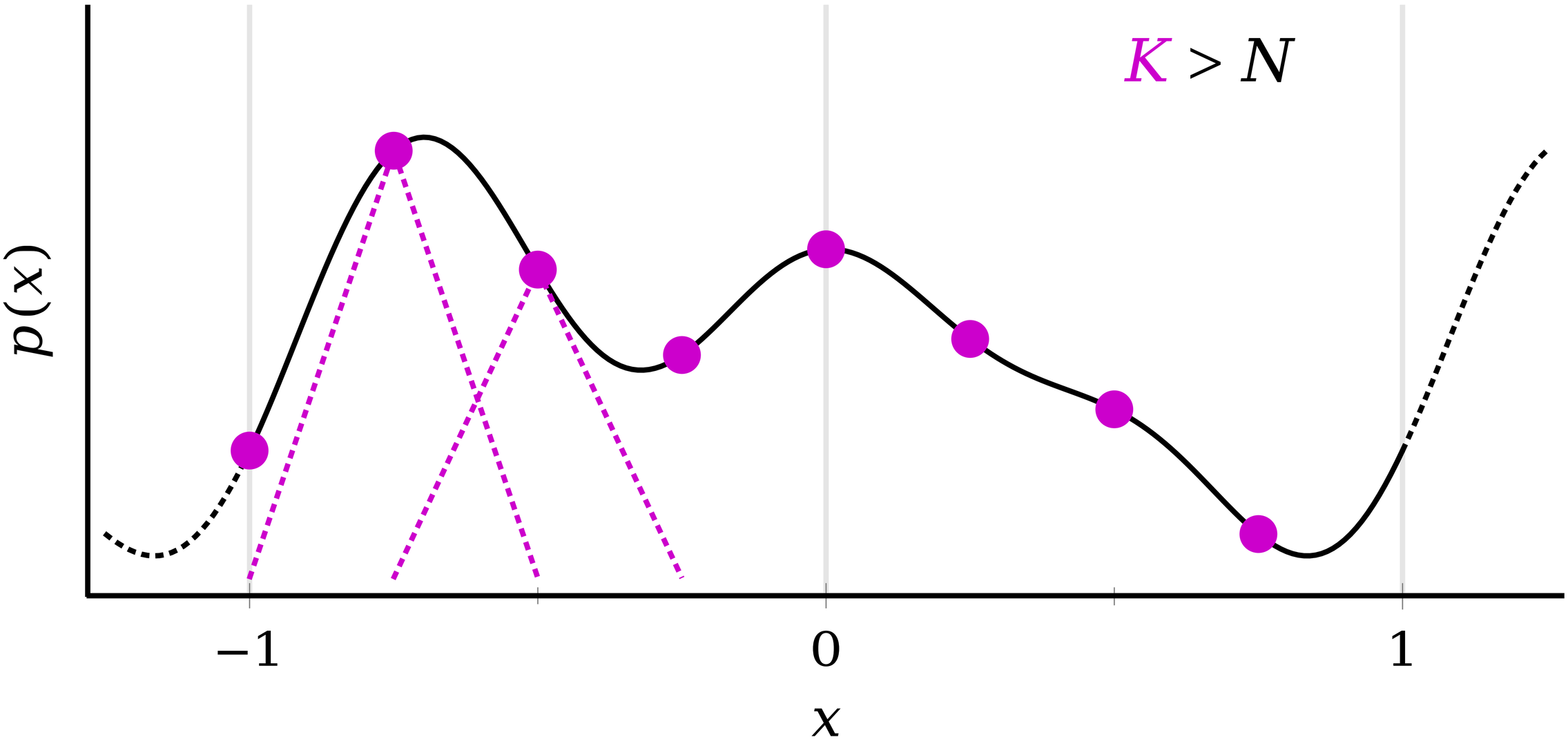
Speaking of sampling…

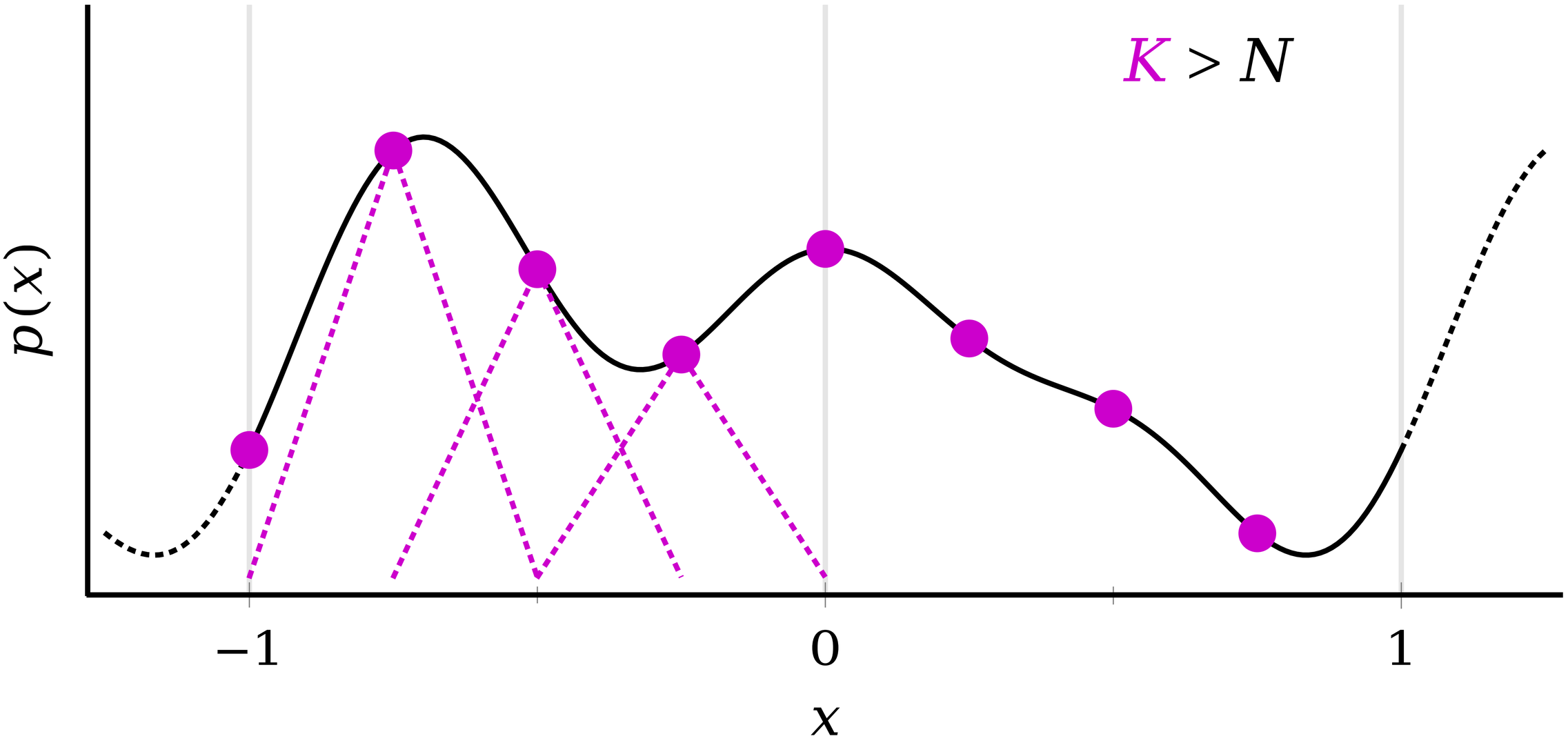
- In **Digital Signal Processing** (DSP), band-limited signals are sampled above the Nyquist rate, so they can be reconstructed with linear filtering.
- Here, the **density** is band-limited, instead of a signal.

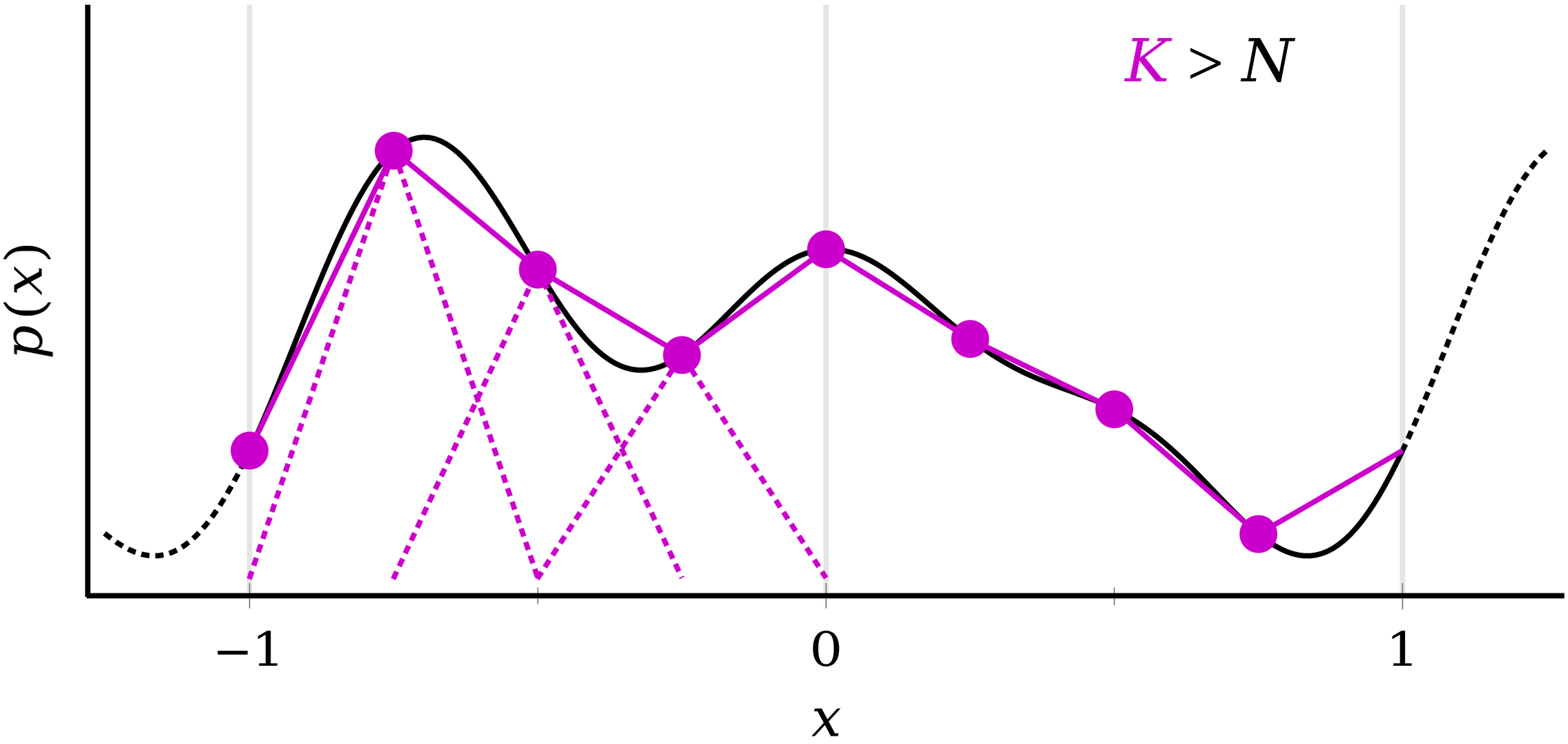

$p(x)$
-1
0
1
x


$K > N$
$p(x)$
-1
0
1
x


$K > N$
$p(x)$
-1
0
1
x

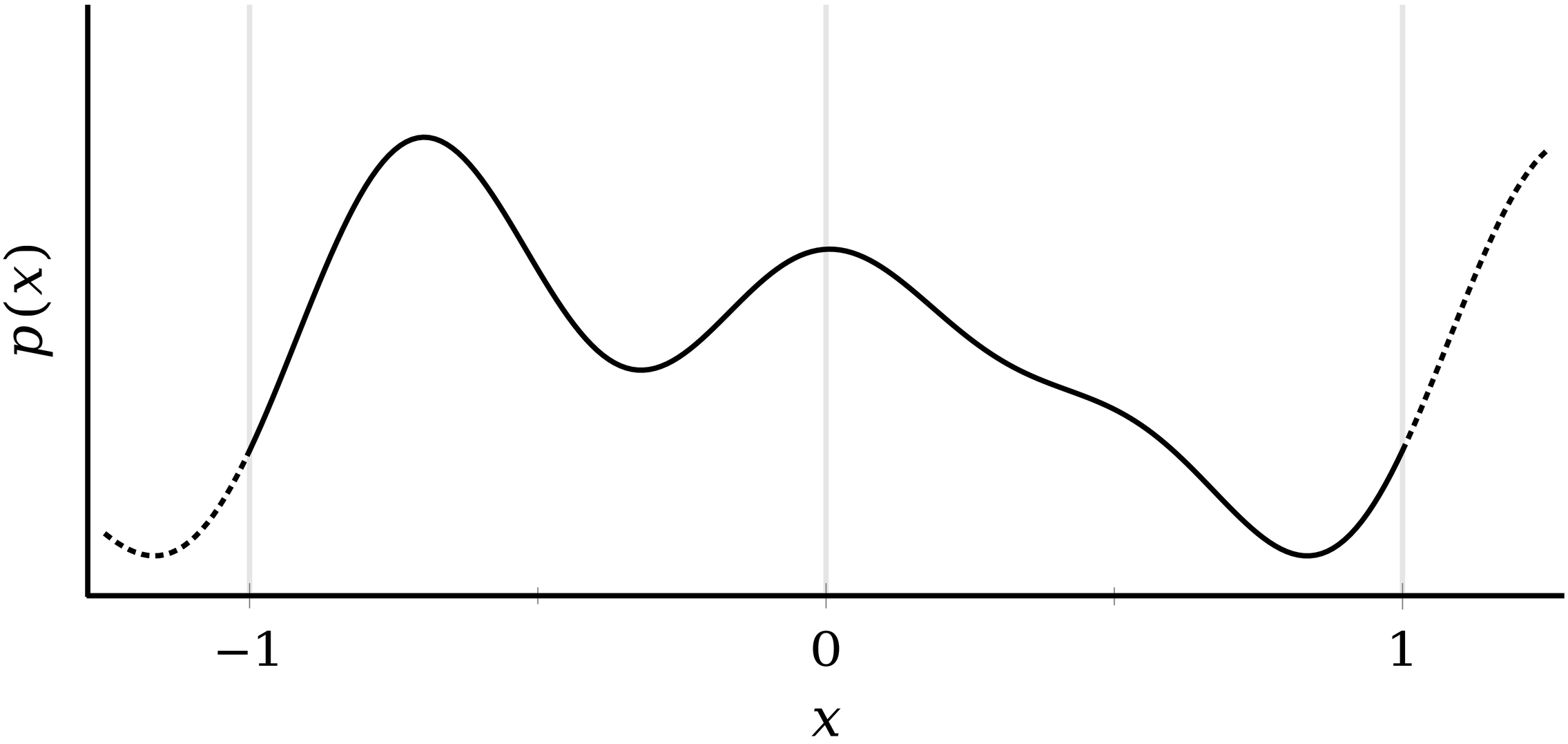

$K > N$
$p(x)$
-1
0
1
x

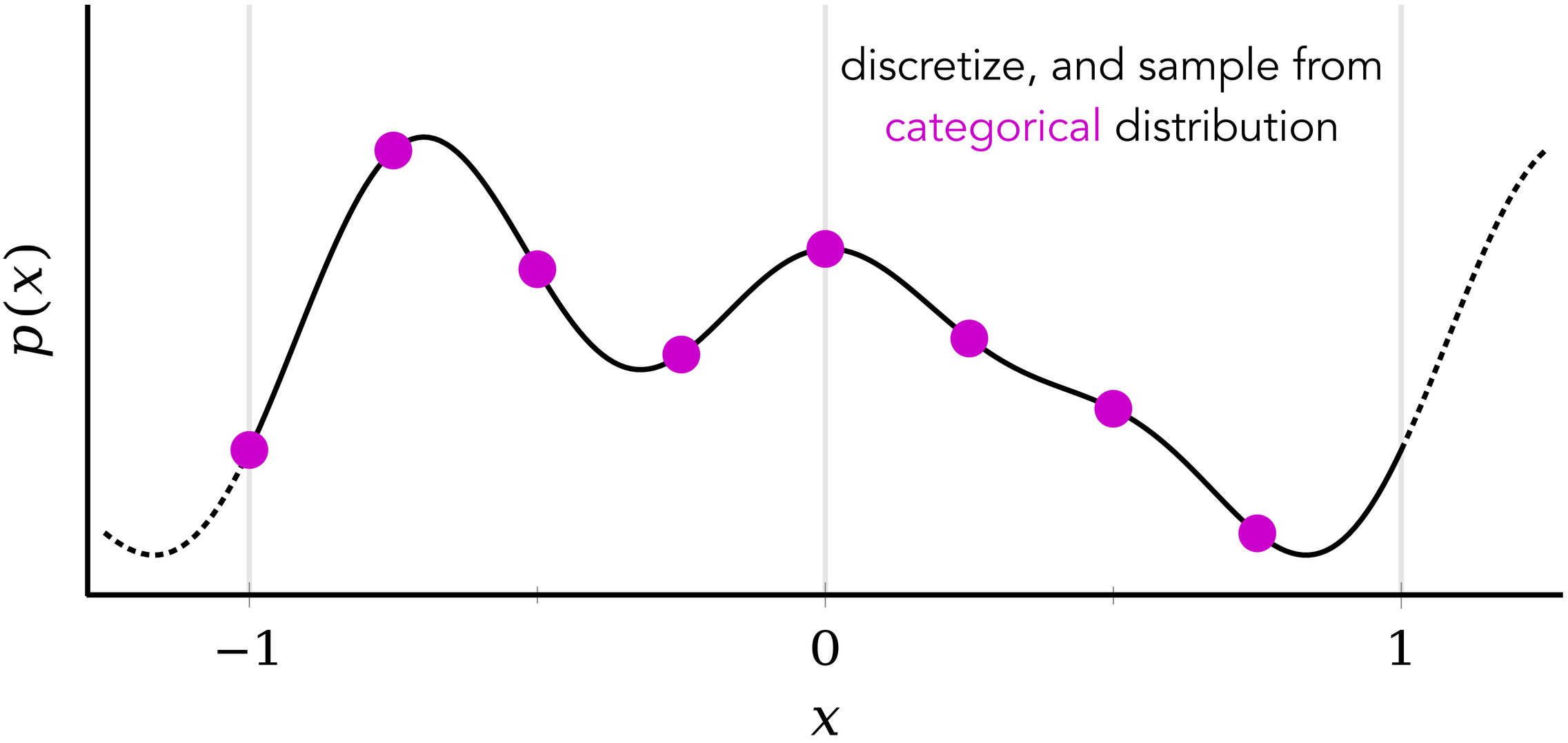

$K > N$
$p(x)$
-1
0
1
x

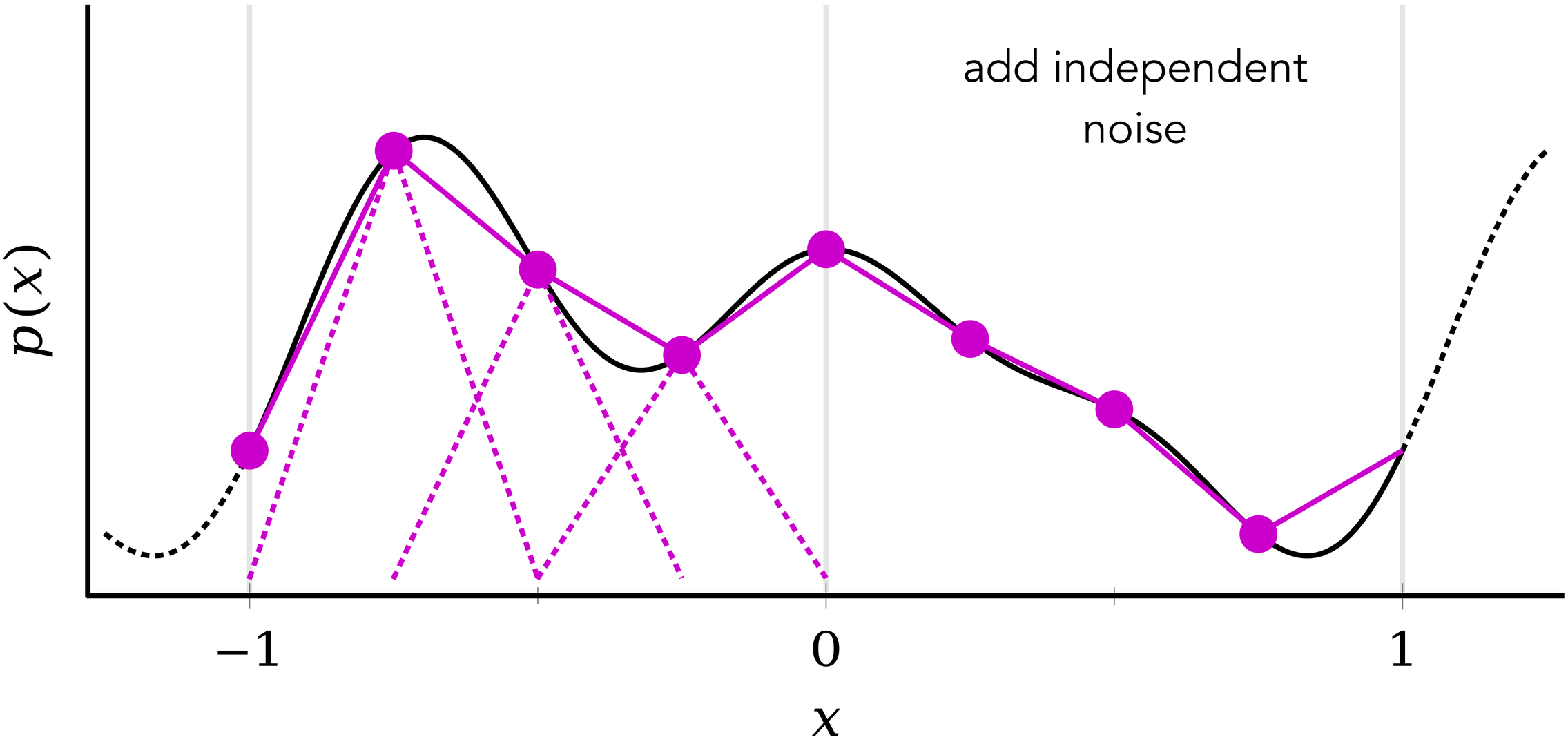

$K > N$
$p(x)$
-1
0
1
x

DSP vs. Discretized Approximate Ancestral Sampling

	DSP	DAAD
1. **discretize**	analog–digital converter	evaluate $p(x)$ (faster via DFT)
2. **process/transmit discrete values**	digital processing	sample from categorical
3. **convolve with lowpass filter** $w(x)$	digital–analog converter	add independent samples drawn from $w(x)$

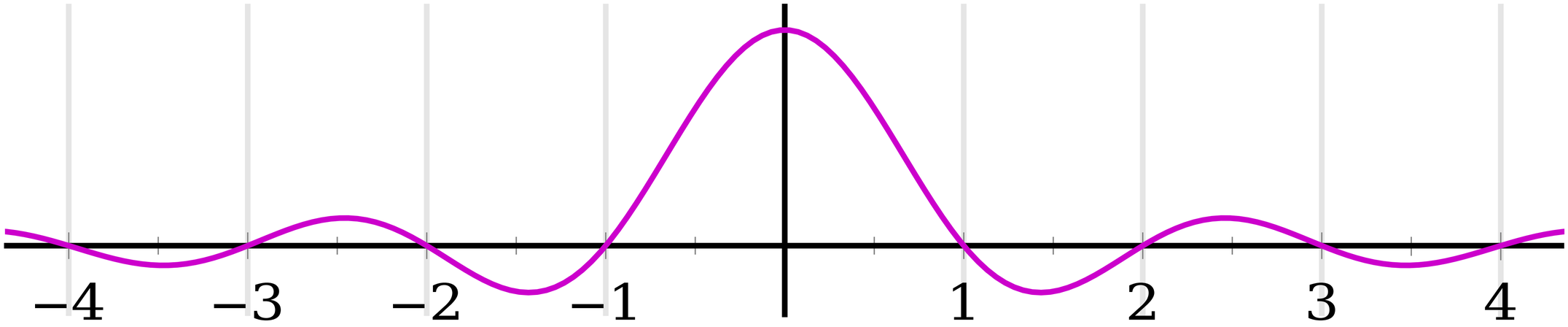

$p(x)$
-1
0
1
x


discretize, and sample from
categorical distribution
$p(x)$
-1
0
1
x


add independent
noise
$p(x)$
-1
0
1
x

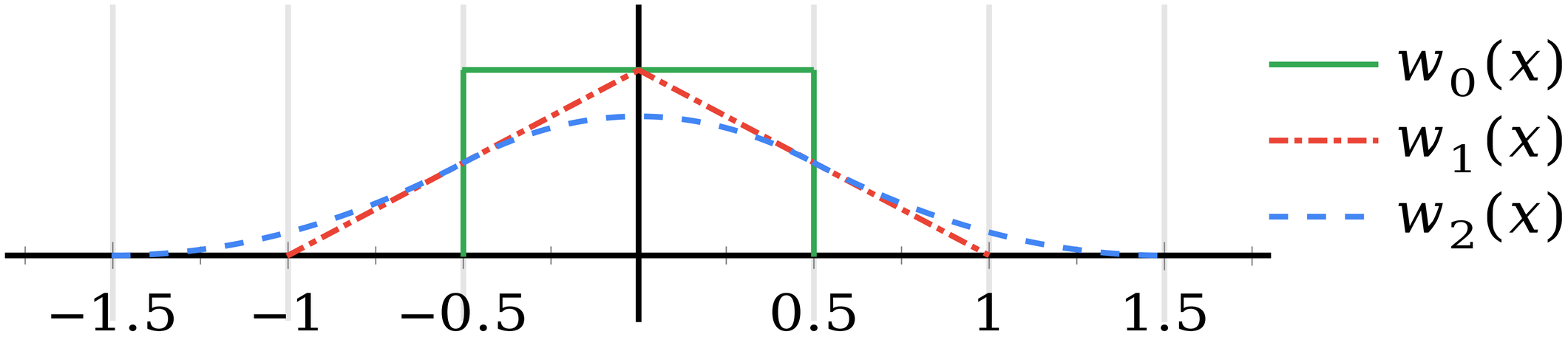
What reconstruction filter/density $w(x)$?

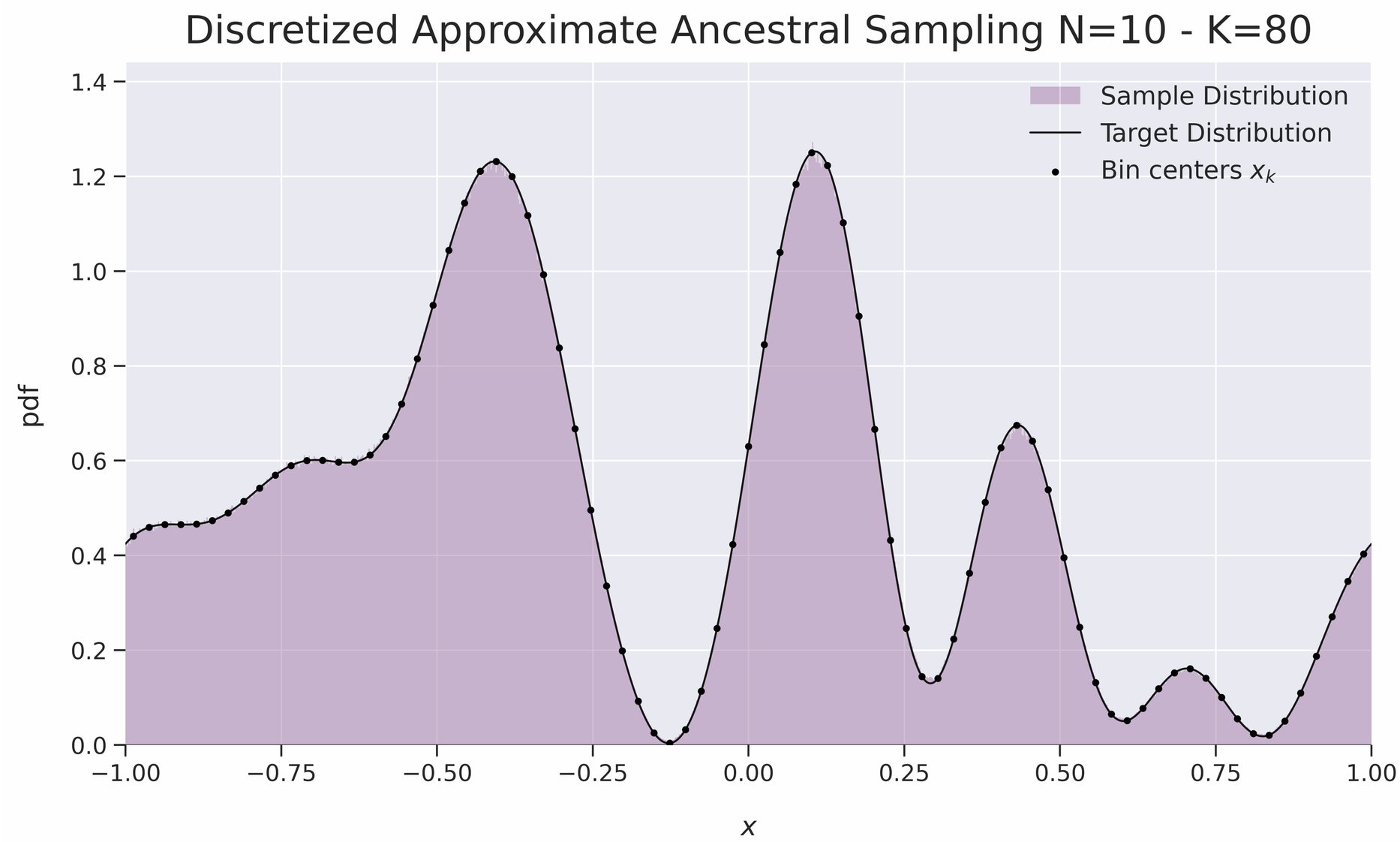
- Generally, we want $w(x)$ to be "close" to a $sinc$ function.
- This is usually not possible → filter design problem.
- In DSP, we need to have finite support.
- In DAAD, we need non-negativity (because it must be a distribution).



What reconstruction filter/density $w(x)$?

- B-splines are a simple and nice choice for DAAD.
- i-th order B-spline kernel: convolve a box function $i - 1$ times with itself.
- Simply take $i - 1$ i.i.d. samples from a uniform distribution, and add them.




Discretized Approximate Ancestral Sampling N=10 - K=80
Sample Distribution
Target Distribution
Bin centers x_k
pdf
x
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

Conclusion

- DAAD is a simple method for sampling from band-limited distributions.
- Strong analogy to Digital Signal Processing, but some of the practical requirements are inherently different (finite support vs. non-negativity).
- I’ve skipped many details here – in the paper, we also have:
  - Accuracy bounds for distribution of samples.
  - Empirical evaluation.
  - Refinement using MCMC with Langevin Dynamics.

Thank you!

https://arxiv.org/abs/2505.06098