Neural Distributed Compressor Does 'Binning'

Ezgi Ozyilkan

Neural Compression Workshop @ ICML 2023
Honolulu, HI | July 29, 2023

Neural Distributed Compressor Does 'Binning'

Ezgi Ozyilkan
Joint work with Johannes Ballé (Google Research)

Neural Compression Workshop @ ICML 2023
Honolulu, HI | July 29, 2023

Google Research

Neural Distributed Compressor Does 'Binning'

Ezgi Ozyilkan
Joint work with Johannes Ballé (Google Research) and Elza Erkip (NYU)

Neural Compression Workshop @ ICML 2023
Honolulu, HI | July 29, 2023

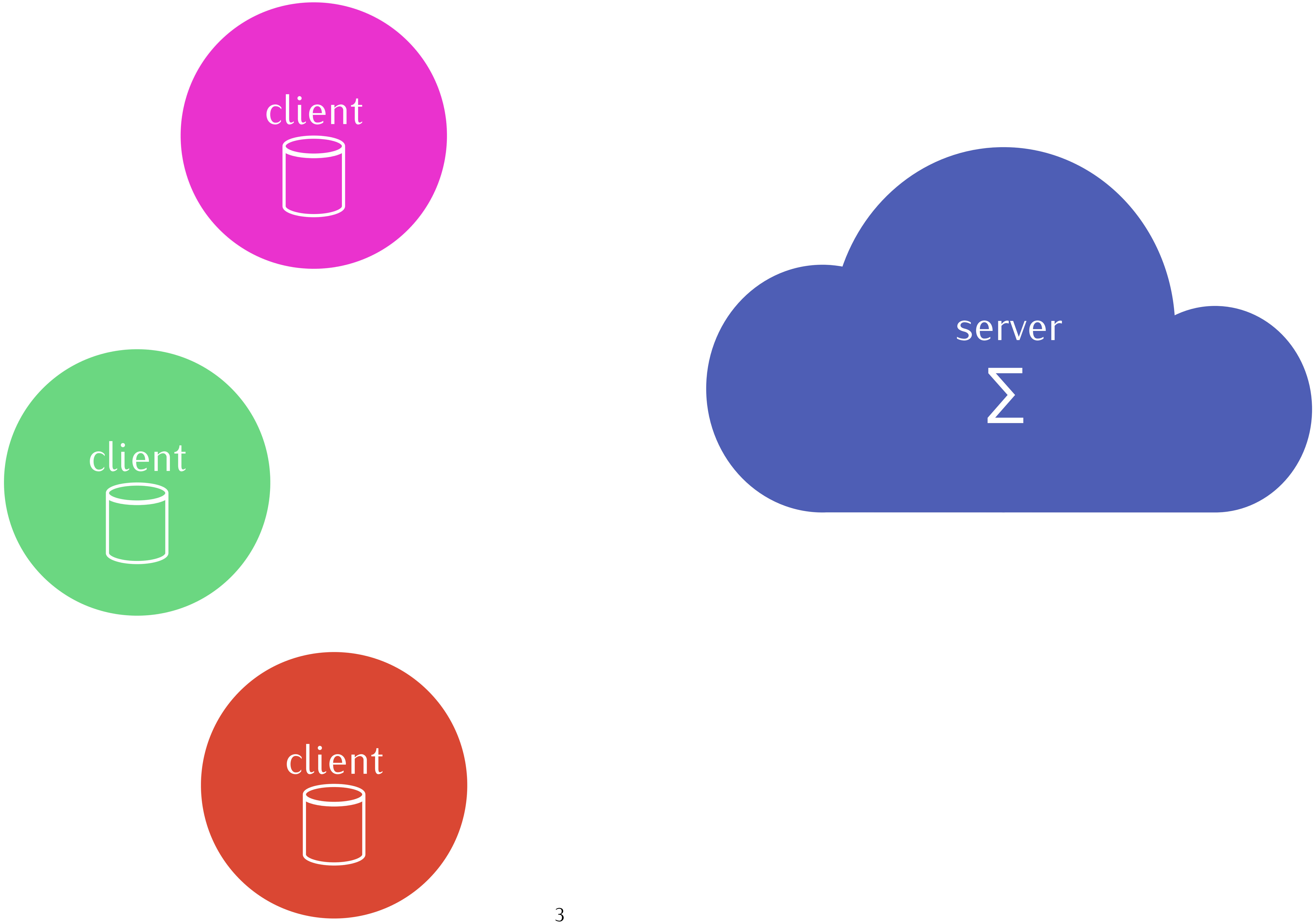
Google Research

NYU | TANDON SCHOOL OF ENGINEERING

Distributed Source Coding

Motivation: Distributed Source Coding

e.g., next-word prediction



Motivation: Distributed Source Coding

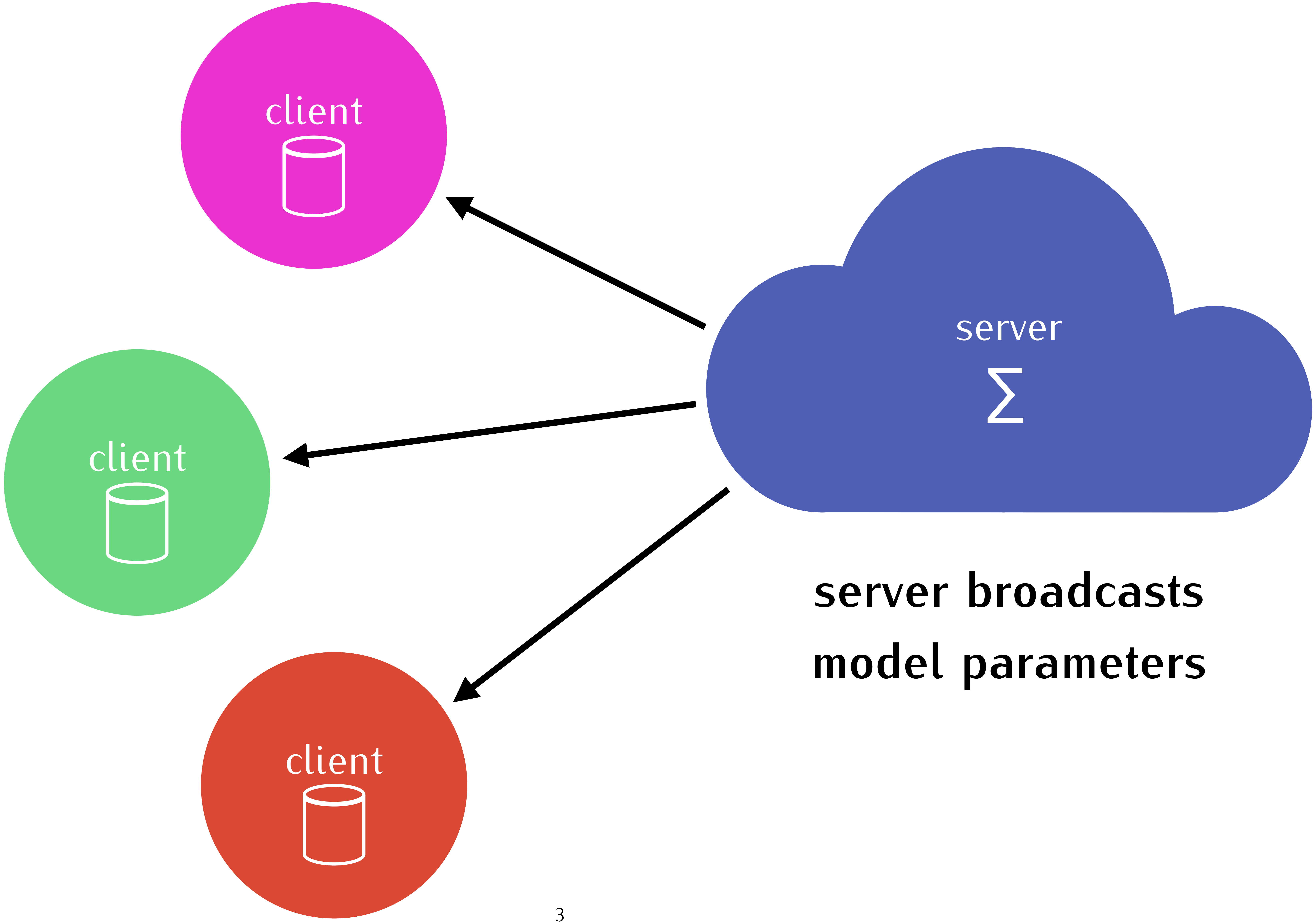
Federated learning.

e.g., next-word prediction

Motivation: Distributed Source Coding

Federated learning.

e.g., next-word prediction

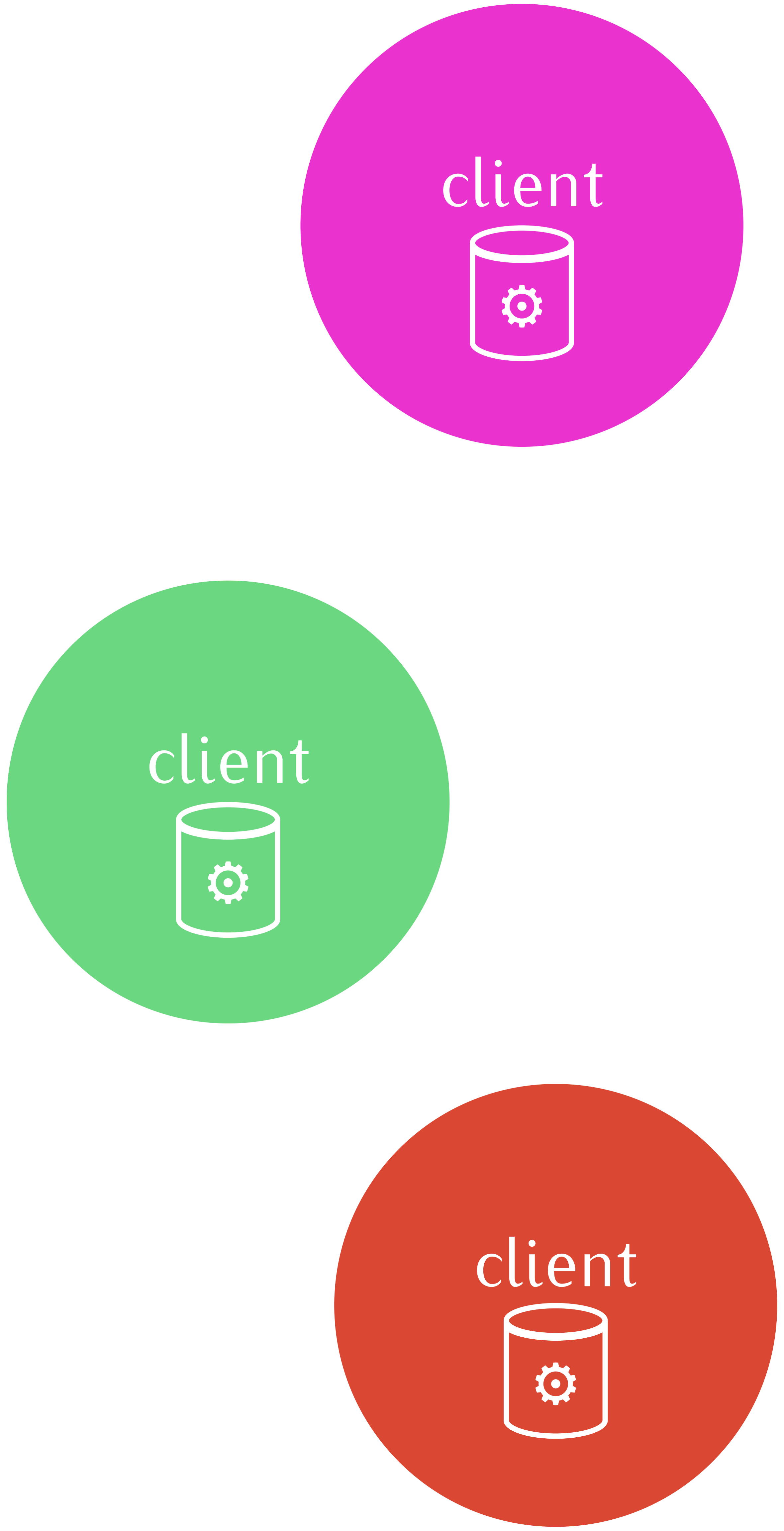


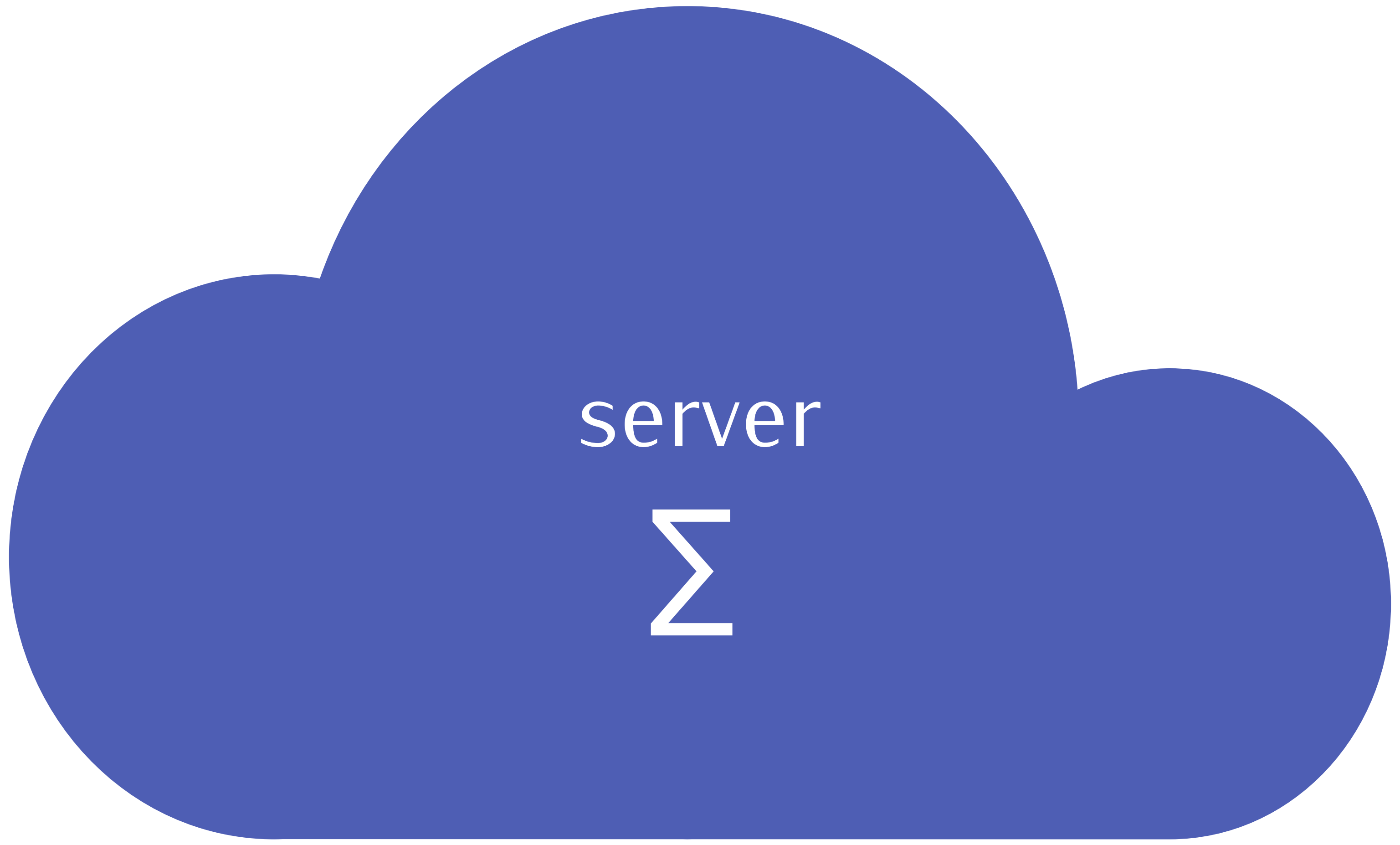
server broadcasts
model parameters

Motivation: Distributed Source Coding

Federated learning.

e.g., next-word prediction



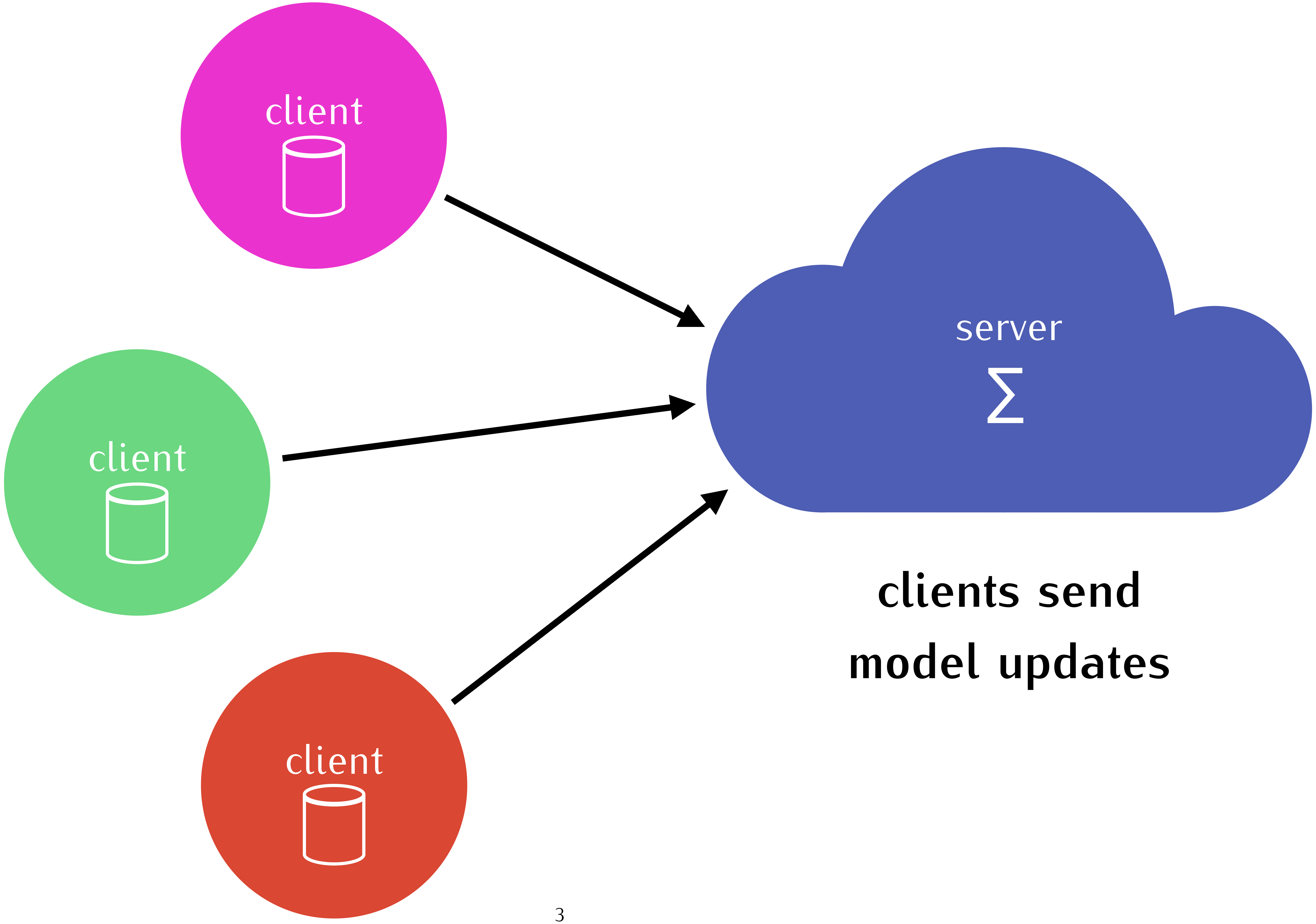


clients update their models based on local data

Motivation: Distributed Source Coding

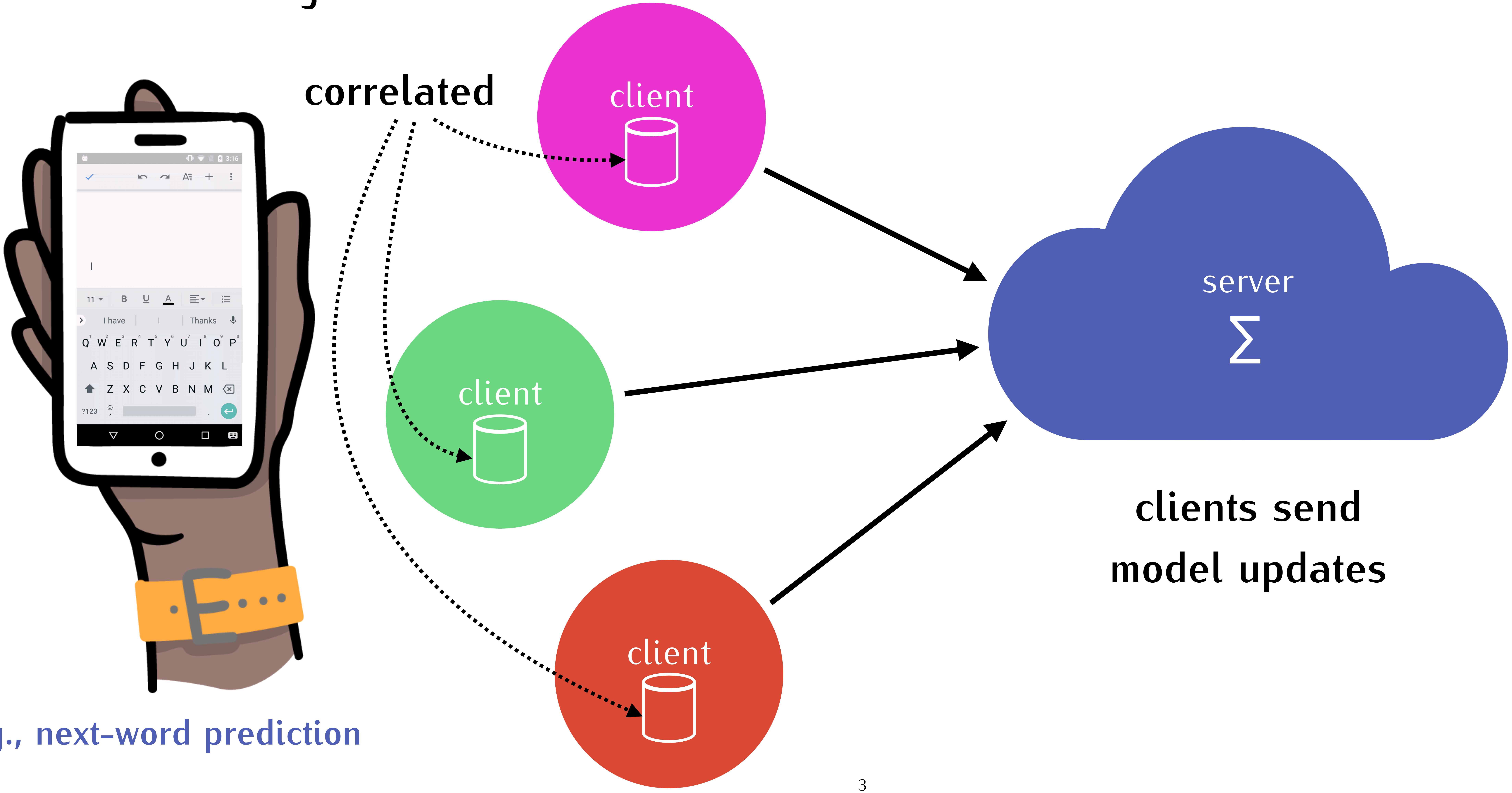
Federated learning.

e.g., next-word prediction



Motivation: Distributed Source Coding

Federated learning.



e.g., next-word prediction

50th year Commemorative Special Issue of the Transactions of Information Theory notes that

50th year Commemorative Special Issue of the Transactions of Information Theory notes that

"[...] despite the existence of potential applications, the conceptual importance of distributed source coding has not been mirrored in **practical data compression**."

S. Verdú, "Fifty years of Shannon theory", *IEEE Transactions on Information Theory*, 1998.

50th year Commemorative Special Issue of the Transactions of Information Theory notes that

"[...] despite the existence of potential applications, the conceptual importance of distributed source coding has not been mirrored in **practical data compression**."

S. Verdú, "Fifty years of Shannon theory", *IEEE Transactions on Information Theory,* 1998.

50th year Commemorative Special Issue of the Transactions of Information Theory notes that

"[...] despite the existence of potential applications, the conceptual importance of distributed source coding has not been mirrored in **practical data compression**."

Still, the case after 25 years.

S. Verdú, "Fifty years of Shannon theory", *IEEE Transactions on Information Theory*, 1998.

50th year Commemorative Special Issue of the Transactions of Information Theory notes that

"[...] despite the existence of potential applications, the conceptual importance of distributed source coding has not been mirrored in **practical data compression**."

Still, the case after 25 years.

Particularly, for *general sources.*

S. Verdú, "Fifty years of Shannon theory", *IEEE Transactions on Information Theory,* 1998.

50th year Commemorative Special Issue of the Transactions of Information Theory notes that

"[...] despite the existence of potential applications, the conceptual importance of distributed source coding has not been mirrored in **practical data compression**."

Still, the case after 25 years.

Particularly, for *general sources.*

Learning-based compressors (e.g., Ballé et al., 2017) may help.

S. Verdú, "Fifty years of Shannon theory", *IEEE Transactions on Information Theory*, 1998.

J. Ballé et al., "End-to-end Optimized Image Compression", *International Conference on Learning Representations* (ICLR), 2017.

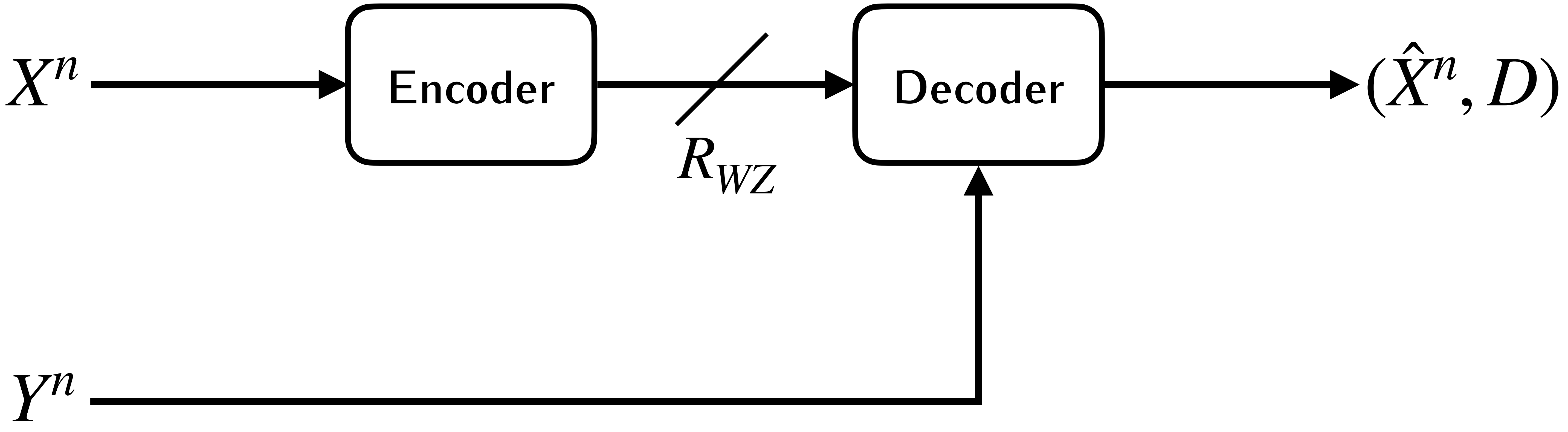
Simpler special case: Rate-distortion (R-D) with side information

Simpler special case: Rate-distortion (R-D) with side information

Known as the Wyner-Ziv (WZ) problem.

Simpler special case: Rate-distortion (R-D) with side information

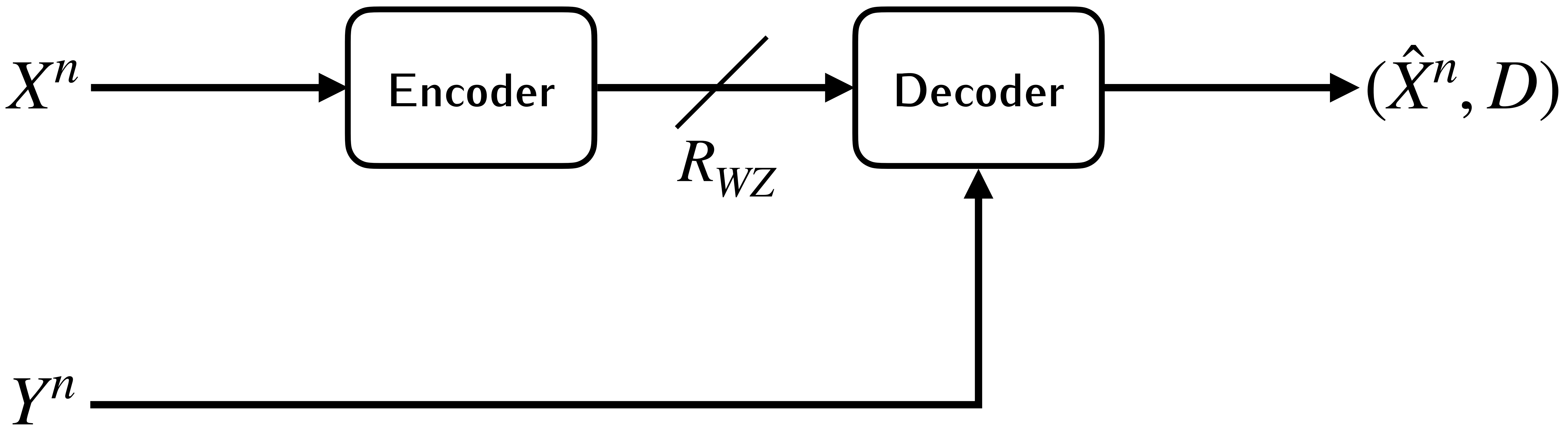
Known as the Wyner-Ziv (WZ) problem.



A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory,* 1976.

Simpler special case: Rate-distortion (R-D) with side information

Known as the Wyner-Ziv (WZ) problem.



Theorem. Let (X, Y) be correlated i.i.d. $\sim p(x, y)$, and let $d(x, \hat{x})$ be a distortion measure. The R-D function for X when Y available at the decoder is:

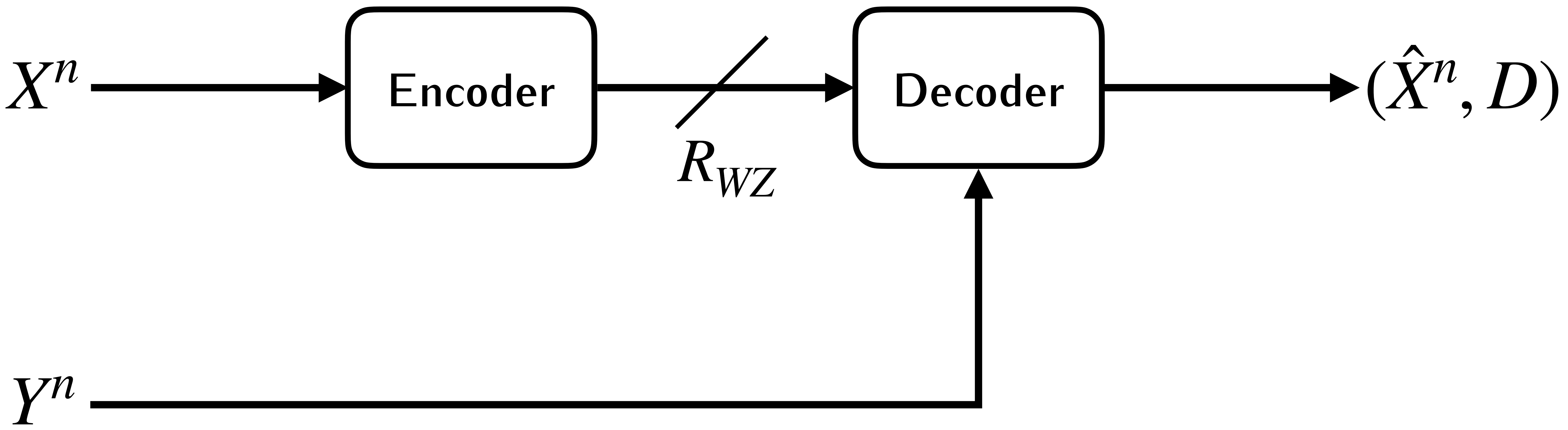
$$R_{WZ}(D) = \min(I(X; U) - I(Y; U)),$$

where the minimization is over all $p(u \mid x)$ and all functions $g(u, y)$ satisfying $\mathbb{E}_{p(x,y)p(u|x)} d(x, g(u, y)) \leq D$.

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory,* 1976.

Simpler special case: Rate-distortion (R-D) with side information

Known as the Wyner-Ziv (WZ) problem.



<u>Theorem.</u> Let (X, Y) be correlated i.i.d. $\sim p(x, y)$, and let $d(x, \hat{x})$ be a distortion measure. The R-D function for X when Y available at the decoder is:

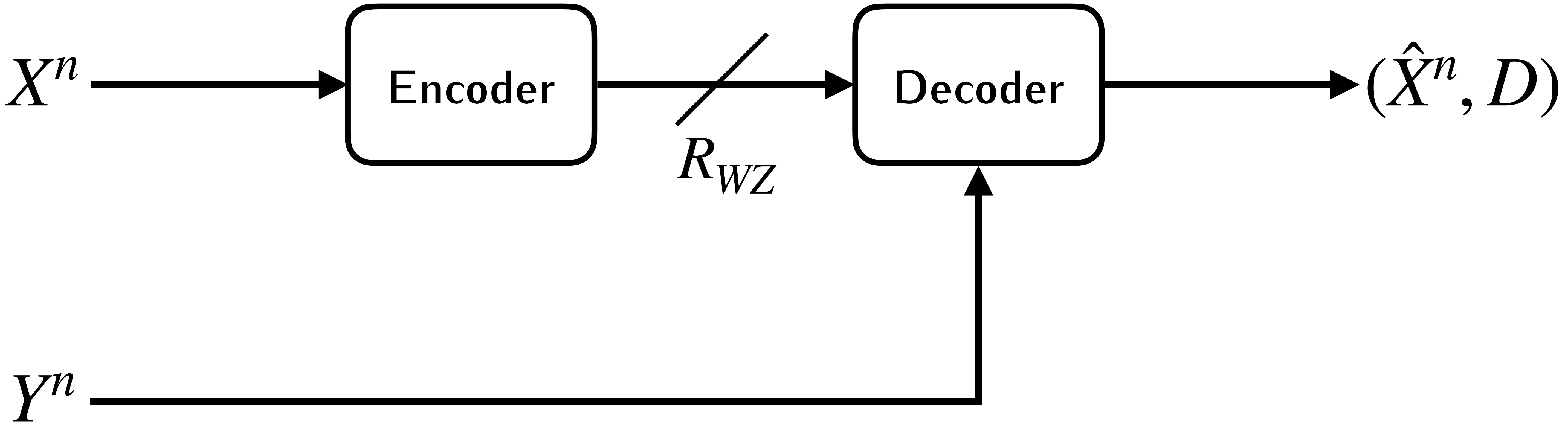
$$R_{WZ}(D) = \min(I(X;U) - I(Y;U)),$$

where the minimization is over all $p(u \mid x)$ and all functions $g(u, y)$ satisfying $\mathbb{E}_{p(x,y)p(u|x)} d(x, g(u, y)) \leq D$.

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory*, 1976.

Simpler special case: Rate-distortion (R-D) with side information

Known as the Wyner-Ziv (WZ) problem.



<u>Theorem.</u> Let (X, Y) be correlated i.i.d. $\sim p(x, y)$, and let $d(x, \hat{x})$ be a distortion measure. The R-D function for X when Y available at the decoder is:

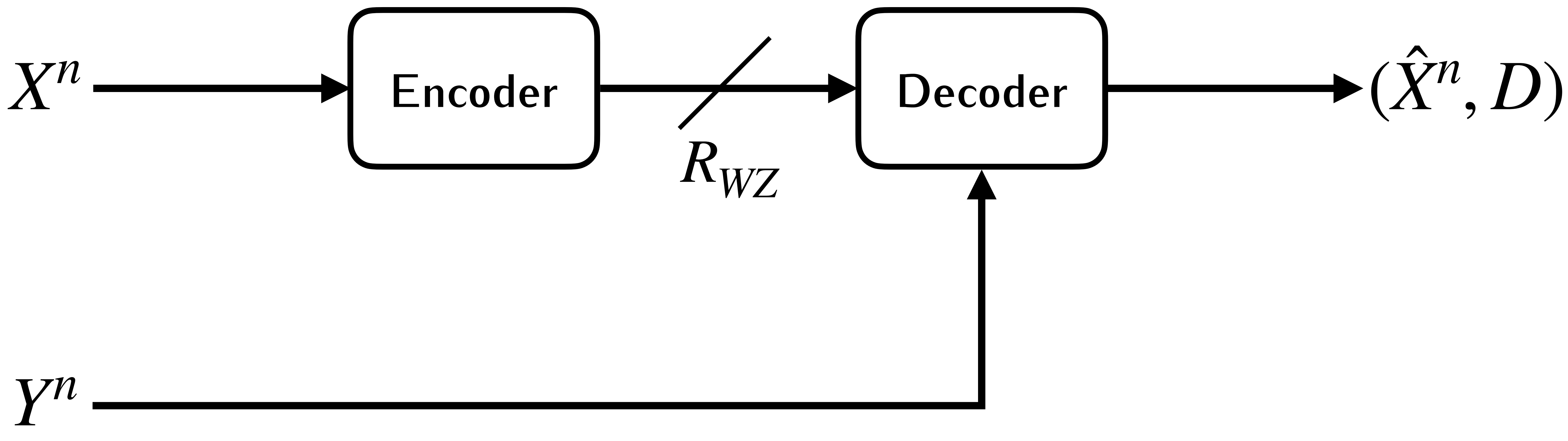
$$R_{WZ}(D) = \min(I(X; U) - I(Y; U)),$$

where the minimization is over all $p(u \mid x)$ and all functions $g(u, y)$ satisfying $\mathbb{E}_{p(x,y)p(u|x)} d(x, g(u, y)) \leq D$.

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder'',
IEEE Transactions on Information Theory, 1976.

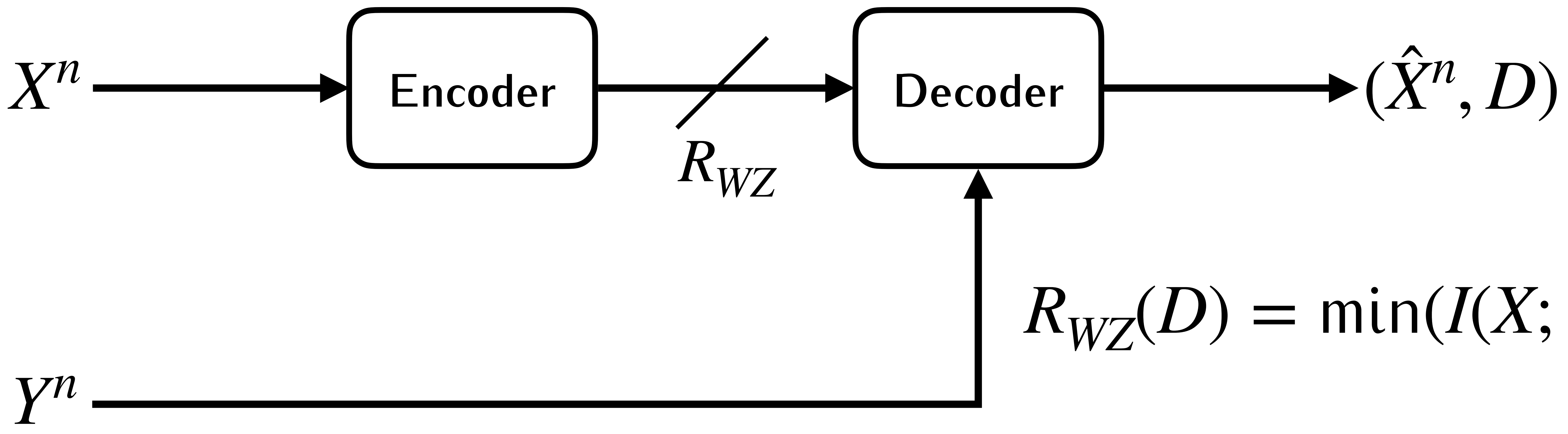
Wyner-Ziv achievability

Wyner-Ziv achievability



A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder",
IEEE Transactions on Information Theory, 1976.

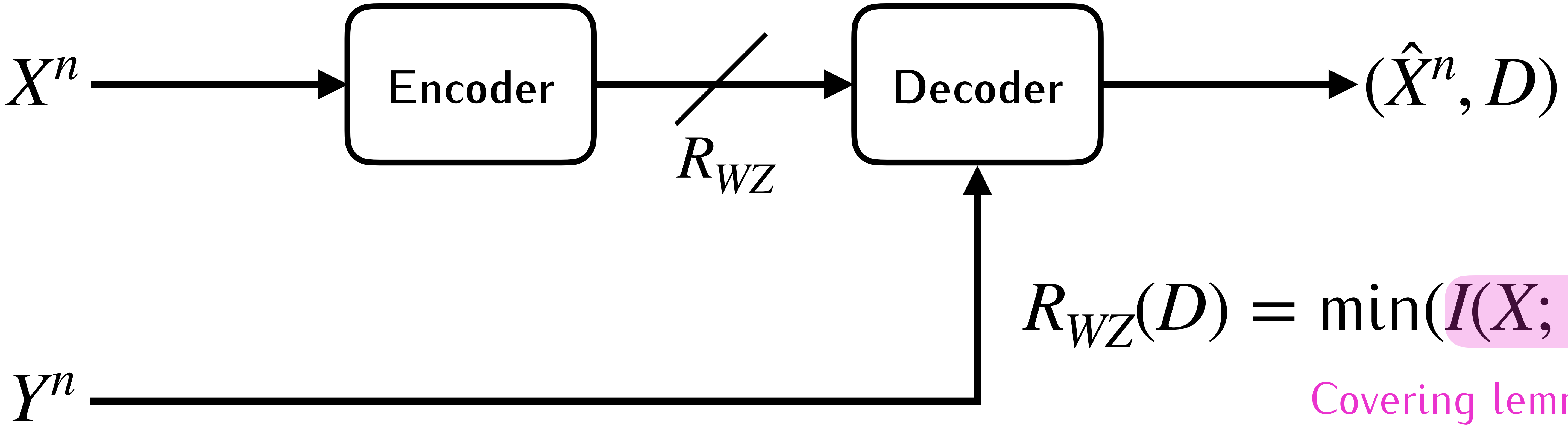
Wyner-Ziv achievability



$$R_{WZ}(D) = \min(I(X;U) - I(Y;U))$$

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory*, 1976.

Wyner-Ziv achievability

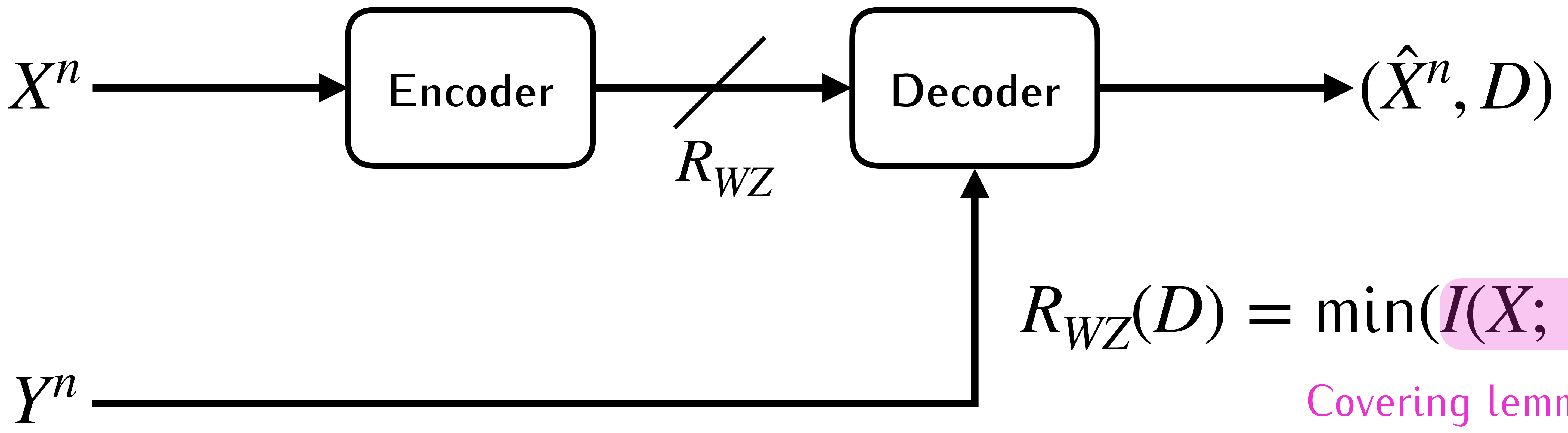


$$R_{WZ}(D) = \min(I(X;U) - I(Y;U))$$

Covering lemma

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory,* 1976.

Wyner-Ziv achievability

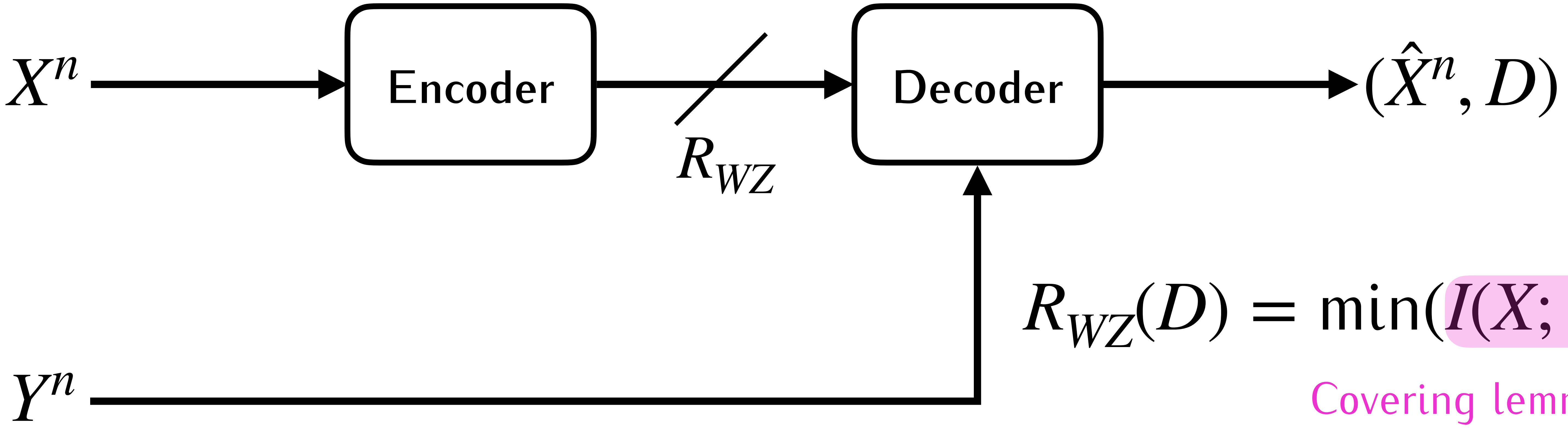


$$R_{WZ}(D) = \min(I(X;U) - I(Y;U))$$

Covering lemma ∝ random binning

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory,* 1976.

Wyner–Ziv achievability



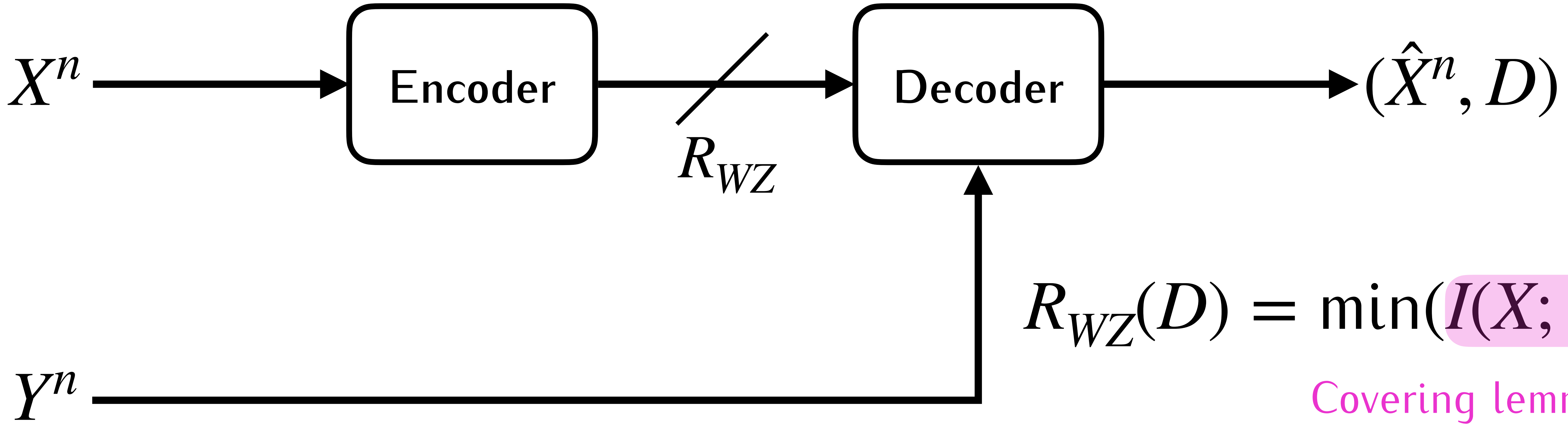
$$R_{WZ}(D) = \min\left(I(X;U) - I(Y;U)\right)$$

Covering lemma   ∝ random binning

'discount'

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory,* 1976.

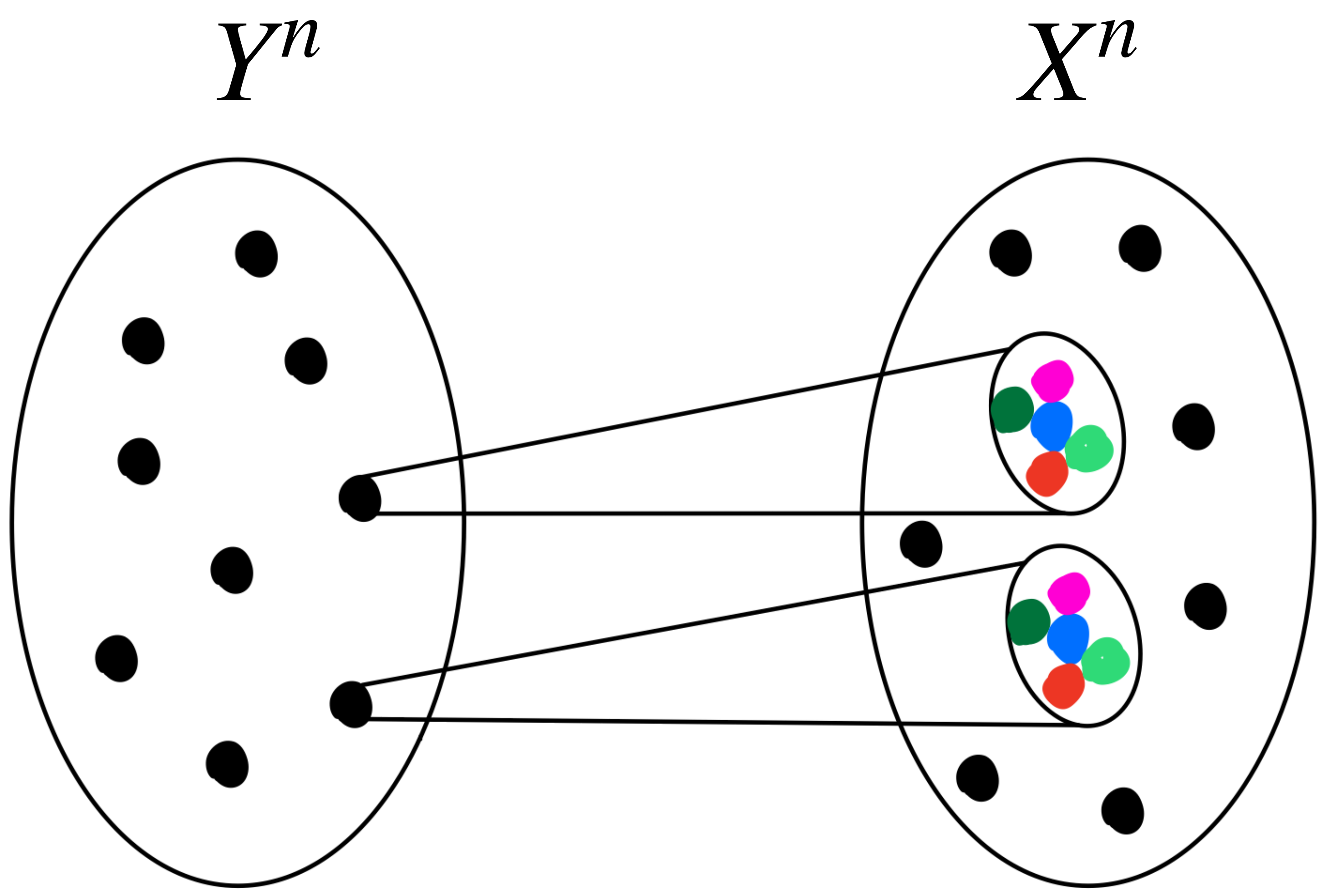
Wyner-Ziv achievability



$$R_{WZ}(D) = \min(I(X;U) - I(Y;U))$$

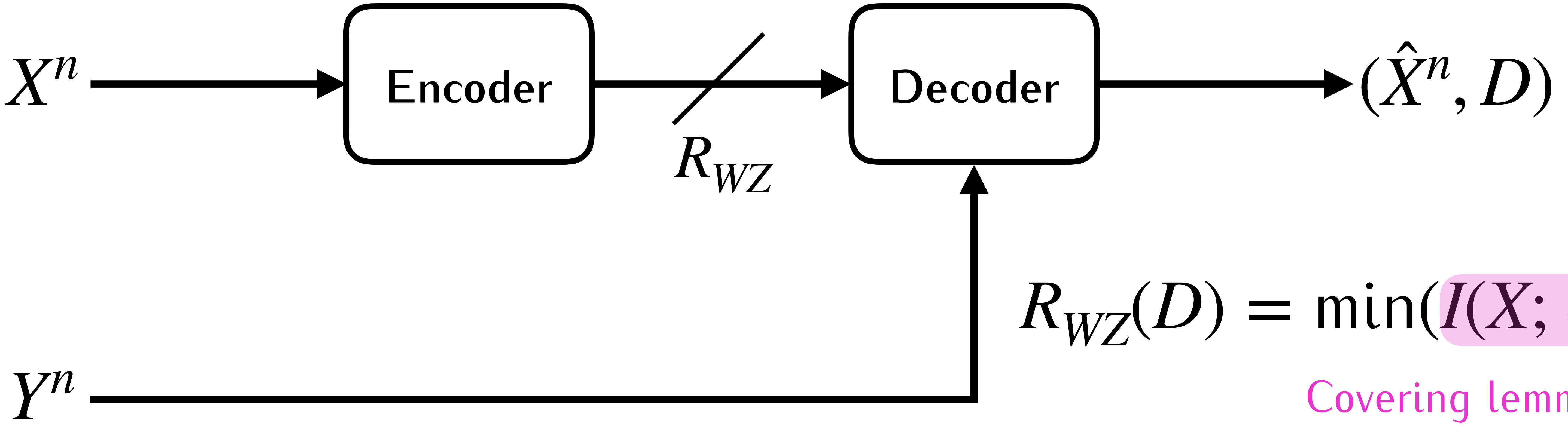
Covering lemma $\quad\propto$ random binning

'discount'



A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory,* 1976.

Wyner-Ziv achievability

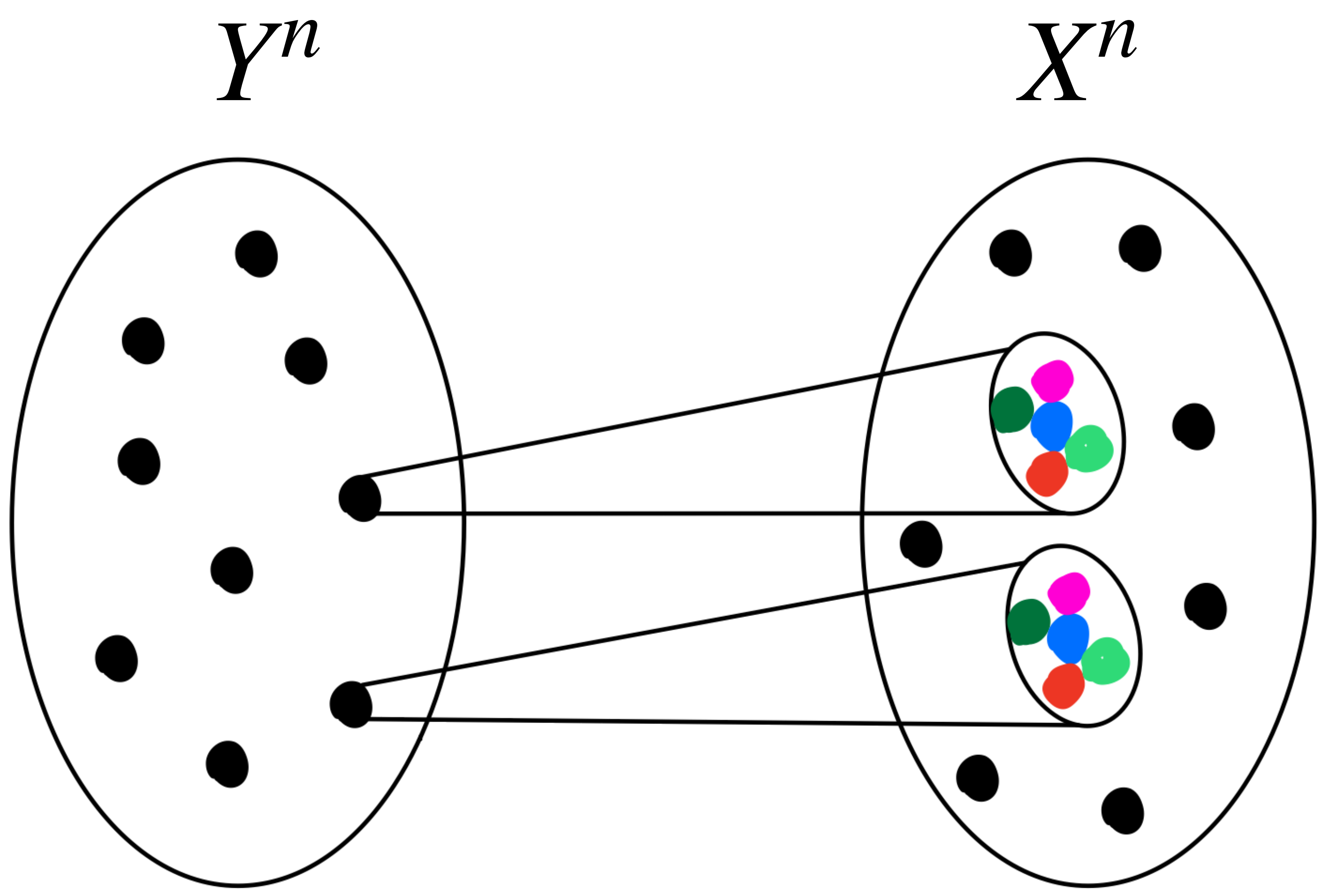


$$R_{WZ}(D) = \min\left(I(X;U) - I(Y;U)\right)$$

Covering lemma

$\propto$ random binning

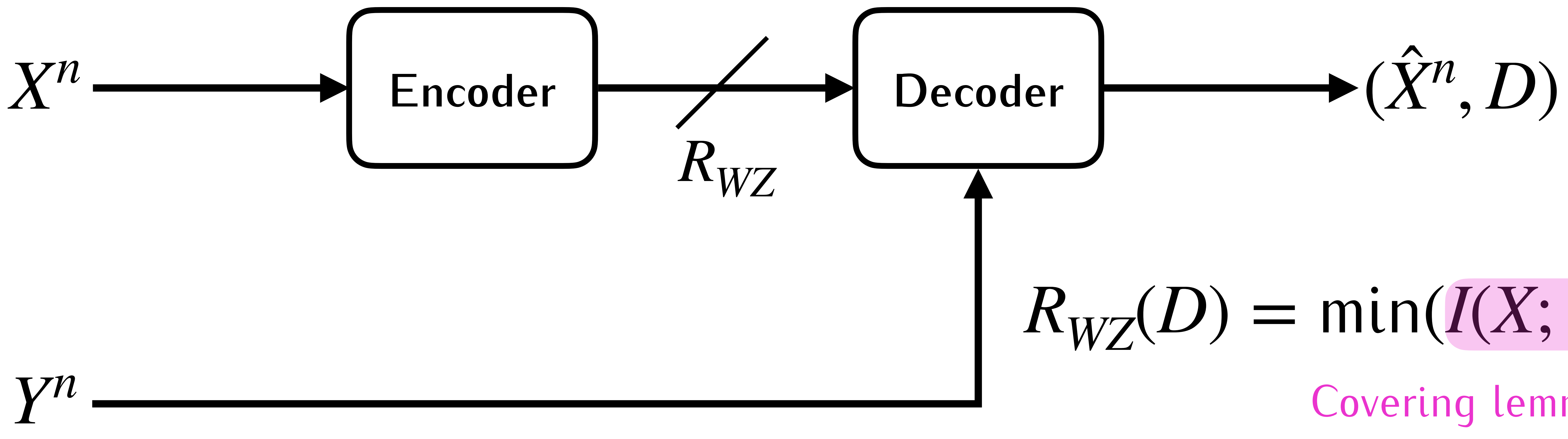
'discount'



For X^n, send the color within the "fan".

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory*, 1976.

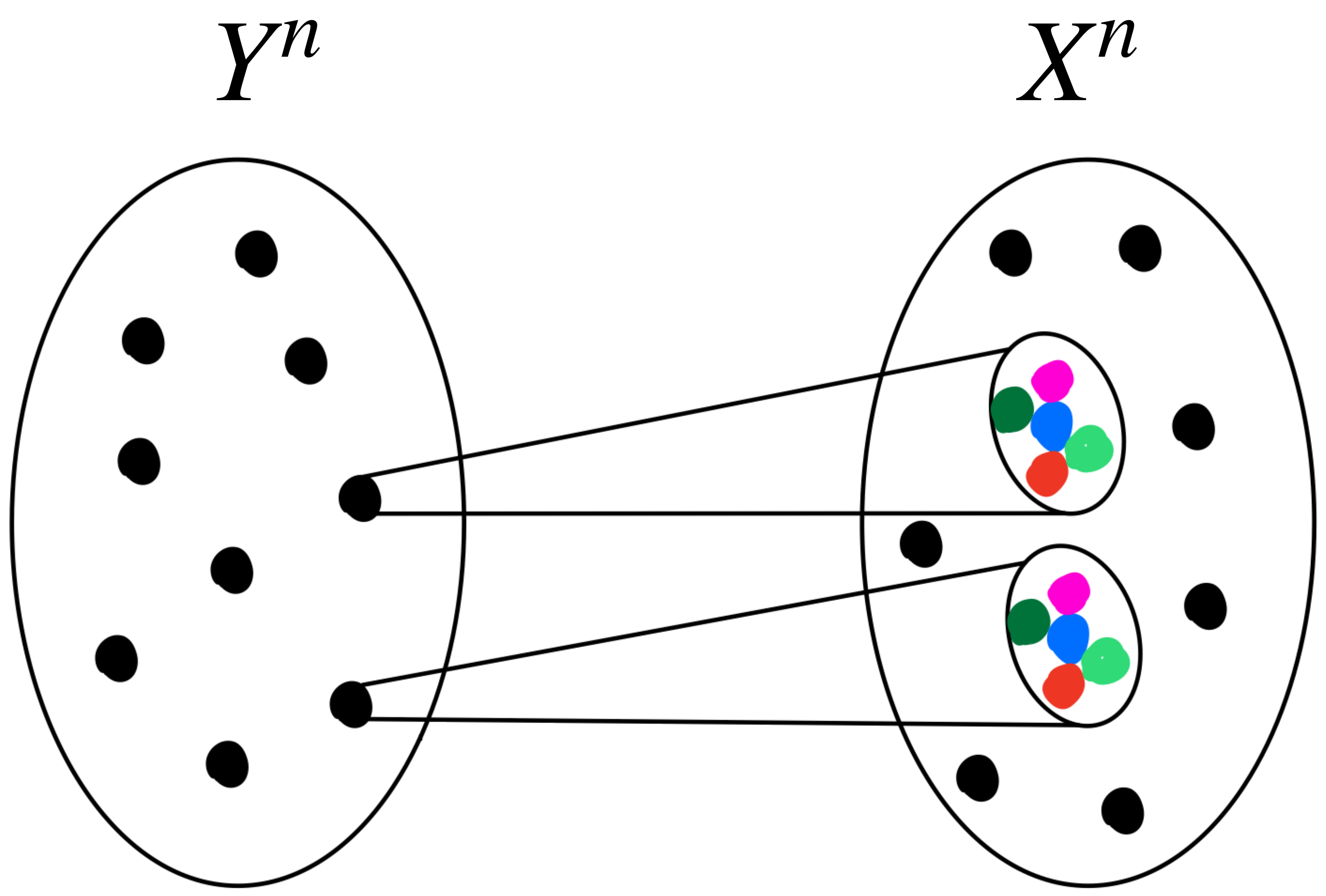
Wyner-Ziv achievability



$$R_{WZ}(D) = \min\big(I(X;U) - I(Y;U)\big)$$

Covering lemma $\propto$ random binning

'discount'

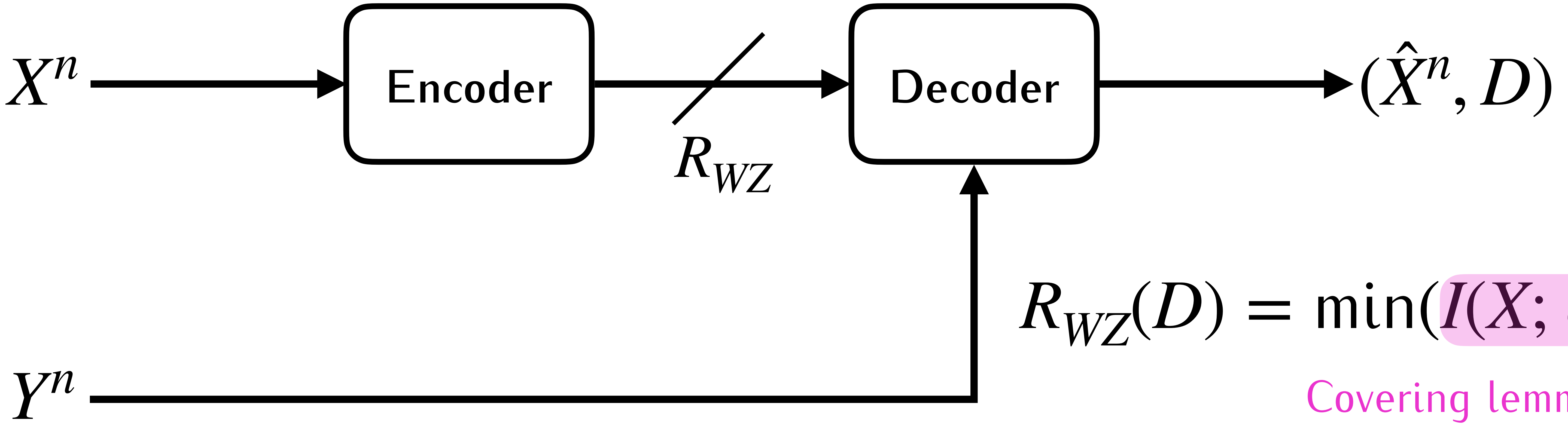


For X^n, send the color within the "fan".

$\Longrightarrow$ **binning**.

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory*, 1976.

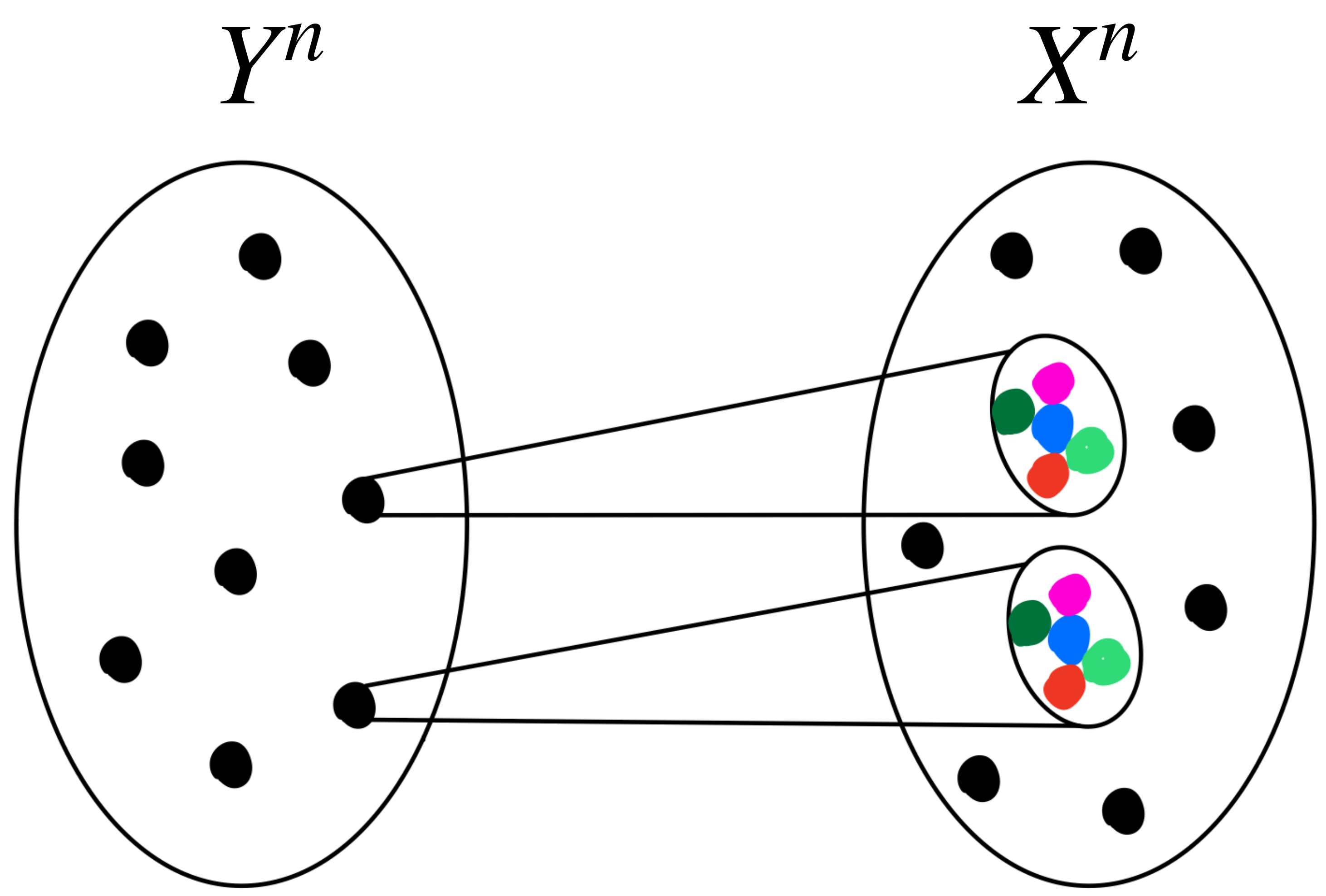
Wyner-Ziv achievability



$$R_{WZ}(D) = \min\big(I(X;U) - I(Y;U)\big)$$

Covering lemma $\propto$ random binning

'discount'



For X^n, send the color within the "fan".

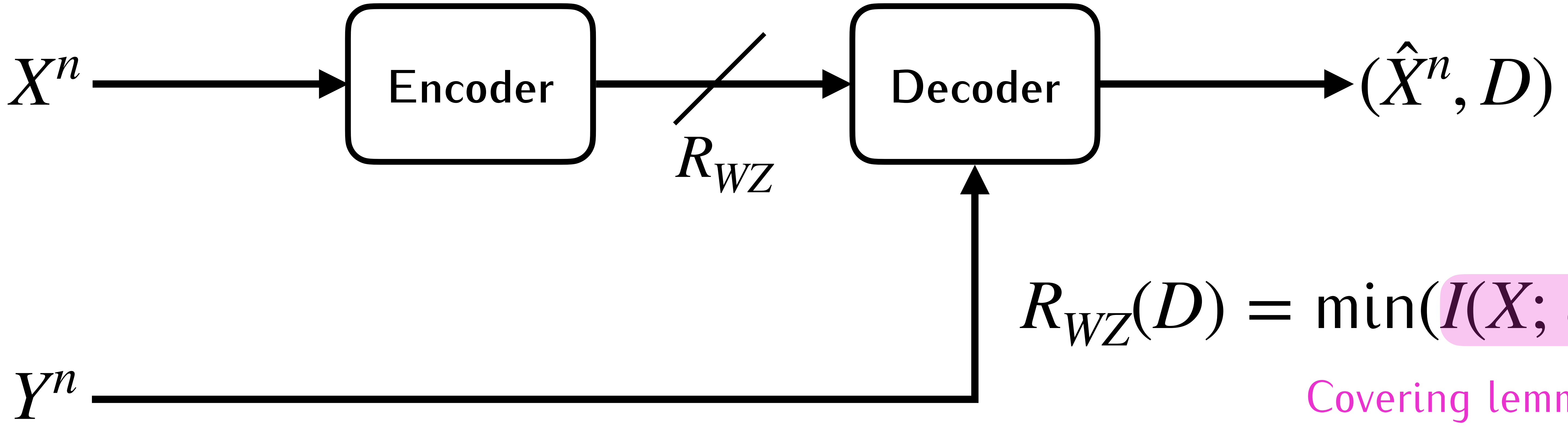
$\Longrightarrow$ **binning**.

In quadratic-Gaussian setup,

$$g(u,y) = a^* y + \beta[u - a^* y]\,.$$

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder",
IEEE Transactions on Information Theory, 1976.

Wyner-Ziv achievability

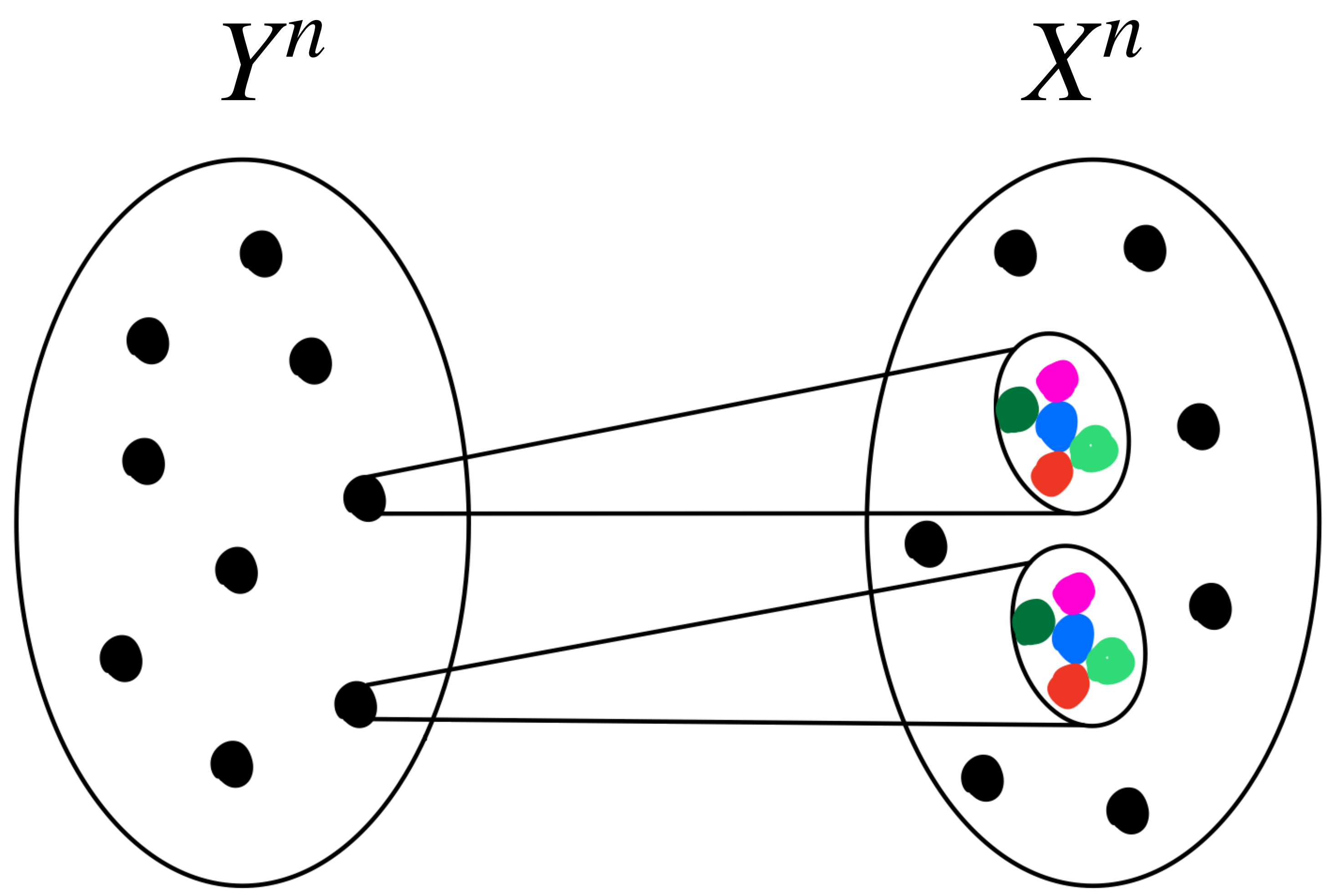


$$R_{WZ}(D) = \min(I(X;U) - I(Y;U))$$

Covering lemma

$\propto$ random binning

'discount'



For X^n, send the color within the "fan".

$\Longrightarrow$ **binning**.

In quadratic-Gaussian setup,

$$g(u, y) = a^*y + \beta[u - a^*y]\,.$$

linear!

A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory*, 1976.

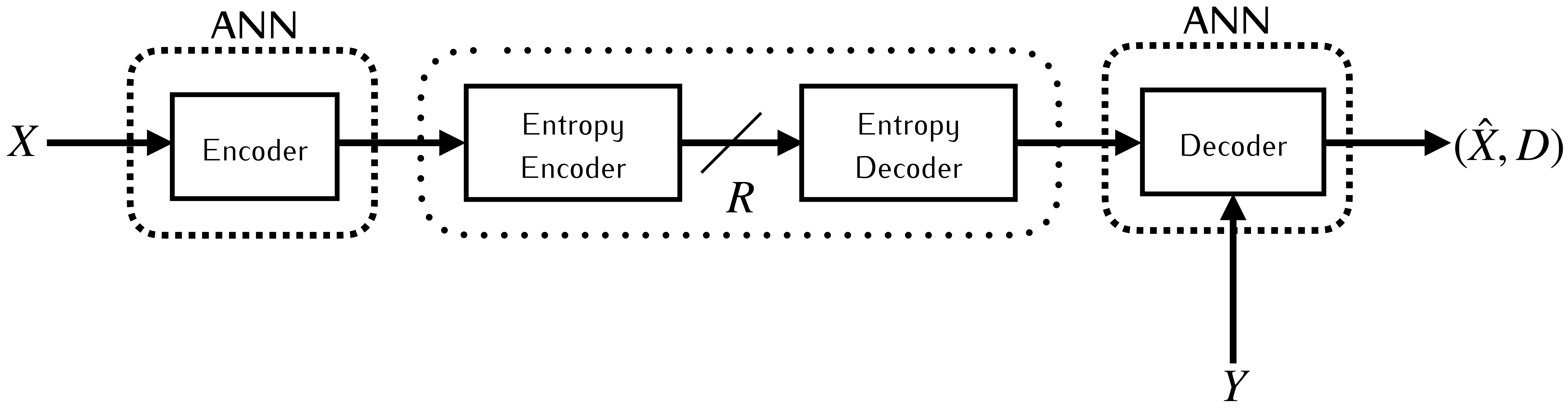
Operational schemes

Operational schemes

With Artificial Neural Networks (ANNs).

Operational schemes

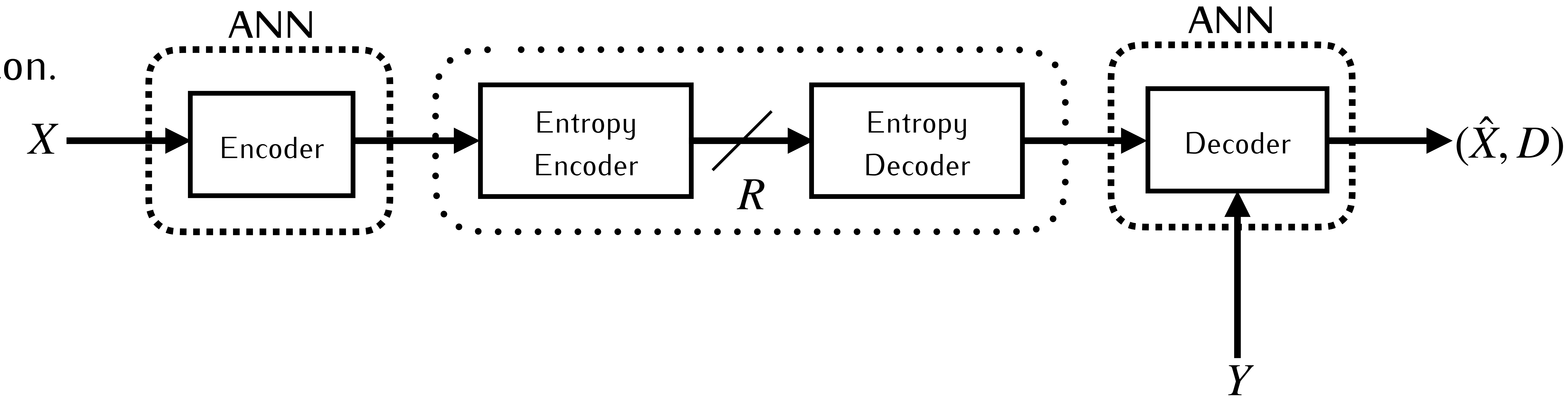
With Artificial Neural Networks (ANNs).



Operational schemes

With Artificial Neural Networks (ANNs).

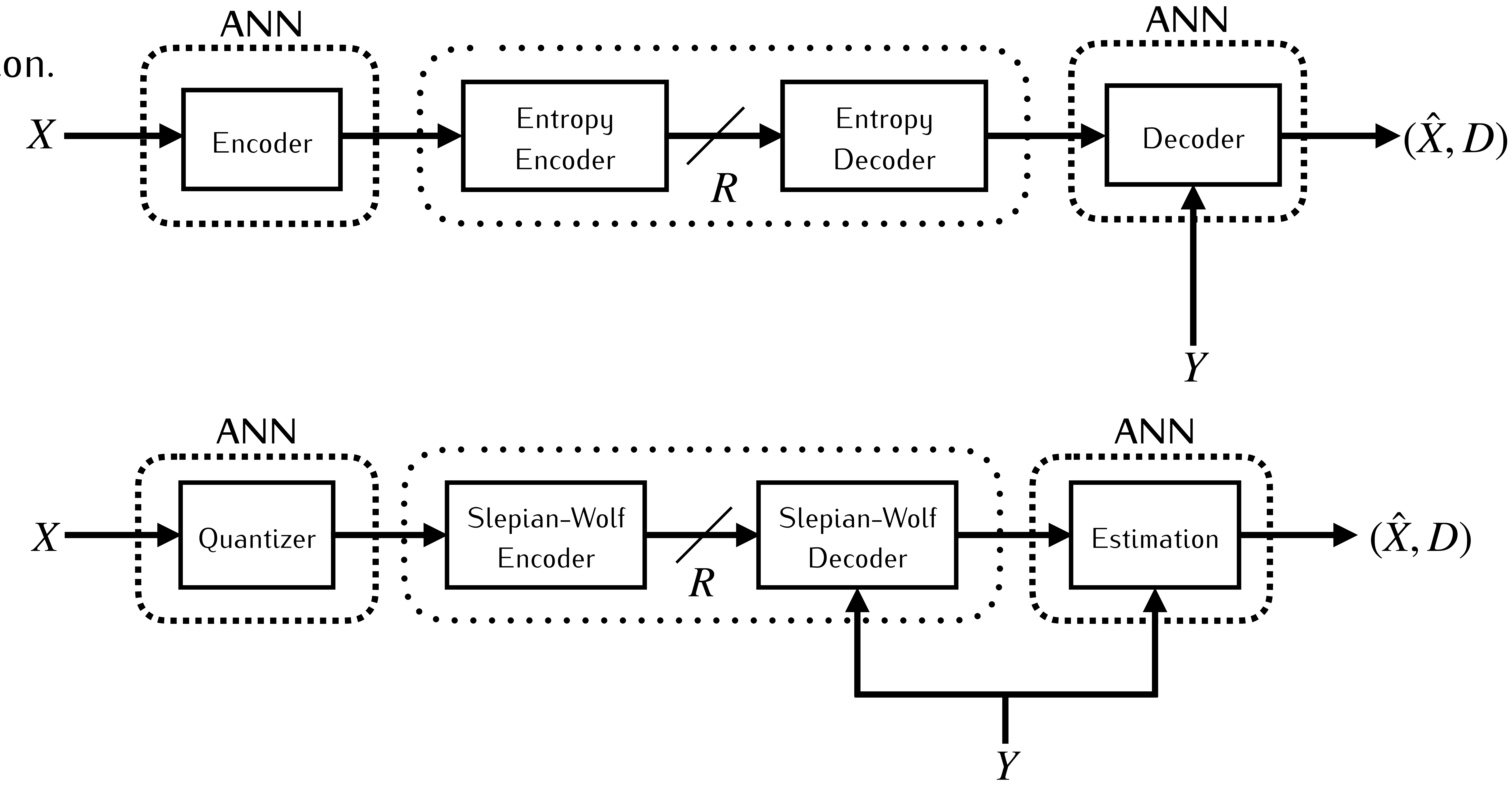
Marginal formulation.



Operational schemes

With Artificial Neural Networks (ANNs).

Marginal formulation.

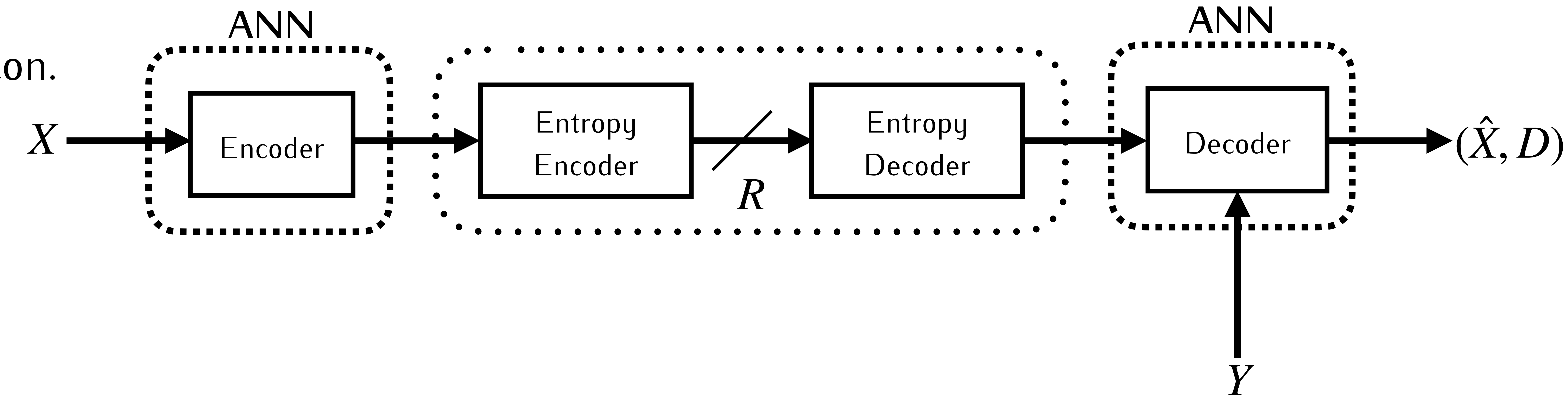


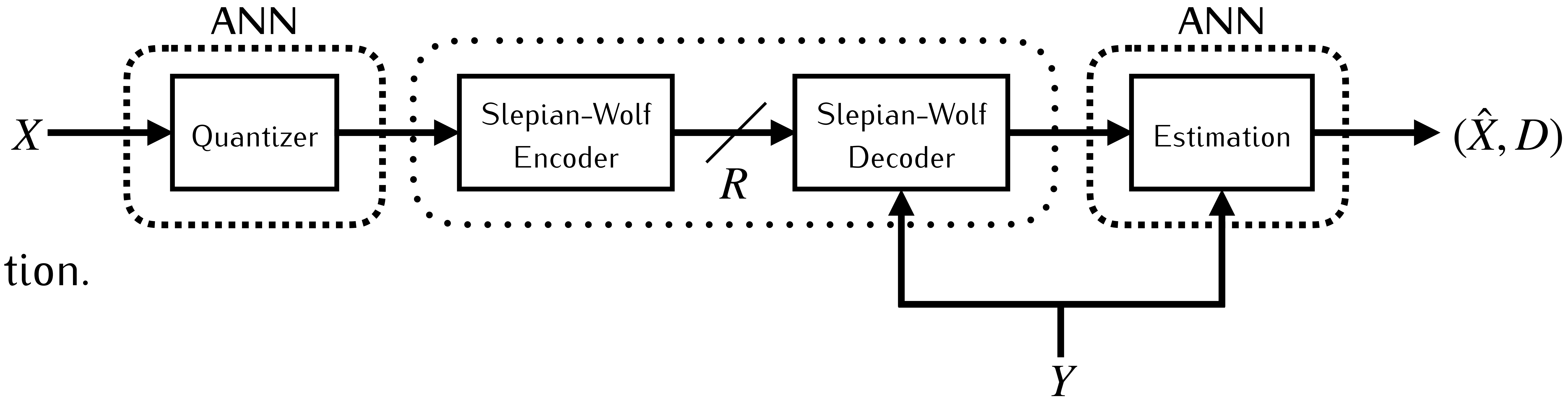
D. Slepian and J. Wolf, "Noiseless coding of correlated information sources", *IEEE Transactions on Information Theory*, 1973.

Operational schemes

With Artificial Neural Networks (ANNs).

Marginal formulation.





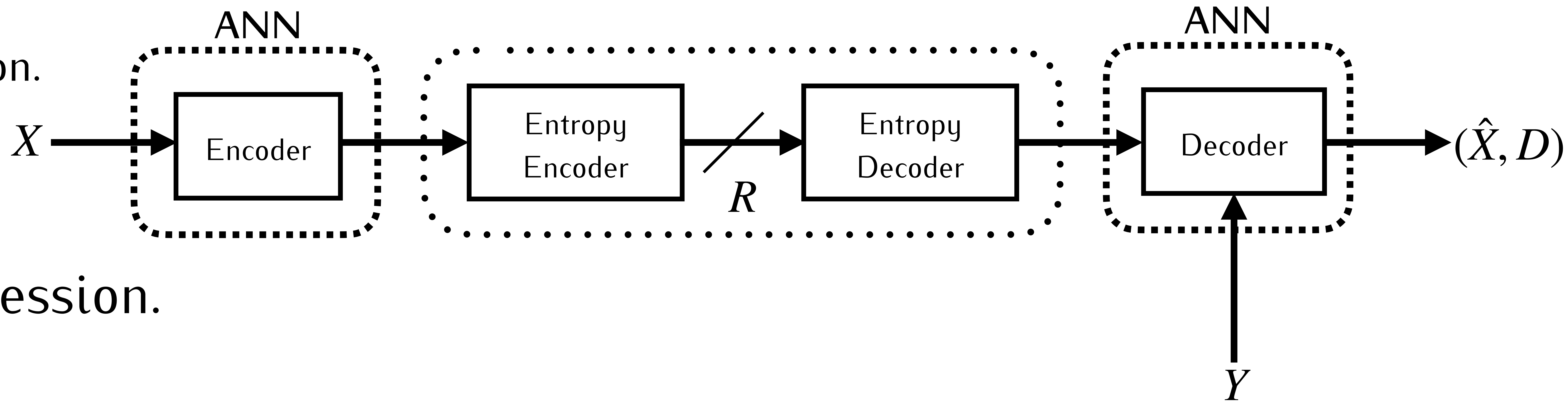
Conditional formulation.

D. Slepian and J. Wolf, "Noiseless coding of correlated information sources", *IEEE Transactions on Information Theory*, 1973.

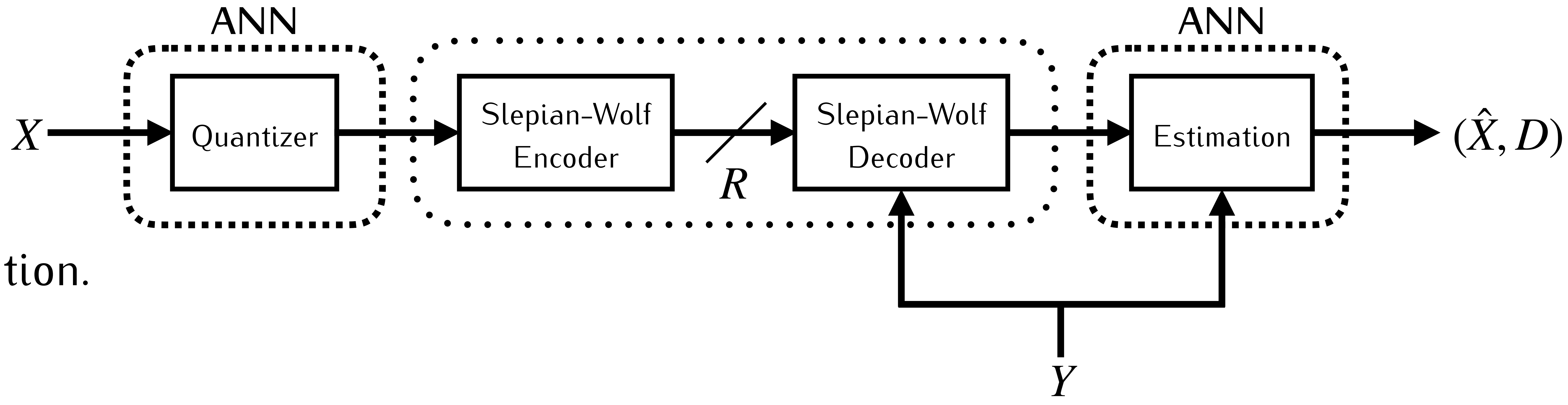
Operational schemes

With Artificial Neural Networks (ANNs).

Marginal formulation.



One-shot compression.



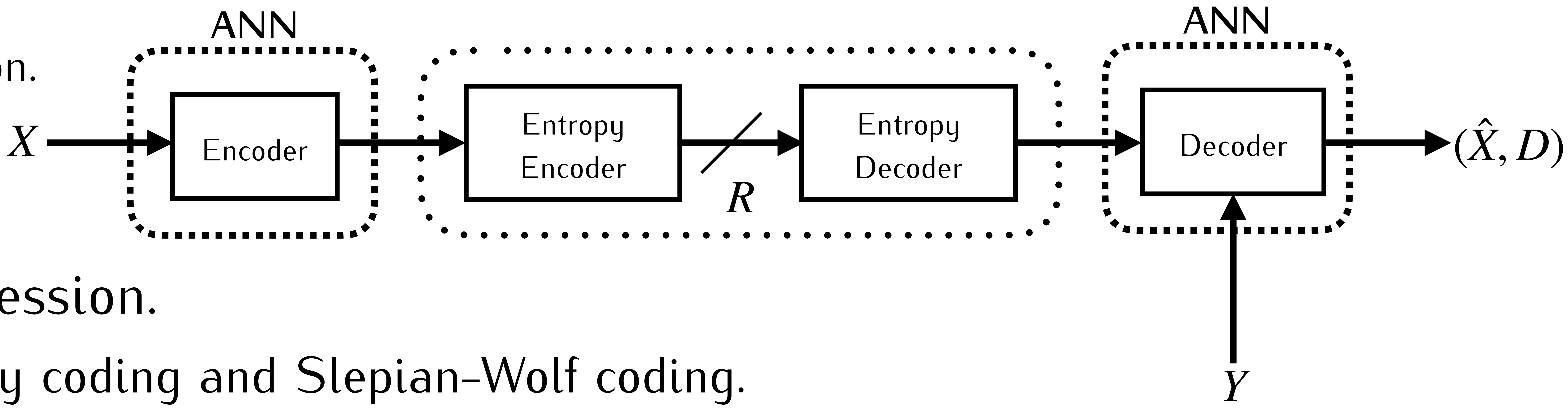
Conditional formulation.

D. Slepian and J. Wolf, "Noiseless coding of correlated information sources", *IEEE Transactions on Information Theory*, 1973.

Operational schemes

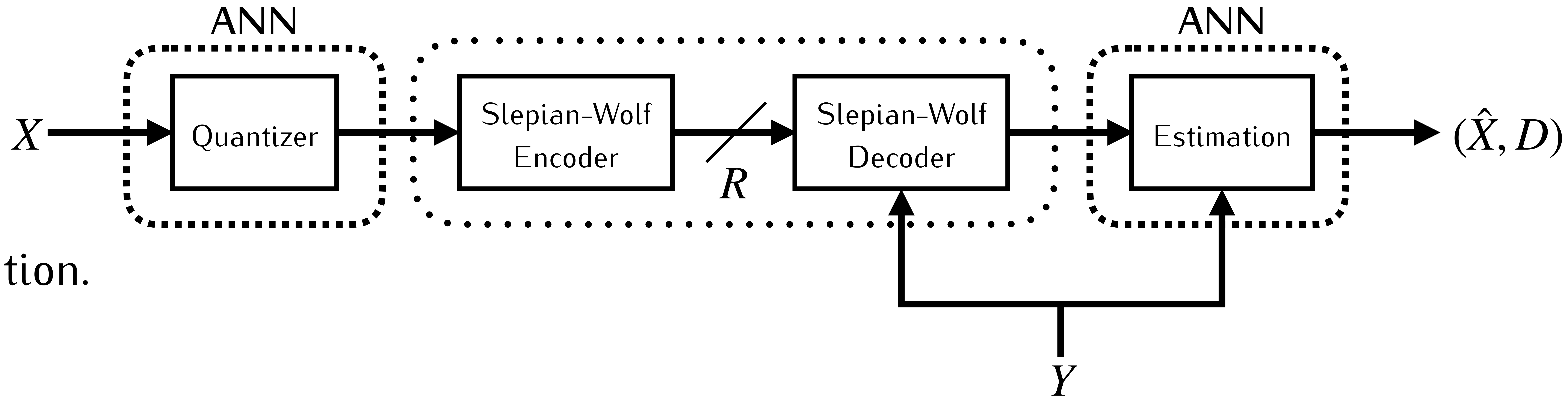
With Artificial Neural Networks (ANNs).

Marginal formulation.



One-shot compression.

High-order entropy coding and Slepian-Wolf coding.



Conditional formulation.

D. Slepian and J. Wolf, "Noiseless coding of correlated information sources", *IEEE Transactions on Information Theory*, 1973.

Neural parametrization for Wyner–Ziv

Neural parametrization for Wyner-Ziv

- Assume that during training, the encoder in achievability is represented by $p_{\boldsymbol{\theta}}(u|x)$,

Neural parametrization for Wyner-Ziv

- Assume that during training, the encoder in achievability is represented by $p_{\boldsymbol{\theta}}(u\,|\,x)$,

$$I(X;U) - I(Y;U) = I(X;U\,|\,Y) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u\,|\,x)}{p(u\,|\,y)}\right].$$

$$U - X - Y$$

Neural parametrization for Wyner-Ziv

- Assume that during training, the encoder in achievability is represented by $p_{\boldsymbol{\theta}}(u\,|\,x)$,

$$I(X;U) - I(Y;U) = I(X;U\,|\,Y) = \mathbb{E}\left[\log\frac{p_{\boldsymbol{\theta}}(u\,|\,x)}{p(u\,|\,y)}\right].$$

$$U - X - Y$$

- For test time, set encoder output as $u = \operatorname{argmax}_v p_{\boldsymbol{\theta}}(v\,|\,x)$, and have U as discrete.

Neural parametrization for Wyner-Ziv

- Assume that during training, the encoder in achievability is represented by $p_{\boldsymbol{\theta}}(u\,|\,x)$,

$$I(X;U) - I(Y;U) = I(X;U\,|\,Y) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u\,|\,x)}{p(u\,|\,y)}\right].$$

$$U - X - Y$$

- For test time, set encoder output as $u = \operatorname{argmax}_v p_{\boldsymbol{\theta}}(v\,|\,x)$, and have U as discrete.

- Choose one of two variational upper bounds:

Neural parametrization for Wyner-Ziv

- Assume that during training, the encoder in achievability is represented by $p_{\boldsymbol{\theta}}(u|x)$,

$$I(X;U) - I(Y;U) = I(X;U|Y) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u|x)}{p(u|y)}\right].$$

$$U - X - Y$$

- For test time, set encoder output as $u = \operatorname{argmax}_v p_{\boldsymbol{\theta}}(v|x)$, and have U as discrete.

- Choose one of two variational upper bounds:

$$I(X;U|Y) \leq \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u|x)}{q_{\boldsymbol{\zeta}}(u)}\right],$$

$$I(X;U|Y) \leq \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u|x)}{q_{\boldsymbol{\xi}}(u|y)}\right].$$

Neural parametrization for Wyner-Ziv

- Assume that during training, the encoder in achievability is represented by $p_{\boldsymbol{\theta}}(u|x)$,

$$I(X;U) - I(Y;U) = I(X;U|Y) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u|x)}{p(u|y)}\right].$$

$$U - X - Y$$

- For test time, set encoder output as $u = \operatorname{argmax}_v p_{\boldsymbol{\theta}}(v|x)$, and have U as discrete.

- Choose one of two variational upper bounds:

$$I(X;U|Y) \leq \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u|x)}{q_{\boldsymbol{\zeta}}(u)}\right],$$

$$I(X;U|Y) \leq \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u|x)}{q_{\boldsymbol{\xi}}(u|y)}\right].$$

Neural parametrization for Wyner-Ziv

- Assume that during training, the encoder in achievability is represented by $p_{\boldsymbol{\theta}}(u \mid x)$,

$$I(X;U) - I(Y;U) = \underset{U-X-Y}{I(X;U \mid Y)} = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{p(u \mid y)}\right].$$

- For test time, set encoder output as $u = \mathrm{argmax}_v p_{\boldsymbol{\theta}}(v \mid x)$, and have U as discrete.

- Choose one of two variational upper bounds:

$$I(X;U \mid Y) \leq \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\zeta}}(u)}\right], \qquad \text{marginal}$$

$$I(X;U \mid Y) \leq \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\xi}}(u \mid y)}\right].$$

Neural parametrization for Wyner-Ziv

- Assume that during training, the encoder in achievability is represented by $p_{\boldsymbol{\theta}}(u \mid x)$,

$$\underset{U-X-Y}{I(X;U) - I(Y;U) = I(X;U \mid Y)} = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{p(u \mid y)}\right].$$

- For test time, set encoder output as $u = \operatorname{argmax}_v p_{\boldsymbol{\theta}}(v \mid x)$, and have U as discrete.

- Choose one of two variational upper bounds:

$$I(X;U \mid Y) \leq \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\zeta}}(u)}\right], \qquad \text{marginal}$$

$$I(X;U \mid Y) \leq \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\xi}}(u \mid y)}\right]. \qquad \text{conditional}$$

Neural parametrization for Wyner-Ziv

Neural parametrization for Wyner-Ziv

- Relax the constrained formulation of Wyner-Ziv theorem using Lagrange multipliers:

Neural parametrization for Wyner-Ziv

- Relax the constrained formulation of Wyner-Ziv theorem using Lagrange multipliers:

$$L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\xi}}(u)} + \lambda \cdot d(x, g_{\boldsymbol{\phi}}(u, y))\right],$$

$$L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta}) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\zeta}}(u \mid y)} + \lambda \cdot d(x, g_{\boldsymbol{\phi}}(u, y))\right].$$

Neural parametrization for Wyner-Ziv

- Relax the constrained formulation of Wyner-Ziv theorem using Lagrange multipliers:

$$L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\xi}}(u)} + \lambda \cdot d(x, g_{\boldsymbol{\phi}}(u, y))\right],$$

$$L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta}) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\zeta}}(u \mid y)} + \lambda \cdot d(x, g_{\boldsymbol{\phi}}(u, y))\right] .$$

Neural parametrization for Wyner-Ziv

- Relax the constrained formulation of Wyner-Ziv theorem using Lagrange multipliers:

$$L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\xi}}(u)} + \lambda \cdot d(x, g_{\boldsymbol{\phi}}(u, y))\right],$$

$$L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta}) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(u \mid x)}{q_{\boldsymbol{\zeta}}(u \mid y)} + \lambda \cdot d(x, g_{\boldsymbol{\phi}}(u, y))\right].$$

Neural parametrization for Wyner–Ziv

- Relax the constrained formulation of Wyner–Ziv theorem using Lagrange multipliers:

$$L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}\left[\log \frac{\overset{\text{encoder}}{p_{\boldsymbol{\theta}}(u \mid x)}}{q_{\boldsymbol{\xi}}(u)} + \lambda \cdot d(x, g_{\boldsymbol{\phi}}(u, y))\right],$$

$$L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta}) = \mathbb{E}\left[\log \frac{\overset{\text{quantizer}}{p_{\boldsymbol{\theta}}(u \mid x)}}{q_{\boldsymbol{\zeta}}(u \mid y)} + \lambda \cdot d(x, g_{\boldsymbol{\phi}}(u, y))\right].$$

Neural parametrization for Wyner-Ziv

- Relax the constrained formulation of Wyner-Ziv theorem using Lagrange multipliers:

$$L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}\left[\log \frac{\overbrace{p_{\boldsymbol{\theta}}(u \mid x)}^{\text{encoder}}}{q_{\boldsymbol{\xi}}(u)} + \lambda \cdot d(x, \overbrace{g_{\boldsymbol{\phi}}(u, y)}^{\text{decoder}})\right],$$

$$L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta}) = \mathbb{E}\left[\log \frac{\overbrace{p_{\boldsymbol{\theta}}(u \mid x)}^{\text{quantizer}}}{q_{\boldsymbol{\zeta}}(u \mid y)} + \lambda \cdot d(x, \overbrace{g_{\boldsymbol{\phi}}(u, y)}^{\text{de-quantizer}})\right].$$

Neural parametrization for Wyner–Ziv

- Relax the constrained formulation of Wyner–Ziv theorem using Lagrange multipliers:

$$\textcolor{blue}{L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi})} = \mathbb{E}\left[\log \frac{\overset{\text{encoder}}{p_{\theta}(u \mid x)}}{q_{\xi}(u)} + \lambda \cdot d(x, \overset{\textbf{decoder}}{g_{\phi}(u, y)})\right],$$

entropy coder

$$\textcolor{red}{L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta})} = \mathbb{E}\left[\log \frac{\overset{\text{quantizer}}{p_{\theta}(u \mid x)}}{q_{\zeta}(u \mid y)} + \lambda \cdot d(x, \overset{\textbf{de-quantizer}}{g_{\phi}(u, y)})\right].$$

Neural parametrization for Wyner-Ziv

- Relax the constrained formulation of Wyner-Ziv theorem using Lagrange multipliers:

$$L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}\left[\log \frac{\overbrace{p_{\boldsymbol{\theta}}(u\,|\,x)}^{\text{encoder}}}{q_{\boldsymbol{\xi}}(u)} + \lambda \cdot d(x, \overbrace{g_{\boldsymbol{\phi}}(u, y)}^{\text{decoder}})\right],$$

$$L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta}) = \mathbb{E}\left[\log \frac{\overbrace{p_{\boldsymbol{\theta}}(u\,|\,x)}^{\text{quantizer}}}{\underbrace{q_{\boldsymbol{\zeta}}(u\,|\,y)}_{\text{entropy coder}}} + \lambda \cdot d(x, \overbrace{g_{\boldsymbol{\phi}}(u, y)}^{\text{de-quantizer}})\right].$$

- Define all models $p_{\boldsymbol{\theta}}(u\,|\,x)$, $q_{\boldsymbol{\xi}}(u)$ and $q_{\boldsymbol{\zeta}}(u\,|\,y)$ as **discrete** distributions with probabilities:

Neural parametrization for Wyner-Ziv

- Relax the constrained formulation of Wyner-Ziv theorem using Lagrange multipliers:

$$L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}\left[\log \frac{\overbrace{p_{\boldsymbol{\theta}}(u \mid x)}^{\text{encoder}}}{q_{\boldsymbol{\xi}}(u)} + \lambda \cdot d(x, \overbrace{g_{\boldsymbol{\phi}}(u, y)}^{\text{decoder}})\right],$$

$$L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta}) = \mathbb{E}\left[\log \frac{\overbrace{p_{\boldsymbol{\theta}}(u \mid x)}^{\text{quantizer}}}{\underbrace{q_{\boldsymbol{\zeta}}(u \mid y)}_{\text{entropy coder}}} + \lambda \cdot d(x, \overbrace{g_{\boldsymbol{\phi}}(u, y)}^{\text{de-quantizer}})\right].$$

- Define all models $p_{\boldsymbol{\theta}}(u \mid x)$, $q_{\boldsymbol{\xi}}(u)$ and $q_{\boldsymbol{\zeta}}(u \mid y)$ as **discrete** distributions with probabilities:

$$P_k = \frac{\exp \alpha_k}{\sum_{i=1}^{K} \exp \alpha_i}.$$

Neural parametrization for Wyner-Ziv

- Relax the constrained formulation of Wyner-Ziv theorem using Lagrange multipliers:

$$L_{\mathrm{m}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}\left[\log \frac{\overbrace{p_{\boldsymbol{\theta}}(u \mid x)}^{\text{encoder}}}{q_{\boldsymbol{\xi}}(u)} + \lambda \cdot d(x, \overbrace{g_{\boldsymbol{\phi}}(u, y)}^{\text{decoder}})\right],$$

$$L_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\zeta}) = \mathbb{E}\left[\log \frac{\overbrace{p_{\boldsymbol{\theta}}(u \mid x)}^{\text{quantizer}}}{\underbrace{q_{\boldsymbol{\zeta}}(u \mid y)}_{\text{entropy coder}}} + \lambda \cdot d(x, \overbrace{g_{\boldsymbol{\phi}}(u, y)}^{\text{de-quantizer}})\right].$$

- Define all models $p_{\boldsymbol{\theta}}(u \mid x)$, $q_{\boldsymbol{\xi}}(u)$ and $q_{\boldsymbol{\zeta}}(u \mid y)$ as **discrete** distributions with probabilities:

$$P_k = \frac{\exp \alpha_k}{\sum_{i=1}^{K} \exp \alpha_i}.$$

- This keeps the parametric families as general as possible, and **<u>does not impose any structure.</u>**

Neural parametrization for Wyner-Ziv

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).
- SGD replaces $\mathbb{E}(\,\cdot\,)$ by averages over batches of samples B.

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).
- SGD replaces $\mathbb{E}(\,\cdot\,)$ by averages over batches of samples B.

For example, $\dfrac{\partial}{\partial \boldsymbol{\theta}}\mathbb{E}[l_{\boldsymbol{\theta}}(x, y)] \approx \dfrac{1}{|B|}\sum_{(x,y)\in B}\dfrac{\partial l_{\boldsymbol{\theta}}(x, y)}{\partial \boldsymbol{\theta}}$, where $l_{\boldsymbol{\theta}}$ is a sample loss with parameters $\boldsymbol{\theta}$.

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).

- SGD replaces $\mathbb{E}(\,\cdot\,)$ by averages over batches of samples B.

  For example, $\dfrac{\partial}{\partial \boldsymbol{\theta}}\mathbb{E}[l_{\boldsymbol{\theta}}(x,y)] \approx \dfrac{1}{|B|}\displaystyle\sum_{(x,y)\in B}\dfrac{\partial l_{\boldsymbol{\theta}}(x,y)}{\partial \boldsymbol{\theta}}$, where $l_{\boldsymbol{\theta}}$ is a sample loss with parameters $\boldsymbol{\theta}$.

- To draw samples u from $p_{\boldsymbol{\theta}}(u\,|\,x)$, use Gumbel-max 'trick' that is:

E. J. Gumbel, "Statistical theory of extreme values and some practical applications: a series of lectures", *US Department of Commerce,* 1954.

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).

- SGD replaces $\mathbb{E}(\,\cdot\,)$ by averages over batches of samples B.

For example, $\dfrac{\partial}{\partial \boldsymbol{\theta}}\mathbb{E}[l_{\boldsymbol{\theta}}(x,y)] \approx \dfrac{1}{|B|}\sum_{(x,y)\in B}\dfrac{\partial l_{\boldsymbol{\theta}}(x,y)}{\partial \boldsymbol{\theta}}$, where $l_{\boldsymbol{\theta}}$ is a sample loss with parameters $\boldsymbol{\theta}$.

- To draw samples u from $p_{\boldsymbol{\theta}}(u\,|\,x)$, use Gumbel-max 'trick' that is:

$$\arg\max_{k\in 1,\ldots,K}\{\alpha_k + G_k\}\ .$$

E. J. Gumbel, "Statistical theory of extreme values and some practical applications: a series of lectures", *US Department of Commerce,* 1954.

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).
- SGD replaces $\mathbb{E}(\,\cdot\,)$ by averages over batches of samples B.

For example, $\dfrac{\partial}{\partial \boldsymbol{\theta}}\mathbb{E}[l_{\boldsymbol{\theta}}(x,y)] \approx \dfrac{1}{|B|}\sum_{(x,y)\in B}\dfrac{\partial l_{\boldsymbol{\theta}}(x,y)}{\partial \boldsymbol{\theta}}$, where $l_{\boldsymbol{\theta}}$ is a sample loss with parameters $\boldsymbol{\theta}$.

- To draw samples u from $p_{\boldsymbol{\theta}}(u\,|\,x)$, use Gumbel-max 'trick' that is:

$$\arg\max_{k\in 1,\ldots,K}\{\alpha_k + G_k\}\ .$$

- **Problem**: the derivative of $\arg\max$ is 0 almost everywhere.

E. J. Gumbel, "Statistical theory of extreme values and some practical applications: a series of lectures", *US Department of Commerce*, 1954.

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).

- SGD replaces $\mathbb{E}(\,\cdot\,)$ by averages over batches of samples B.

For example, $\dfrac{\partial}{\partial \boldsymbol{\theta}}\mathbb{E}[l_{\boldsymbol{\theta}}(x,y)] \approx \dfrac{1}{|B|}\sum_{(x,y)\in B}\dfrac{\partial l_{\boldsymbol{\theta}}(x,y)}{\partial \boldsymbol{\theta}}$, where $l_{\boldsymbol{\theta}}$ is a sample loss with parameters $\boldsymbol{\theta}$.

- To draw samples u from $p_{\boldsymbol{\theta}}(u\,|\,x)$, use Gumbel-max 'trick' that is:

$$\arg\max_{k\in 1,\ldots,K}\{\alpha_k + G_k\}\ .$$

- **Problem**: the derivative of $\arg\max$ is 0 almost everywhere.

- Need continuous relaxation of $\arg\max$ during training.

E. J. Gumbel, "Statistical theory of extreme values and some practical applications: a series of lectures", *US Department of Commerce,* 1954.

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).
- SGD replaces $\mathbb{E}(\,\cdot\,)$ by averages over batches of samples B.

For example, $\dfrac{\partial}{\partial \boldsymbol{\theta}}\mathbb{E}[l_{\boldsymbol{\theta}}(x,y)] \approx \dfrac{1}{|B|}\sum_{(x,y)\in B}\dfrac{\partial l_{\boldsymbol{\theta}}(x,y)}{\partial \boldsymbol{\theta}}$, where $l_{\boldsymbol{\theta}}$ is a sample loss with parameters $\boldsymbol{\theta}$.

- To draw samples u from $p_{\boldsymbol{\theta}}(u\,|\,x)$, use Gumbel-max 'trick' that is:

$$\arg\max_{k\in 1,\ldots,K}\{\alpha_k + G_k\}\ .$$

- **Problem**: the derivative of $\arg\max$ is 0 almost everywhere.
- Need <u>continuous relaxation</u> of $\arg\max$ during training.
  - Opt for *softmax* (differentiable!).

E. J. Gumbel, "Statistical theory of extreme values and some practical applications: a series of lectures", *US Department of Commerce,* 1954.

Neural parametrization for Wyner-Ziv

- Optimize learnable parameters with stochastic gradient descent (SGD).
- SGD replaces $\mathbb{E}(\,\cdot\,)$ by averages over batches of samples B.

For example, $\frac{\partial}{\partial \boldsymbol{\theta}}\mathbb{E}[l_{\boldsymbol{\theta}}(x,y)] \approx \frac{1}{|B|}\sum_{(x,y)\in B} \frac{\partial l_{\boldsymbol{\theta}}(x,y)}{\partial \boldsymbol{\theta}}$, where $l_{\boldsymbol{\theta}}$ is a sample loss with parameters $\boldsymbol{\theta}$.

- To draw samples u from $p_{\boldsymbol{\theta}}(u\,|\,x)$, use Gumbel-max 'trick' that is:

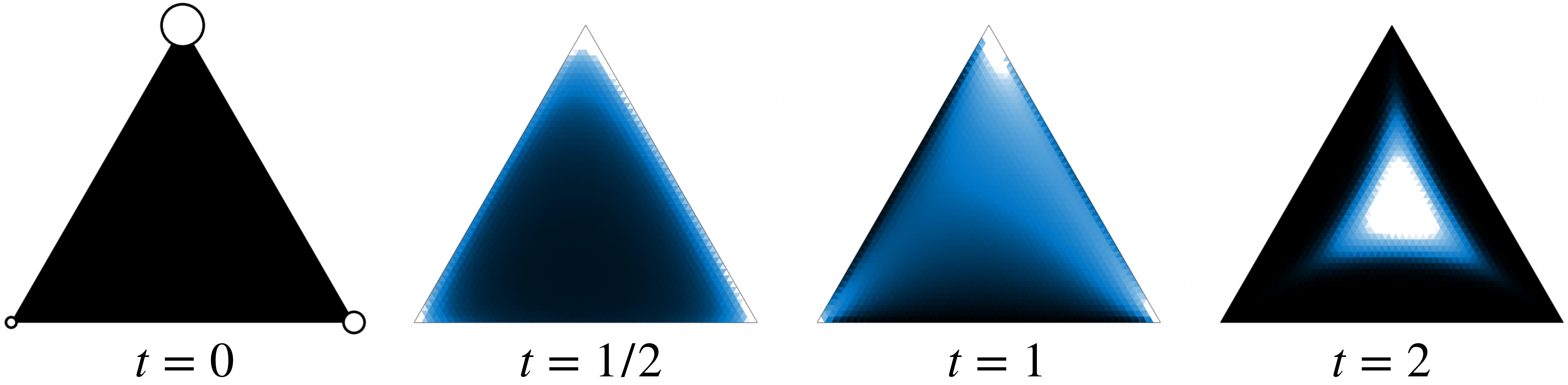
$$\arg\max_{k\in 1,\ldots,K}\{\alpha_k + G_k\} \ .$$

- **Problem**: the derivative of arg max is 0 almost everywhere.
- Need <u>continuous relaxation</u> of arg max during training.
  - Opt for *softmax* (differentiable!).
  - Use Gumbel-softmax 'trick' by Maddison et al.

E. J. Gumbel, "Statistical theory of extreme values and some practical applications: a series of lectures", *US Department of Commerce,* 1954.
C. Maddison et al., "The concrete distribution: a continuous relaxation of discrete random variables", *ICLR,* 2017.

Gumbel-softmax ‘trick’

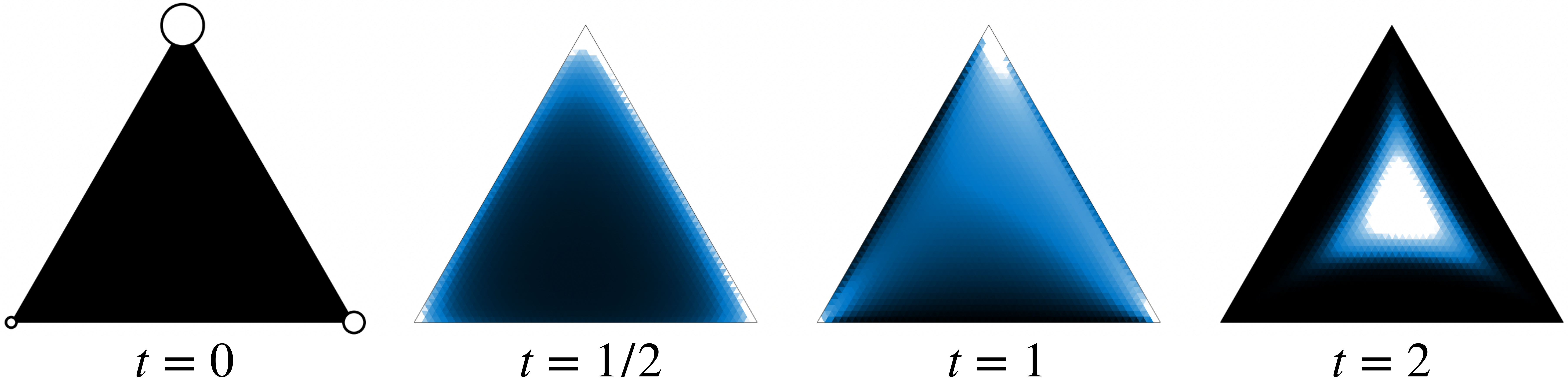
Gumbel-softmax 'trick'



- Concrete distribution (with temperature t) relaxes sampling from a discrete distribution.

Figure taken from C. Maddison et al., "The concrete distribution: a continuous relaxation of discrete random variables", *ICLR,* 2017.

Gumbel-softmax 'trick'

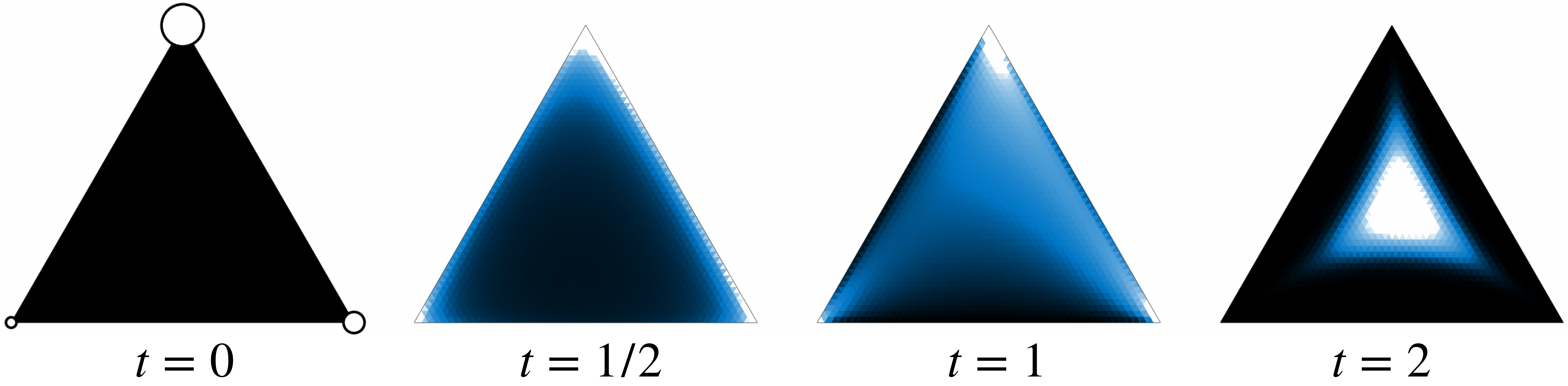


- Concrete distribution (with temperature t) relaxes sampling from a discrete distribution.

- Rather than sampling an index U, sample a vector $\mathbf{U}$:

$$U_k = \frac{\exp((\alpha_k + G_k) \;/\; t)}{\sum_{i=1}^{n} \exp((\alpha_i + G_i) \;/\; t)} \quad .$$

Gumbel-softmax 'trick'



- Concrete distribution (with temperature t) relaxes sampling from a discrete distribution.

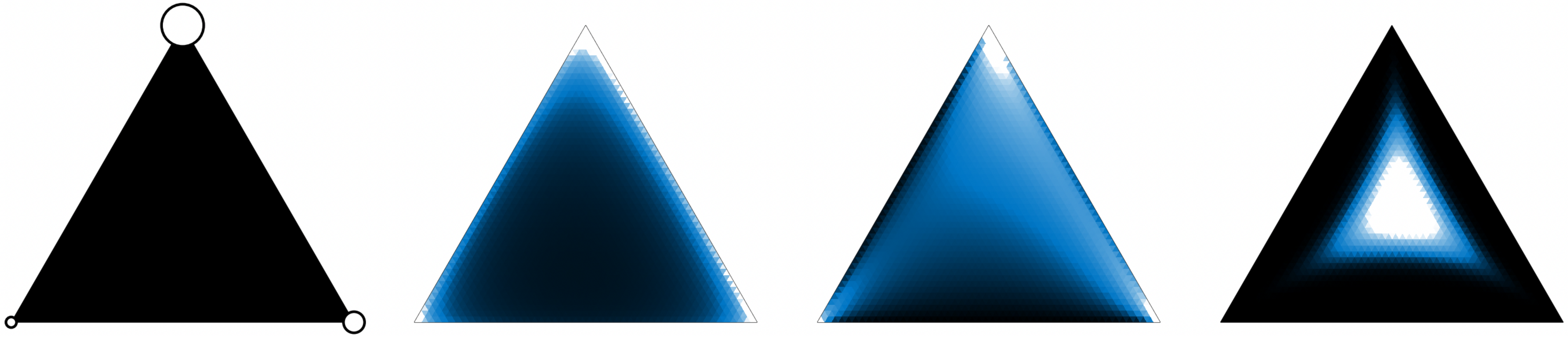
- Rather than sampling an index U, sample a vector $\mathbf{U}$:

$$U_k = \frac{\exp((\alpha_k + G_k) \,/\, t)}{\sum_{i=1}^{n} \exp((\alpha_i + G_i) \,/\, t)} \quad .$$

$\propto$ soft max

Figure taken from C. Maddison et al., "The concrete distribution: a continuous relaxation of discrete random variables", *ICLR*, 2017.

Gumbel-softmax 'trick'



- Concrete distribution (with temperature t) relaxes sampling from a discrete distribution.

- Rather than sampling an index U, sample a vector $\mathbf{U}$:

$$U_k = \frac{\exp((\alpha_k + G_k) \,/\, t)}{\sum_{i=1}^{n} \exp((\alpha_i + G_i) \,/\, t)} \quad .$$

$\propto$ soft max

- As $t \to 0^+$, soft max $\to$ arg max.
  - Concrete distribution $\to$ discrete distribution.

Figure taken from C. Maddison et al., "The concrete distribution: a continuous relaxation of discrete random variables", *ICLR*, 2017.

Evaluation

Evaluation

- Wyner-Ziv formula has a **closed-form solution in few special cases**.

Evaluation

- Wyner-Ziv formula has a **closed-form solution in few special cases**.
- To evaluate how close we can get to the R-D bound, we choose:

Evaluation

- Wyner-Ziv formula has a **closed-form solution in few special cases**.

- To evaluate how close we can get to the R-D bound, we choose:

  - Let X and Y be correlated, zero-mean and stationary Gaussian memoryless sources.

  - Let $d(\cdot)$ be mean-squared error.

Evaluation

- Wyner-Ziv formula has a **closed-form solution in few special cases**.
- To evaluate how close we can get to the R-D bound, we choose:
  - Let X and Y be correlated, zero-mean and stationary Gaussian memoryless sources.
  - Let $d(\cdot)$ be mean-squared error.
- Wyner-Ziv R-D function then is:

$$R_{WZ}(D) = \frac{1}{2}\log\left(\frac{\sigma^2_{x|y}}{D}\right),\ 0 \leq D \leq \sigma^2_{x|y}\ .$$

Evaluation

- Wyner-Ziv formula has a **closed-form solution in few special cases**.
- To evaluate how close we can get to the R-D bound, we choose:
  - Let X and Y be correlated, zero-mean and stationary Gaussian memoryless sources.
  - Let $d(\cdot)$ be mean-squared error.
- Wyner-Ziv R-D function then is:

$$R_{WZ}(D) = \frac{1}{2} \log\left(\frac{\sigma^2_{x|y}}{D}\right), \ 0 \leq D \leq \sigma^2_{x|y} \ .$$

- Consider correlation patterns of $X = Y + N$ and $Y = X + N$.

Evaluation

- Wyner-Ziv formula has a **closed-form solution in few special cases**.
- To evaluate how close we can get to the R-D bound, we choose:
  - Let X and Y be correlated, zero-mean and stationary Gaussian memoryless sources.
  - Let $d(\cdot)$ be mean-squared error.
- Wyner-Ziv R-D function then is:

$$R_{WZ}(D) = \frac{1}{2}\log\left(\frac{\sigma_{x|y}^2}{D}\right),\ 0 \leq D \leq \sigma_{x|y}^2\ .$$

- Consider correlation patterns of $X = Y + N$ and $Y = X + N$.
- The neural compressor **does not make any assumptions** on the source distribution.

Evaluation

- Wyner-Ziv formula has a **closed-form solution in few special cases**.
- To evaluate how close we can get to the R-D bound, we choose:
  - Let X and Y be correlated, zero-mean and stationary Gaussian memoryless sources.
  - Let $d(\cdot)$ be mean-squared error.
- Wyner-Ziv R-D function then is:

$$R_{WZ}(D) = \frac{1}{2}\log\left(\frac{\sigma_{x|y}^2}{D}\right), \; 0 \leq D \leq \sigma_{x|y}^2 \; .$$

- Consider correlation patterns of $X = Y + N$ and $Y = X + N$.
- The neural compressor **does not make any assumptions** on the source distribution.
  - The model parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}, \boldsymbol{\zeta}\}$ are learned in a data-driven way.

Results

Learned compressor recovers binning.

Results

Learned compressor recovers binning.

Learned encoder:

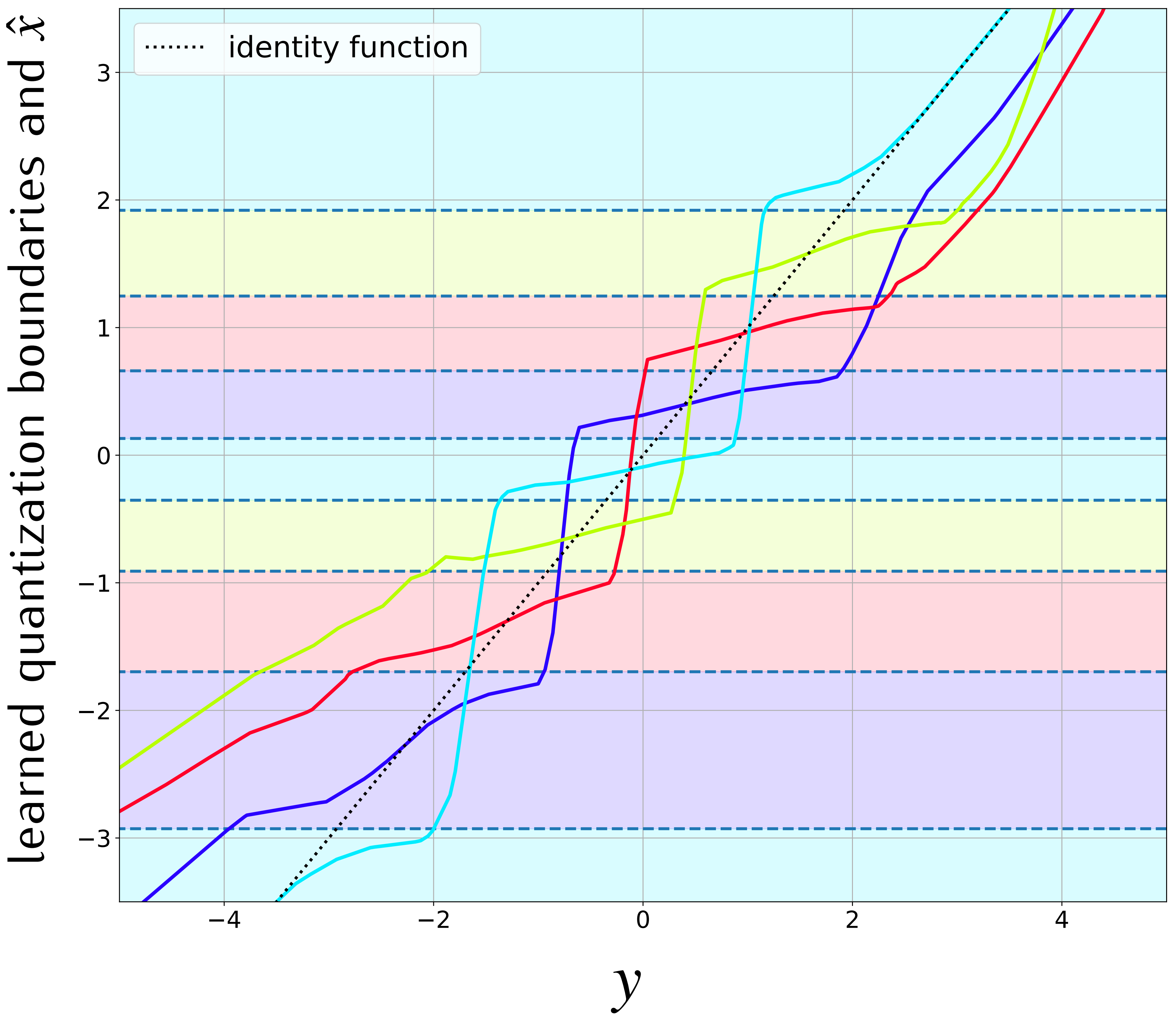
$u = \arg\max_v p_{\boldsymbol{\theta}}(v \mid x)$

Results

Learned compressor recovers binning.

Learned encoder:

$u = \arg\max_v p_{\boldsymbol{\theta}}(v \mid x)$



Marginal formulation.

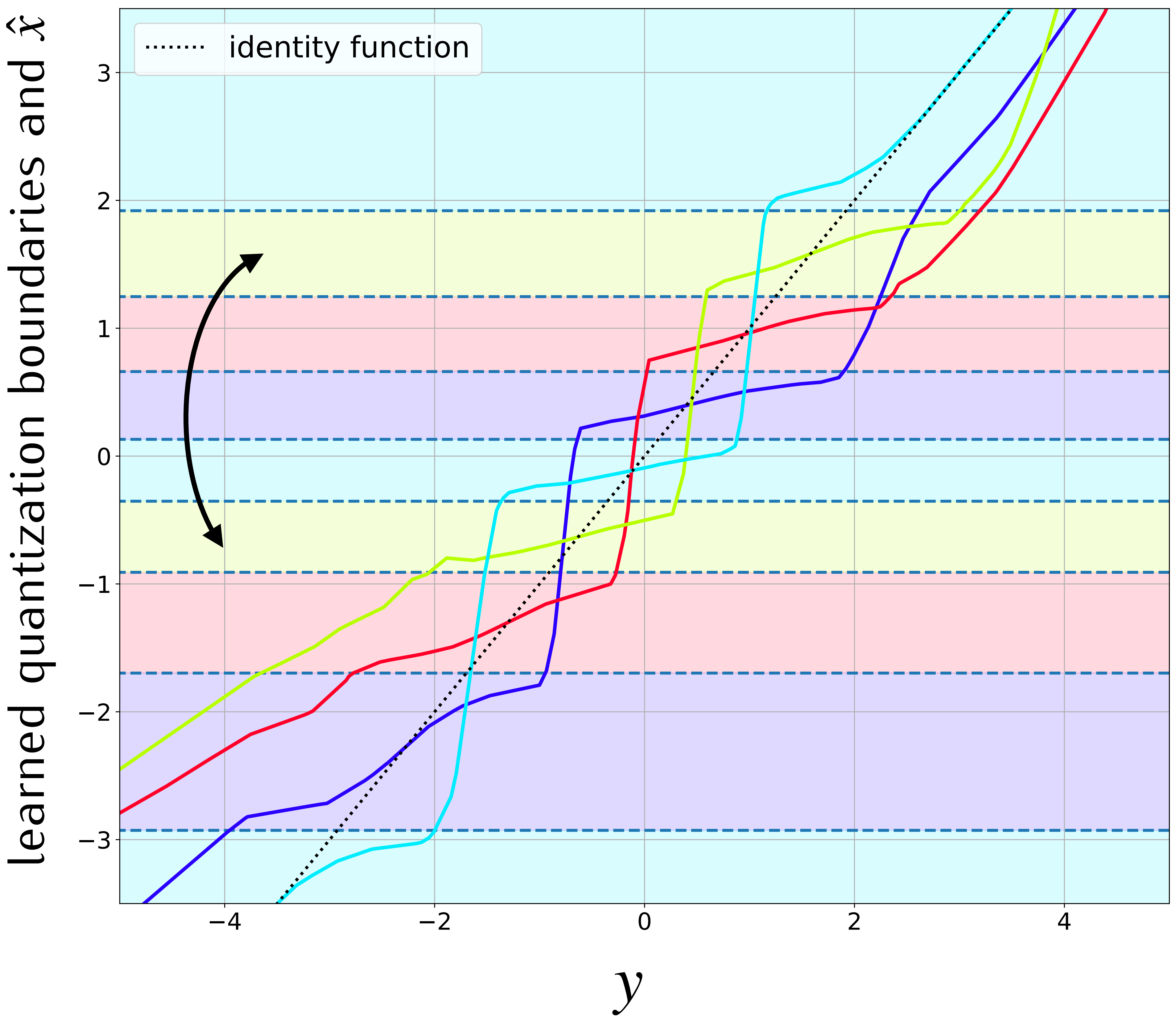
$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.

Results

Learned compressor recovers binning.

Learned encoder:

$u = \arg\max_v p_{\boldsymbol{\theta}}(v \mid x)$



Marginal formulation.

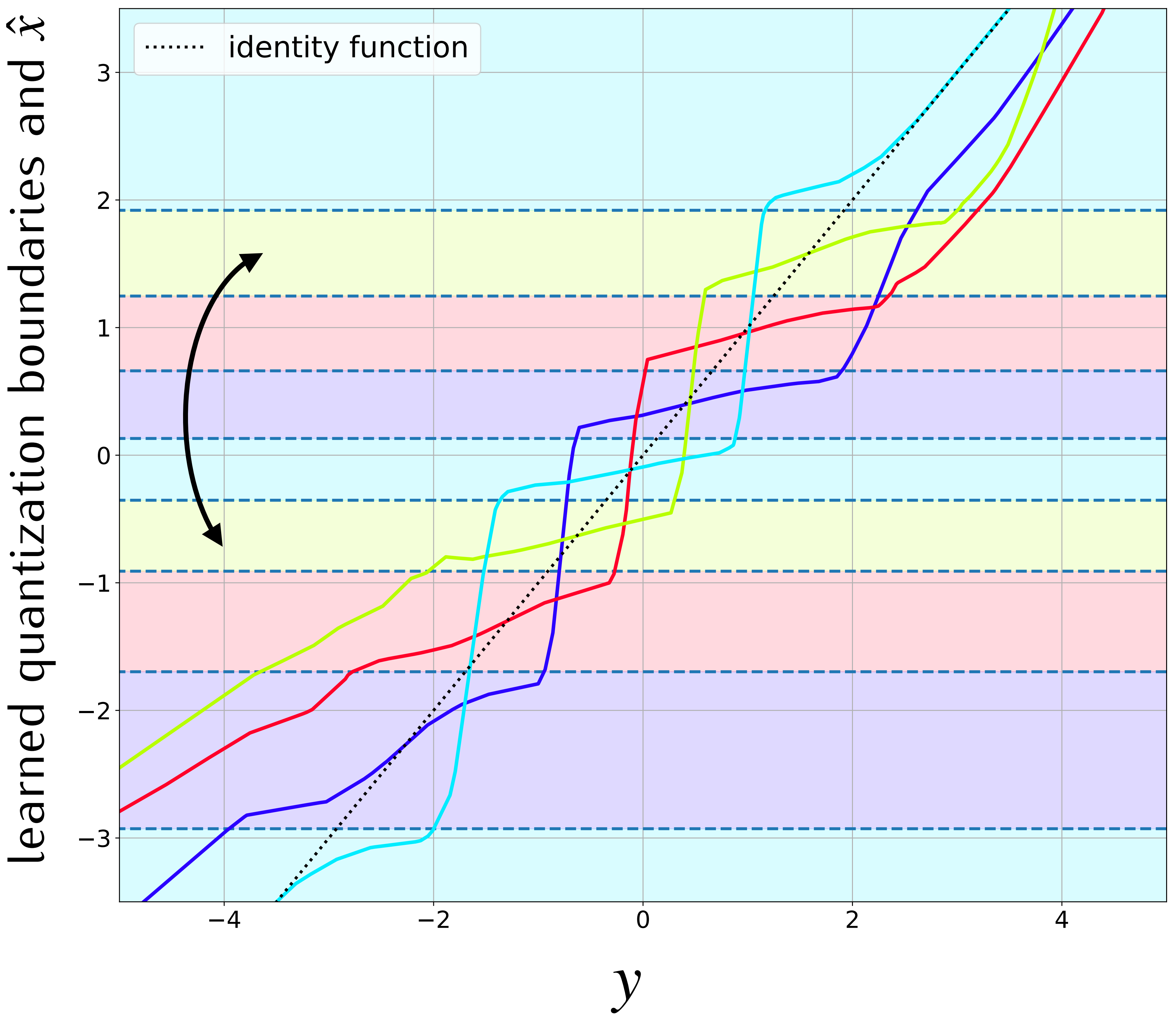
$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.

Results

Learned compressor recovers binning.

Learned encoder:

$u = \arg\max_v p_{\boldsymbol{\theta}}(v \,|\, x)$



same index

Marginal formulation.

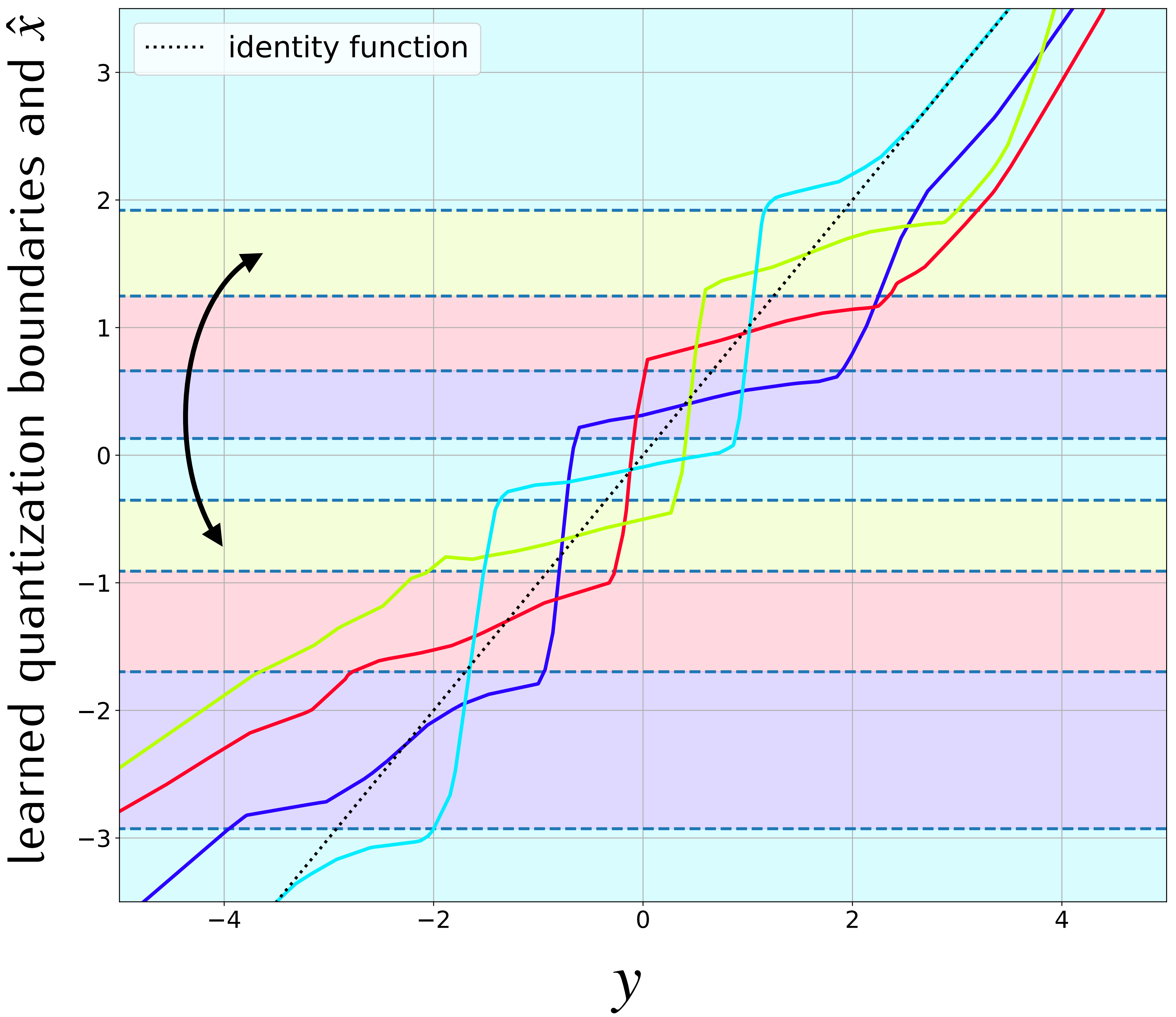
$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.

Results

Learned compressor recovers binning.

Learned encoder:

$u = \arg\max_v p_{\boldsymbol{\theta}}(v \mid x)$



same index

$\Longrightarrow$ binning.

Marginal formulation.

$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.

Results

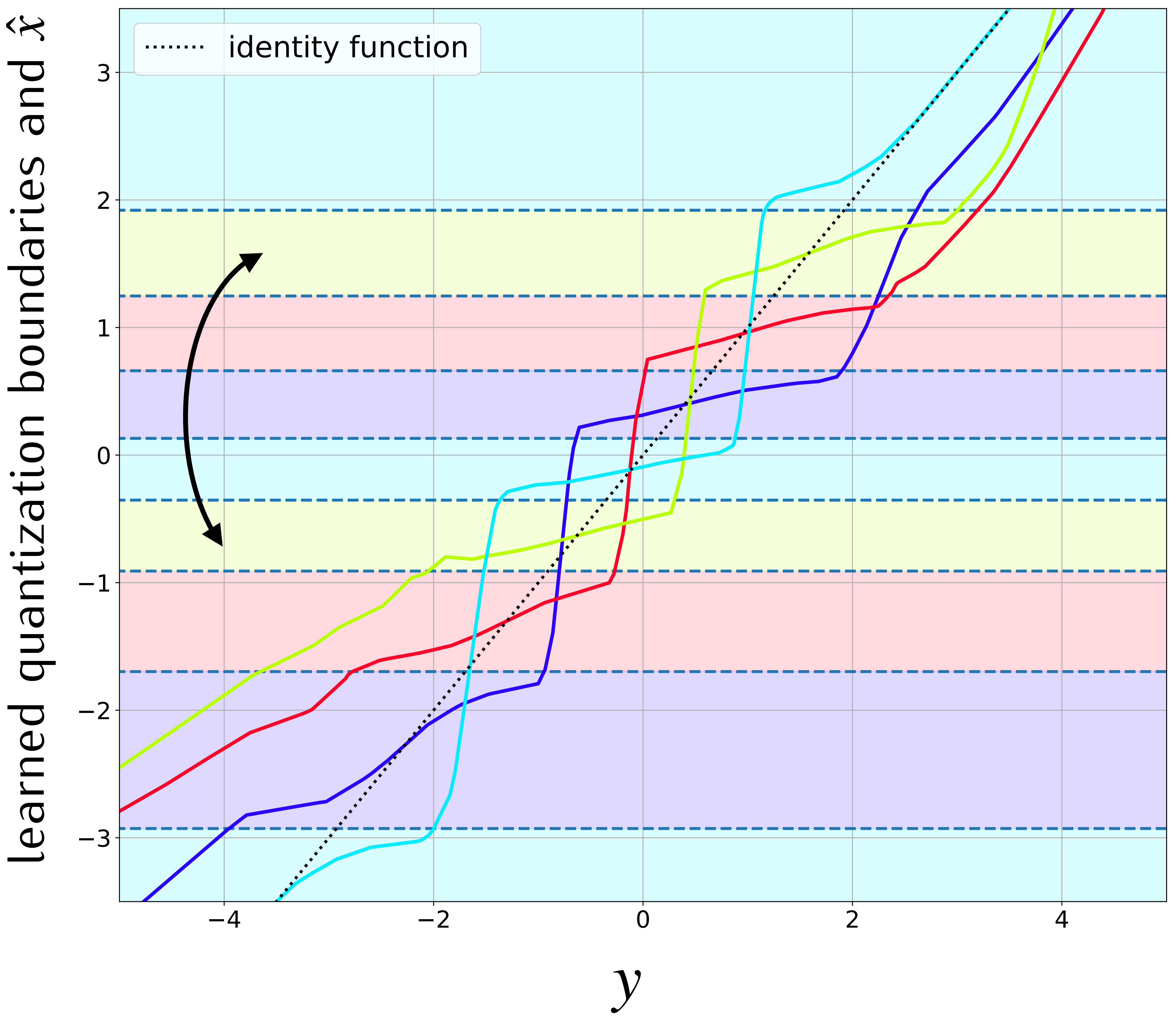
Learned compressor recovers binning.

Learned decoder:

$\hat{x} = g_{\boldsymbol{\phi}}(u, y)$

Learned encoder:

$u = \arg\max_v p_{\boldsymbol{\theta}}(v \,|\, x)$



same index

$\Longrightarrow$ binning.

Marginal formulation.

$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.

Results

Learned compressor recovers binning.

Learned decoder:

$\hat{x} = g_{\boldsymbol{\phi}}(u, y)$

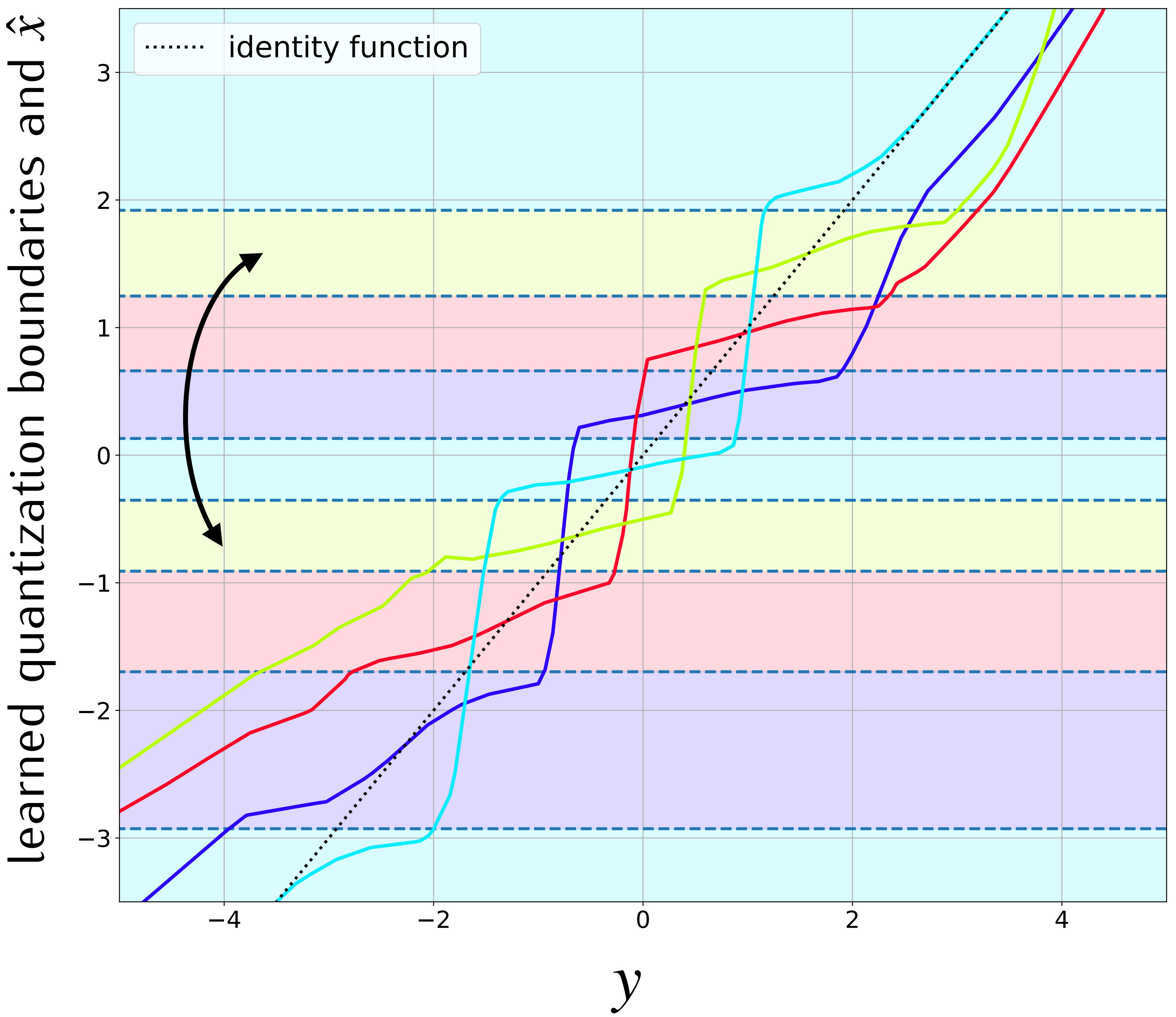
In quadratic-Gaussian WZ setup, the optimal decoder does:

$$\hat{x} = (1-\beta)\cdot y + \beta \cdot u,$$

where $\beta \propto \sigma_n^2$.

Learned encoder:

$u = \arg\max_v p_{\boldsymbol{\theta}}(v \mid x)$



same index

$\Longrightarrow$ binning.

Marginal formulation.

$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.

Results

Learned compressor recovers binning.

Learned decoder:

$\hat{x} = g_{\boldsymbol{\phi}}(u, y)$

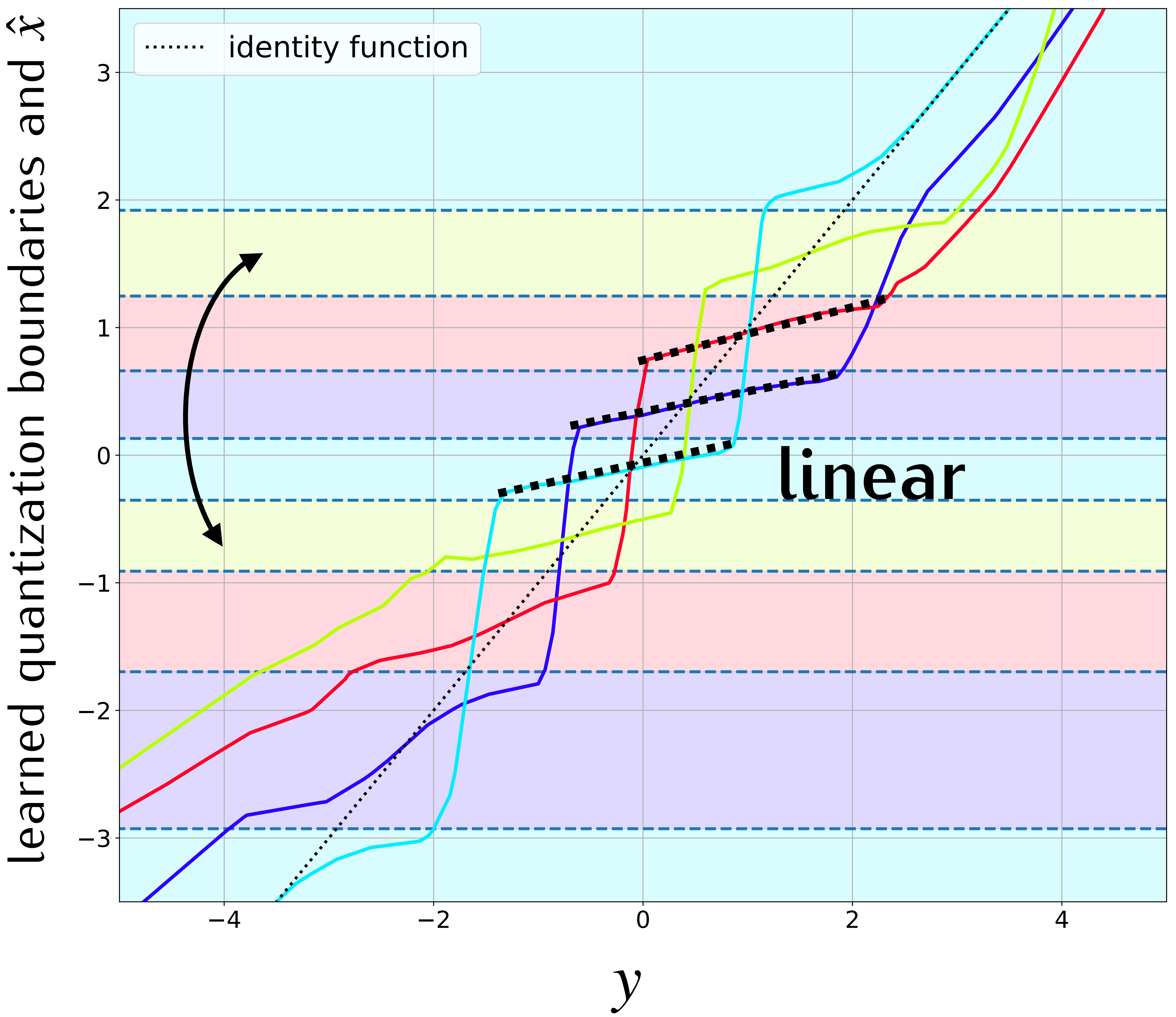
In quadratic-Gaussian WZ setup, the optimal decoder does:

$$\hat{x} = (1 - \beta) \cdot y + \beta \cdot u,$$

where $\beta \propto \sigma_n^2$.

Learned encoder:

$u = \arg\max_v p_{\boldsymbol{\theta}}(v \mid x)$



same index

$\Longrightarrow$ **binning.**

Marginal formulation.

$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.

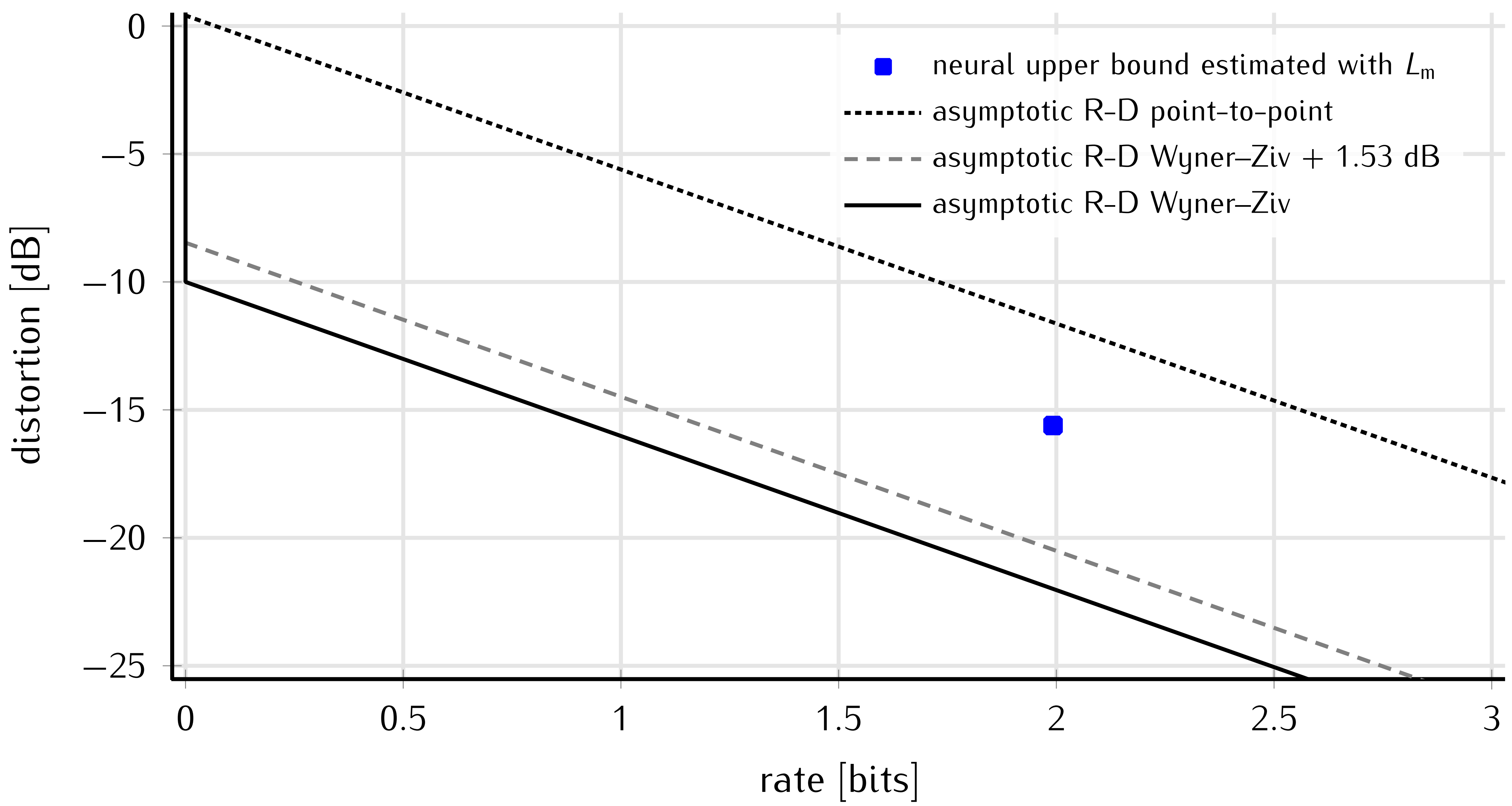
Recovers optimal reconstruction function.

Results

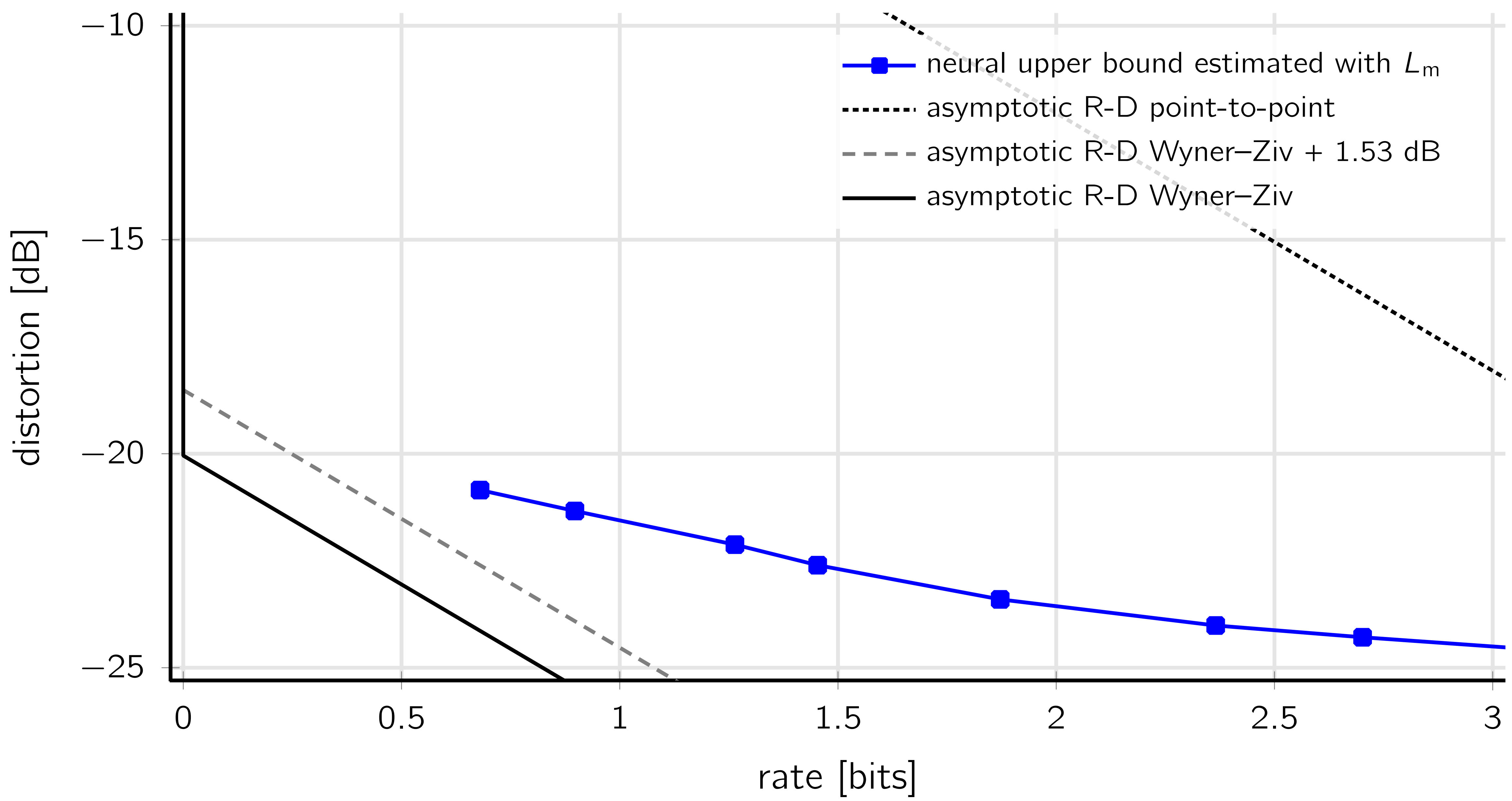
R-D performances.

Results

R-D performances.



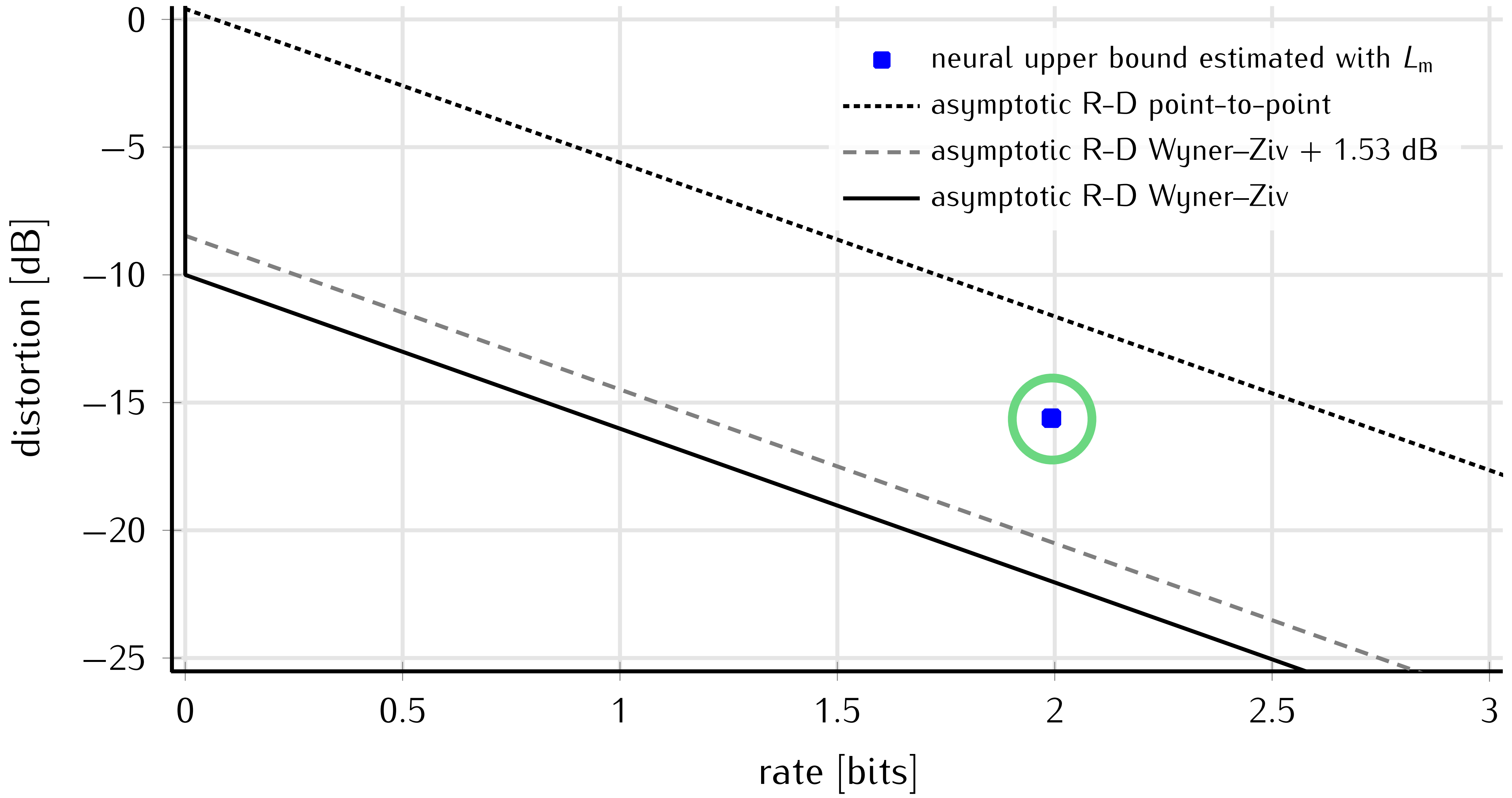
$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.



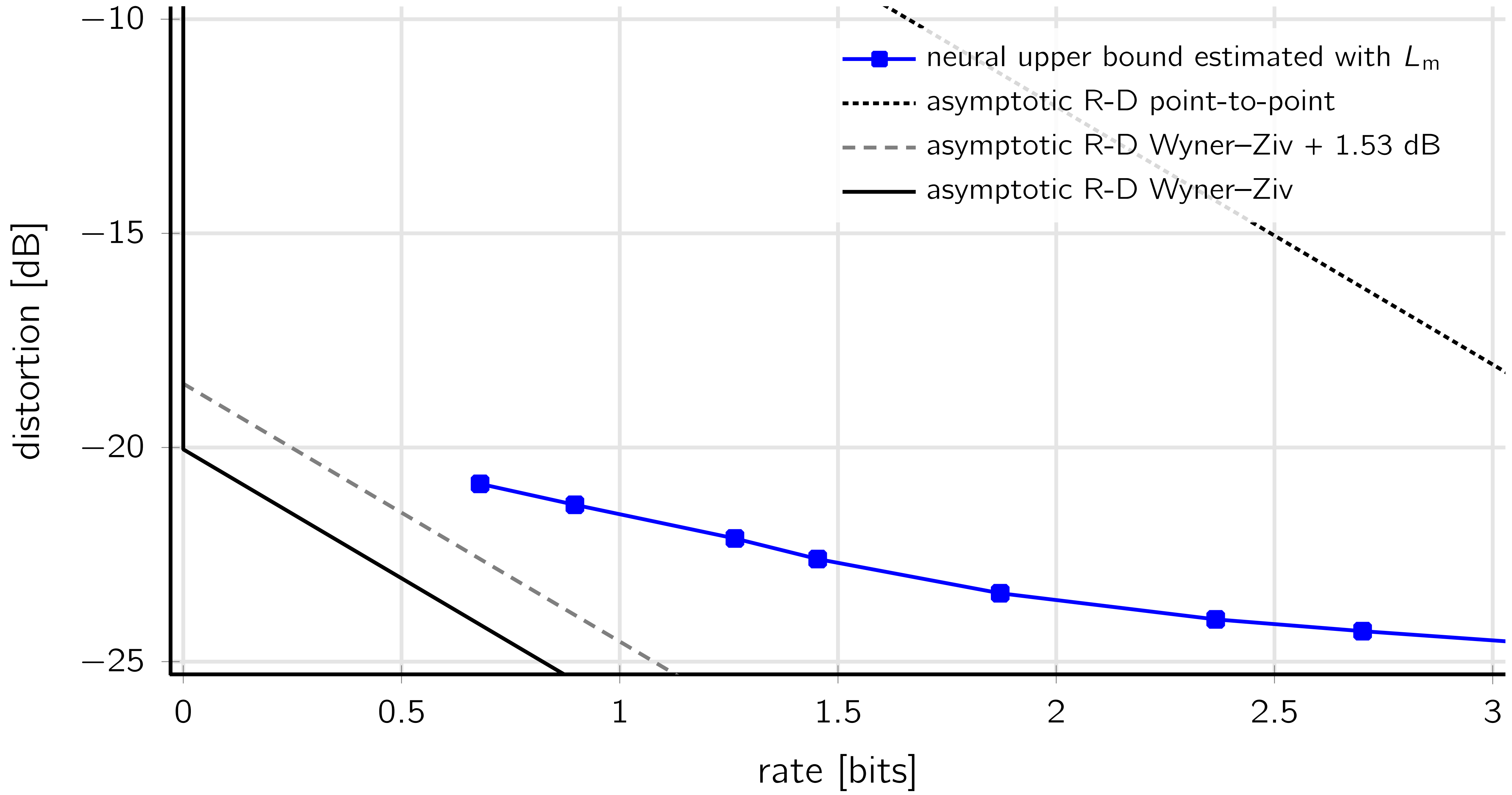
$Y = X + N$ with $X \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-2})$.

Results

R-D performances.



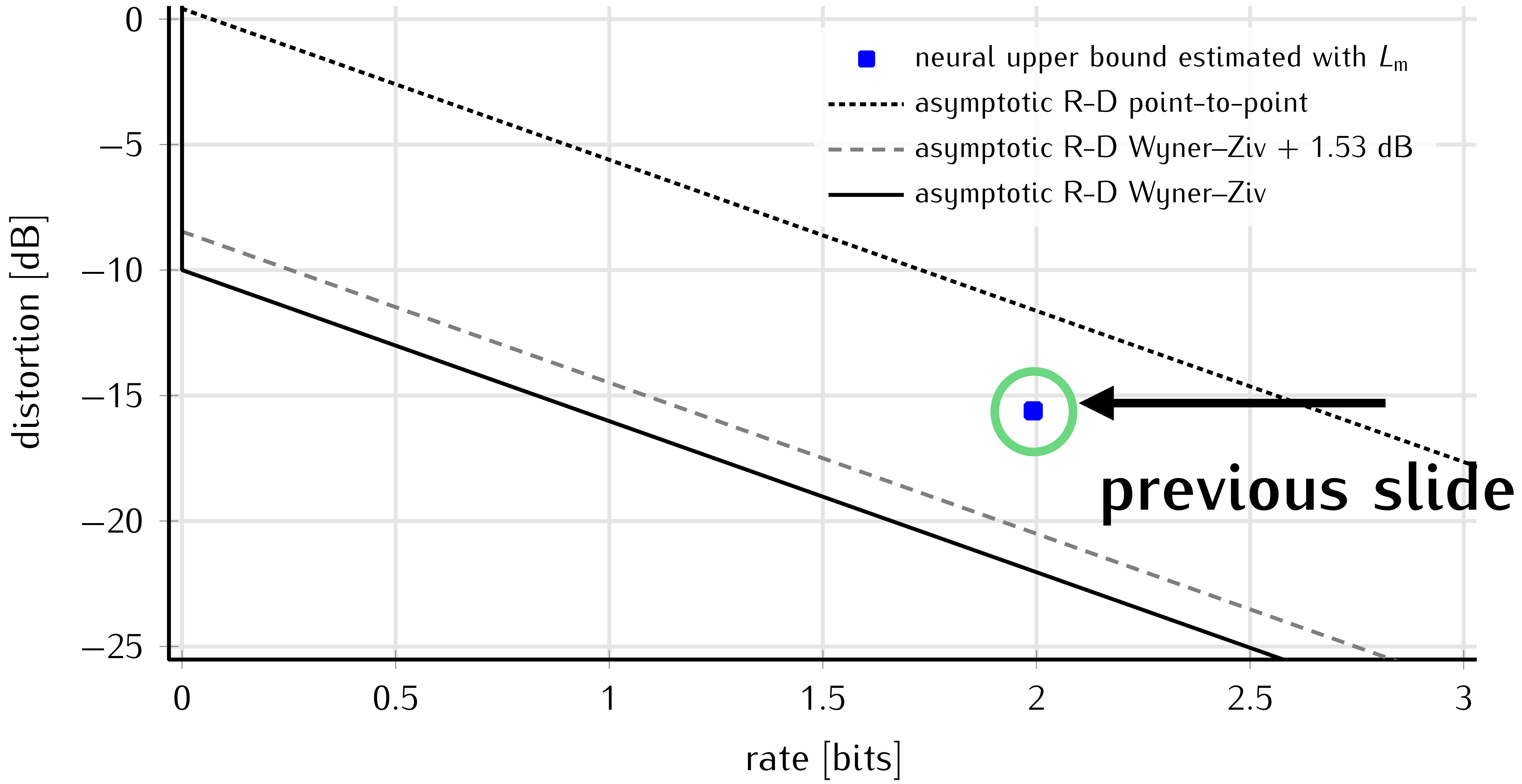
$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.



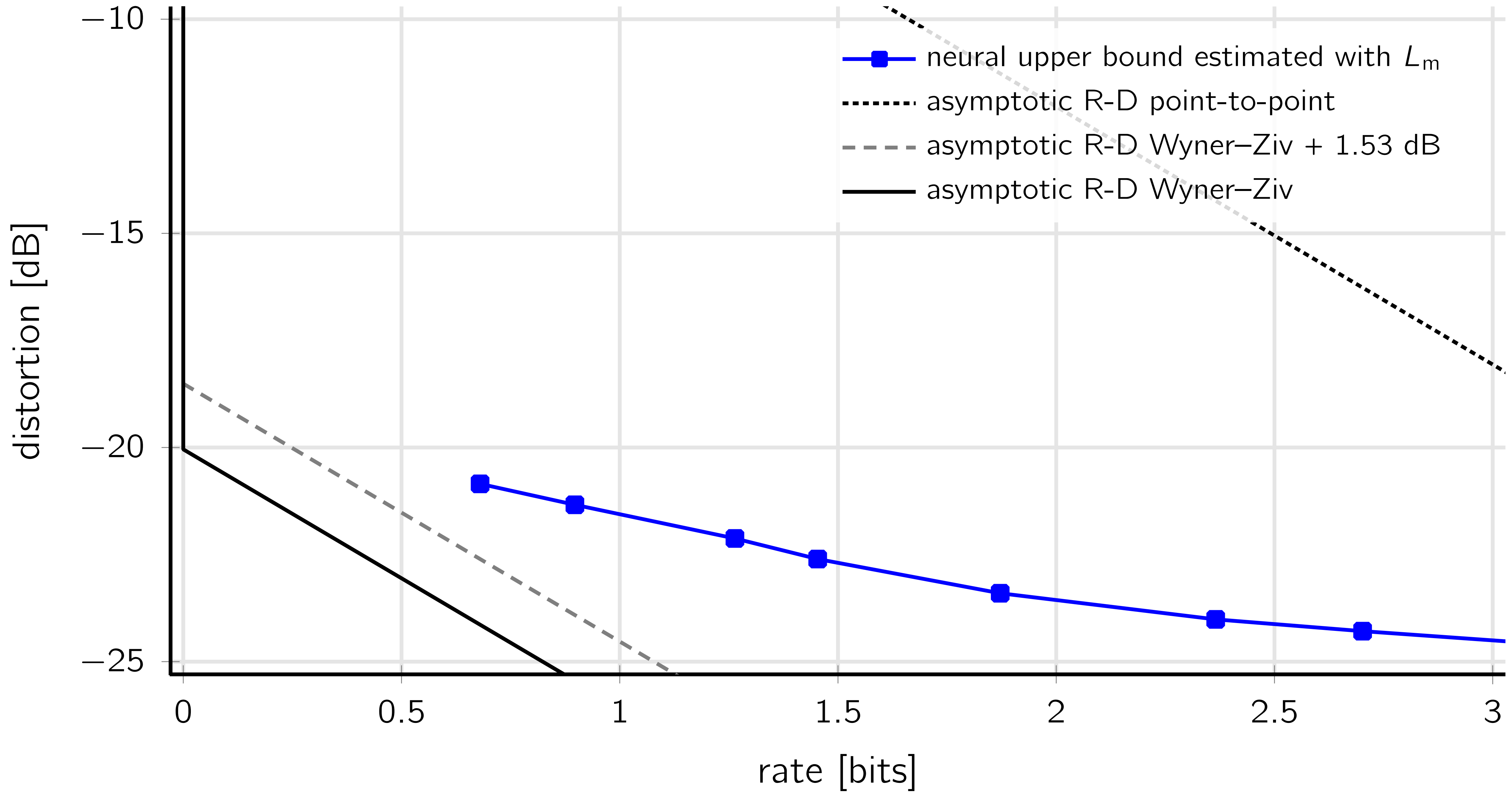
$Y = X + N$ with $X \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-2})$.

Results

R-D performances.



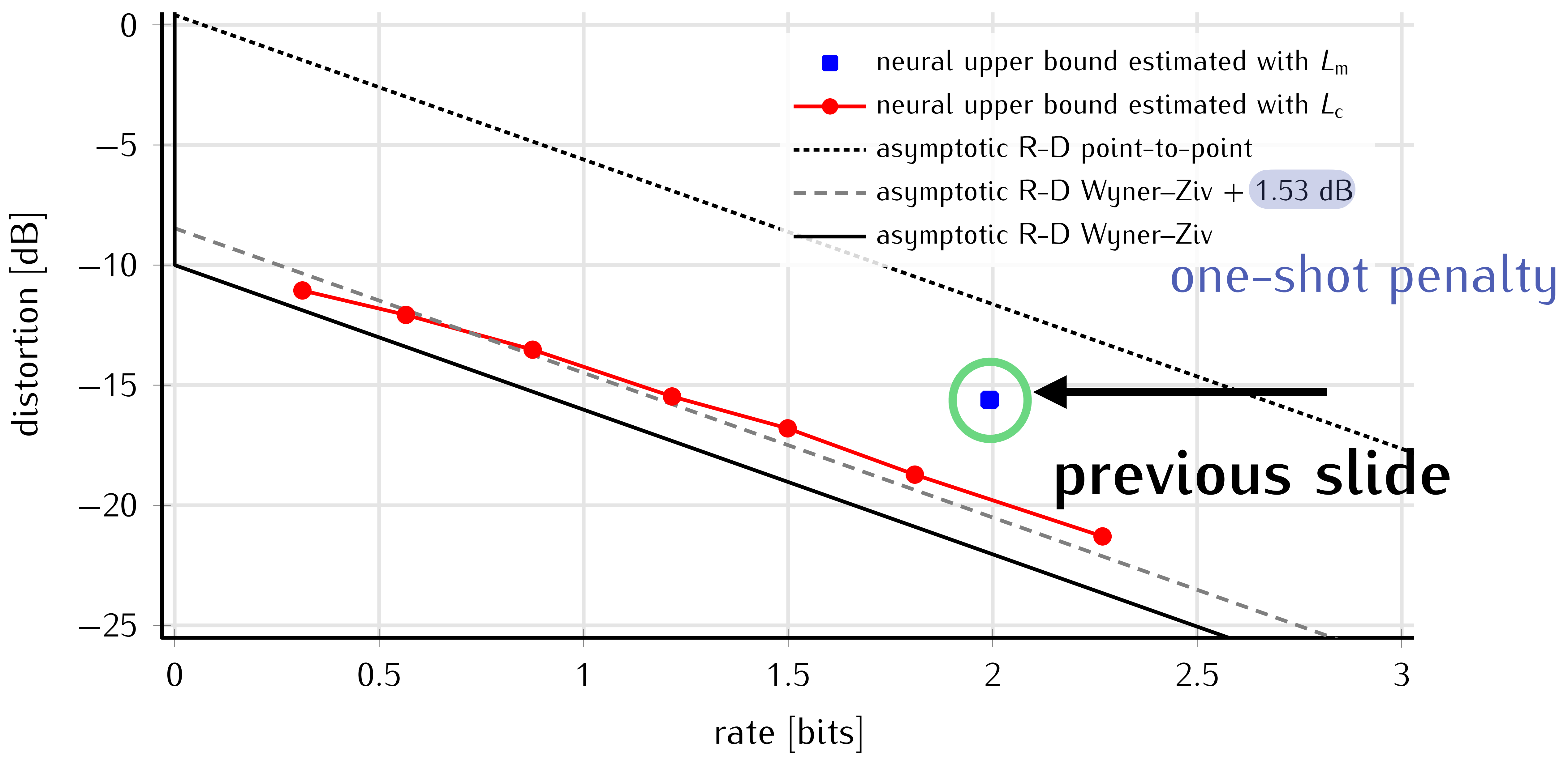
$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.



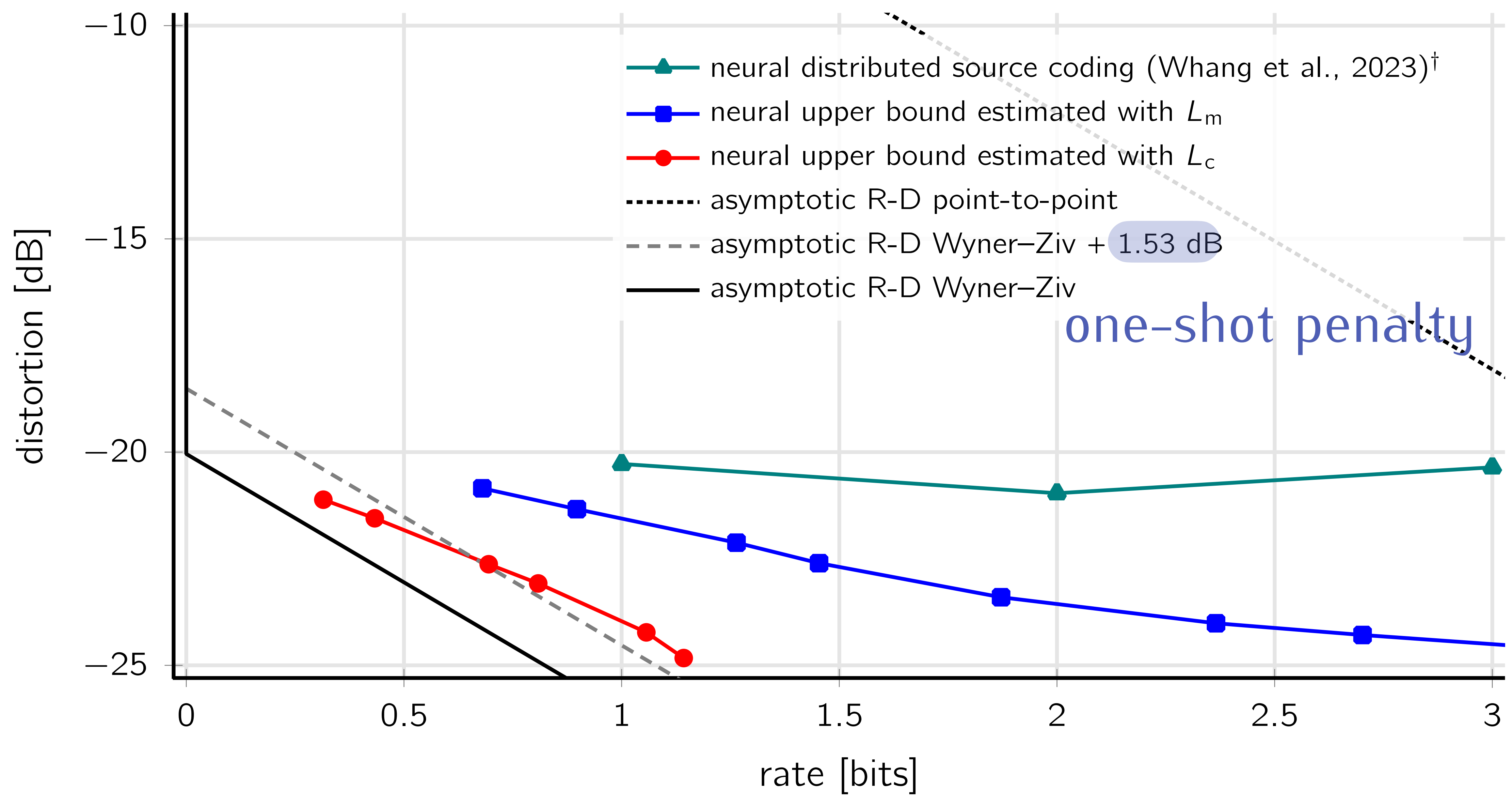
$Y = X + N$ with $X \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-2})$.

Results

R-D performances.



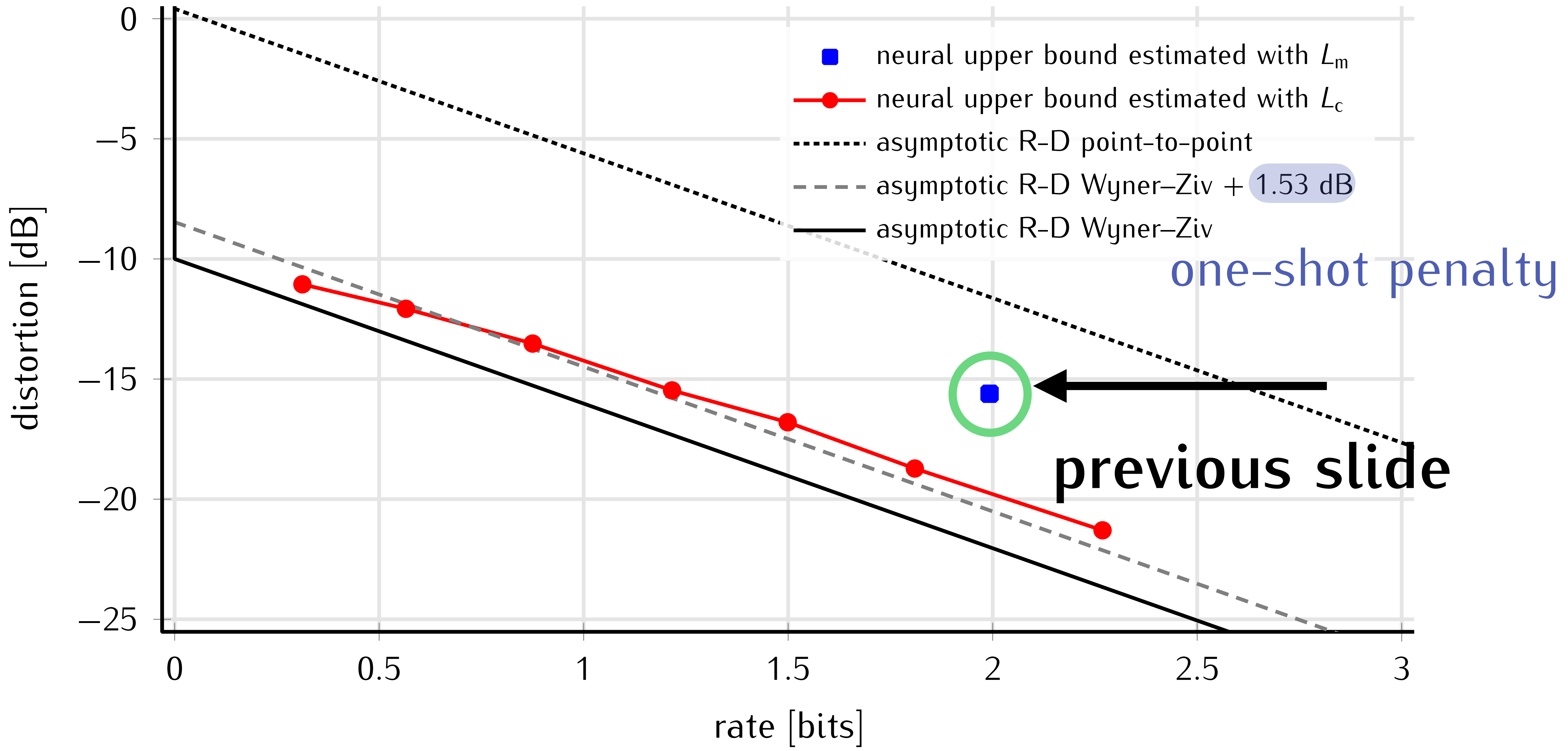
$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.



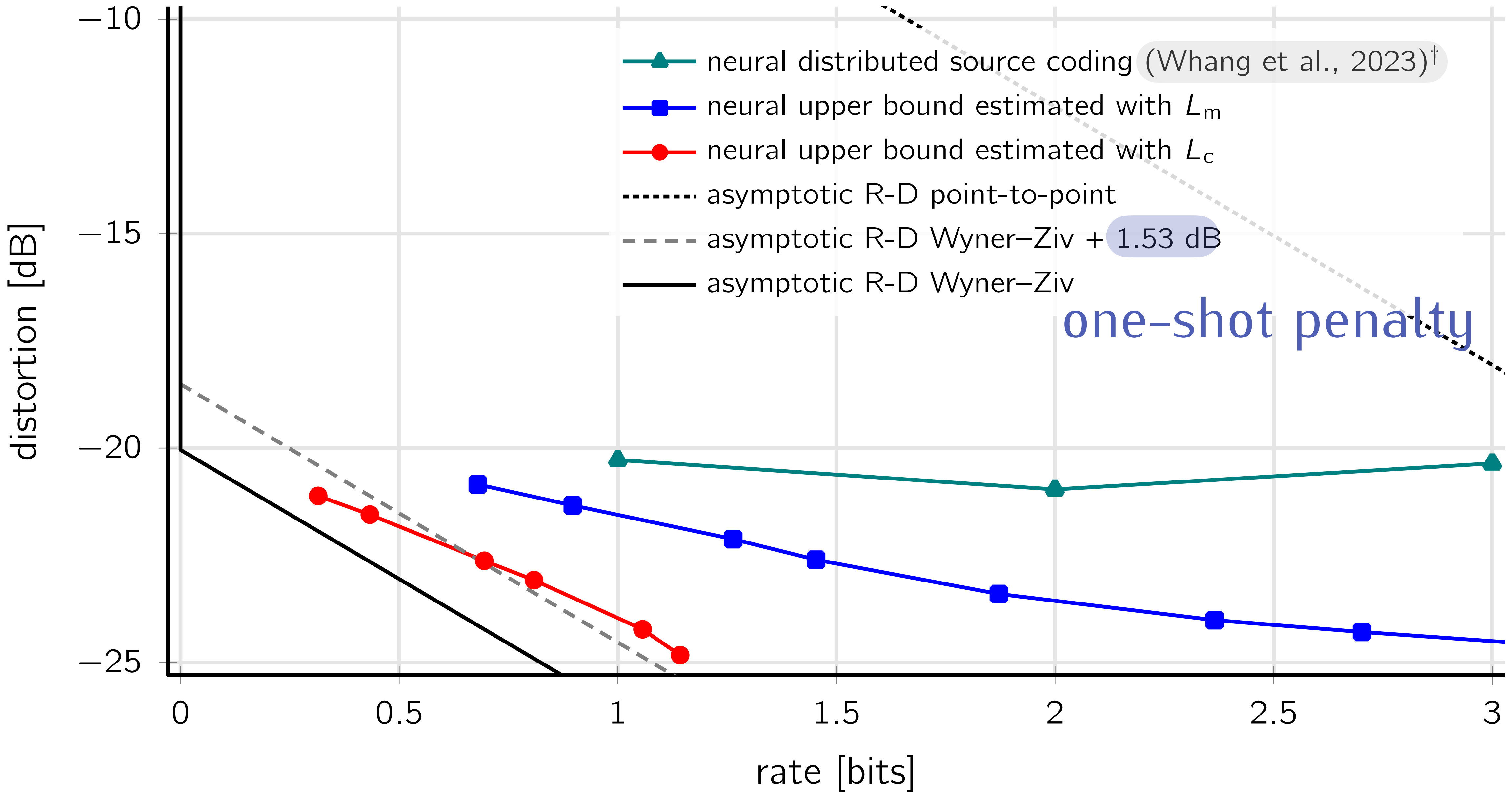
$Y = X + N$ with $X \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-2})$.

Results

R-D performances.



$X = Y + N$ with $Y \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-1})$.



$Y = X + N$ with $X \sim N(0,1)$ and $\mathrm{N} \sim N(0,10^{-2})$.

†J. Whang, A. Nagle, A. Acharya, H. Kim, and A. G. Dimakis, "Neural distributed source coding", https://arxiv.org/abs/2106.02797, 2023.

Take-home messages

Take-home messages

- To close the gap between theory and practice in distributed source coding, **learned compression is a promising approach**.

Take-home messages

- To close the gap between theory and practice in distributed source coding, **learned compression is a promising approach**.
- In quadratic-Gaussian case, learned compressors recover some elements of the optimal theoretical solution.

Take-home messages

- To close the gap between theory and practice in distributed source coding, **learned compression is a promising approach**.
- In quadratic-Gaussian case, learned compressors recover some elements of the optimal theoretical solution.
  - Binning in the source space and linear decoding functions.

Take-home messages

- To close the gap between theory and practice in distributed source coding, **learned compression is a promising approach**.
- In quadratic-Gaussian case, learned compressors recover some elements of the optimal theoretical solution.
  - Binning in the source space and linear decoding functions.
  - First-time **binning** emerges from learning.

Thank you. Questions?

Neural Distributed Compressor Does Binning

Ezgi Özyılkan*, Johannes Ballé†, Elza Erkip*
*NYU, †Google Research
ezgi.ozyilkan@nyu.edu

Neural Compression Workshop @ ICML 2023
Honolulu, HI | July 29, 2023

Thank you. Questions?

Neural Distributed Compressor Does Binning

Ezgi Özyılkan*, Johannes Ballé†, Elza Erkip*
*NYU, †Google Research
ezgi.ozyilkan@nyu.edu

Google Research

NYU | TANDON SCHOOL OF ENGINEERING

Neural Compression Workshop @ ICML 2023
Honolulu, HI | July 29, 2023

Presented at ***International Symposium of Information Theory (ISIT)*** *2023.*

References

- S. Verdú, “Fifty years of Shannon theory’’, *IEEE Transactions on Information Theory,* vol. 2, no. 5, p. 359–366, 1998.
- J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression”, *International Conference on Learning Representations*, 2017.
- A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder”, *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.
- D. Slepian and J. Wolf, “Noiseless coding of correlated information sources’’, *IEEE Transactions on Information Theory,* vol. 19, no. 4, pp. 471– 480, 1973.
- E. J. Gumbel, “Statistical theory of extreme values and some practical applications: a series of lectures”, *US Department of Commerce*, 1954.
- C. J. Maddison, A. Mnih, and Y. W. Teh, “The concrete distribution: a continuous relaxation of discrete random variables”, *International Conference on Learning Representations*, 2017.
- J. Whang, A. Nagle, A. Acharya, H. Kim, and A. G. Dimakis, “Neural distributed source coding”, `https://arxiv.org/abs/2106.02797`, 2023.