Autoencoders for Probabilistic Constellation Shaping

Thang K. Nguyen

May. 24, 2024

Contents

- Autoencoders
  - Basic idea
  - Training
  - Application
- My current work
  - Visible light communication (VLC)
  - Probabilistic constellation shaping (PCS)
  - Autoencoder-based PCS design
  - Extension and direction

Autoencoders

- ❑ Basic idea
- ❑ Training
- ❑ Application

Basic idea

Autoencoders

↓

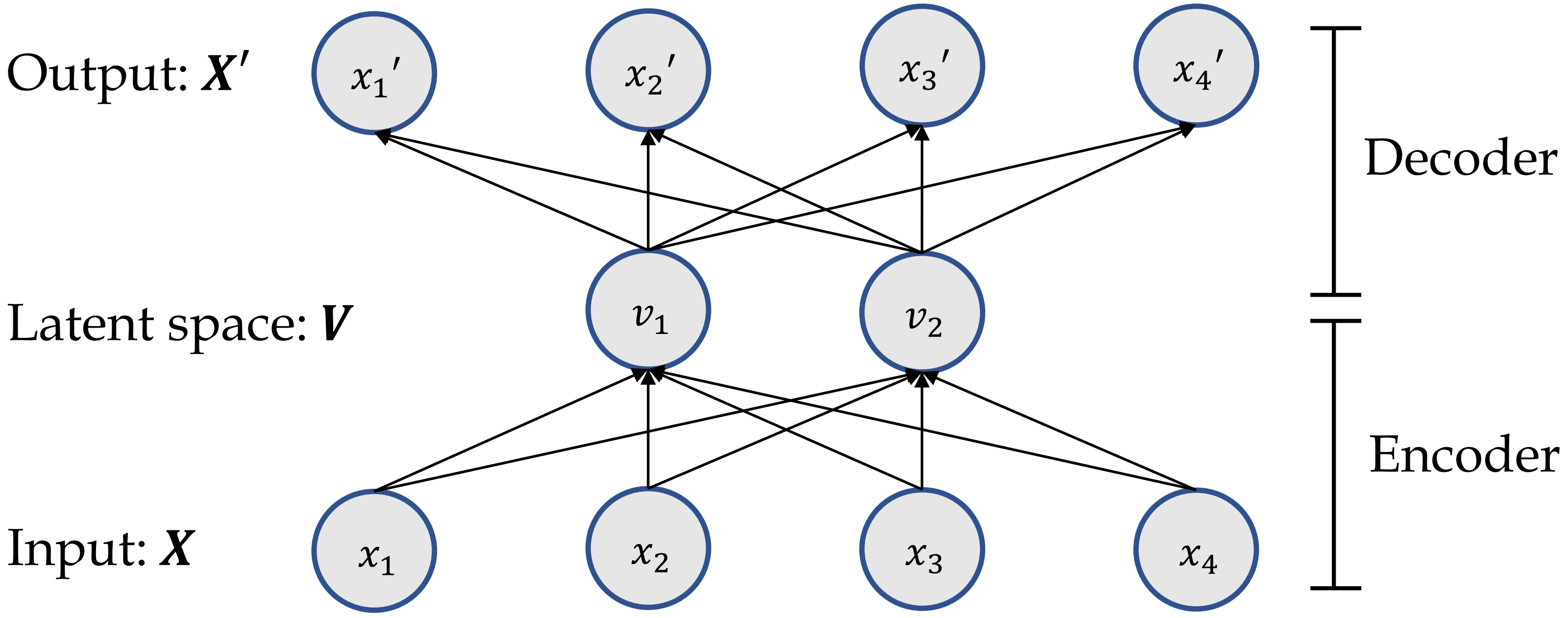
Unsupervised learning

↓

Representation learning:

Learning **patterns** in data

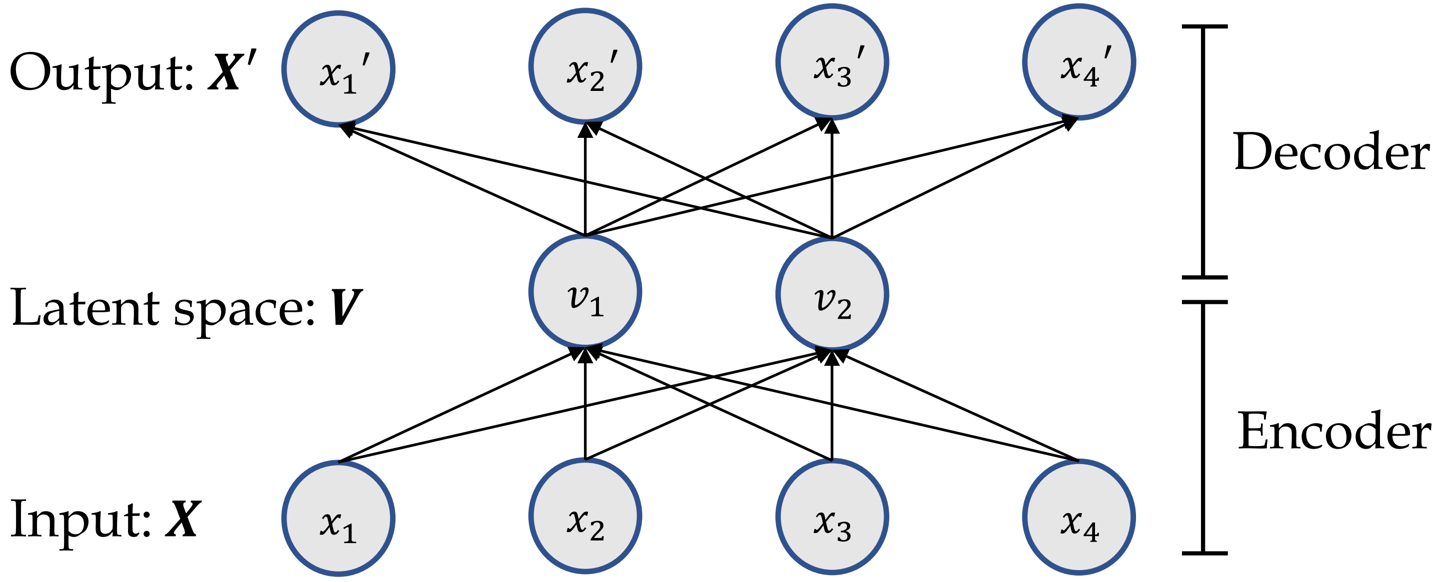
- An **autoencoder** is a network trained to learn the **identity function:** Output ≈ Input



- **Encoder** $f(\boldsymbol{X})$ maps input to a **lower-dimensional representation** (latent space $\boldsymbol{V}$)
- **Decoder** $g(\boldsymbol{V})$ decompress representation back to original domain ($\boldsymbol{X}'$)

Basic idea

- Basic idea: Create an architecture with a latent space (**bottleneck layer**), which ensures a **lower-dimensional representation** of the original data
- The latent space keeps the most important attributes of the input data



Why autoencoders?

- Map high-dimensional data to low dimensions for visualization
- Compression
- Learn abstract features of data

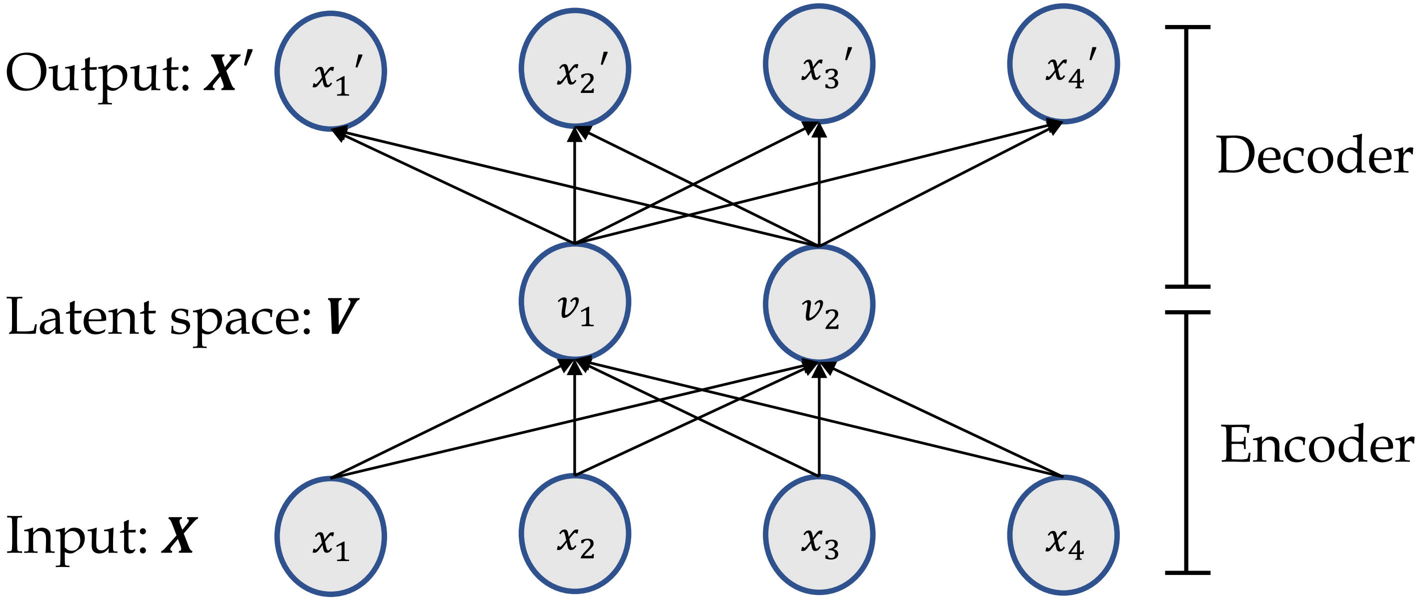
But how do we train an autoencoder?

Training

- Minimize reconstruction error:

$$E(\boldsymbol{X}, \boldsymbol{X}')$$

Original data   Reconstructed data



- Backpropagation algorithm

- Requirement for an autoencoder:

  1. Sensitive enough to input data to reconstruct it

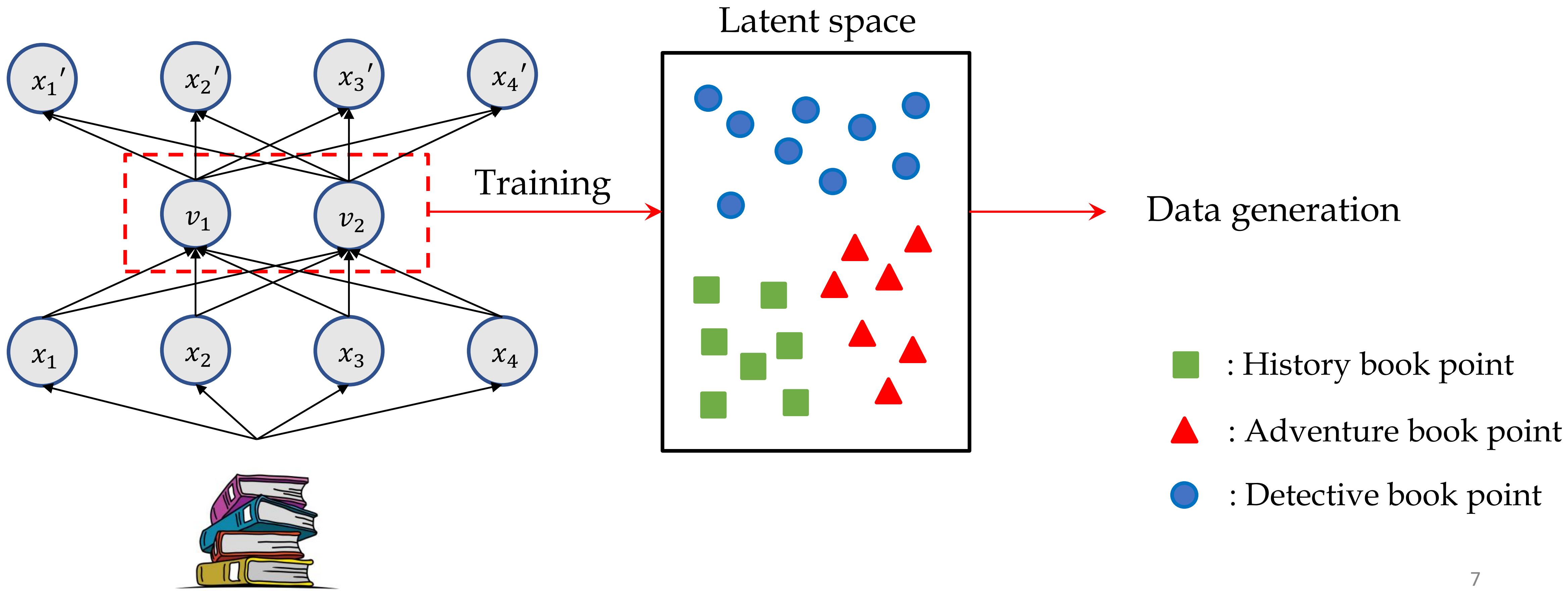
  2. Insensitive enough to input data **not** to overfit it

$$E(\boldsymbol{X}, \boldsymbol{X}') + \text{regularization}$$

Application

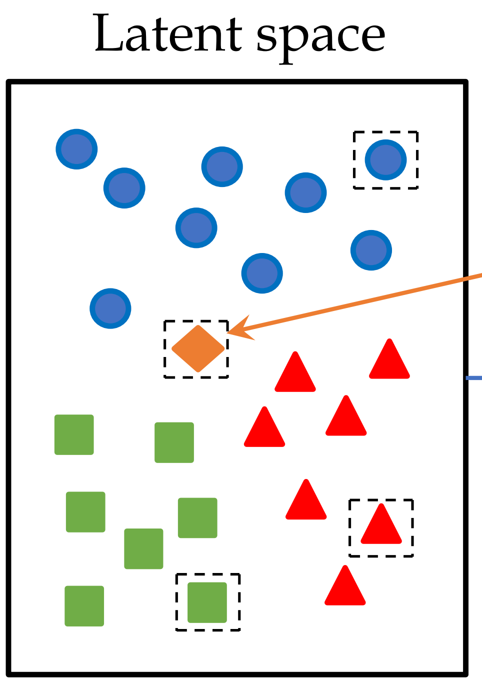
❑ The latent space keeps the most important attributes of the input data

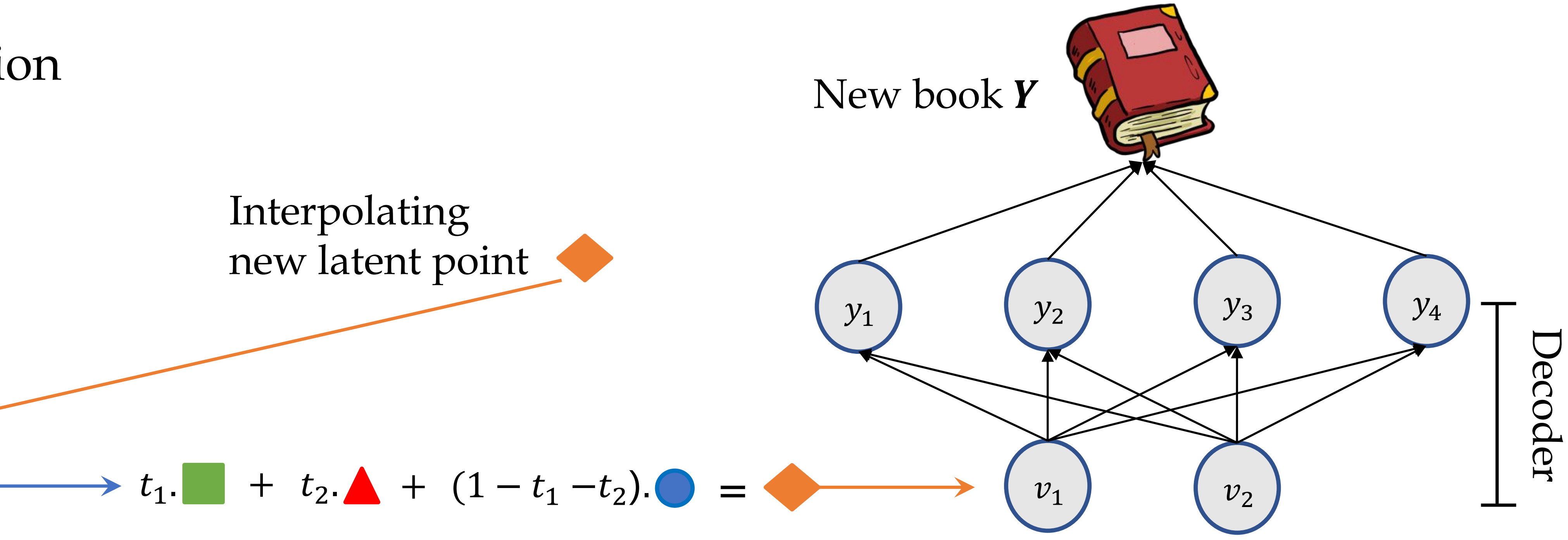
➡ We can leverage the latent space to perform several interesting tasks



Application

❑ Data generation





: History book point

: Adventure book point

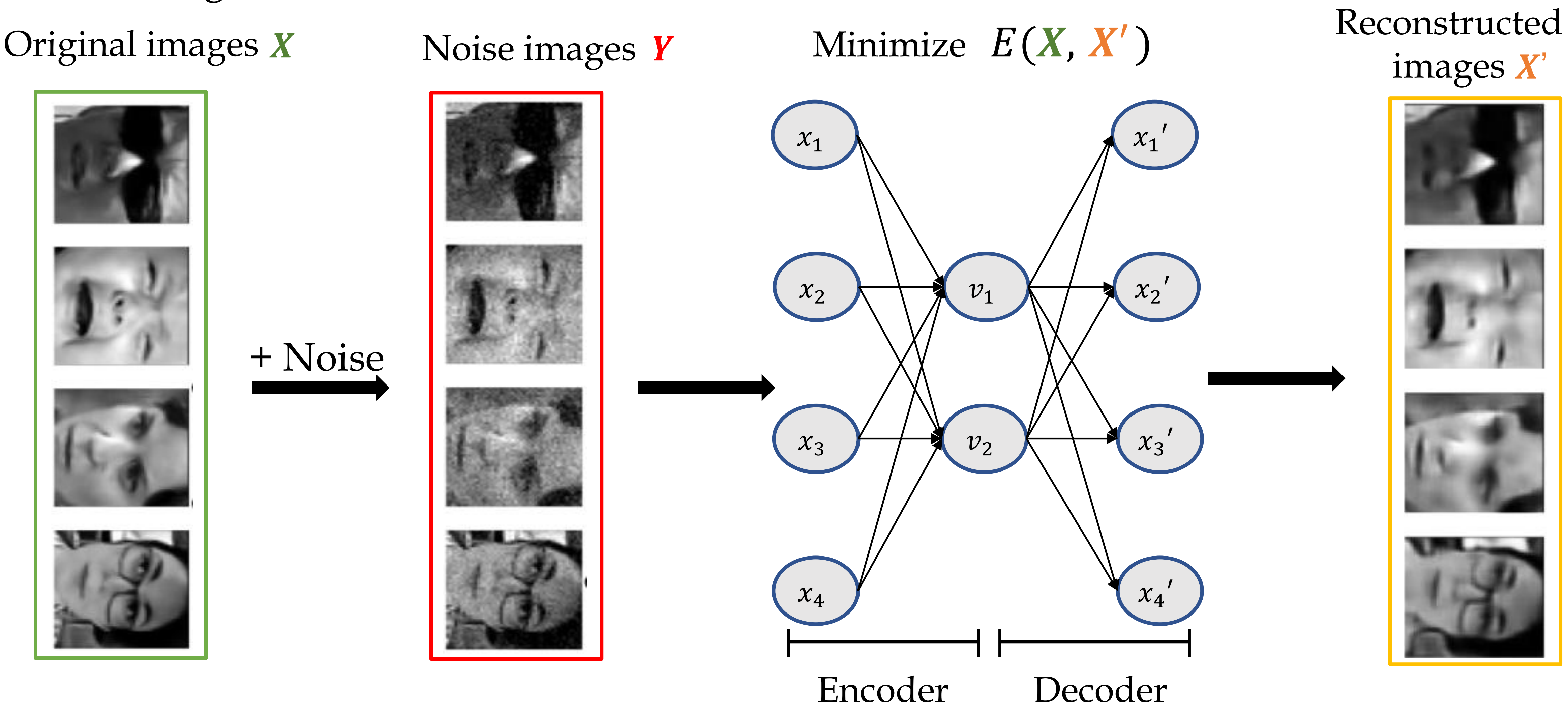
: Detective book point

New book $\boldsymbol{Y}$ could describe:

- A detective adventure that takes place during a historical event
- Discuss the historical event with a detective adventurous narrative style

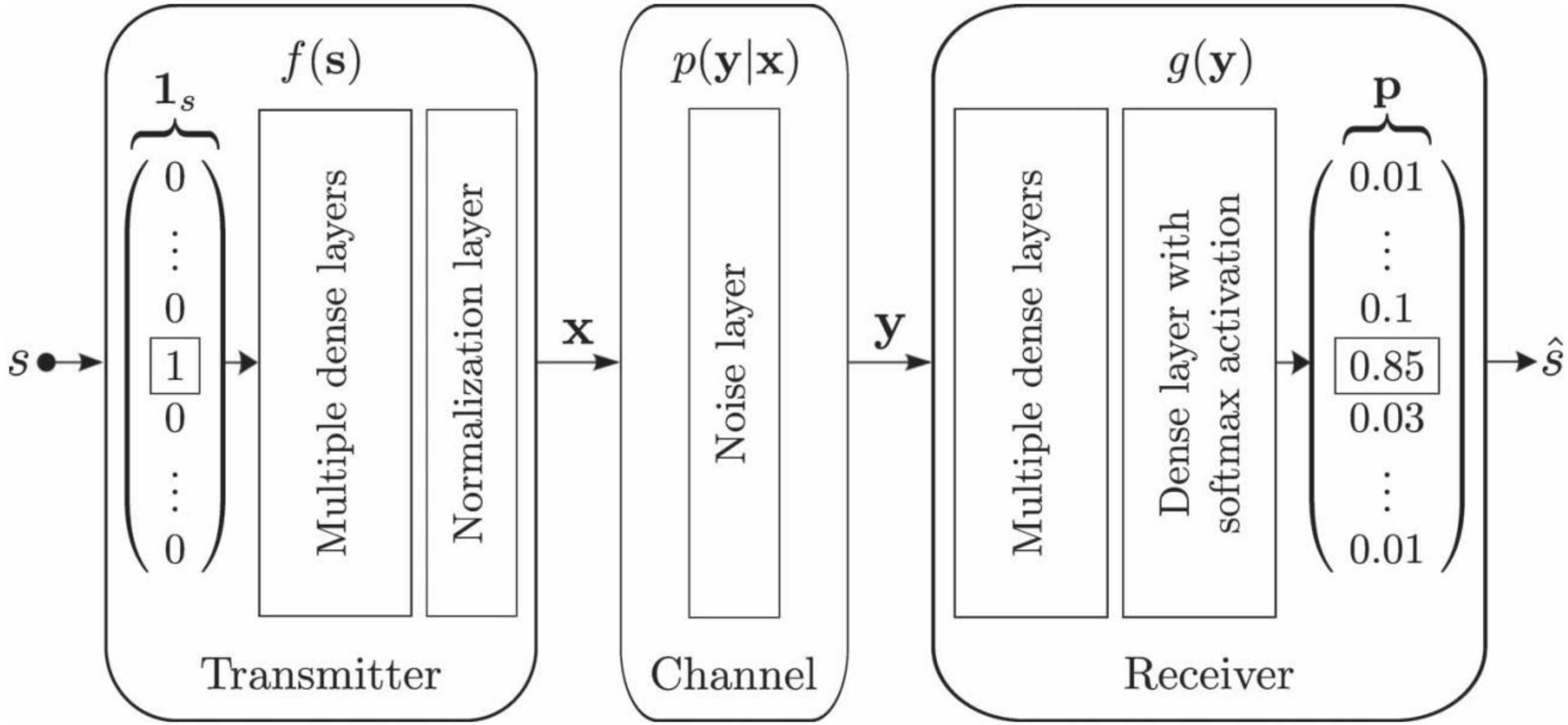
Application

- Denoising



➡ This concept can be applied to **an end-to-end communication system**

Application



A communication system over an AWGN channel represented as an autoencoder [1]

- Input $\boldsymbol{s}$: The transmitted symbol $\boldsymbol{s} \ \in \{1, \dots, M\}$
- Output $\boldsymbol{p}$: Probability vector over all possible symbol
- $\hat{\boldsymbol{s}}$: Estimated the transmitted symbol

- The autoencoder is trained on the set of all possible transmitted symbol $\boldsymbol{s}$
- Loss function is the categorical cross entropy between $\mathbf{1}_{\boldsymbol{s}}$ and $\mathbf{p}$
- It learns an **intermediate representation x** of **s** (constellation, coding) robust to channel perturbations

[1] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563-575, Dec. 2017, doi: 10.1109/TCCN.2017.2758370.

My current work

- Visible light communication (VLC)
- Probabilistic constellation shaping (PCS)
- Autoencoder-based PCS design
- Extension and direction

Visible light communication

❑ It uses the emitted light from the light-emitted diode (LED) as a transmission medium

❖ Dual functionality
  - Illumination (primary)
  - Communication (secondary)

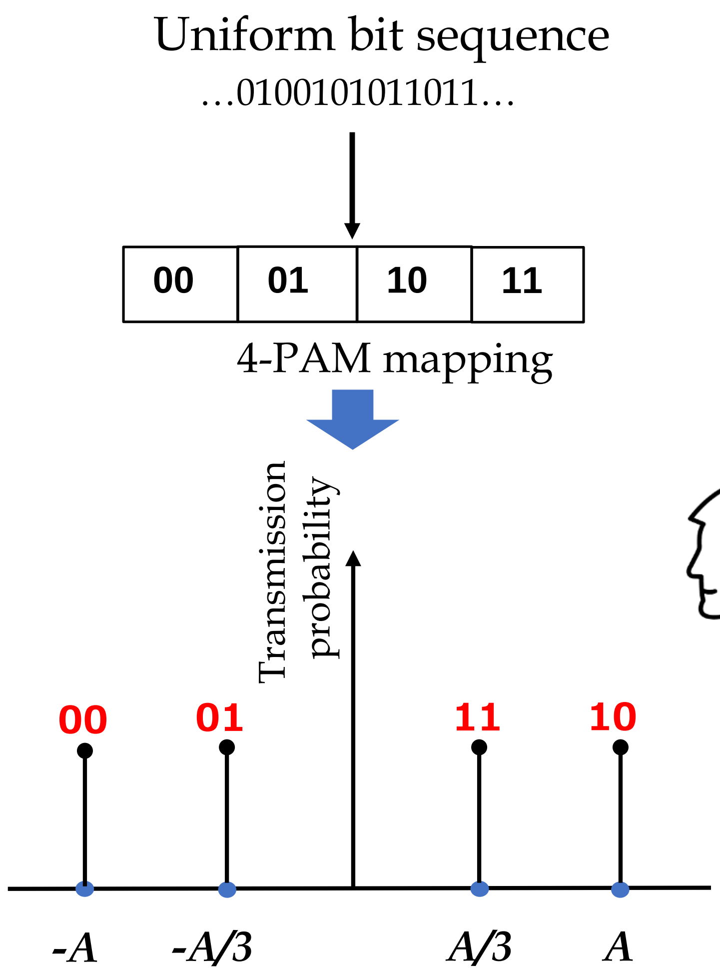
❖ Immunity to interference from other electromagnetic sources

❖ Environment friendly
  - Hospital
  - Airplane

However, the practical deployment of VLC systems faces challenges in **achieving high transmission rates** under peak amplitude constraints of the LEDs

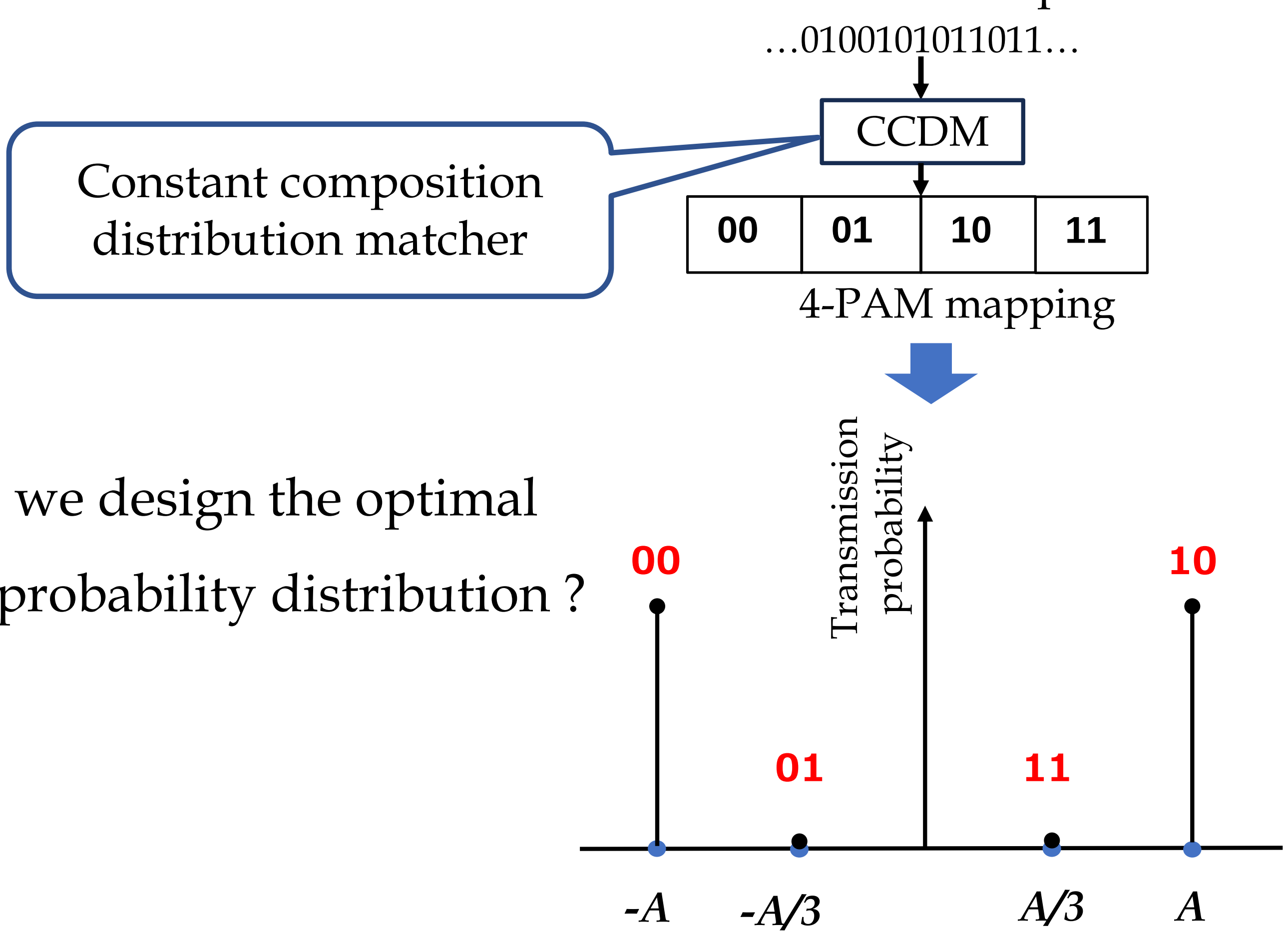
Probabilistic constellation shaping

- Probabilistic constellation shaping is an approach to improve the achievable rate by **designing the symbol transmission probability**



Uniform signaling

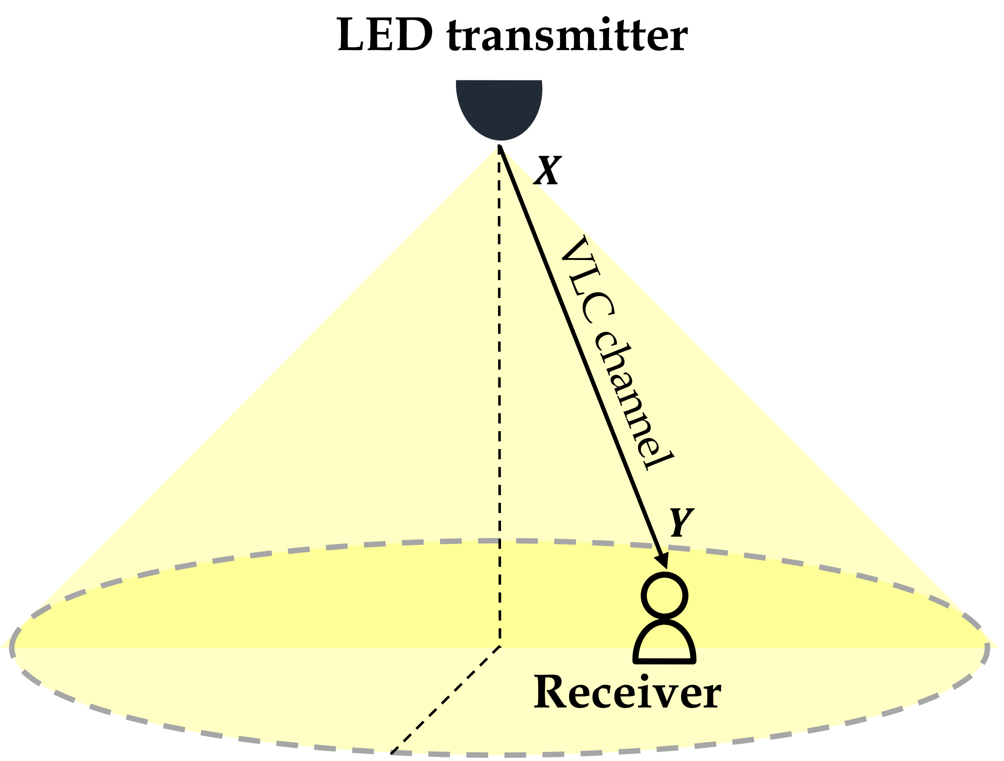
How do we design the optimal symbol probability distribution ?



Probabilistic Constellation Shaping

Probabilistic constellation shaping

How to design the optimal symbol probability distribution?



Single input single output (SISO) visible light (VLC) communication system

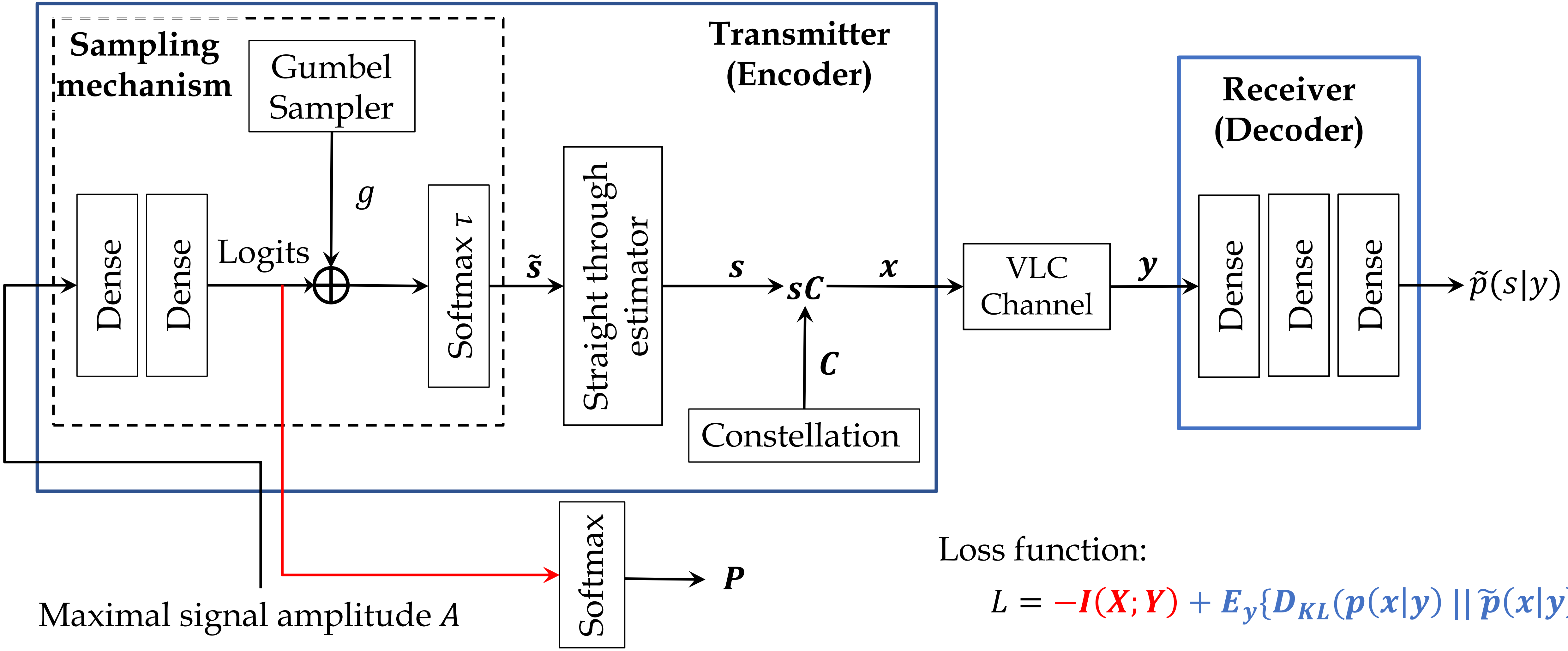
- **Purpose**: Maximize the achievable rate between the LED transmitter and receiver
- **Variable**: Symbol probability distribution ***P*** of M-PAM constellation

❖ Achievable rate $I(X,Y)$:

- Amount of information that one random variable ***Y*** contains about another random variable ***X***
- Measure the reduction in uncertainty about ***X*** given the knowledge of ***Y***

➡ Representing the **end-to-end SISO VLC system** as **an autoencoder** is an approach to learning the optimal symbol distribution

Autoencoder-based PCS design



Loss function:

$$L = -\boldsymbol{I}(\boldsymbol{X};\boldsymbol{Y}) + \boldsymbol{E}_y\{\boldsymbol{D}_{KL}(\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y})\ ||\ \widetilde{\boldsymbol{p}}(\boldsymbol{x}|\boldsymbol{y})\}$$

- The autoencoder learns **a sampling mechanism** (***P***) of transmitted symbols to **maximize the achievable rate** and **minimize the error construction** of transmitted symbol ***s*** at the receiver

Autoencoder-based PCS design

❑ Current results

Achievable rate versus maximal signal amplitude A

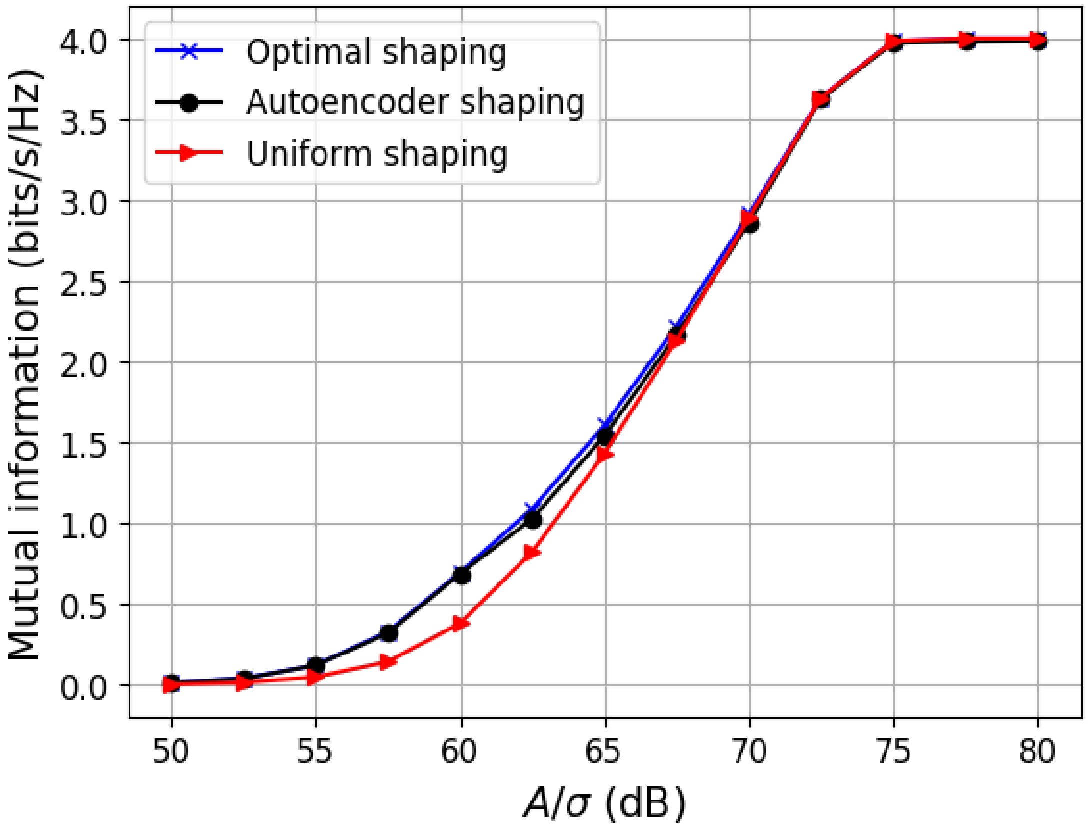


Figure 1

Learning behavior of autoencoder with $A/\sigma = 60$ (dB)

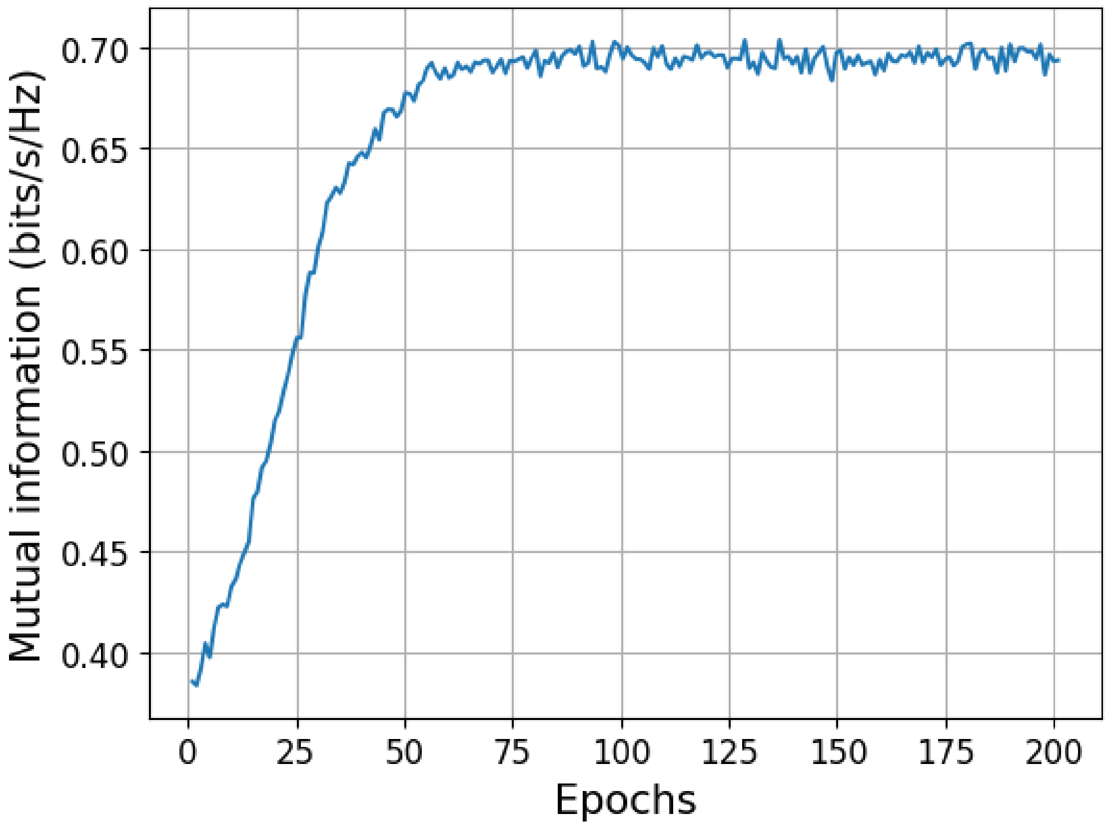
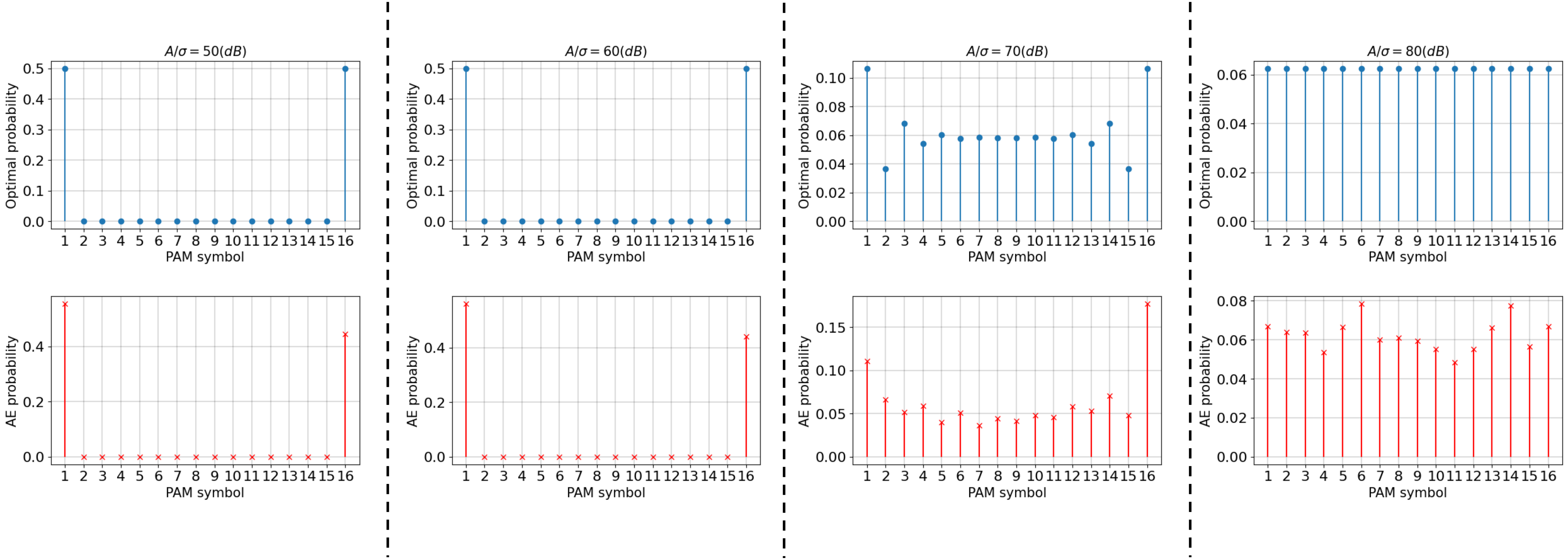


Figure 2

Autoencoder-based PCS design

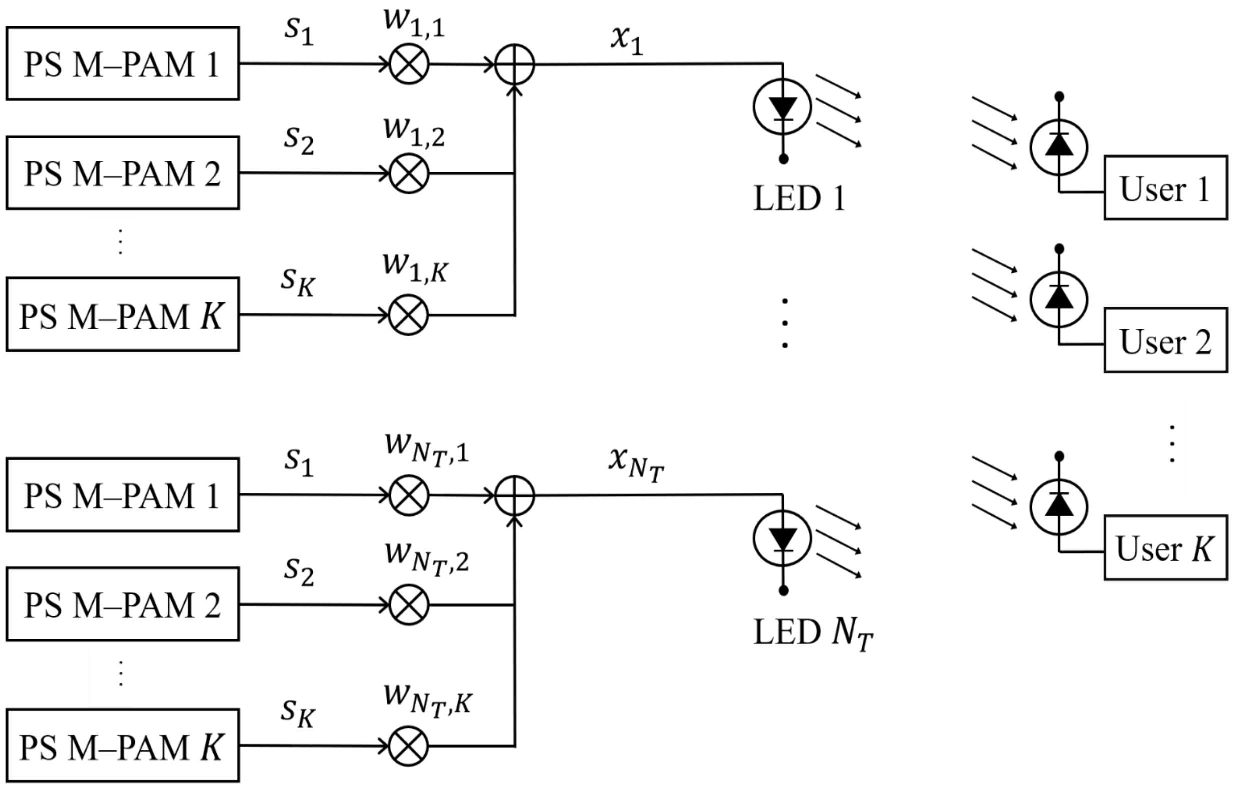
❑ Current results

Symbol probability distribution



Extension and direction

❑ Extend the PCS design for multiple transmitters and multiple users scenario



- $\boldsymbol{s} = [s_1 \;\; s_2 \cdots \;\; s_K]$: vector of data symbols for K users.
- $s_k \sim \{s_{k,m_k}\}$: data symbol for user k, drawn from the PS M-PAM k.

$$k \in \{1, 2, \ldots, K\}, m_k \in \{1, 2, \ldots, M\}$$

- $\boldsymbol{w}_k = [w_{k,1} \;\; w_{k,2} \cdots \;\; w_{k,N_T}]$ is the precoding vector for user k.

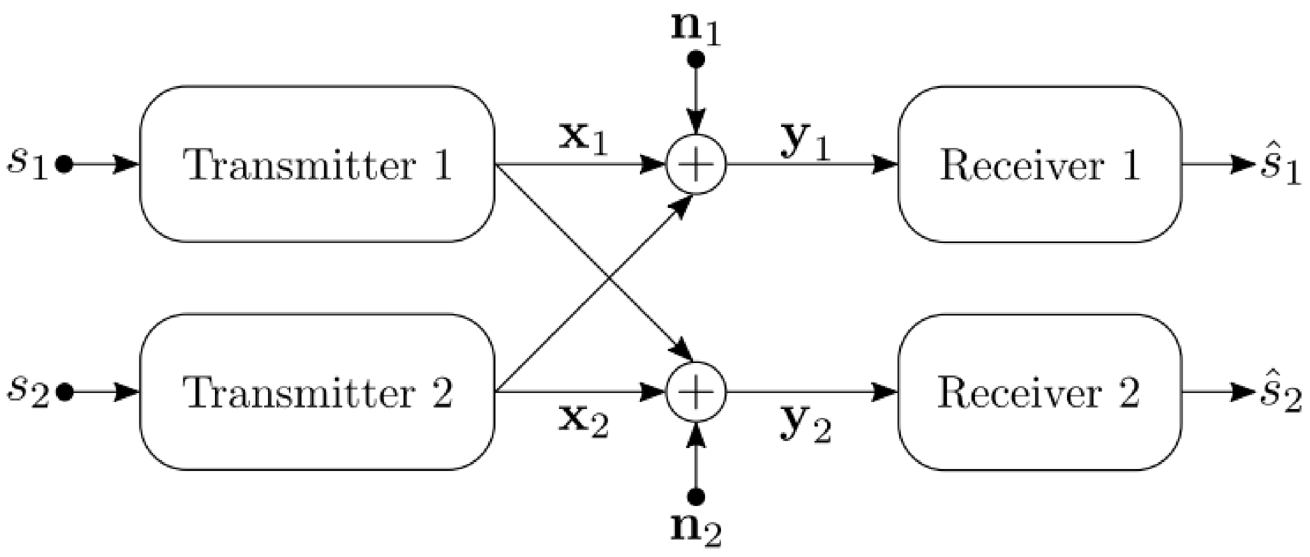
➢ Besides symbol distribution $\boldsymbol{P}$, the precoding matrix $\boldsymbol{W}$ is another optimization variable

➢ The autoencoder network needs to be modified

Extension and direction

❑ Extend the PCS design for multiple transmitters and multiple users scenario

- Both transmitter-receiver pairs are implemented as autoencoders



- The two-user interference channel is seen as a combination of **two interfering autoencoders** that try to maximize the sum of achievable rates and reconstruct their respective symbols
- The two autoencoders will be trained to learn both the **sampling mechanism (P)** and the **internal representation** of transmitted symbol with the **precoding matrix (W)**

Extension and direction

❑Extend the PCS design for multiple transmitters and multiple user scenarios with channel uncertainty

- Current work: PCS and precoding design with the **perfect knowledge** about the channel state information (CSI) in the transmitter
- Practical scenario: Due to users movements, CSI is **imperfect** due to outdated feedback or erroneous channel estimation

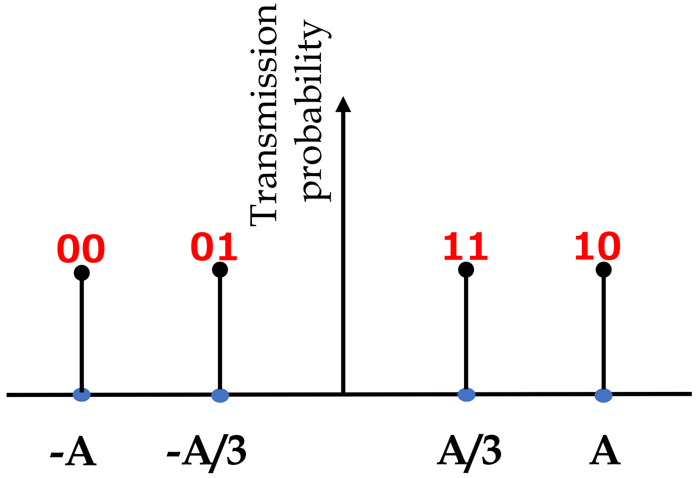
➡ Robust designs needed to be addressed

- Worst-case problem: guarantee **a certain performance** level for all possible channel realizations
- Average problem: guarantees **an average performance** over possible error realizations

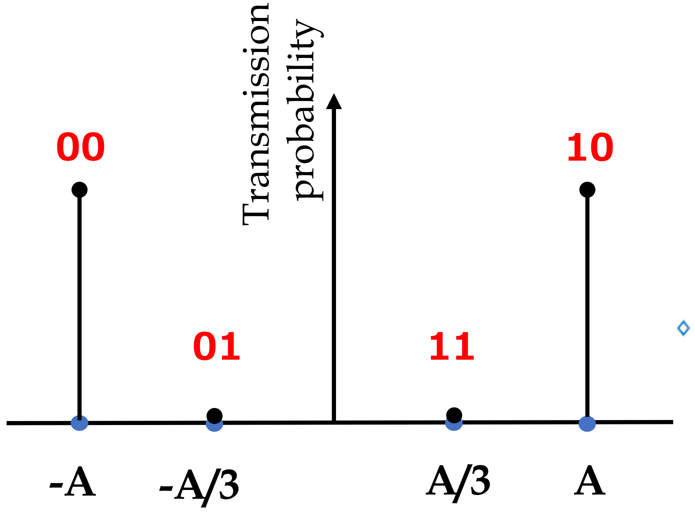
Thank you for listening!

Q & A

Probabilistic constellation shaping



Uniform signaling



Probabilistic Constellation Shaping

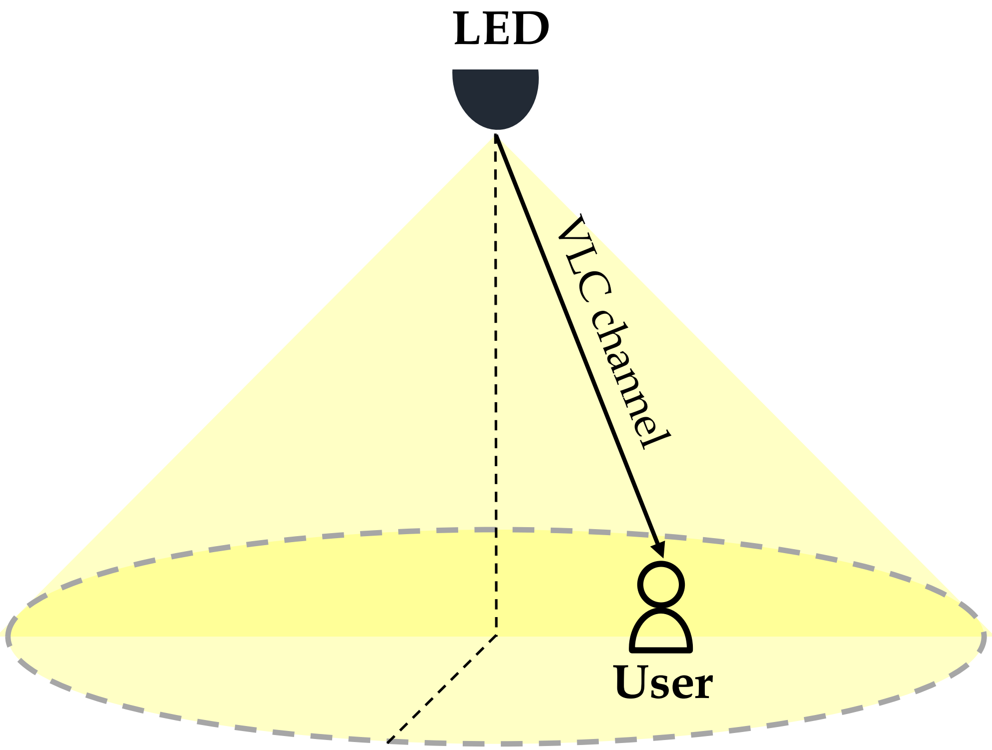
Received signal at the receiver

- Received signals when using PCS are more diverse with the advantage of signal power to noise ratio
- Entropy of received signals will increase leading to the increase of achievable rate

Received signal at the receiver

Probabilistic constellation shaping

- Let $s_1, s_2, ..., s_M$ be M bipolar M-PAM symbols generated from CCDM and constellation mapping
  - $a_m (0 < a_m < A)$, p_m are the amplitude and transmission probability of s_m



Single input single output (SISO) visible light (VLC) communication system

- Received electrical signal at user:

$$y_{\mathrm{U}} = h_{\mathrm{U}} \gamma \eta (s + I_{\mathrm{DC}}) + n_{\mathrm{U}}$$

h_{U}: line-of-sight (LoS) channel gain

γ: photodetector's responsivity

η: LED's electrical-to-optical conversion factor

I_{DC}: DC bias

$n_{\mathrm{U}} \sim \mathcal{N}(0, \sigma_{\mathrm{U}}^2)$: Gaussian reicever nosie

Removing the DC bias which carries no information

$$\overline{y}_{\mathrm{U}} = h_{\mathrm{U}} \gamma \eta s + n_{\mathrm{U}}$$