Image and video compression with deep neural networks (and other works)

Luis Herranz

Computer Vision Center
Universitat Autònoma de Barcelona

August 2023

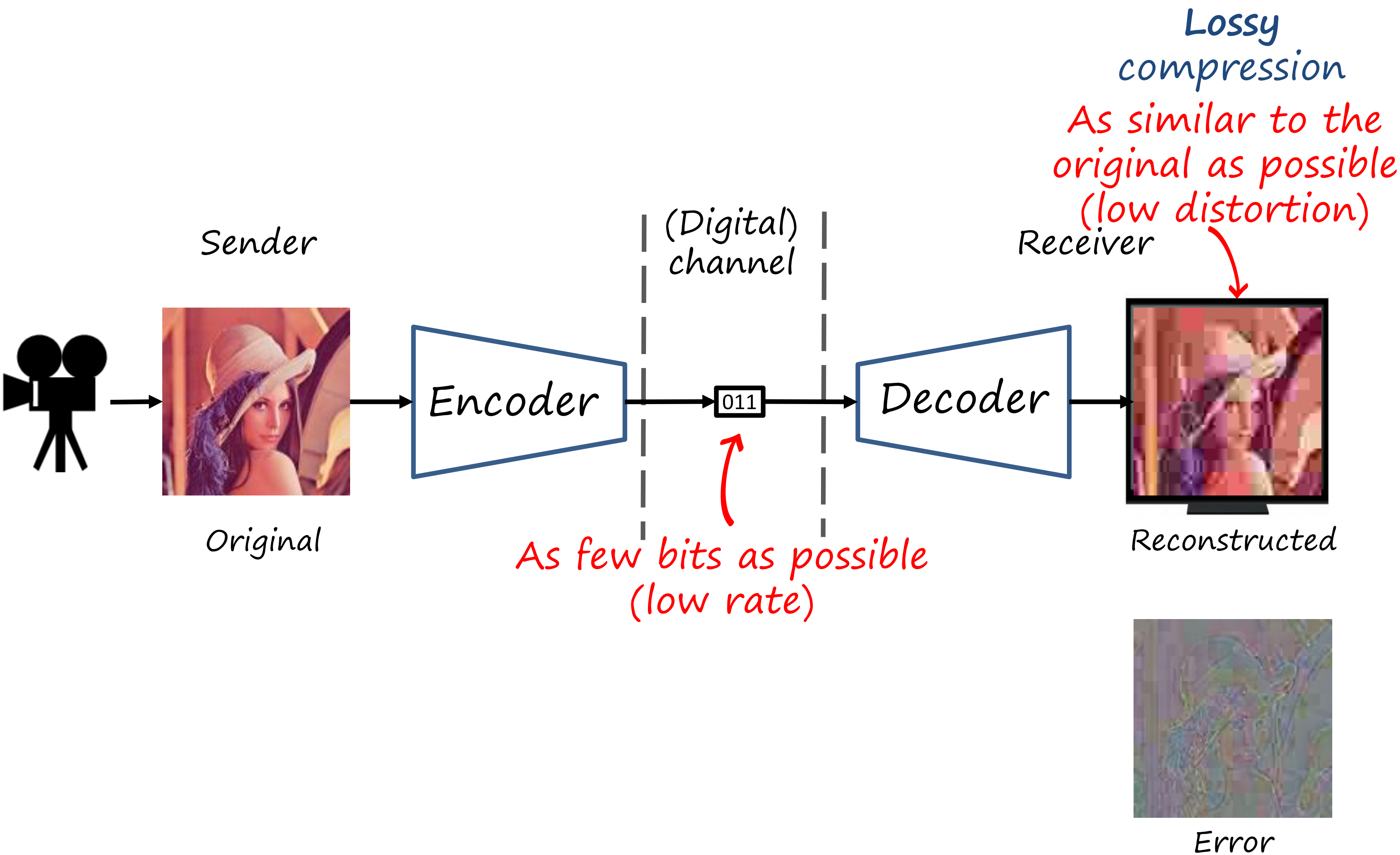
Outline

- Neural image/video compression: a walkthrough
- Our work on neural image/video compression
- Briefly: other works

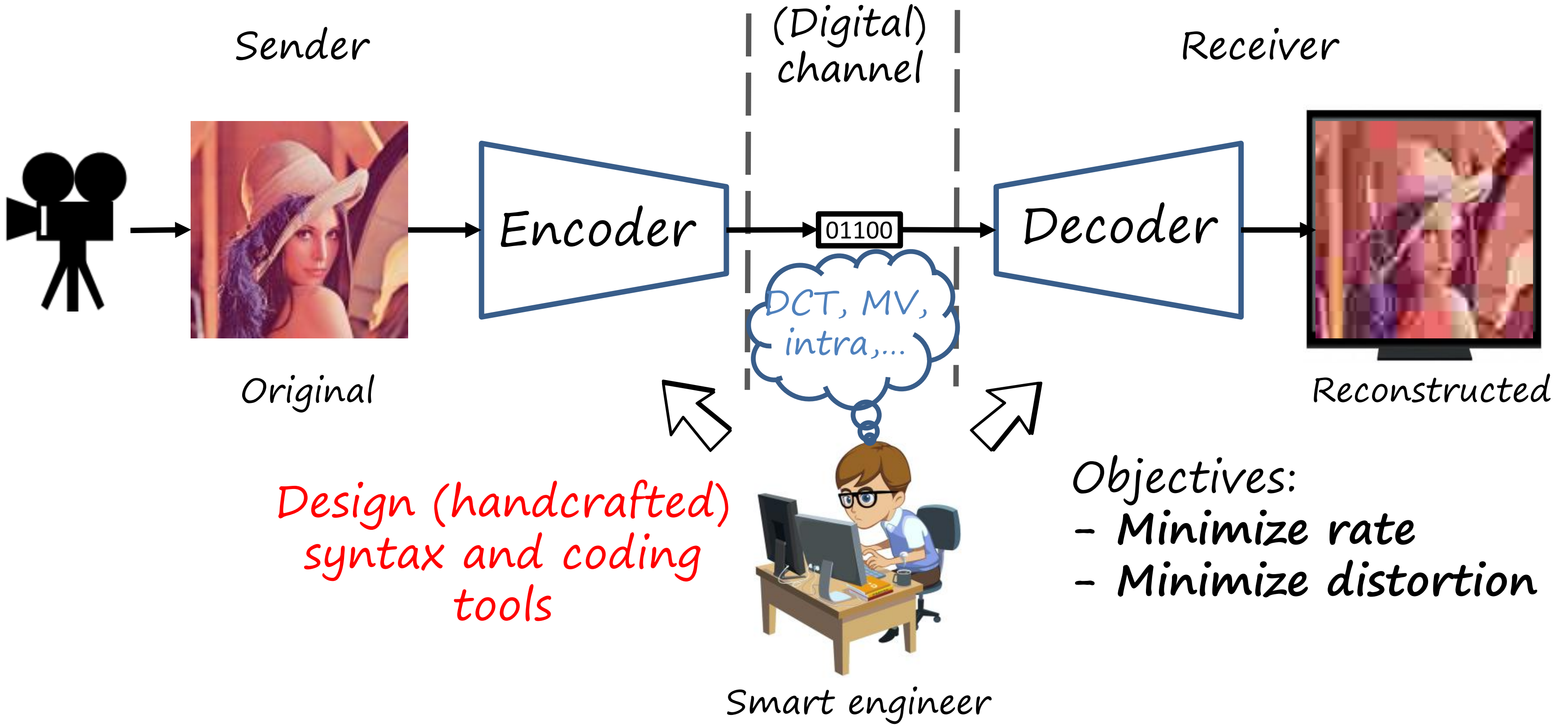
Outline

- Neural image/video compression: a walkthrough
  - Image compression
  - Video compression
- Our work on neural image/video compression
- Other works

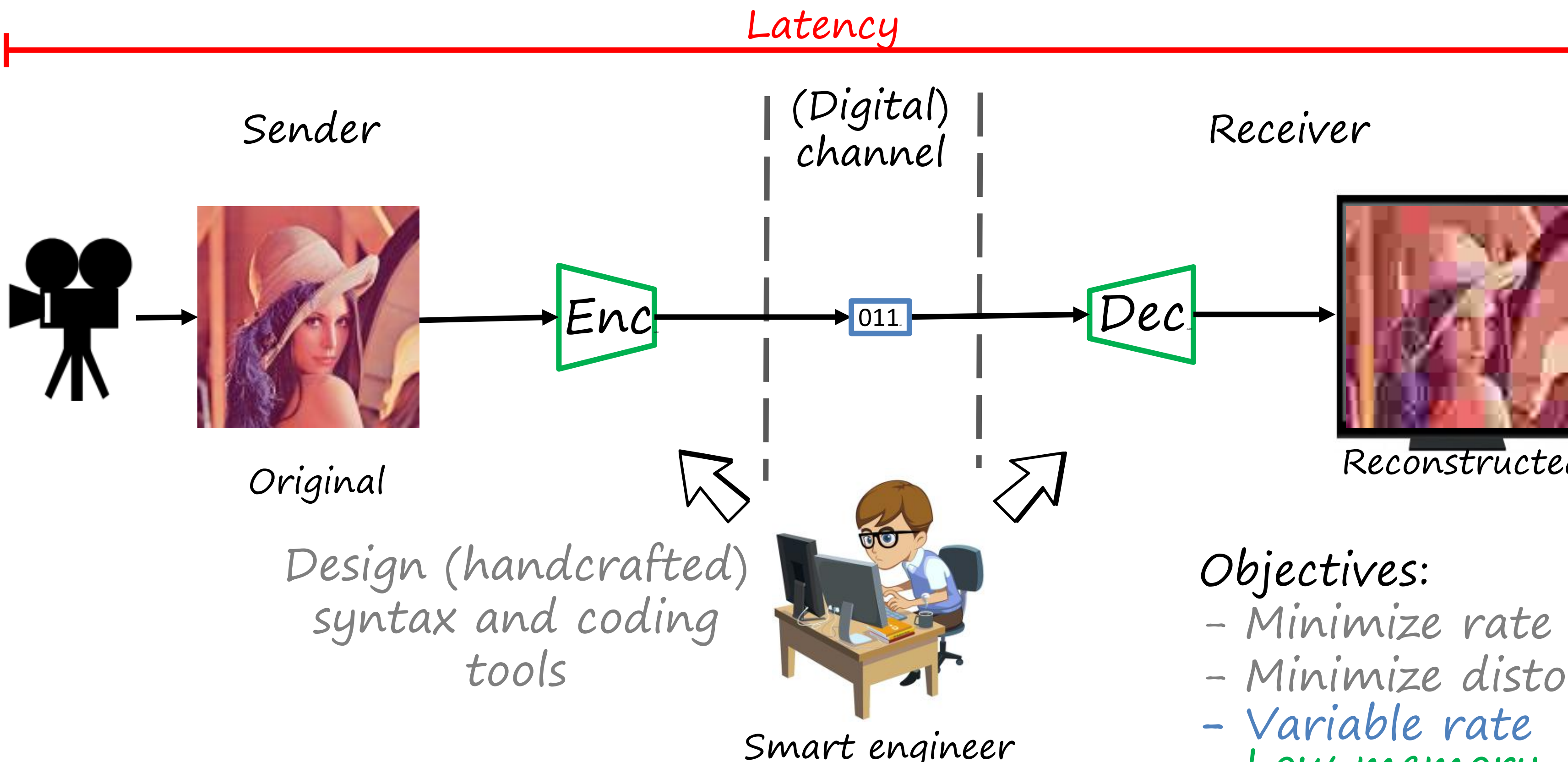
The visual communication problem



Developing traditional image/video codecs

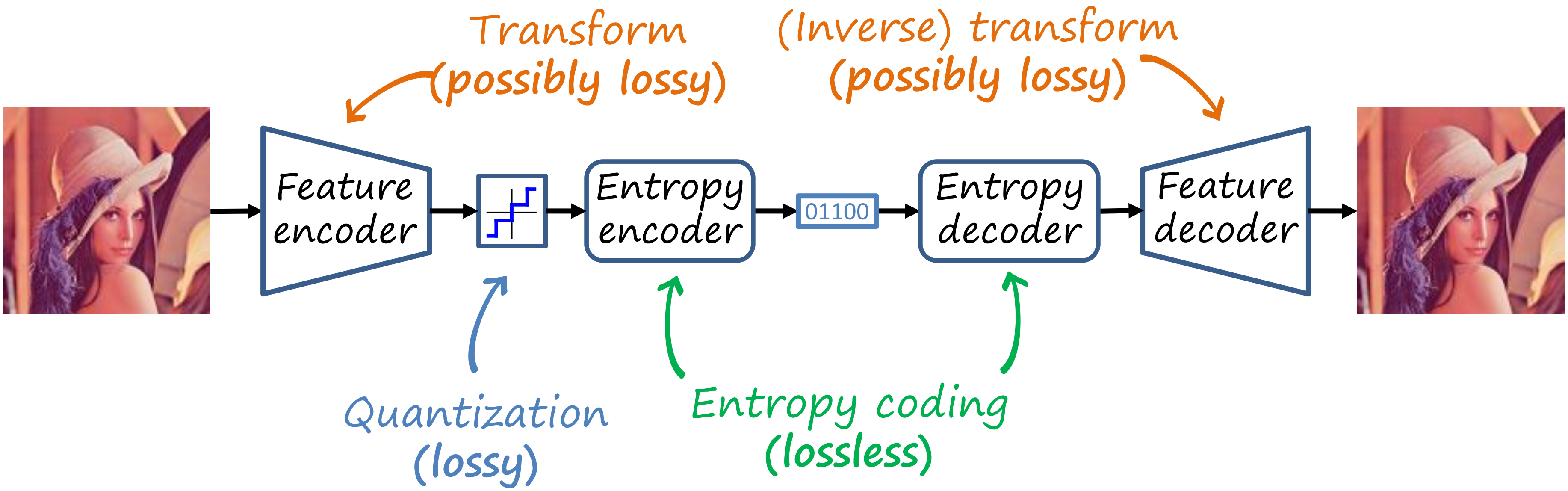


… for practical applications

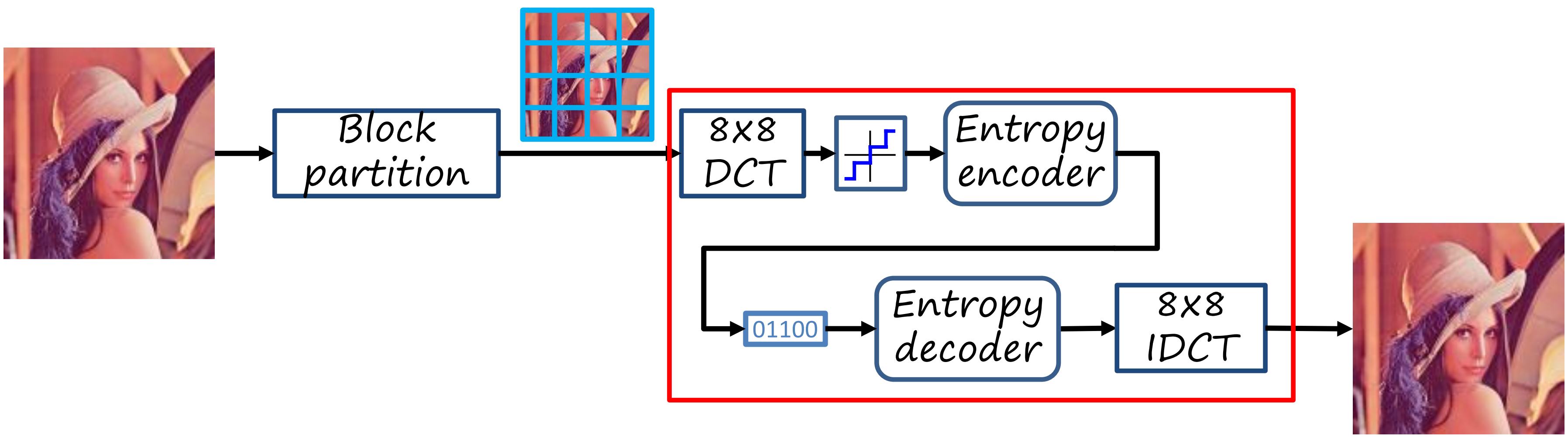


Objectives:
- Minimize rate
- Minimize distortion
- Variable rate
- Low memory
- Low computation
- Low latency
- **Compatibility**
- **Domain-specific**

Transform coding pipeline

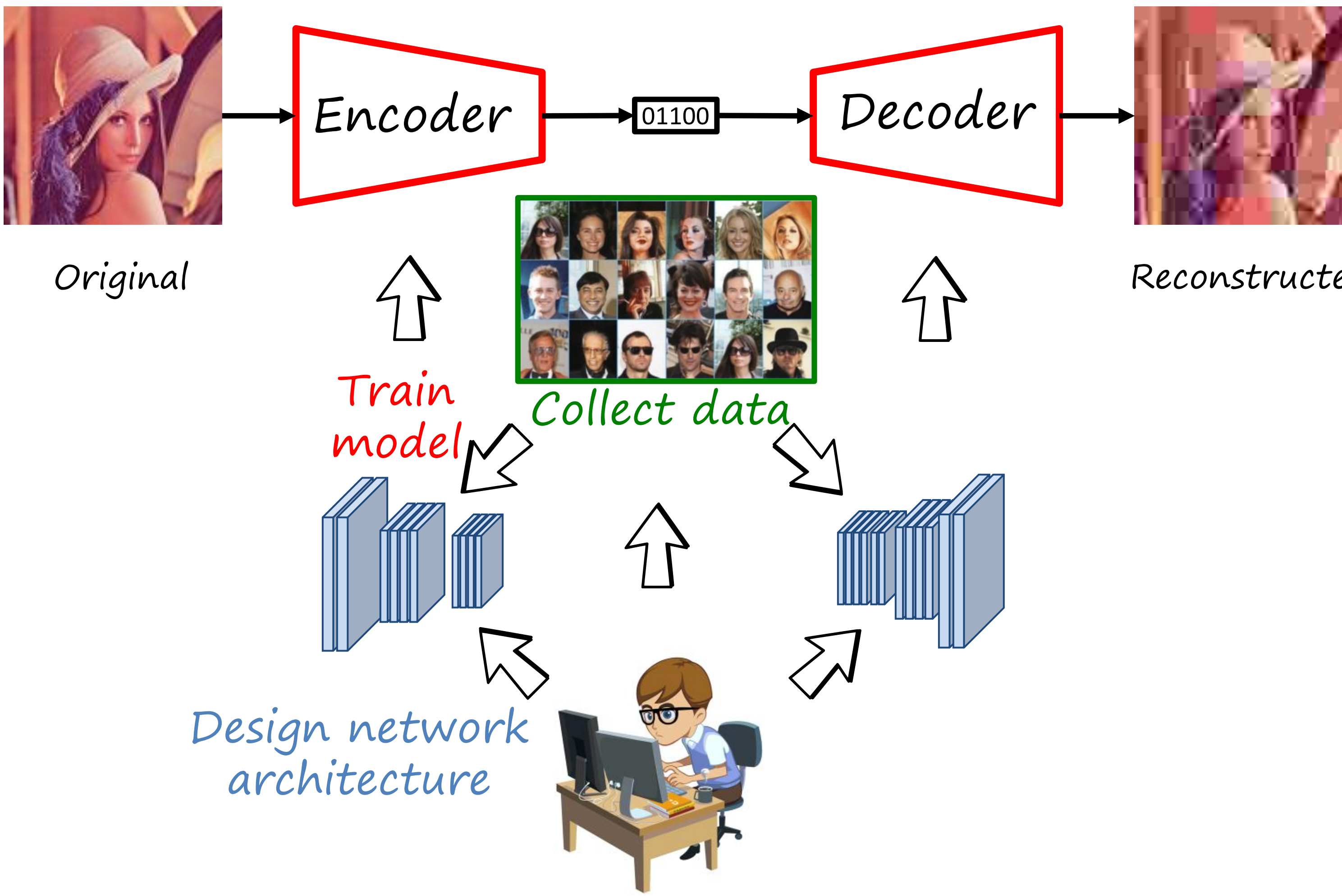


Example: block-based transform coding (e.g. JPEG, MPEG-2, H.264)



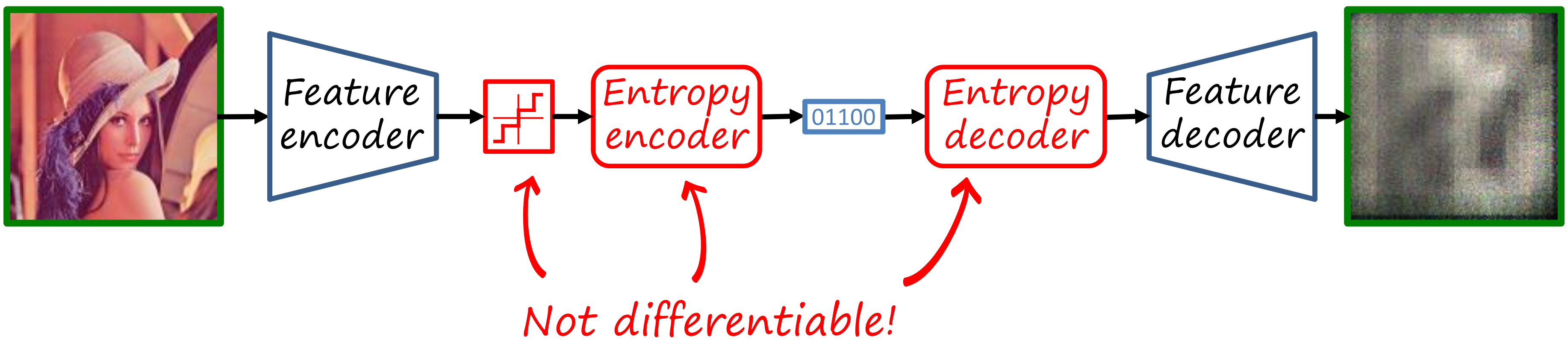
Neural image codecs

- Coding tools and syntax are **parametric** and **learned**
- Encoders/decoders and probability models are **deep neural networks**



Typical pipeline

Compressive autoencoder (CAE) [Theis2017, Balle2017]
(autoencoder+quantization+entropy coding)

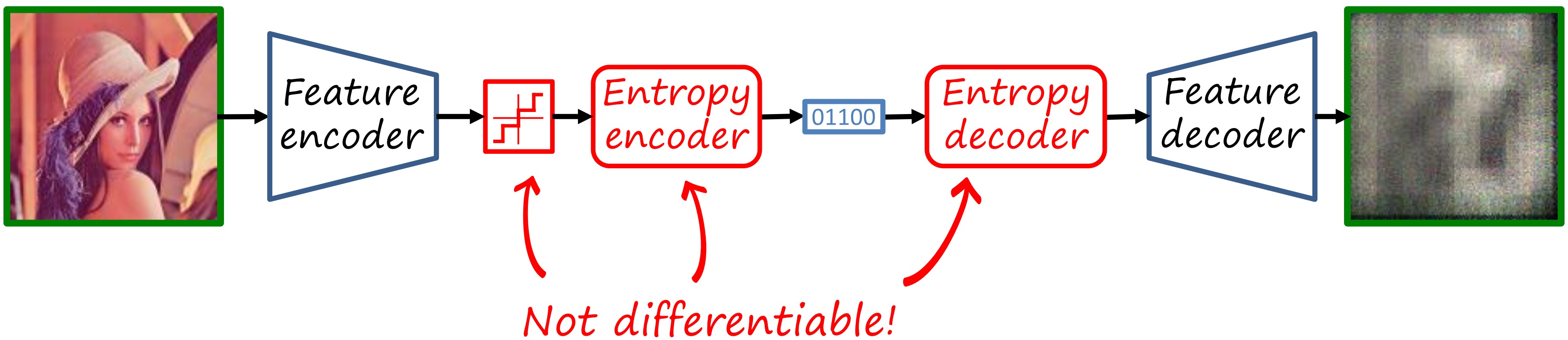


[Balle et al. End-to-end Optimized Image Compression,](#) ICLR 2017
[Theis et al., Lossy Image Compression with Compressive Autoencoders,](#) ICLR 2017

Typical pipeline

Observation 1:
- Entropy coding is reversible: bypass it
- Entropy is a tight lower bound to the rate: use as approximation

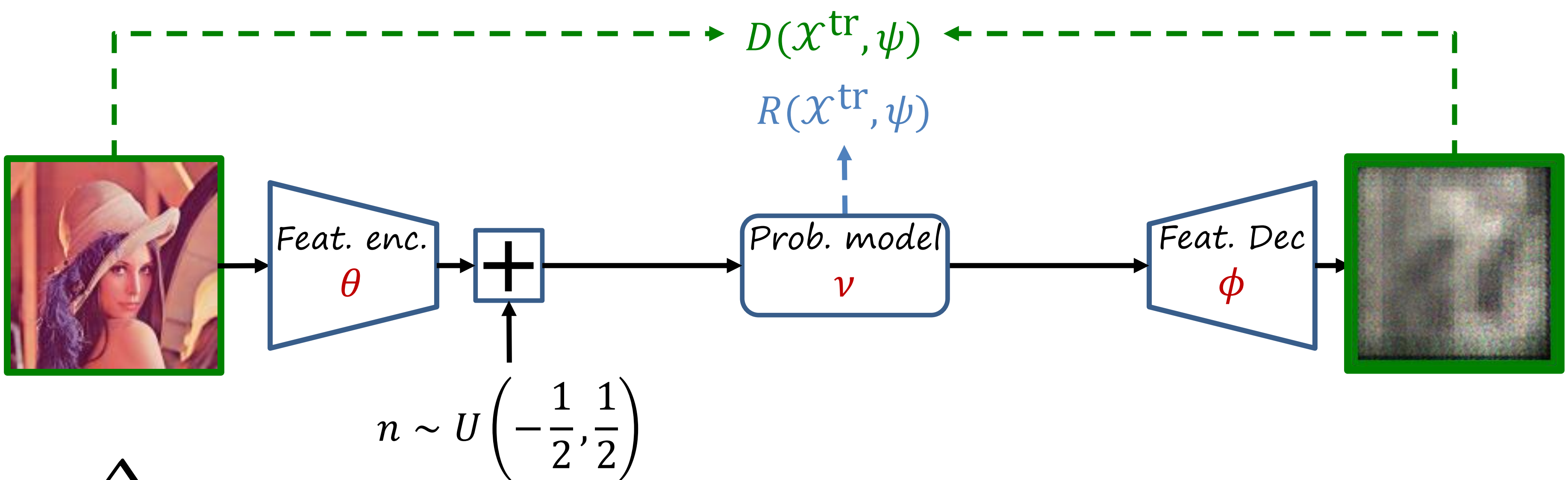
Observation 2:
- Quantization (i.e. rounding) introduces a uniform error

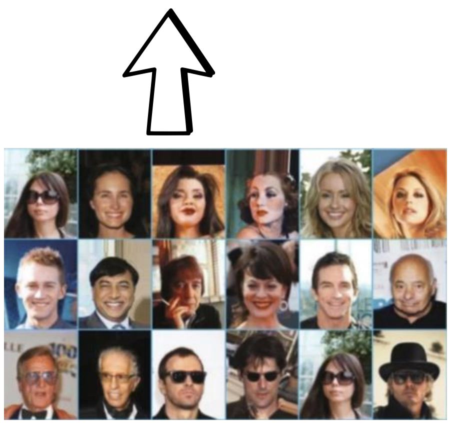


[Balle et al. End-to-end Optimized Image Compression,](#) ICLR 2017
[Theis et al., Lossy Image Compression with Compressive Autoencoders,](#) ICLR 2017

Architecture (training)

Use differentiable proxies for end-to-end training



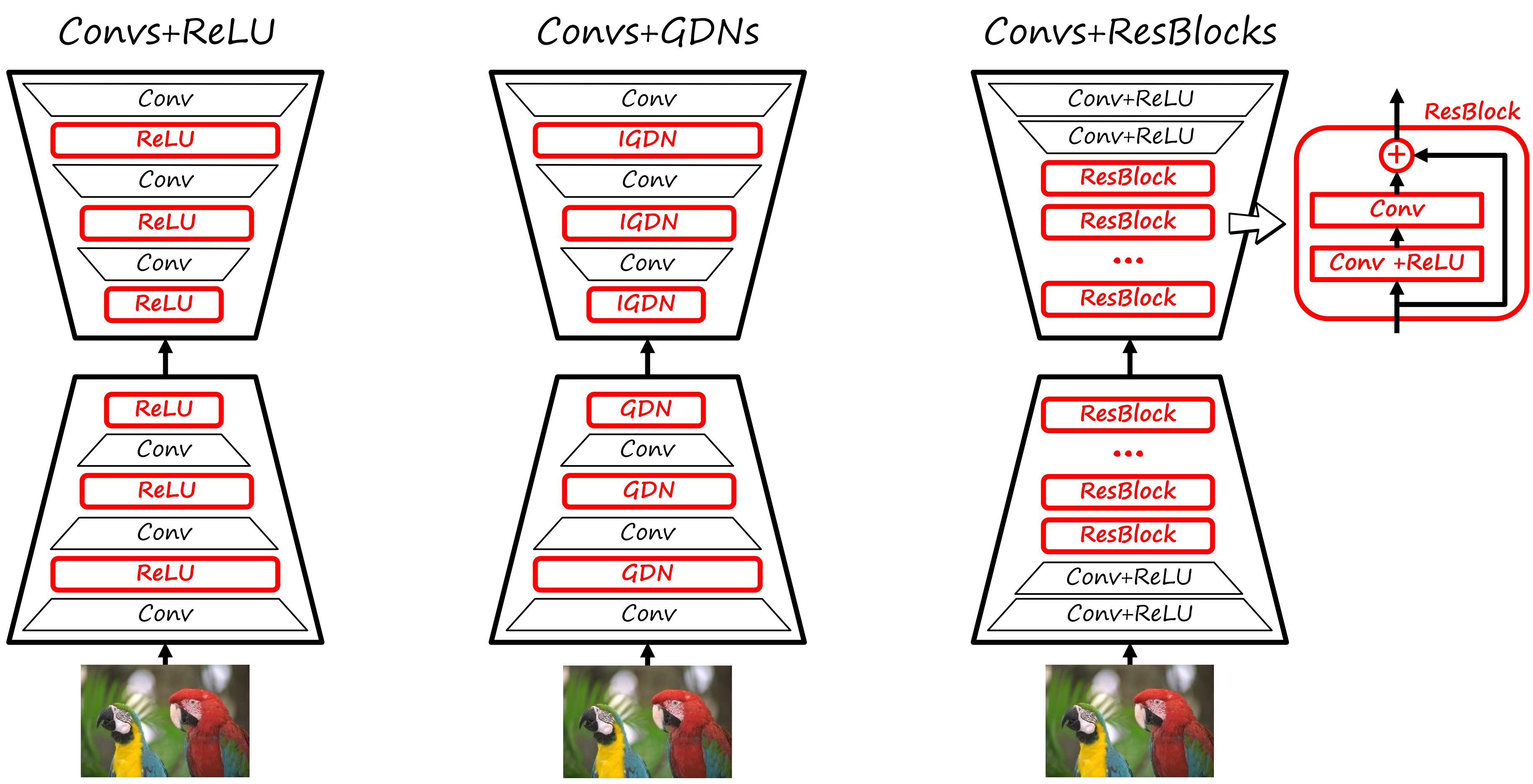


Training data $\mathcal{X}^{\mathrm{tr}}$

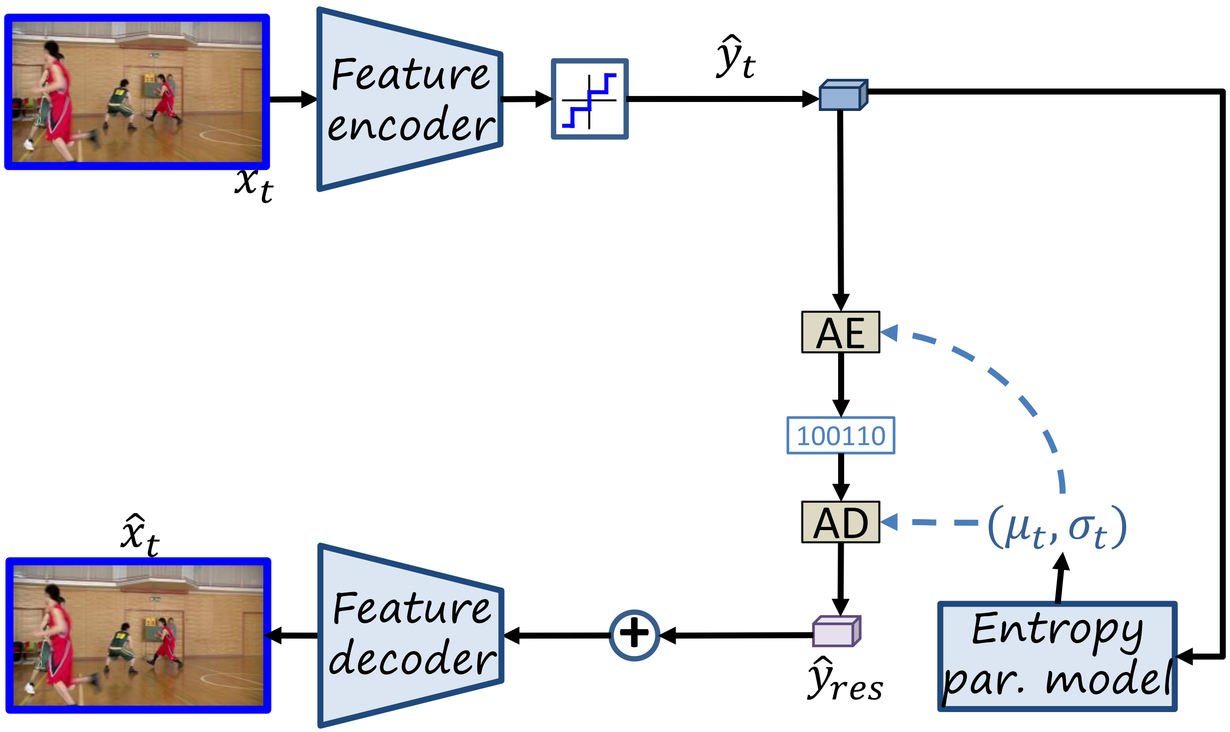
Model parameters	$\psi = (\theta, \phi, \nu)$
Loss	$J(\mathcal{X}^{\mathrm{tr}}, \psi; \lambda) = R(\mathcal{X}^{\mathrm{tr}}, \psi) + \lambda D(\mathcal{X}^{\mathrm{tr}}, \psi)$
Optimization problem	$\psi^* = \min_{\psi} J(\mathcal{X}^{\mathrm{tr}}, \psi; \lambda)$

[Balle et al. End-to-end Optimized Image Compression,](#) ICLR 2017
[Theis et al., Lossy Image Compression with Compressive Autoencoders,](#) ICLR 2017

Autoencoder architecture

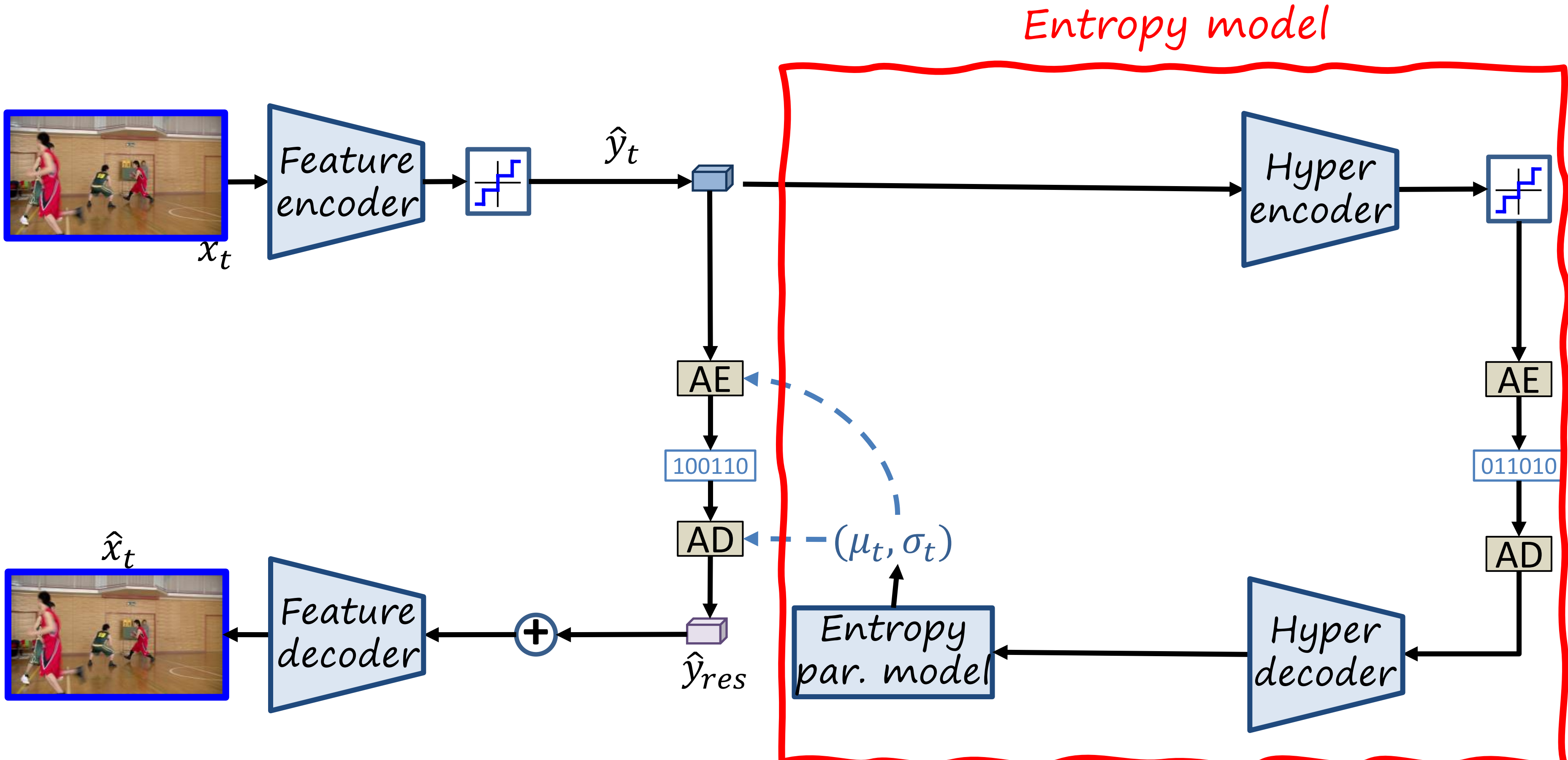


Simple entropy model



More complex entropy models

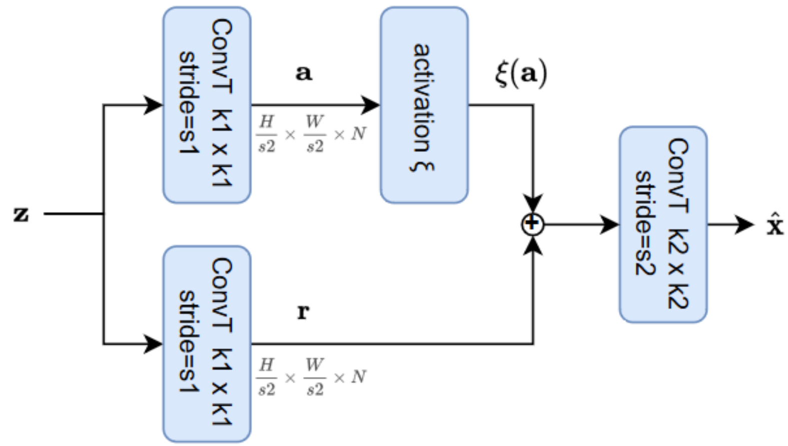
E.g. hyperprior [Balle 2018]

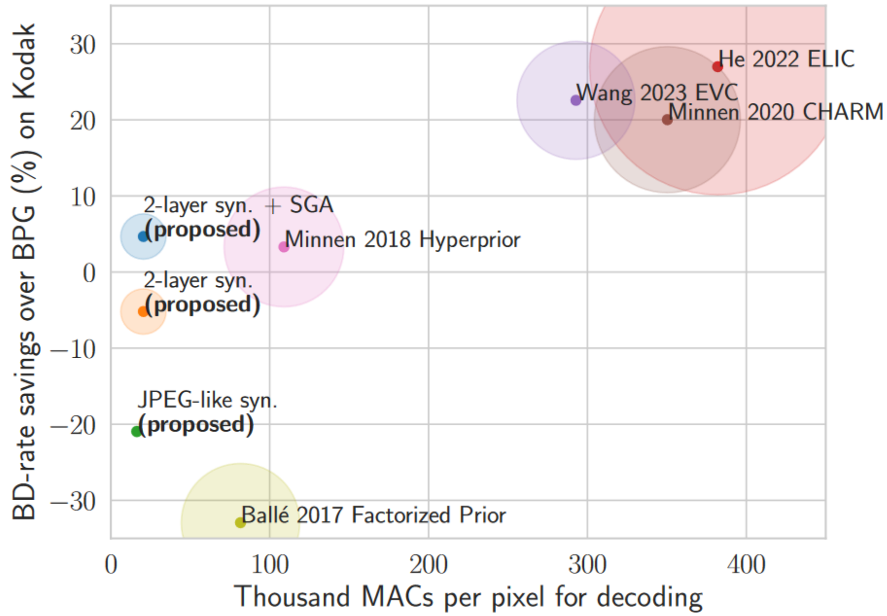


Balle et al. Variational image compression with a scale hyperprior ICLR 2018

Reducing decoding cost: shallow decoders

E.g. 2 layer-decoder [Yang2023]



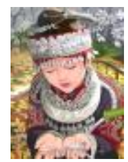


Method	Computational complexity (KMAC)						Syn. param count (Mil.)	BD rate savings (%) ↑
	f	f_h	enc. tot.	g	g_h	dec. tot.		
He 2022 ELIC [20]	255.42	6.73	262.15	255.42	126.57	381.99	7.34	26.98
Minnen 2020 CHARM [30]	93.79	5.90	99.70	93.79	256.51	350.30	4.18	20.02
Wang 2023 EVC [41]	263.25	1.86	265.11	257.94	34.82	292.76	3.38	22.56
Minnen 2018 Hyperprior [29]	93.79	6.73	100.52	93.79	15.18	108.97	3.43	3.30
Ballé 2017 Factorized Prior [2]	81.63	0.00	81.63	81.63	0.00	81.63	3.39	-32.93
2-layer syn. + SGA (proposed)	255.42	6.73	262.15	**5.34**	15.18	20.52	1.30	4.67
2-layer syn. (proposed)	255.42	6.73	262.15	**5.34**	15.18	20.52	1.30	-5.19
JPEG-like syn. (proposed)	255.42	6.73	262.15	**1.22**	15.18	16.39	0.31	-20.95

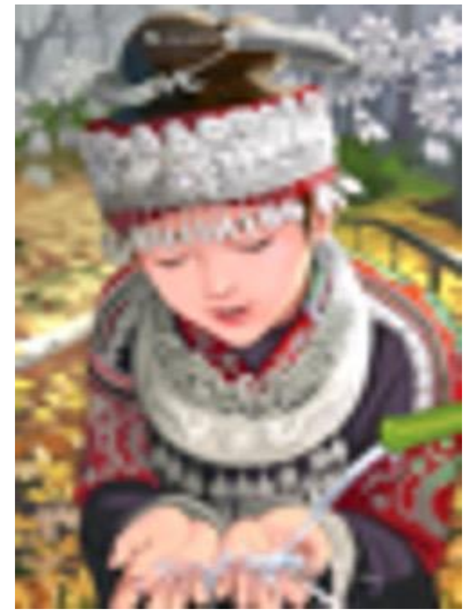
Yang and Mandt, Asymmetrically-powered Neural Image Compression with Shallow Decoders ICCV 2023

Perception vs distortion

Downsampling
(25%)



Upsampling
(bicubic 4x)



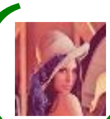
Note: lossy
(lost information
can't be recovered)

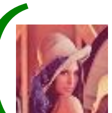
Perception vs distortion

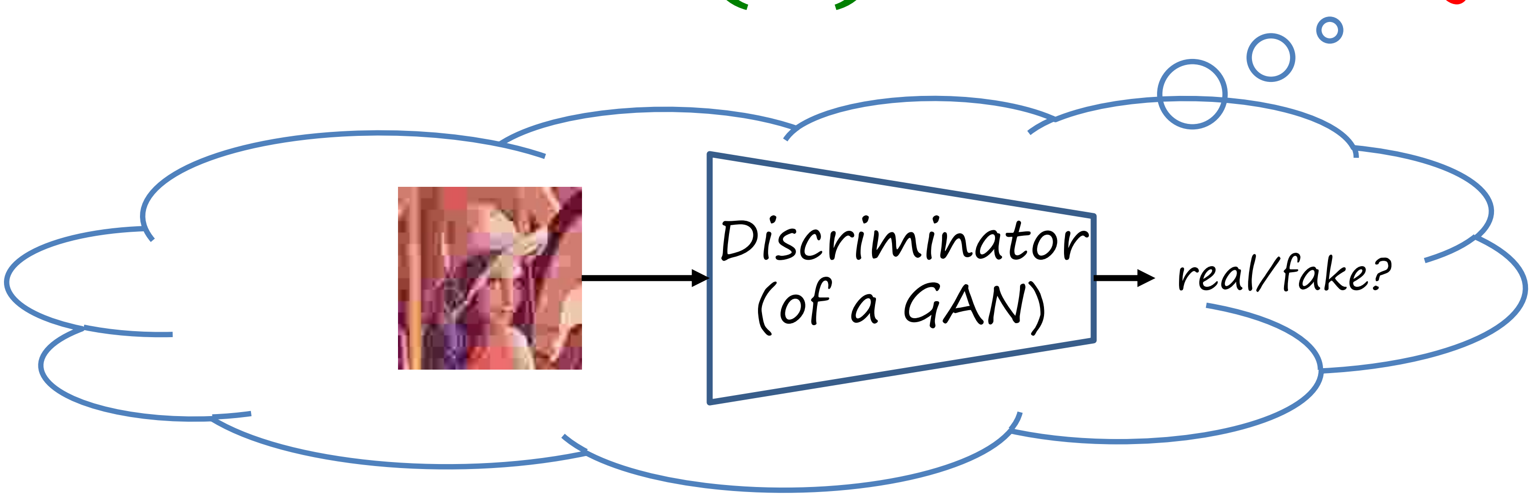
Is (MSE/PSNR) distortion a good quality metric?

Bicubic	SRResNet (MSE)	SRGAN	Original
PNSR 21.59 dB	PNSR 23.53 dB	PNSR 21.15 dB	

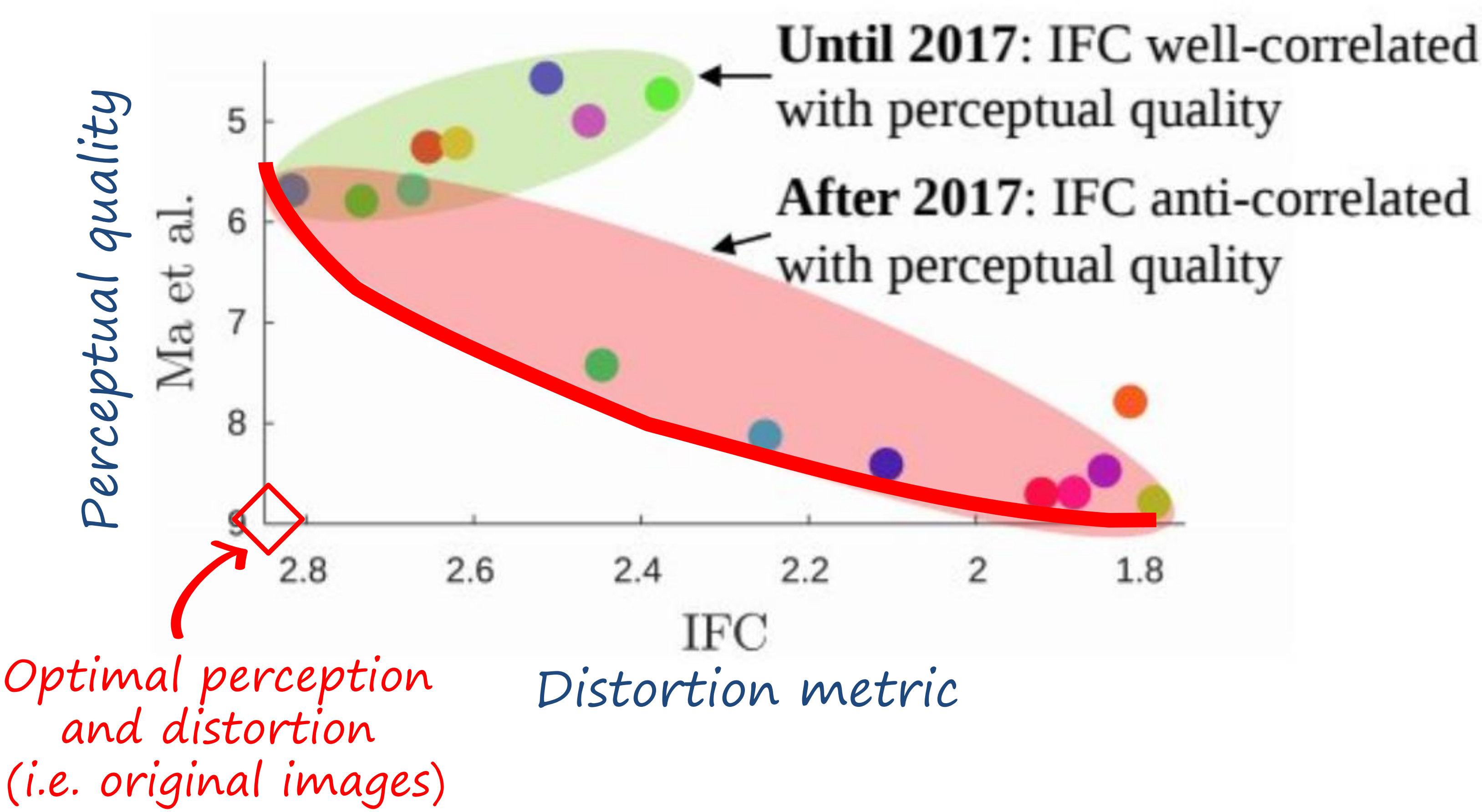
Perception vs distortion

Distortion metric (full-reference) $D($  ,  $)$ How close is the image to the original one?

Perceptual metric (no-reference) $P($  $)$ How realistic is the image?

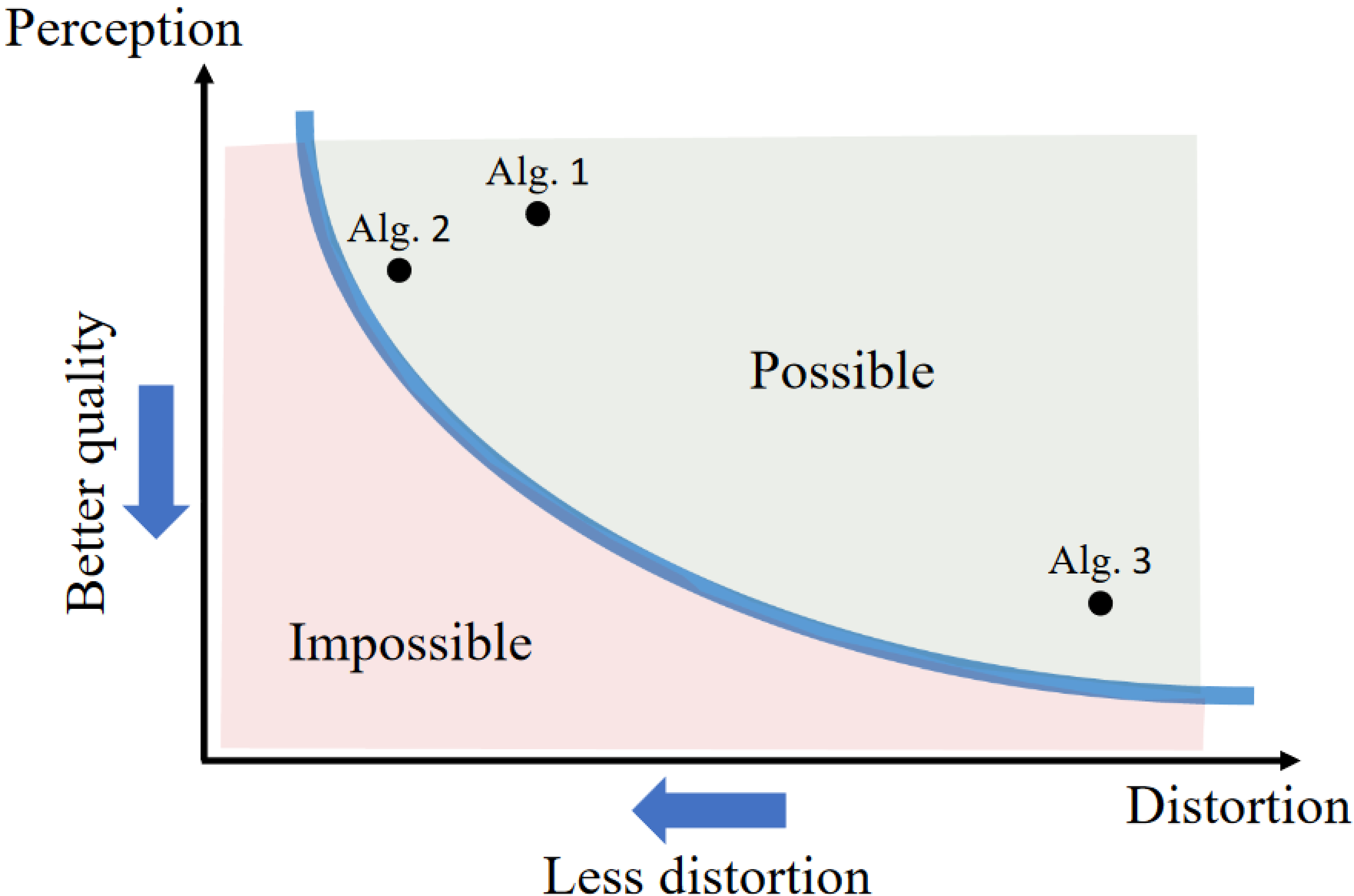


Perception-distortion in image superresolution methods



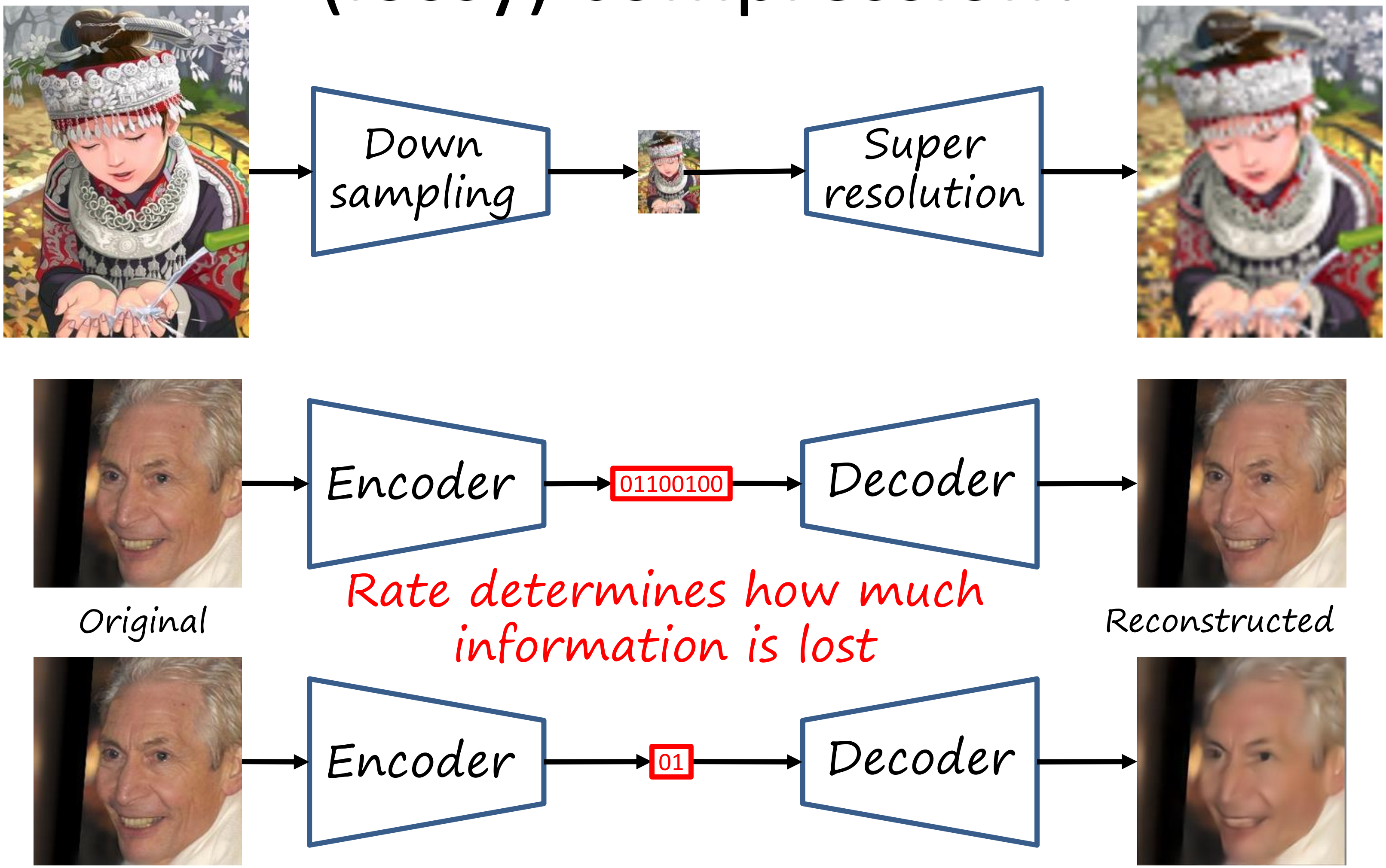
Slide adapted from Y. Blau

Perception-distortion tradeoff

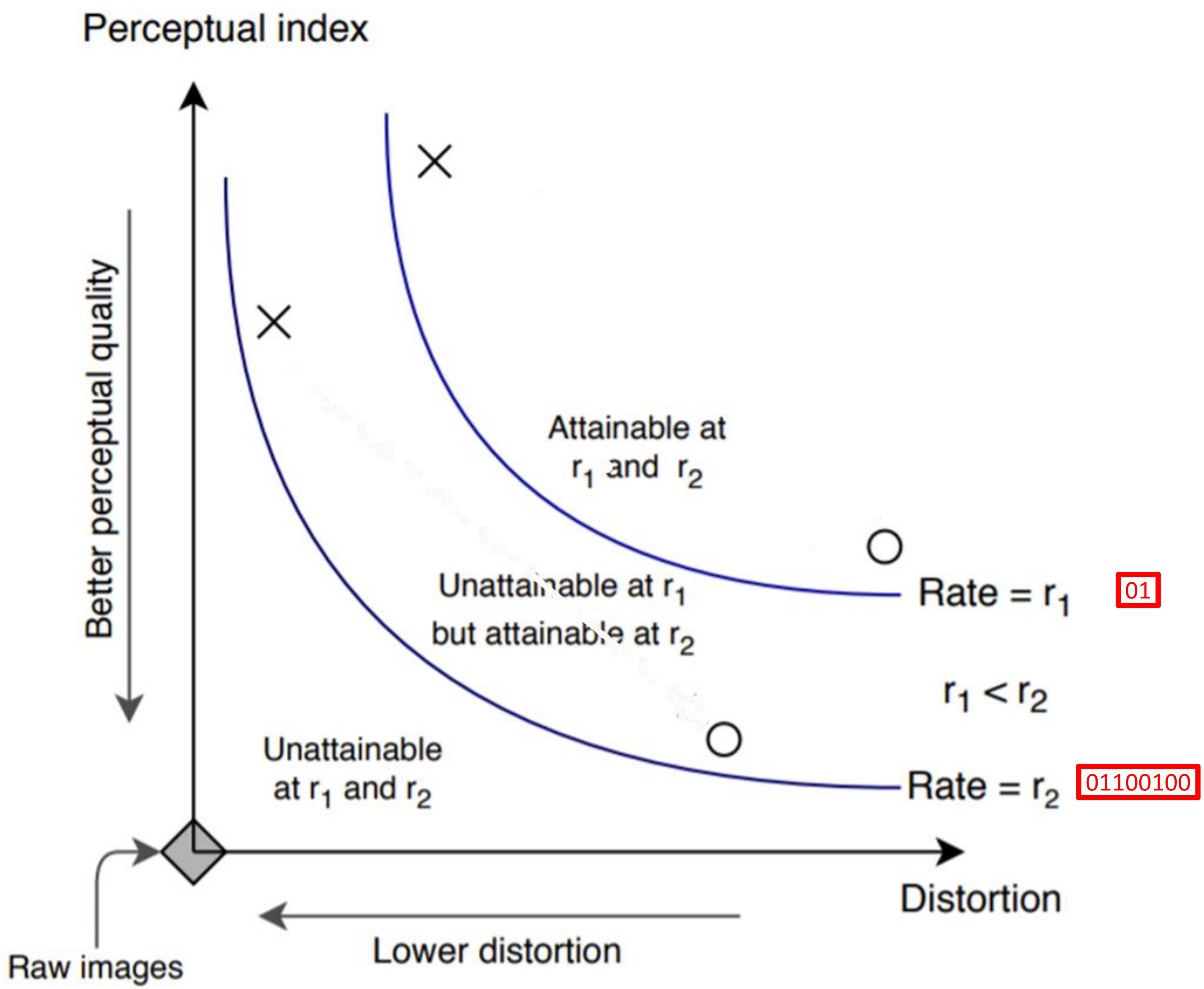


The Perception-Distortion Tradeoff, CVPR 2018

Perception vs distortion in (lossy) compression?



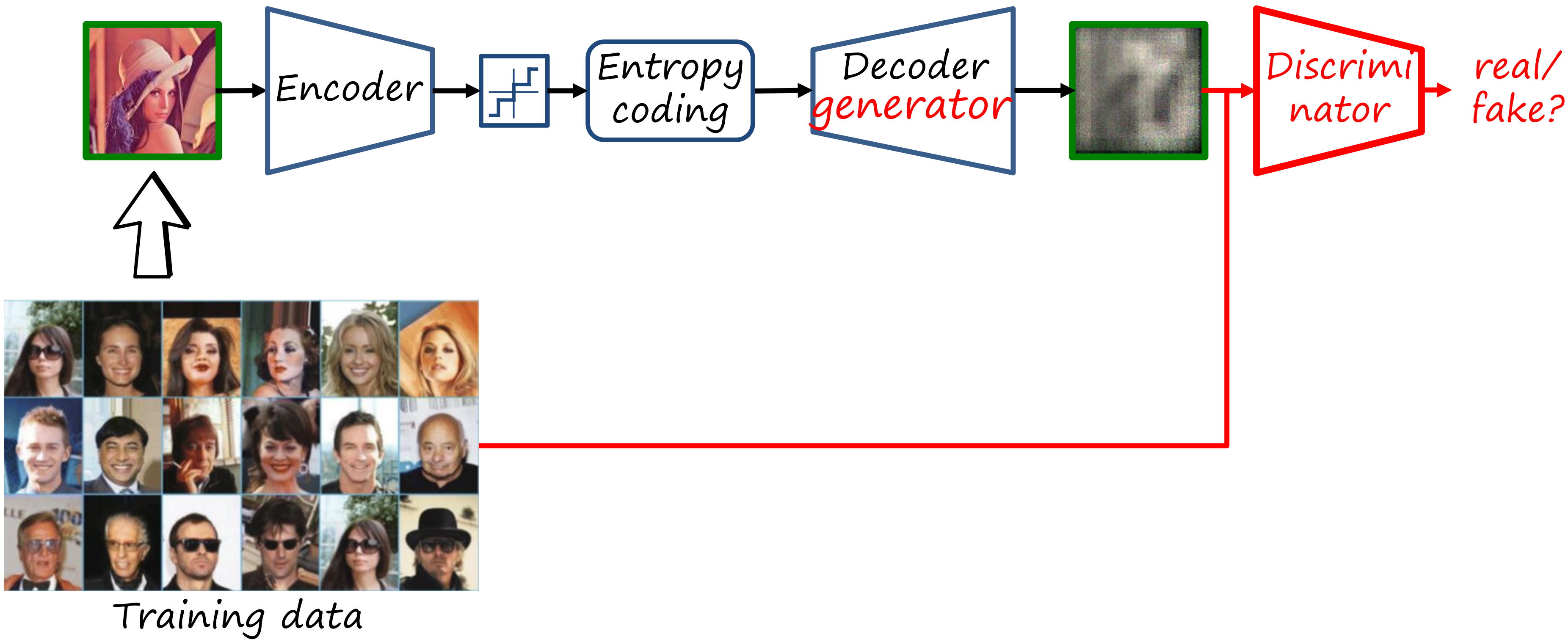
Rate-distortion-perception tradeoff



[Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff](), ICML 2019

Optimizing for perception: generative lossy compression

Optimize perception using a discriminator and adversarial loss
The decoder acts as generator of a conditional GAN



High-Fidelity Generative Image Compression, NeurIPS 2020

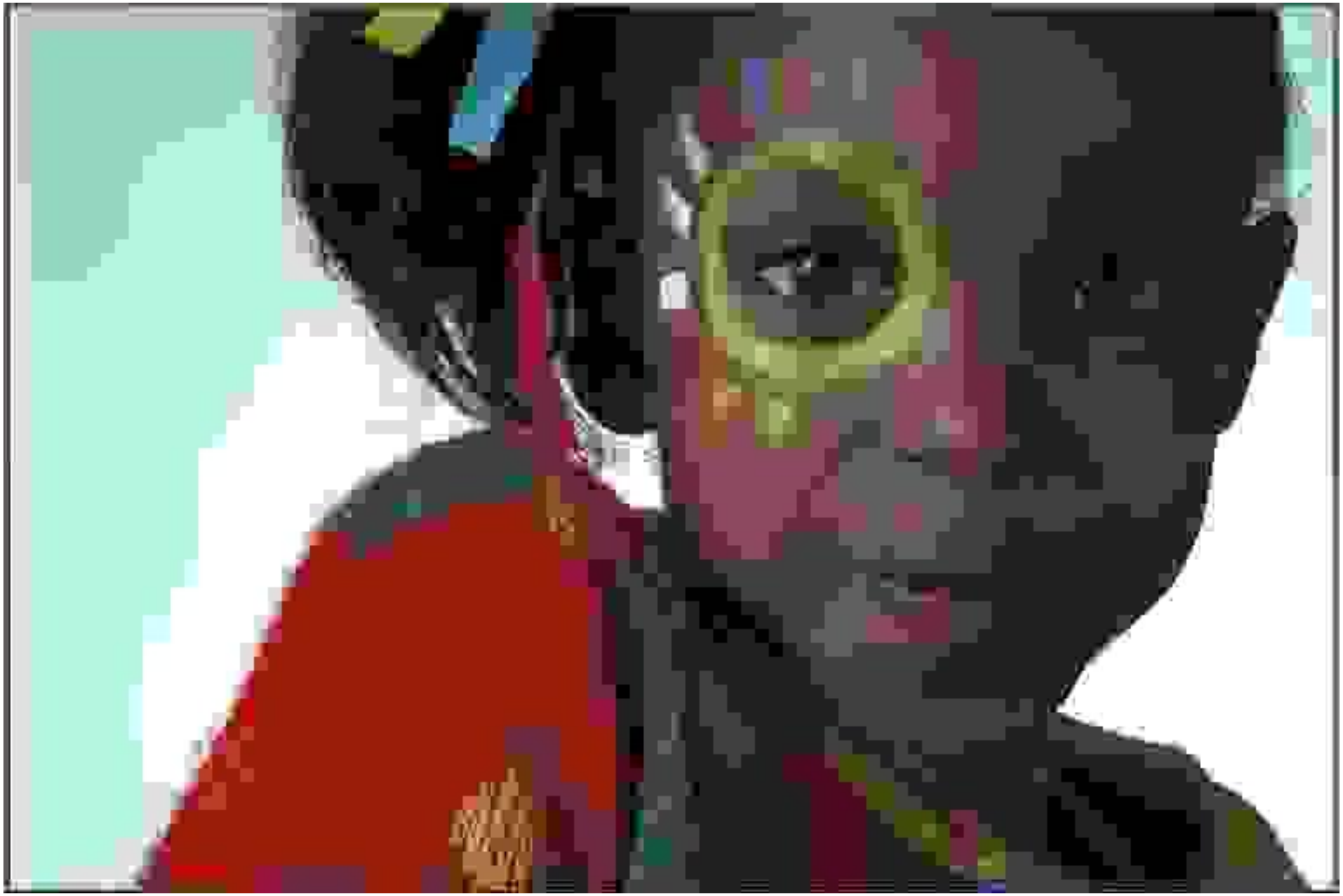
Generative lossy compression

Original (768x512 pixels – 1.18 MB)

High-Fidelity Generative Image Compression, NeurIPS 2020

Generative lossy compression

JPEG (8 kB)



[High-Fidelity Generative Image Compression](), NeurIPS 2020

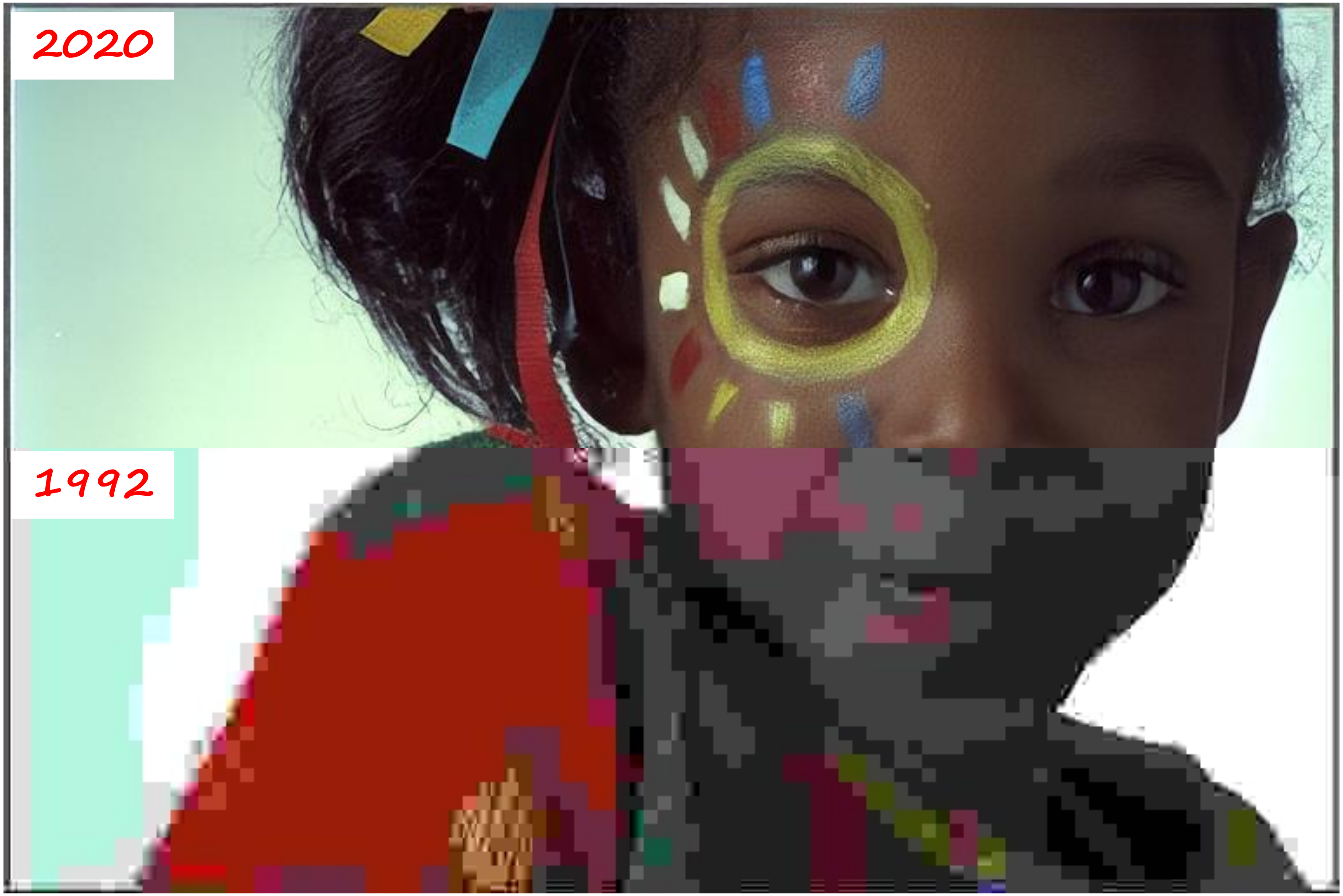
Generative lossy compression

HiFiC (7 kB)

High-Fidelity Generative Image Compression, NeurIPS 2020

Generative lossy compression

HiFiC
(7 kB)

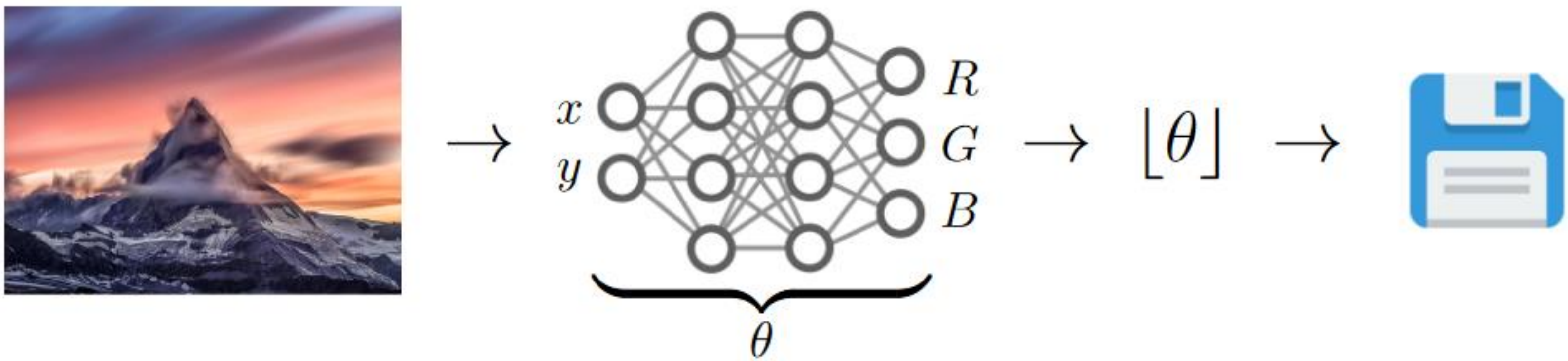


JPEG
(8 kB)

High-Fidelity Generative Image Compression, NeurIPS 2020

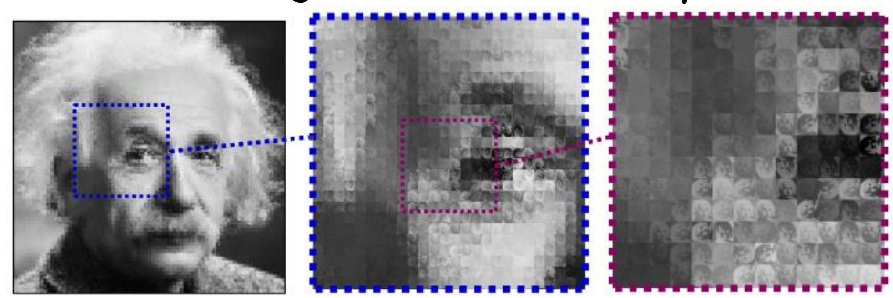
Other learned-based image compression approaches

Implicit representations (e.g. COIN, COIN++)



Diffusion models

Neural collages (fractal compression)

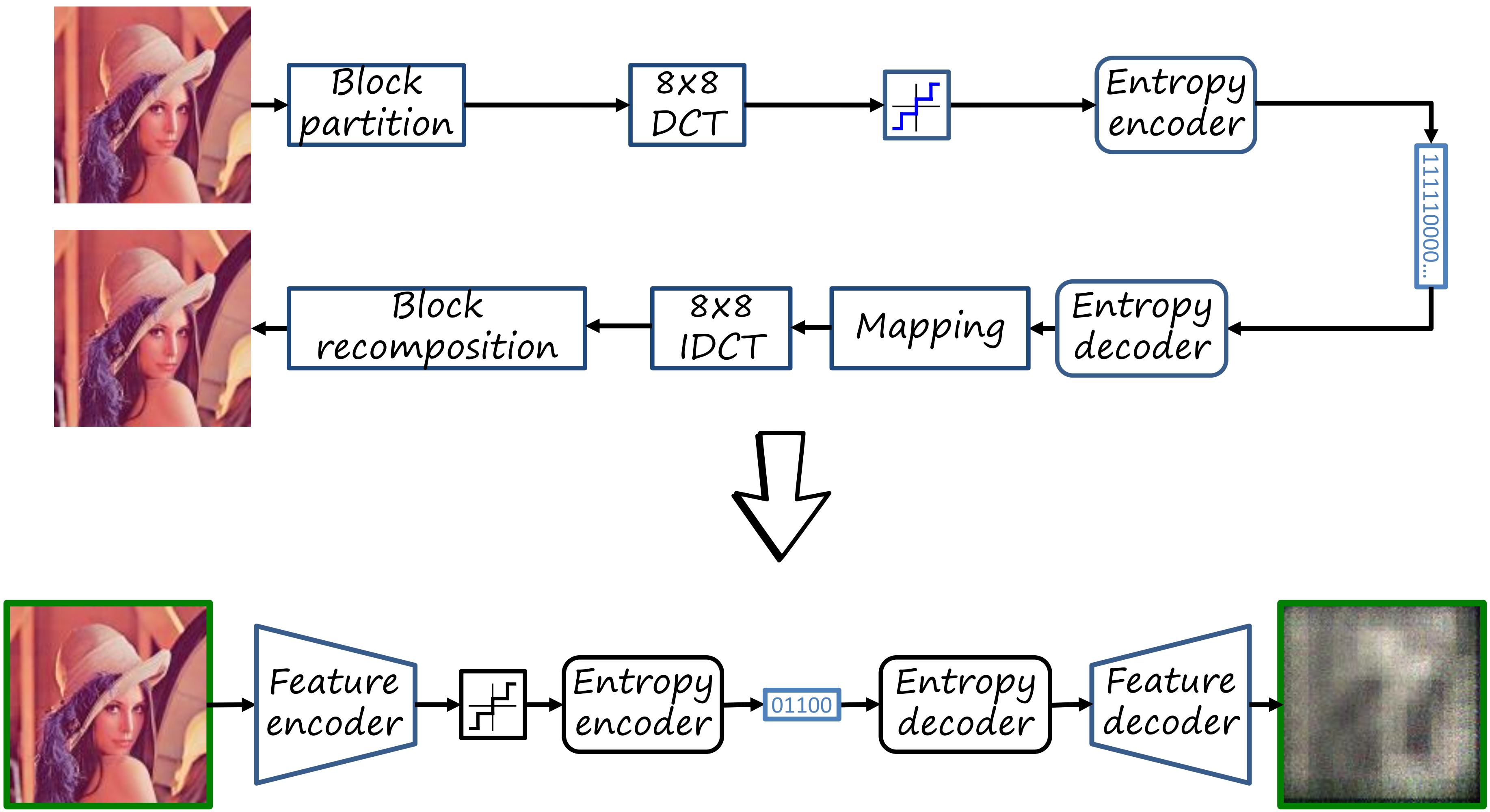


[Dupont et al. COIN: COmpression with Implicit Neural representations](#) arxiv 2021
[Dupont et al. COIN++: Neural Compression Across Modalities](#) TMLR 2022
[Yang et al. Lossy Image Compression with Conditional Diffusion Models](#) arxiv 2022
[Poli et al., Self-Similarity Priors: Neural Collages as Differentiable Fractal Representations](#) NeurIPS 2022

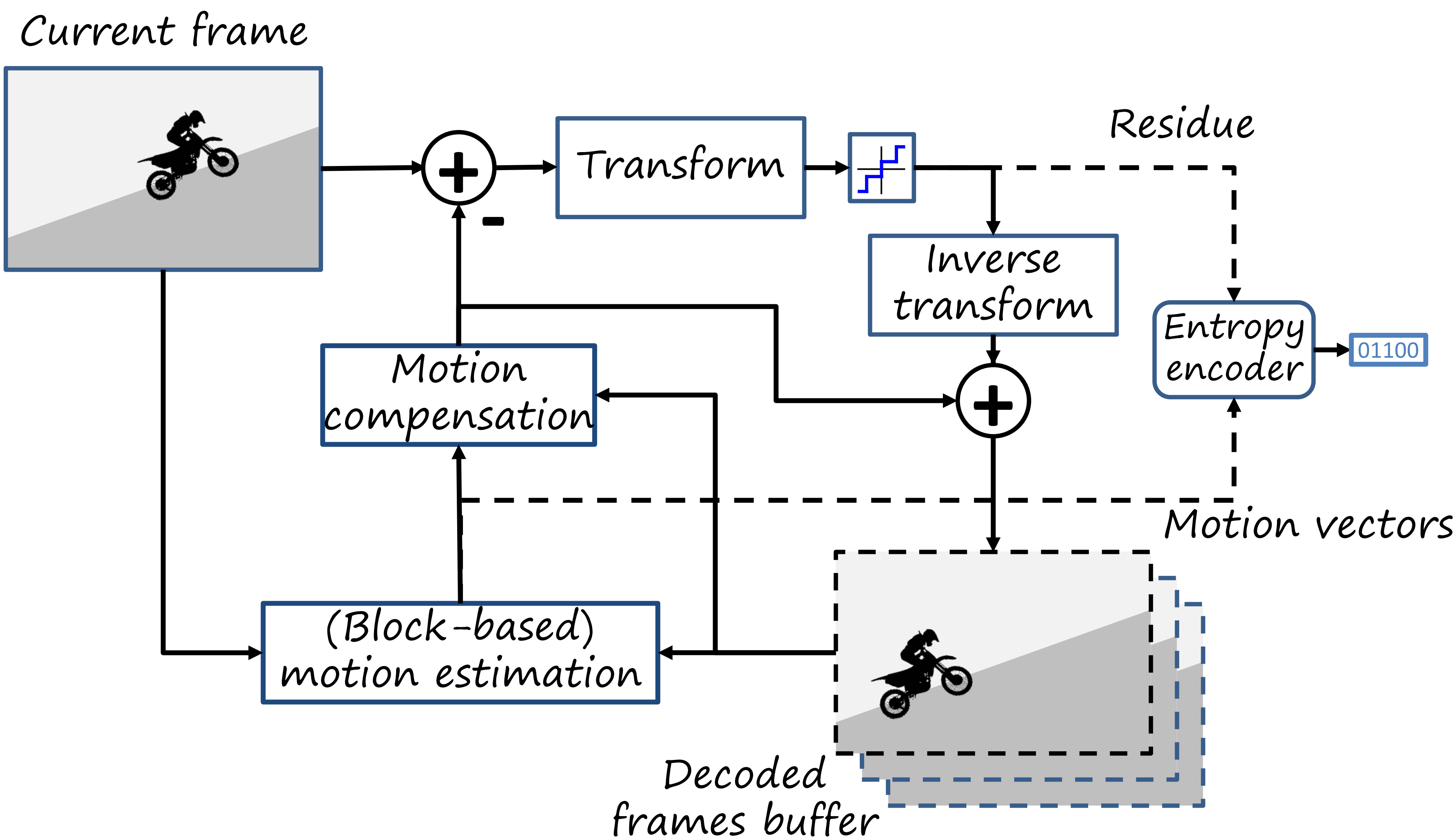
Outline

- Neural image/video compression: a walkthrough
  - Image compression
  - Video compression
- Our work on neural image/video compression
- Other works

From transform coding to neural image coding

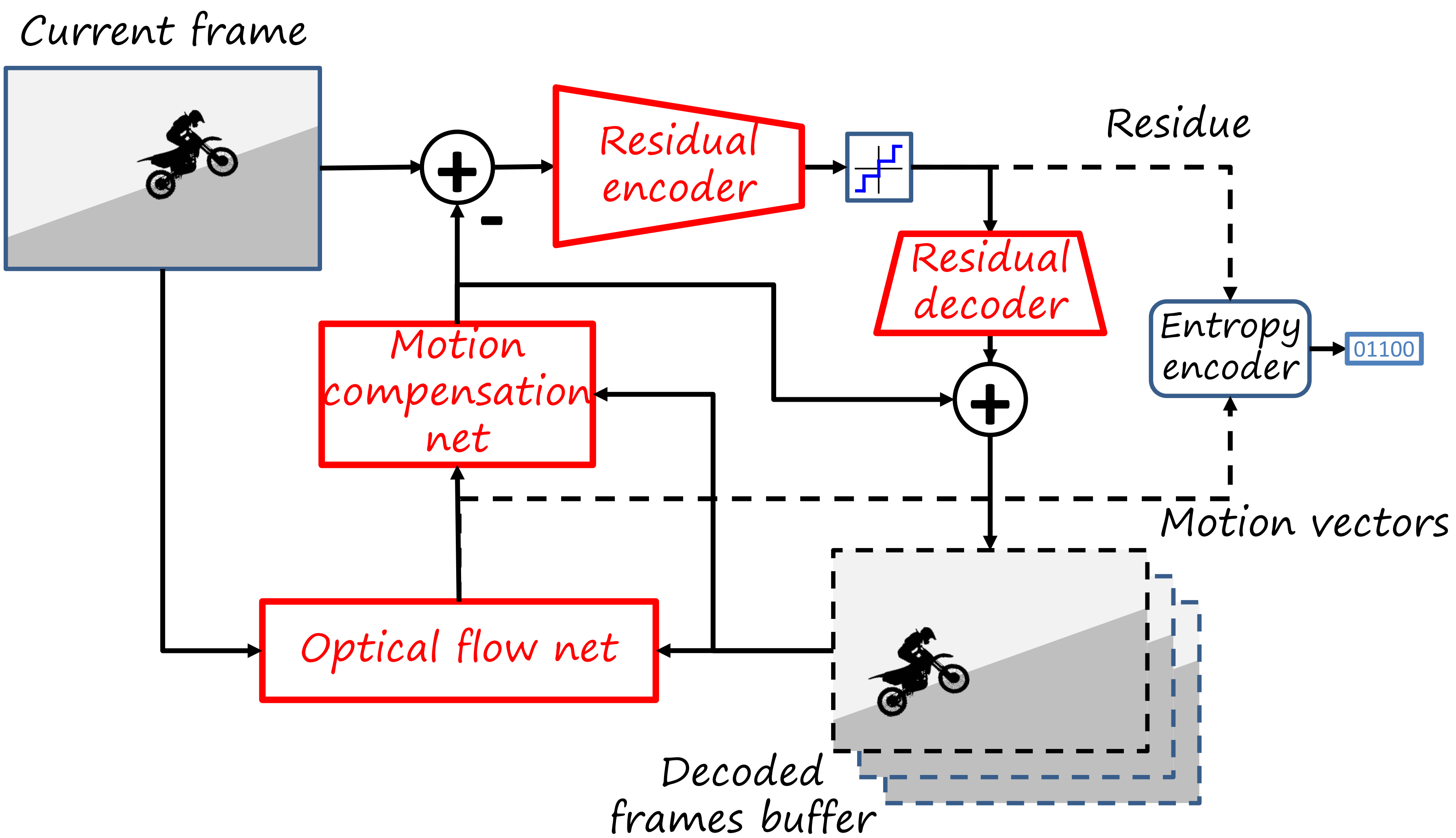


Traditional video compression (motion-compensated transform coding)



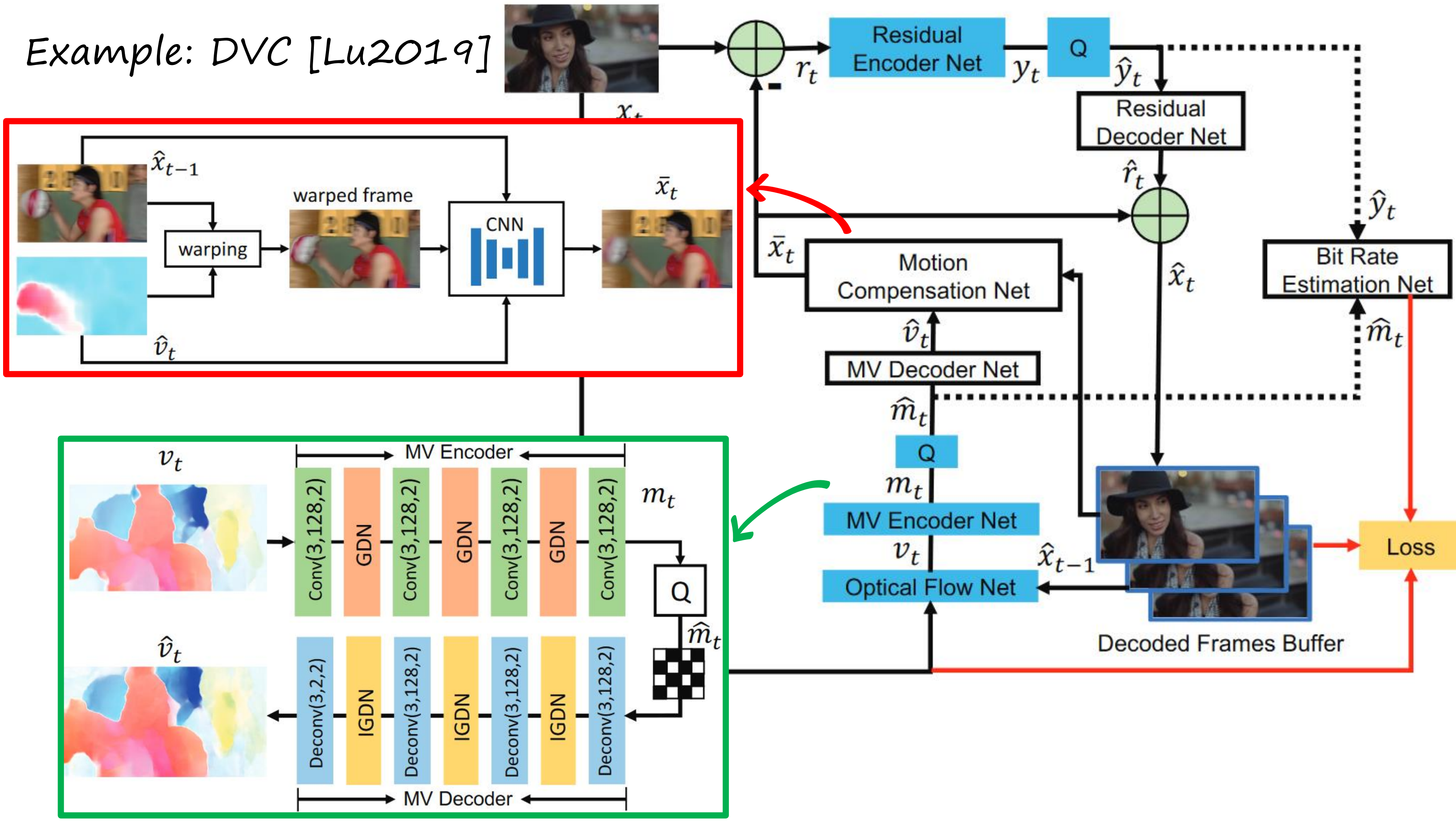
Neural video compression

Idea: replace modules by trainable neural networks



Motion-compensated neural video compression

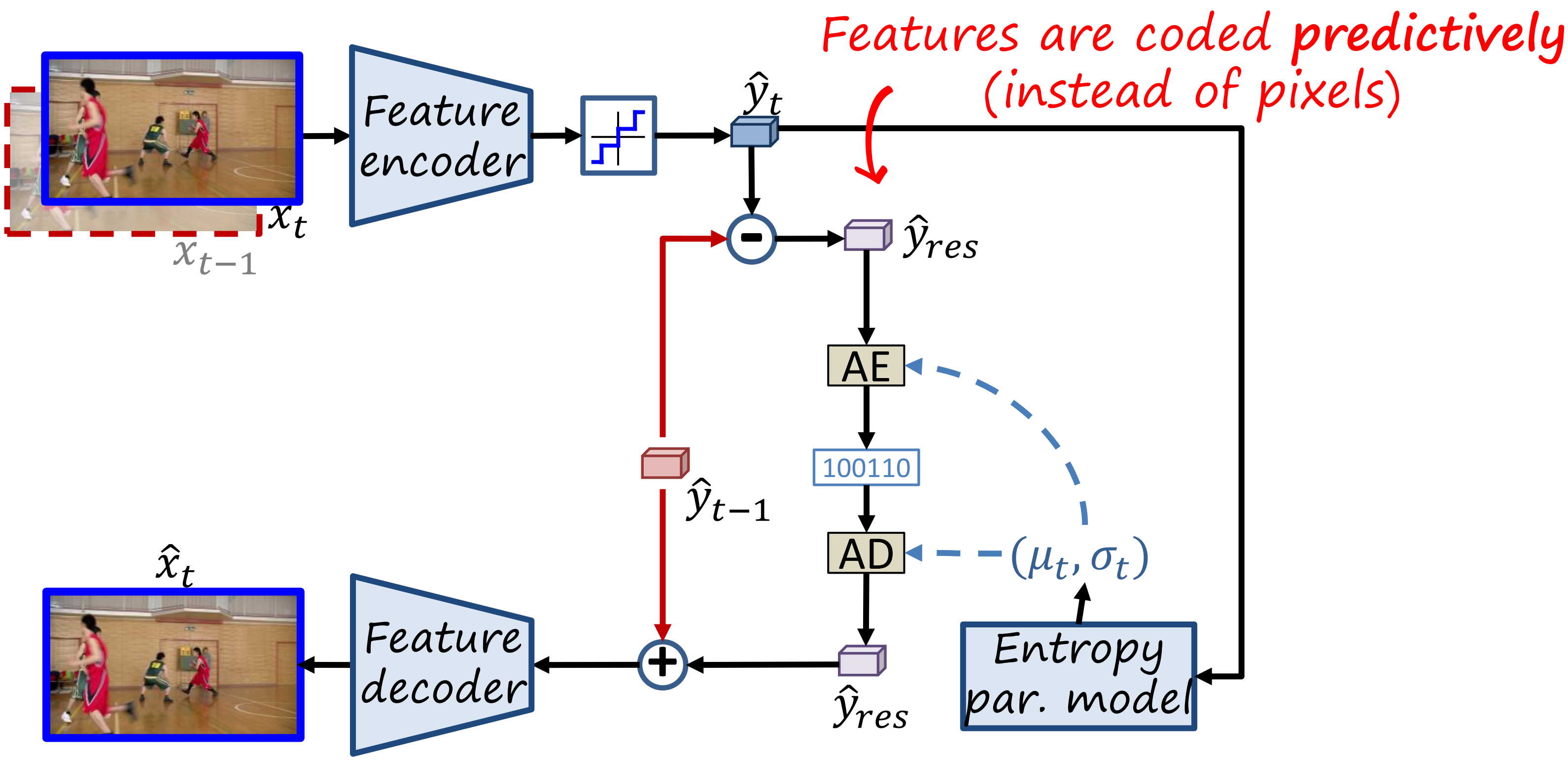
Example: DVC [Lu2019]



Lu et al. DVC: An End-To-End Deep Video Compression Framework CVPR 2019

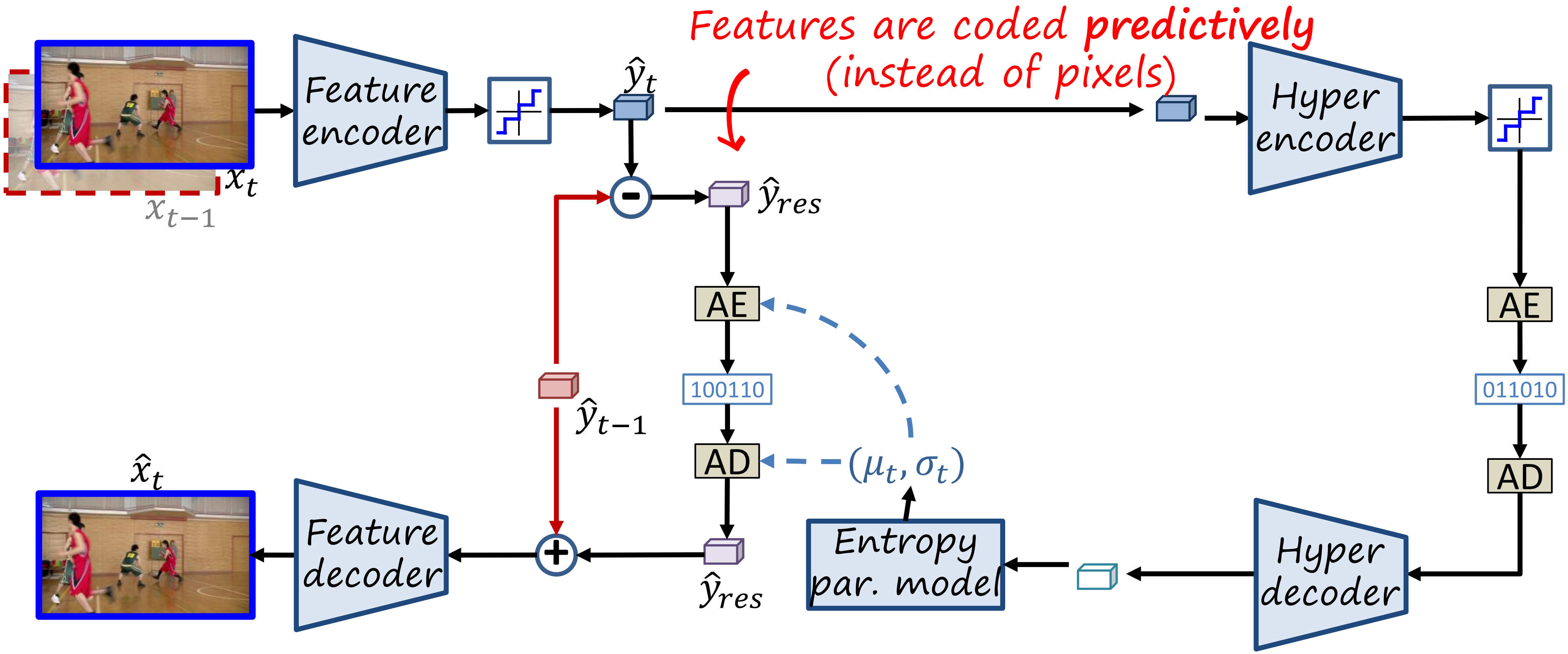
Predictive feature coding

Exploit temporal redundancy directly in the latent space
- More flexible, since it is not constrained by the characteristics of pixel space



Predictive feature coding

Exploit temporal redundancy directly in the latent space
- More flexible, since it is not constrained by the characteristics of pixel space

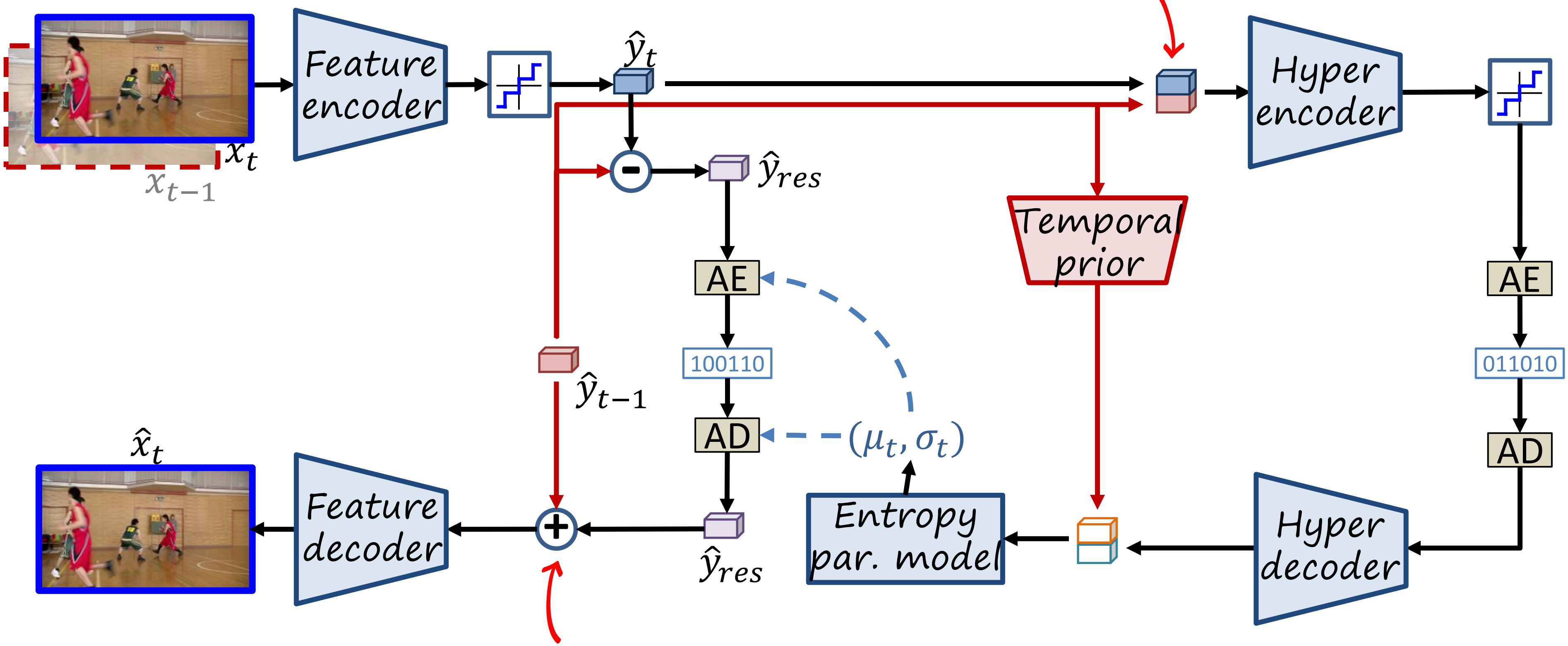


Conditional entropy model

Condition on previous feature to exploit temporal redundancy for entropy modeling

Examples: [Liu2020], STEM [Sun2021]

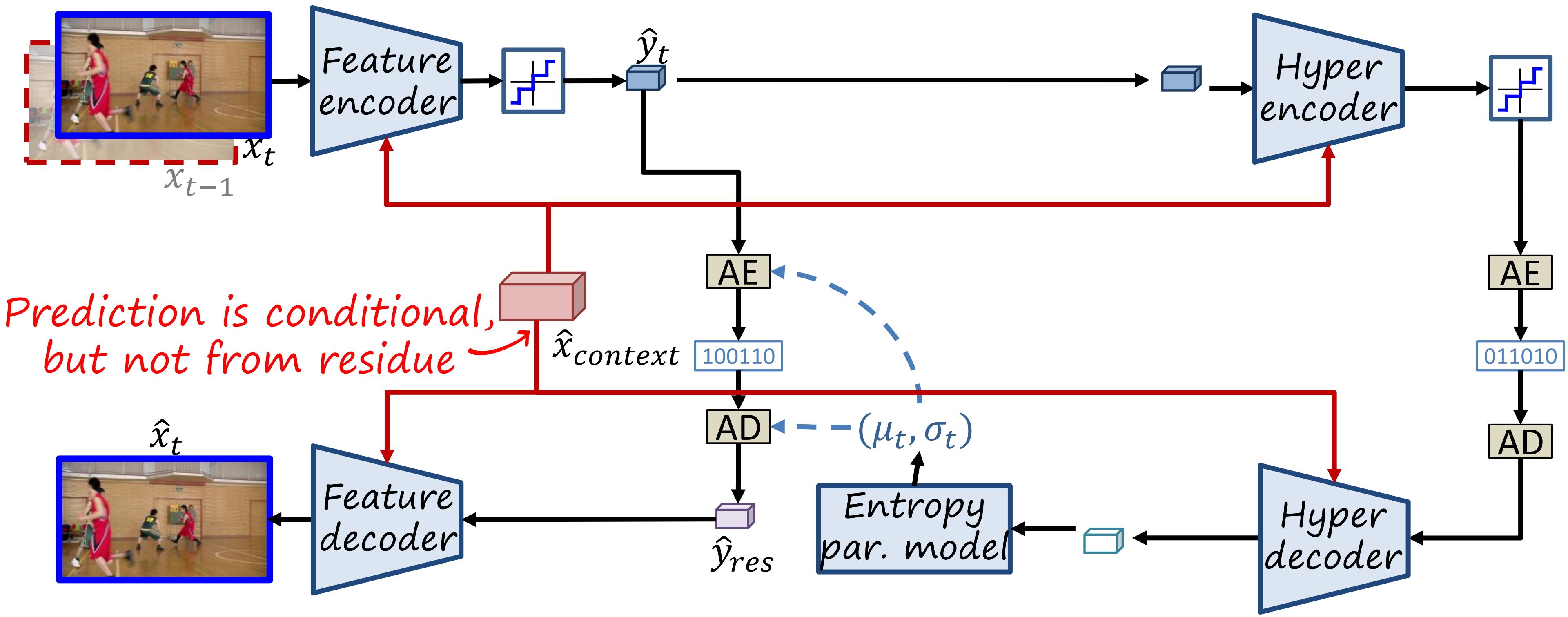
The probability model processes pairs of frame **features** (i.e. conditional)



Features are still coded **predictively**

Conditional video compression

Condition on a contextual feature to exploit temporal redundancy



Conditional video compression

Why residual coding is suboptimal compared to conditional coding?

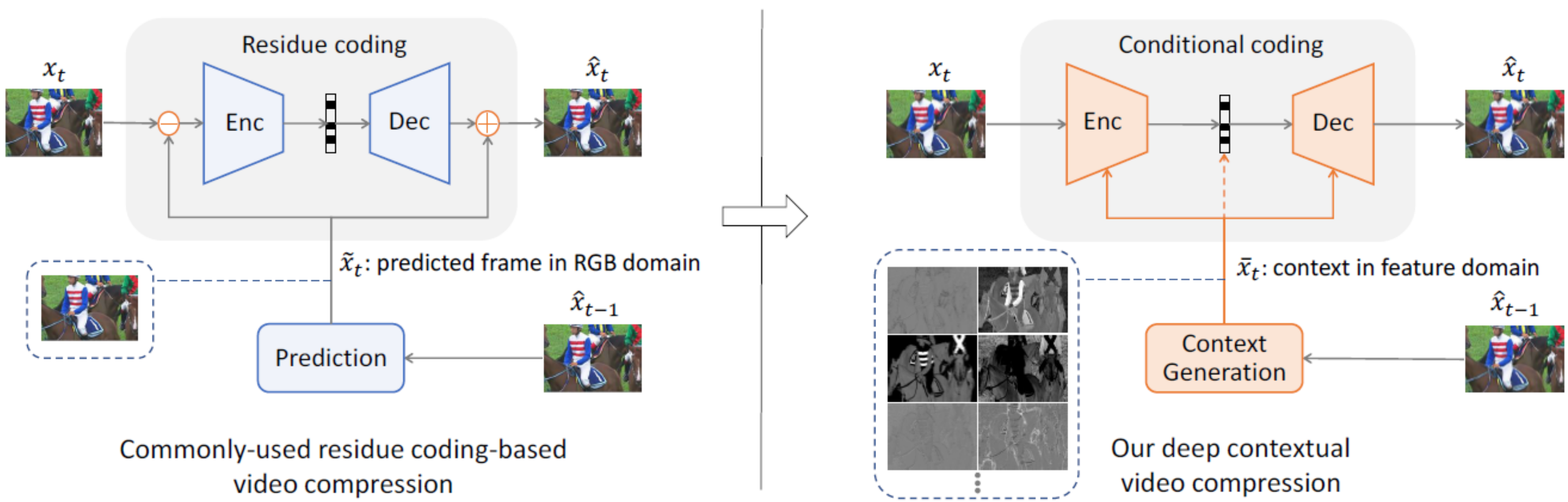
- Let's consider we want to encode x_t given context $\tilde{x}_t$,
- We are interested in estimating $\mathrm{p}(x_t|\tilde{x}_t)$
- Residual coding is a particular case of conditional coding $\mathrm{p}(x_t|\tilde{x}_t) = \mathrm{p}(x_t - \tilde{x}_t)$

  i.e. substraction is a particular fixed (not learnable) operation to predict x_t

- Residual entropy is higher than conditional entropy, so less compressible

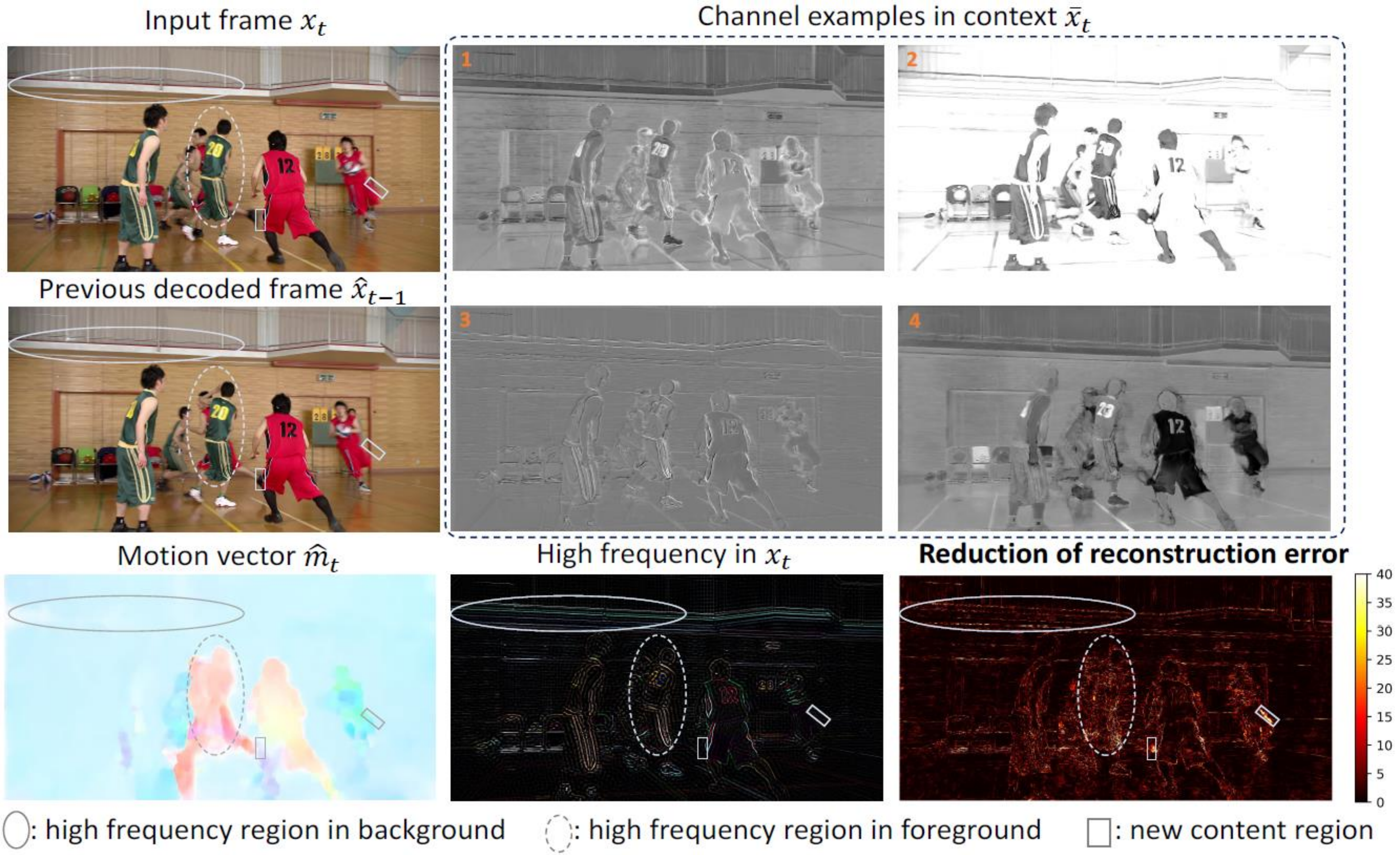
  i.e. $\mathrm{H}(x_t - \tilde{x}_t) \geq \mathrm{H}(x_t|\tilde{x}_t)$

- $\tilde{x}_t$ doesn't need to be a frame, could be a more flexible learned context



Li et al. Deep Contextual Video Compression NeurIPS 2021

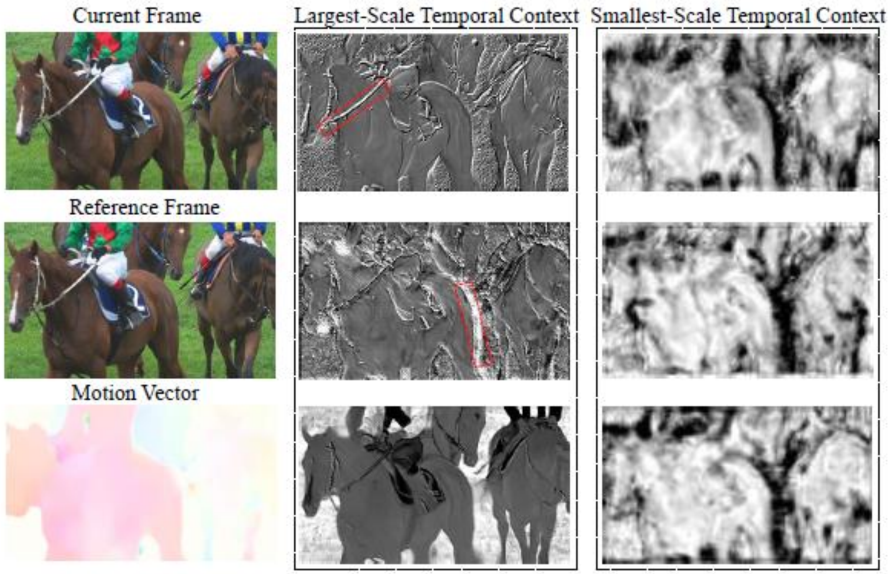
Conditional video compression



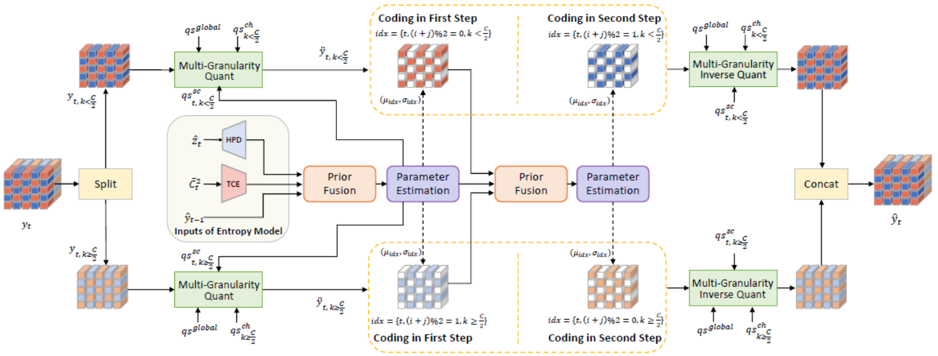
Li et al. Deep Contextual Video Compression NeurIPS 2021

Richer contextual models

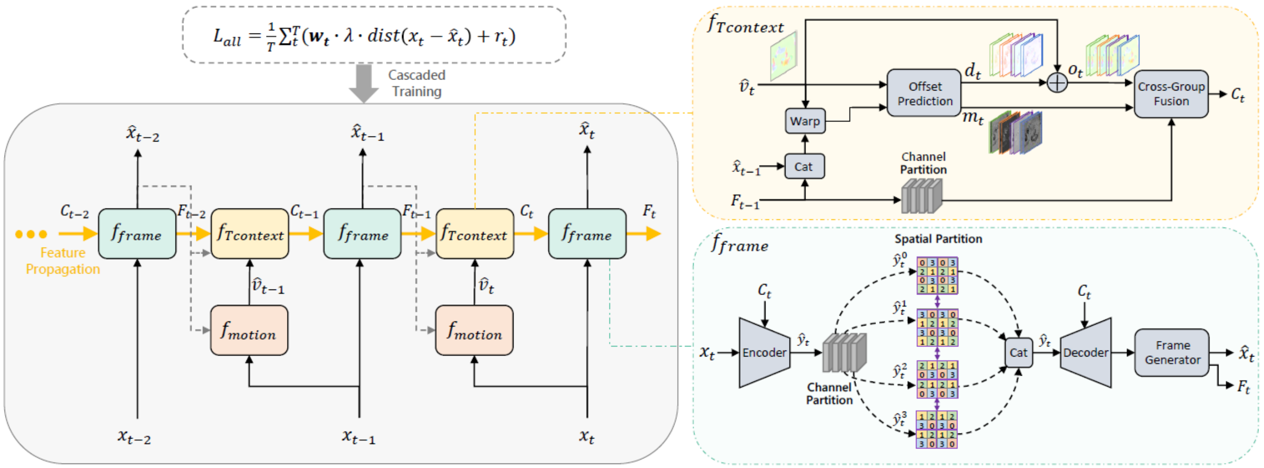
Multi-scale temporal contexts [DCVC-TCM]



Hybrid spatio-temporal entropy modeling [DCVC-HEM]



More diverse contexts [DCVC-DC]

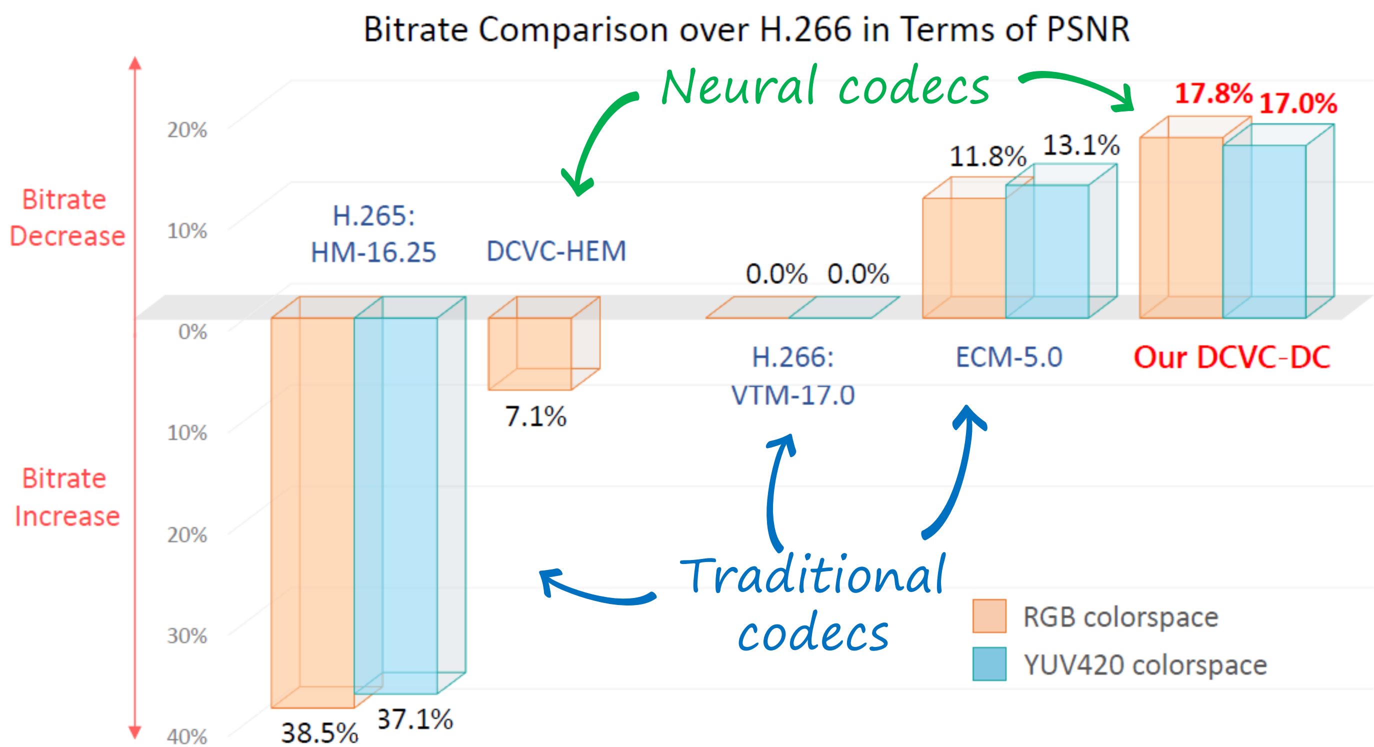


DCVC-TCM: Sheng et al. Temporal Context Mining for Learned Video Compression arxiv 2021/TMM 2023

DCVC-HEM: Li et al. Deep Contextual Video Compression ACM Multimedia 2022

DCVC-DC: Li et al. Neural Video Compression with Diverse Contexts arxiv 2023

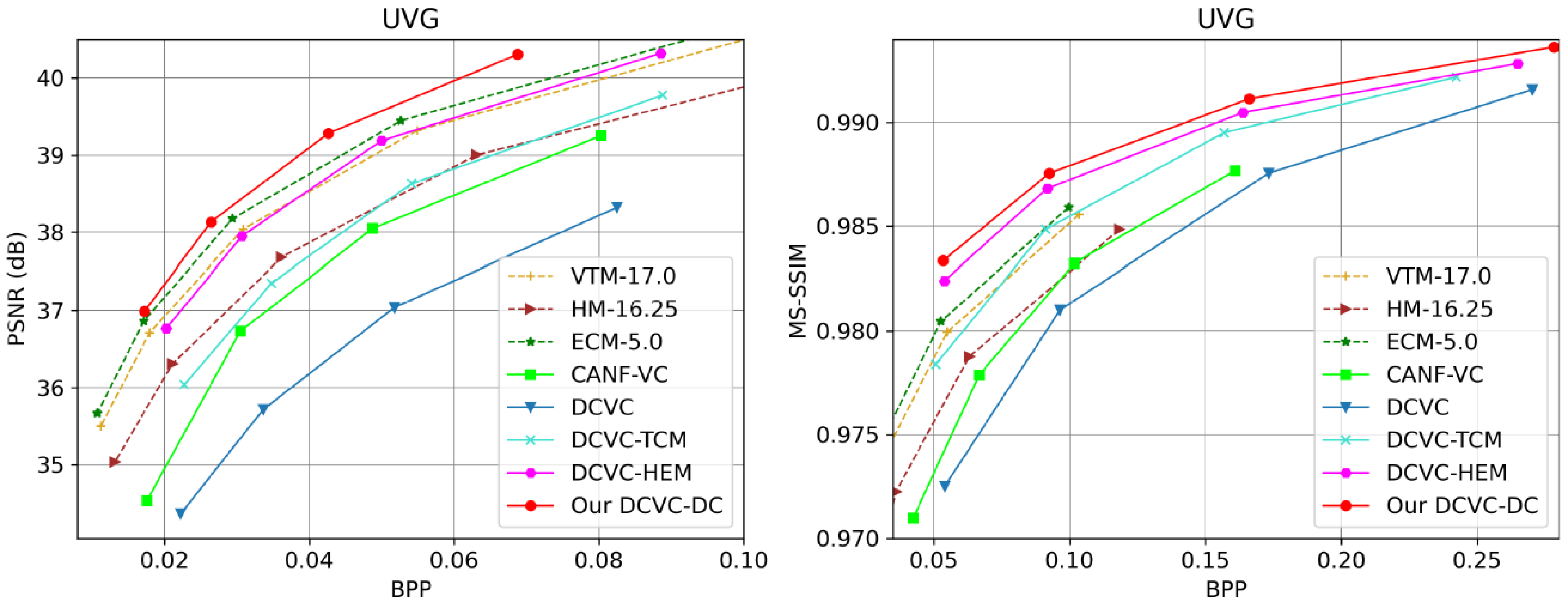
Current SOTA in video compression



*HM, VTM, ECM use their best compression ratio configurations for low delay

Li et al. Neural Video Compression with Diverse Contexts arxiv 2023

Current SOTA in video compression



	MACs	Encoding Time	Decoding Time
DCVC-HEM [29]	3279G	890ms	652ms
Our DCVC-DC	2642G	1005ms	765ms

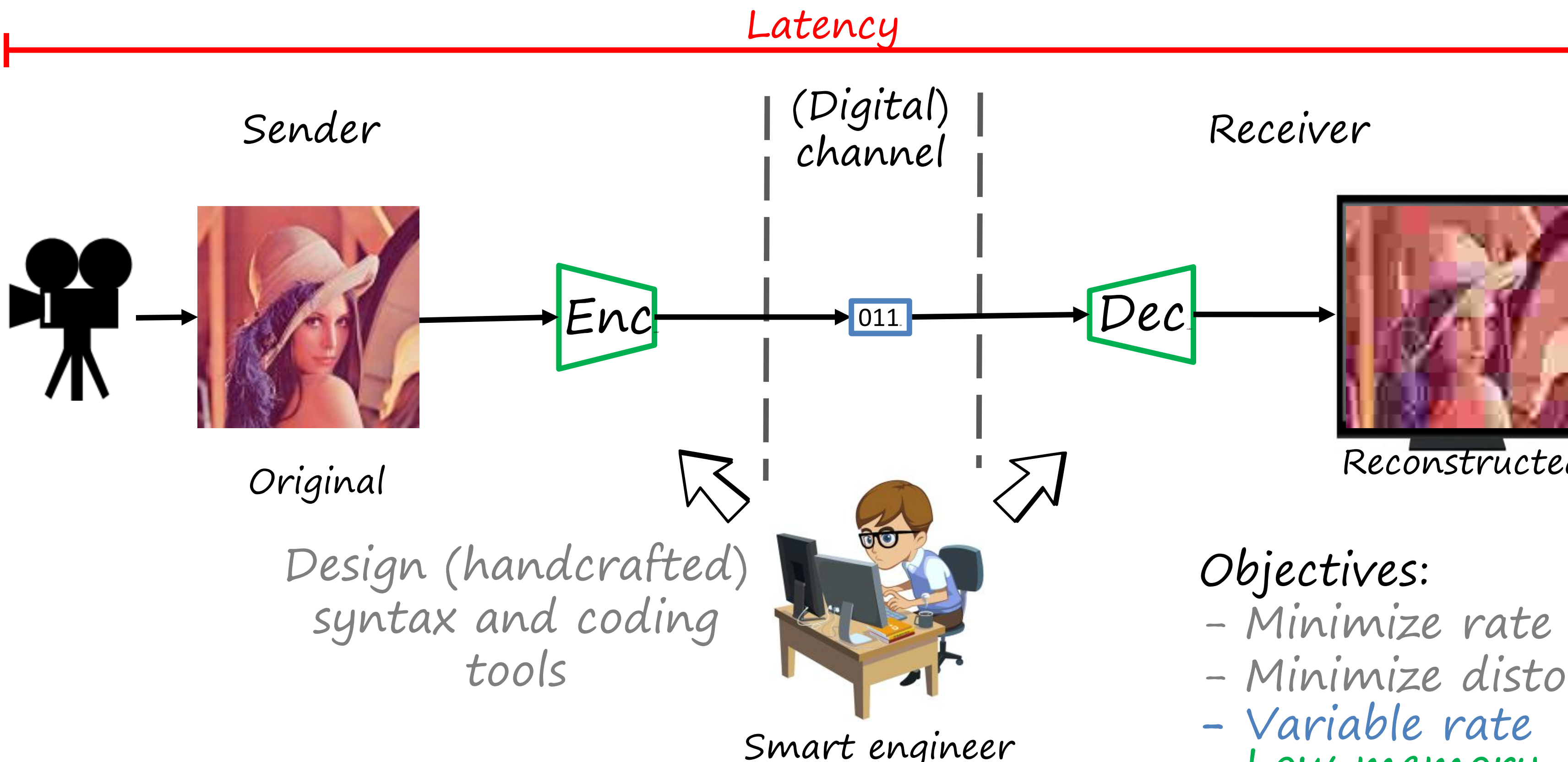
Note: Tested on NVIDIA 2080TI with using 1080p as input.

Li et al. Neural Video Compression with Diverse Contexts arxiv 2023

Outline

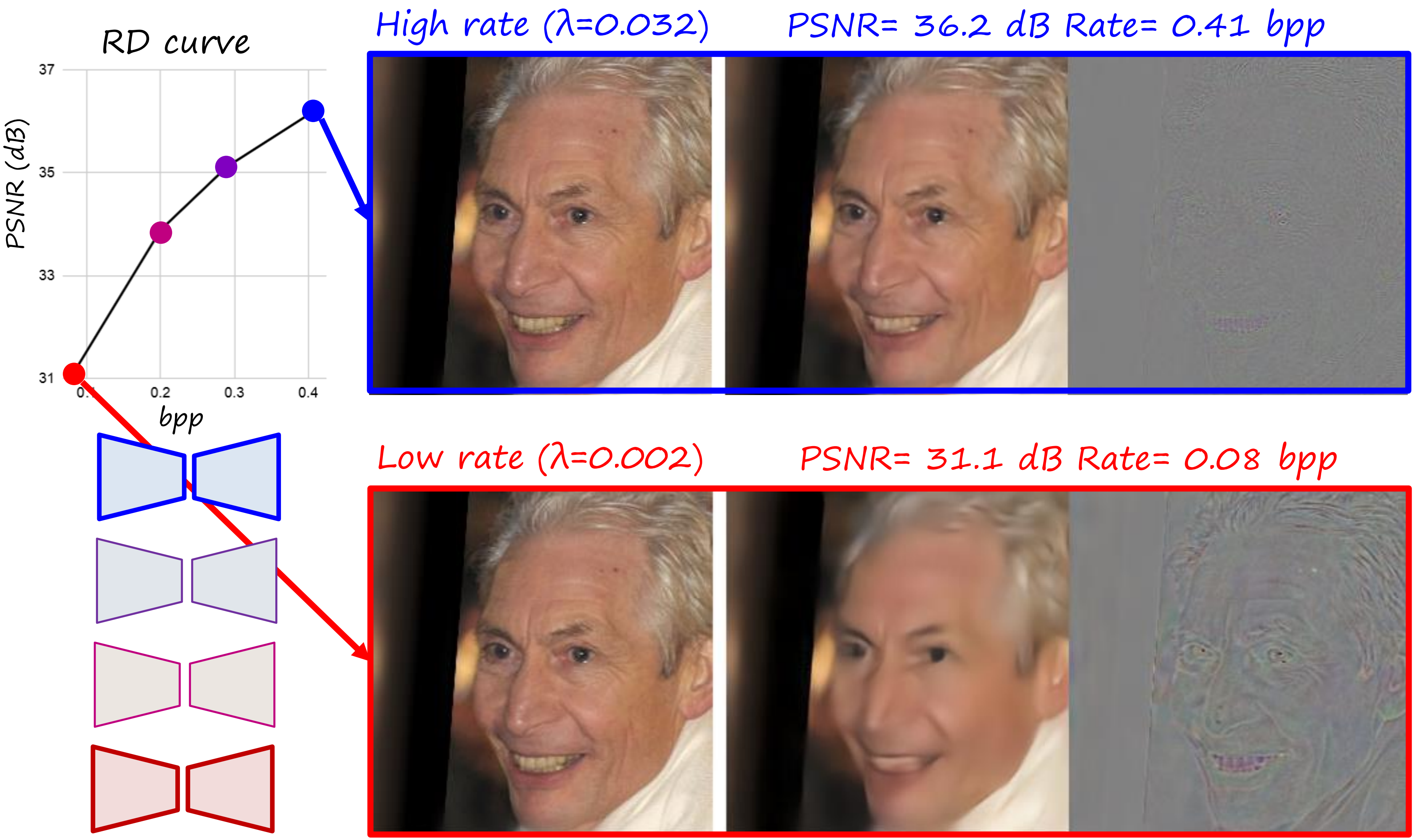
- Neural image/video compression: a walkthrough
- Our work on neural image/video compression
  - Practical neural image/video compression
  - Neural image compression for machines
- Other works

Practical image/video compression



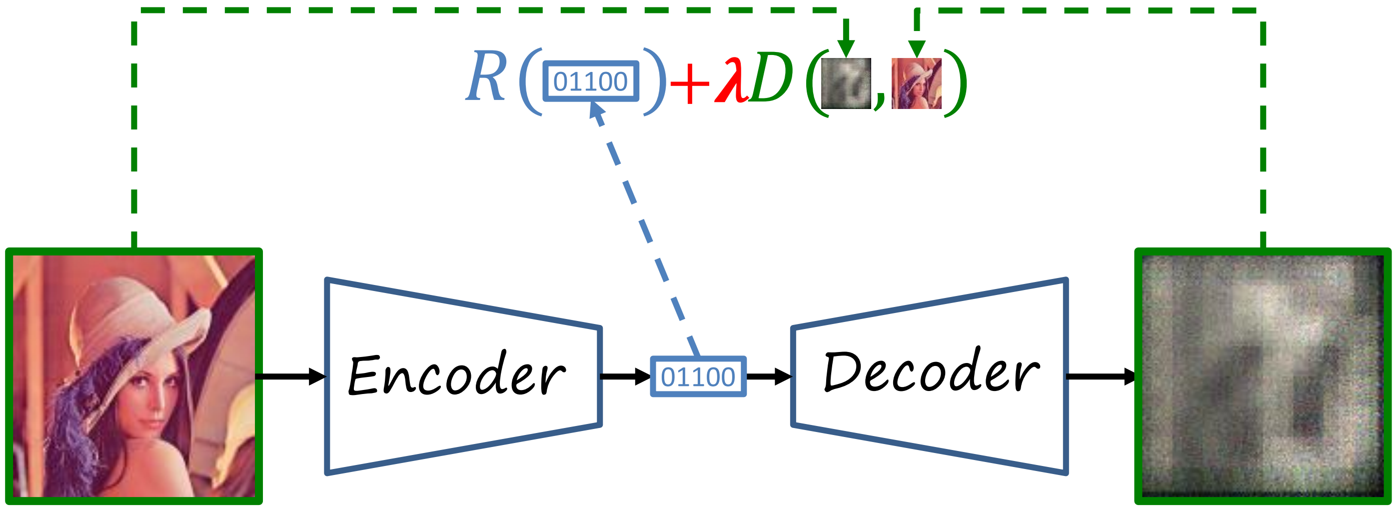
Objectives:
- Minimize rate
- Minimize distortion
- Variable rate
- Low memory
- Low computation
- Low latency
- **Compatibility**
- **Domain-specific**
- ...

Rate-distortion tradeoff λ in NIC



Problems: total memory, total training time

Is neural image compression practical?



Practical neural image compression?
- Minimize rate ✓
- Minimize distortion ✓
- Variable rate ✗
- Low memory ✗
- Low computation ✗
- Low latency ✗

MAE [SPL2020]

SlimCAE [CVPR2021]

Limitations
- λ is fixed
- Heavy encoders/decoders

Other practical considerations
- **Domain-specific codecs** (e.g. videoconference, screencast)
- **Back./forw. compatibility** (with legacy encoders/decoders)

DANICE [CLIC2021]

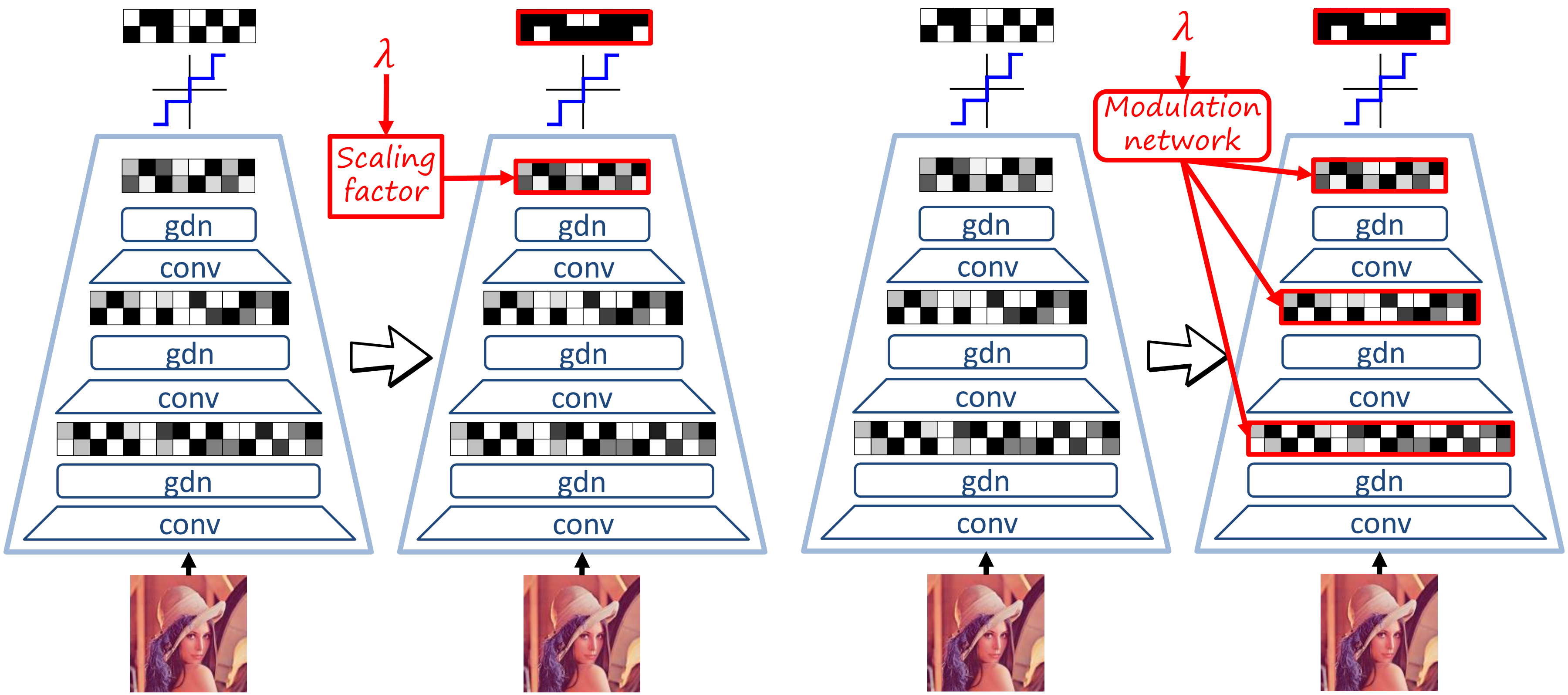
[SPL2020] Variable Rate Deep Image Compression with Modulated Autoencoder, Signal Processing Letters 2020
[CVPR2021] Slimmable compressive autoencoders for practical imaga compression, CVPR 2021
[CLIC2021] DANICE: Domain adaptation without forgetting in neural image compression, CLIC 2021 at CVPR 2021

Variable rate with modulated autoencoders

Objective: one single model for multiple λ

Bottleneck scaling [Theis2017]

Feature modulation [MAE, cAE]

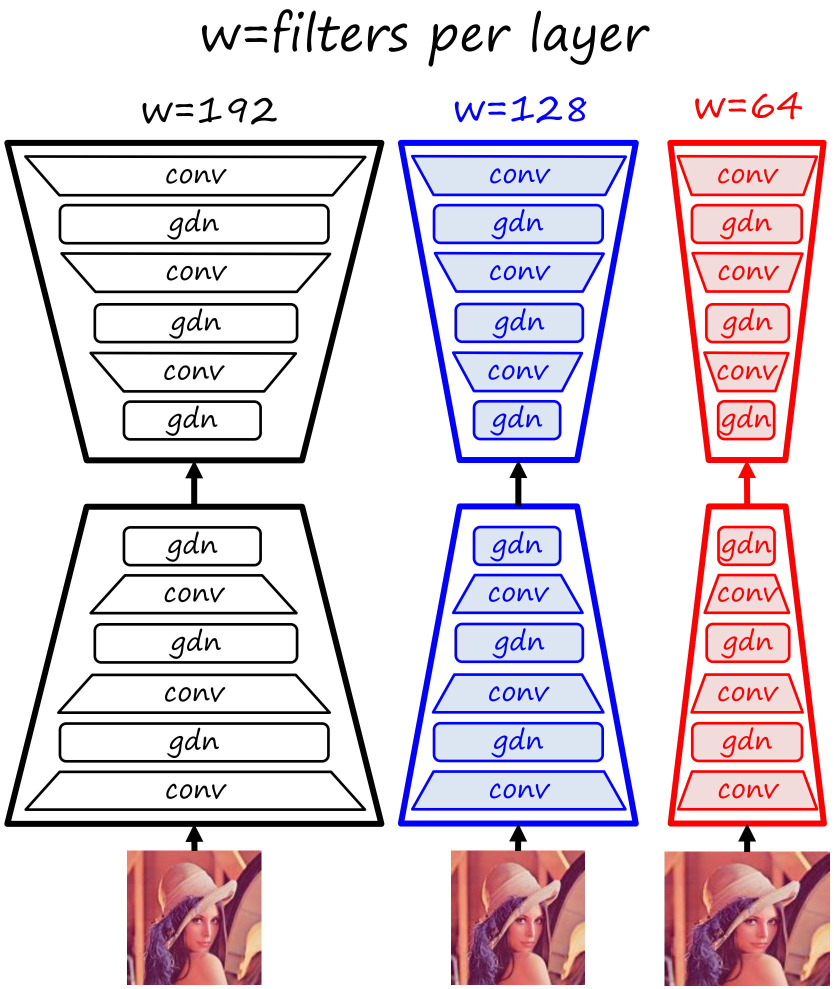


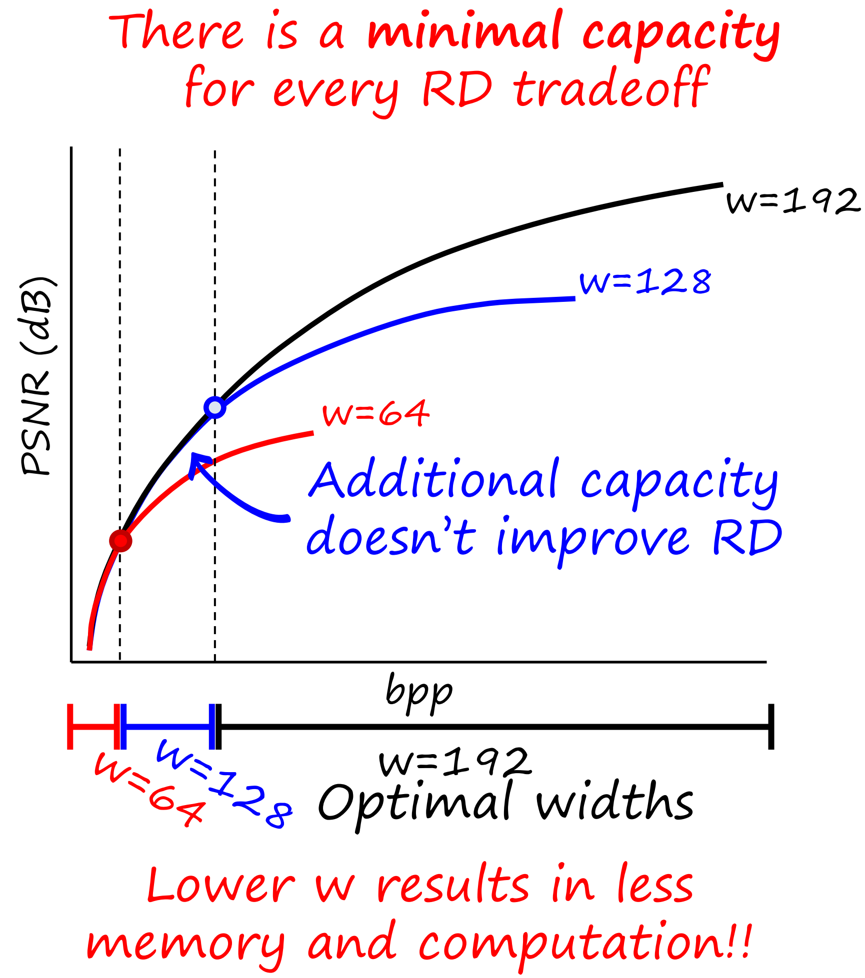
- Minimize rate ✓
- Minimize distortion ✓
- Variable rate ✓

- Low memory ✗
- Low computation ✗
- Low latency ✗

cAE: conditional autoencoder [Choi2019]
MAE: modulated autoencoder [Yang2020]

Model capacity and rate-distortion

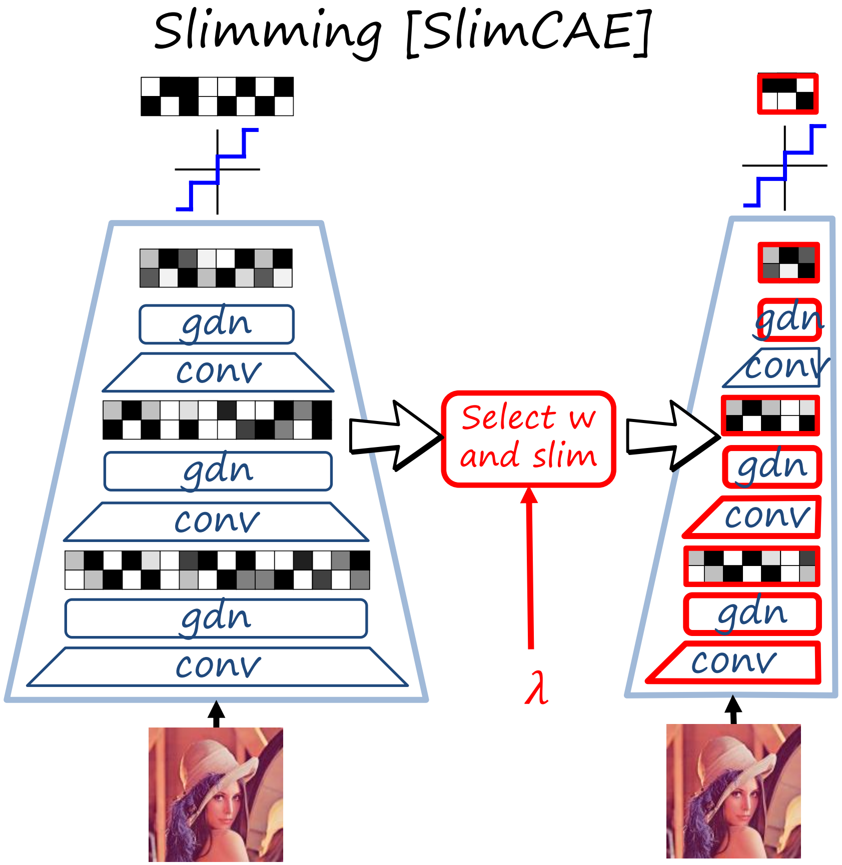




[Slimmable compressive autoencoders for practical image compression](), CVPR 2021

Slimmable compressive autoencoder

Approach: slim the network to the minimal capacity for a given λ

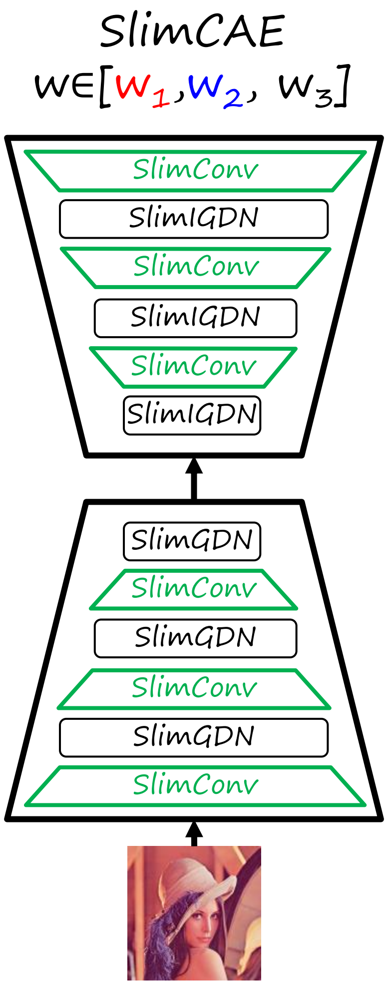


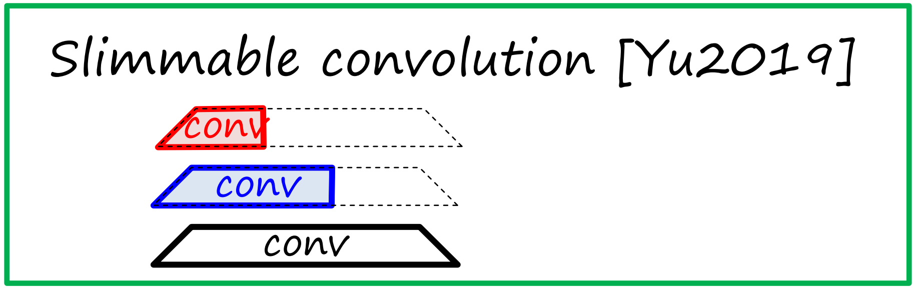
- Minimize rate ✓
- Minimize distortion ✓
- Variable rate ✓
- Lower memory ✓
- Lower computation ✓
- Lower latency ✓

(for low-mid rates)

Slimmable compressive autoencoders for practical image compression, CVPR 2021

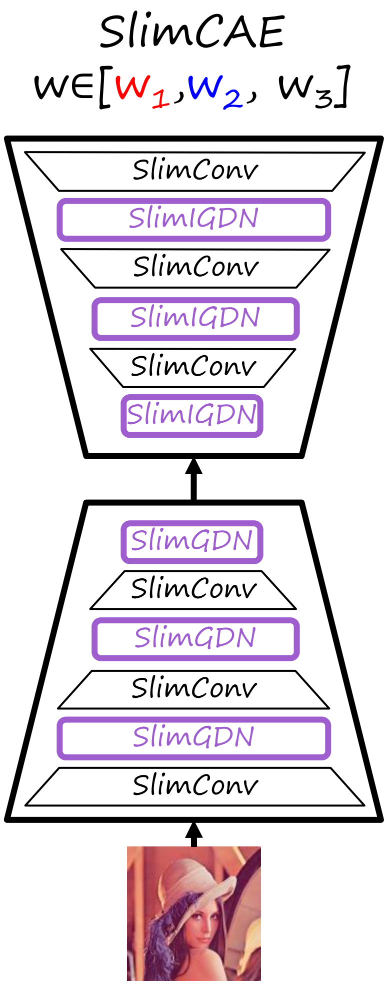
Slimmable layers in SlimCAE

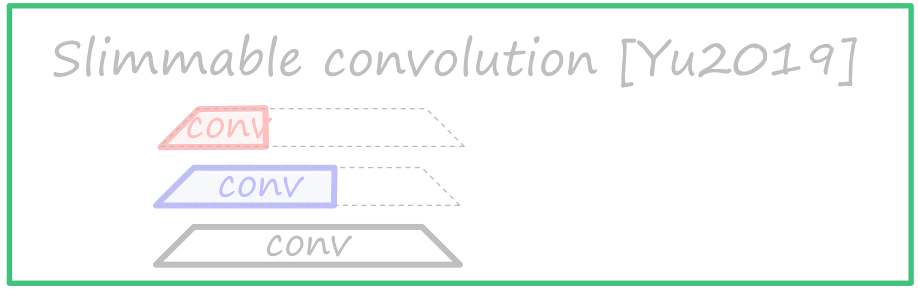


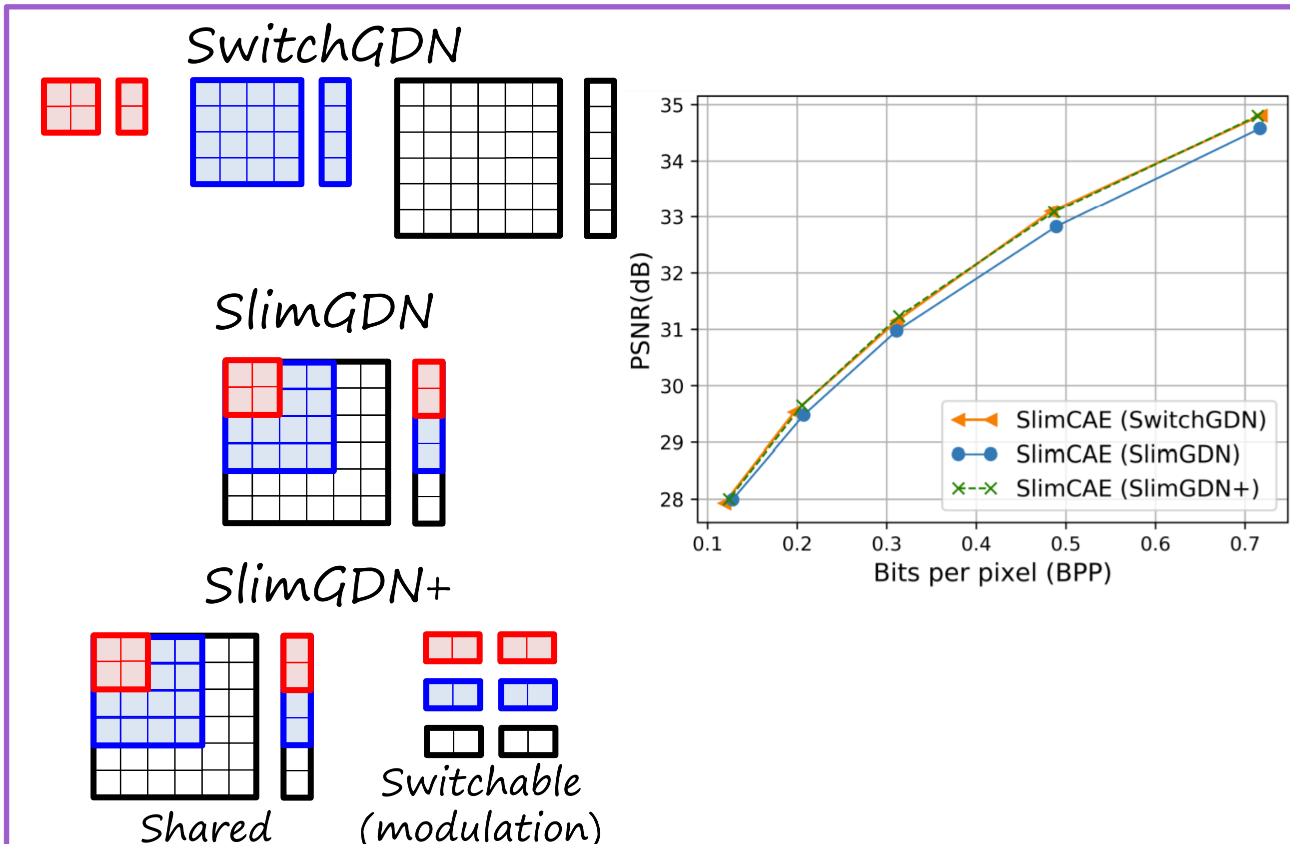


[Slimmable compressive autoencoders for practical image compression](), CVPR 2021

Slimmable layers in SlimCAE





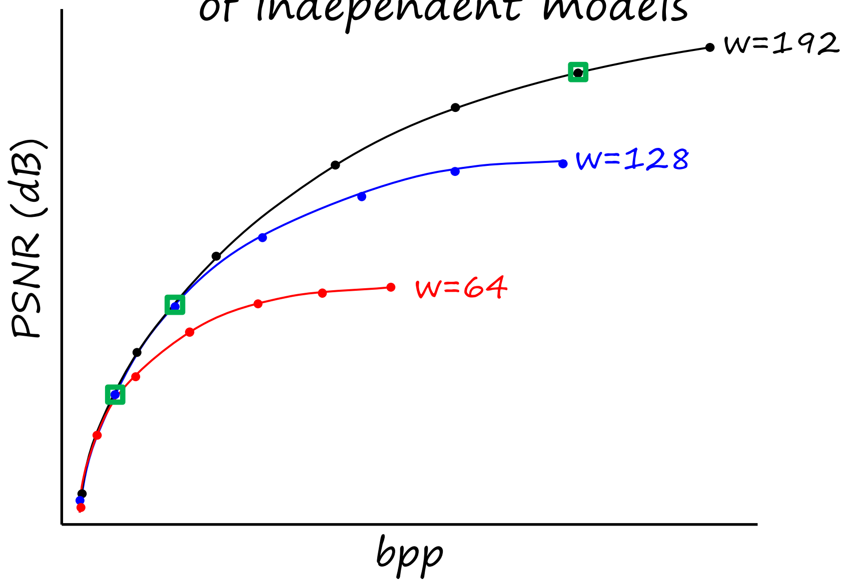


Slimmable compressive autoencoders for practical image compression, CVPR 2021

Training SlimCAE

Problem: we need the ***optimal* λs** to train the SlimCAE

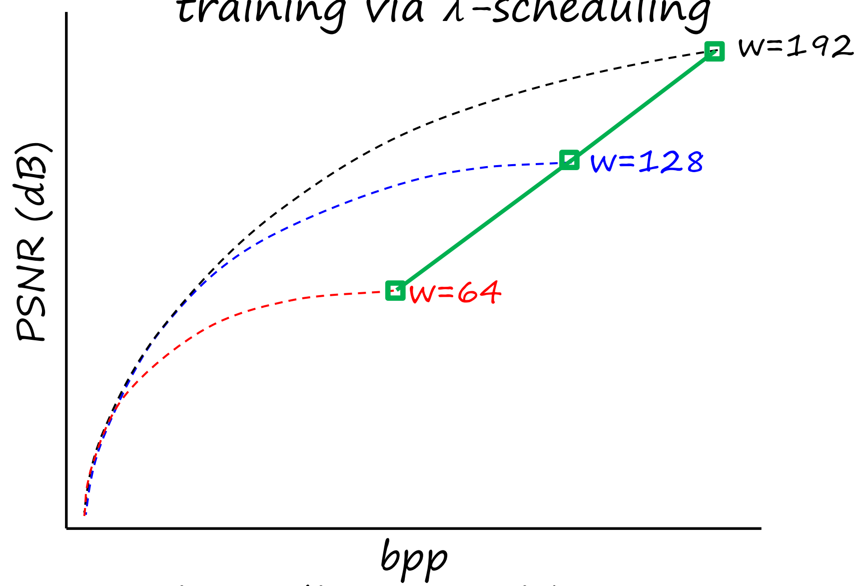
Estimate from RD curves of independent models



1. Train several independent models for different w
2. Plot RD curves and find critical points
3. Estimate optimal λs from trained models

Problem: extremely expensive!

Automatically estimate during training via λ-scheduling



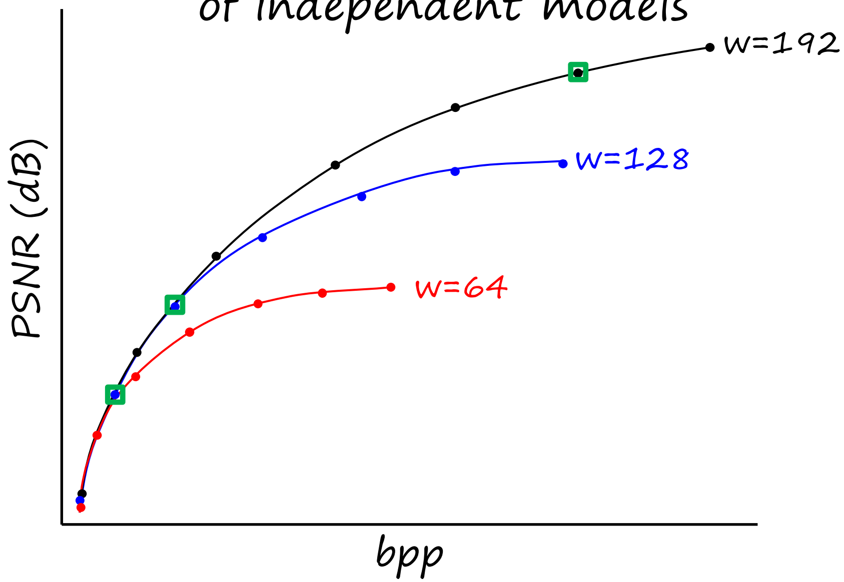
1. Train a SlimCAE with $\lambda_1 = \lambda_2 = \lambda_3$
2. While not converged do
   - Update λ s according to schedule
   - Optimize CAE

Slimmable compressive autoencoders for practical image compression, CVPR 2021

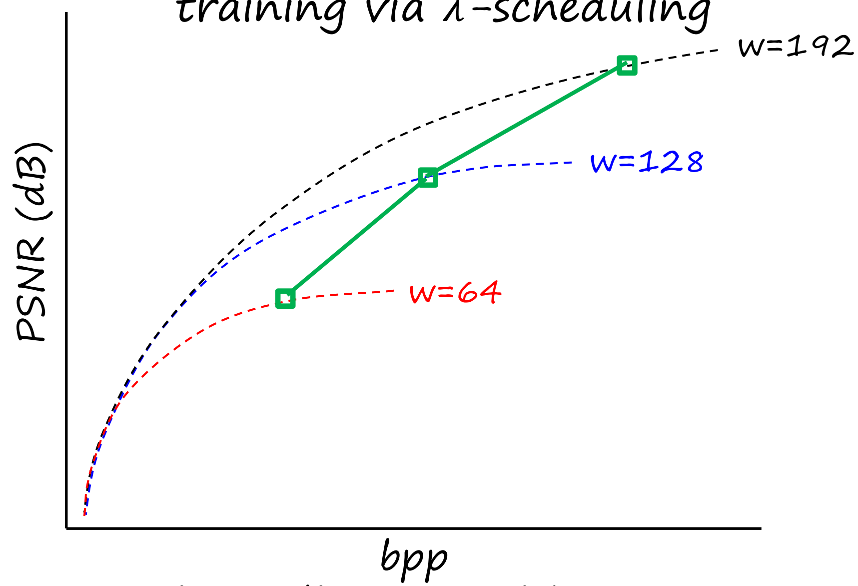
Training SlimCAE

Problem: we need the optimal λs to train the SlimCAE

Estimate from RD curves of independent models



Automatically estimate during training via λ-scheduling



1. Train several independent models for different w
2. Plot RD curves and find critical points
3. Estimate optimal λs from trained models

Problem: extremely expensive!

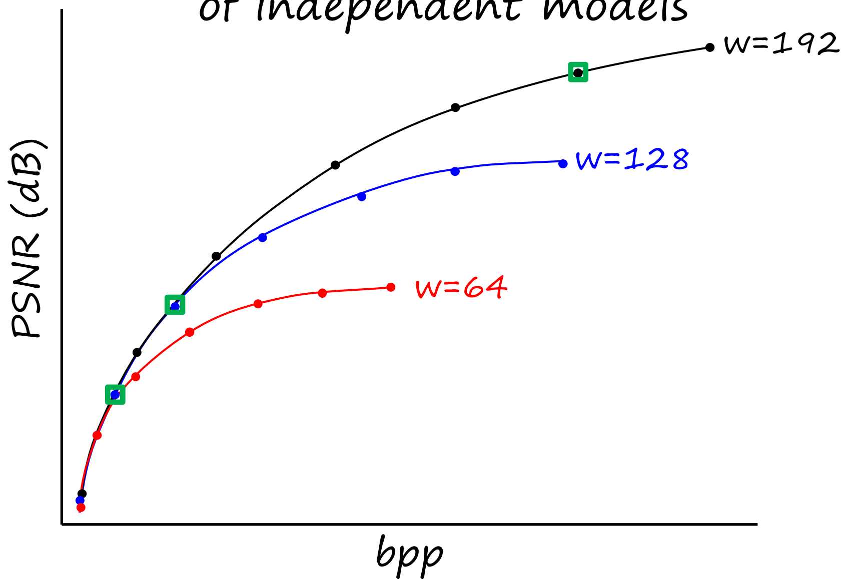
1. Train a SlimCAE with $\lambda_1 = \lambda_2 = \lambda_3$
2. While not converged do
   - Update λs according to schedule
   - Optimize CAE

Slimmable compressive autoencoders for practical image compression, CVPR 2021

Training SlimCAE

Problem: we need the optimal λs to train the SlimCAE

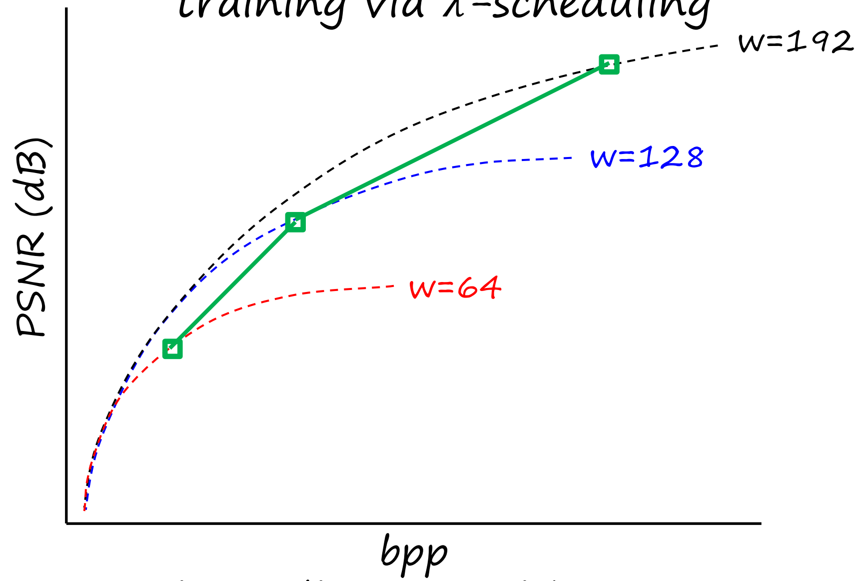
Estimate from RD curves of independent models



1. Train several independent models for different w
2. Plot RD curves and find critical points
3. Estimate optimal λs from trained models

Problem: extremely expensive!

Automatically estimate during training via λ-scheduling



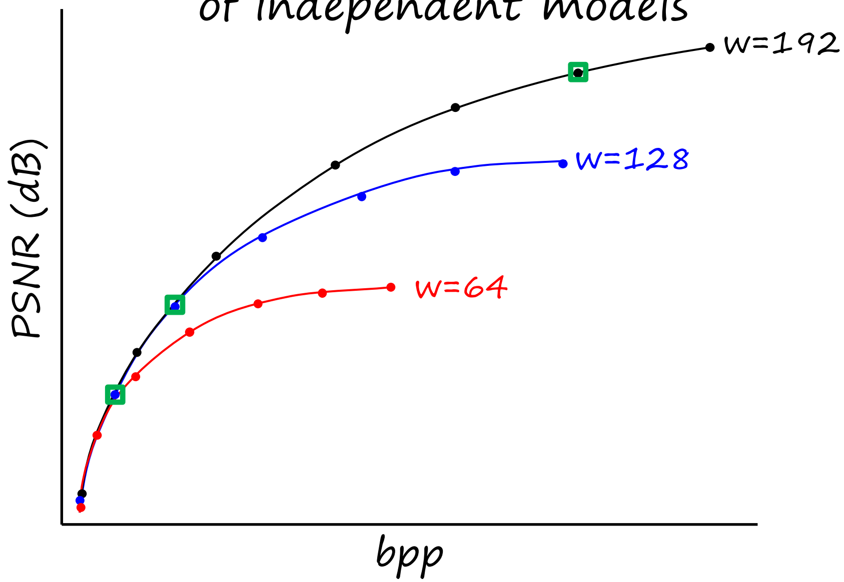
1. Train a SlimCAE with $\lambda_1 = \lambda_2 = \lambda_3$
2. While not converged do
   - Update λs according to schedule
   - Optimize CAE

[Slimmable compressive autoencoders for practical image compression](), CVPR 2021

Training SlimCAE

Problem: we need the optimal λs to train the SlimCAE

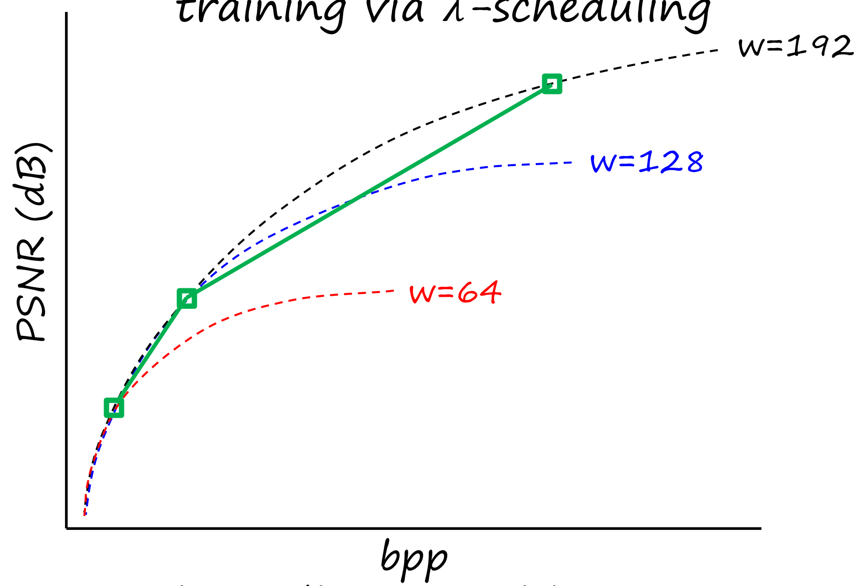
Estimate from RD curves of independent models



1. Train several independent models for different w
2. Plot RD curves and find critical points
3. Estimate optimal λs from trained models

Problem: extremely expensive!

Automatically estimate during training via λ-scheduling

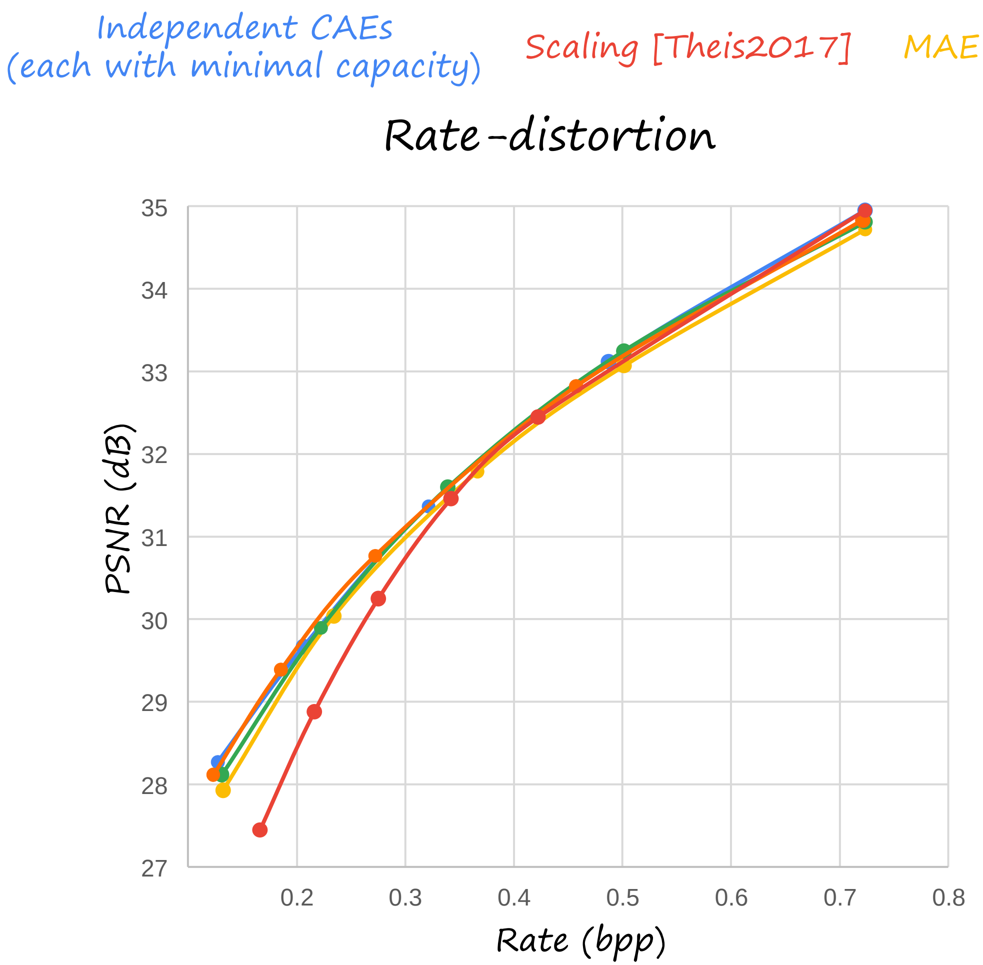


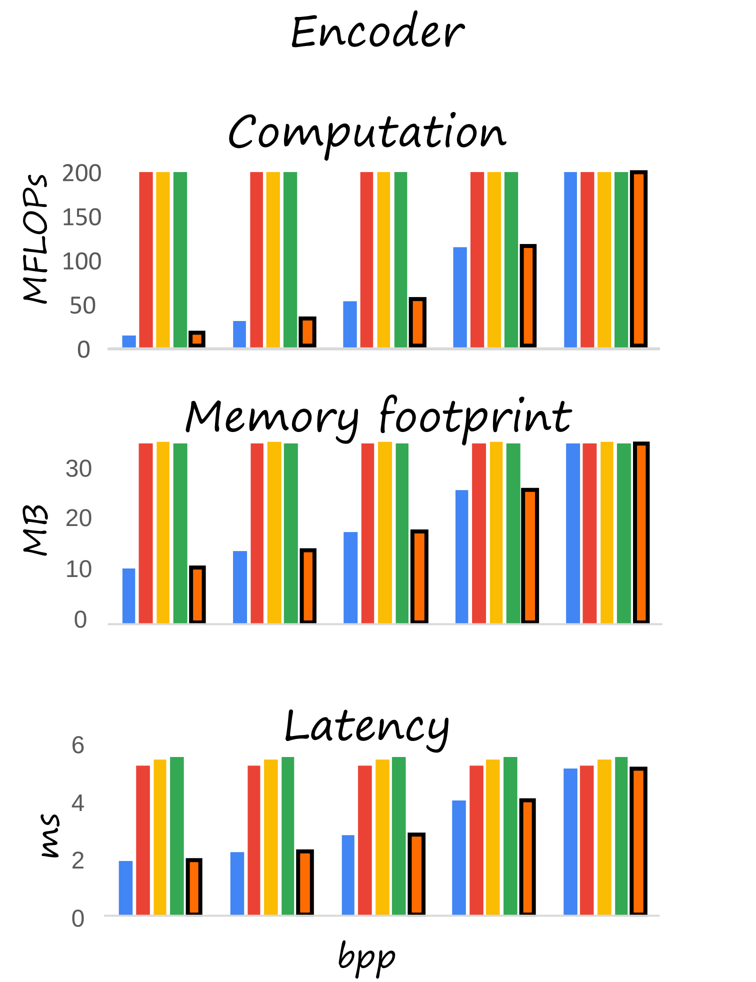
1. Train a SlimCAE with $\lambda_1 = \lambda_2 = \lambda_3$
2. While not converged do
   - Update λs according to schedule
   - Optimize CAE

Directly train one model!

[Slimmable compressive autoencoders for practical image compression](), CVPR 2021

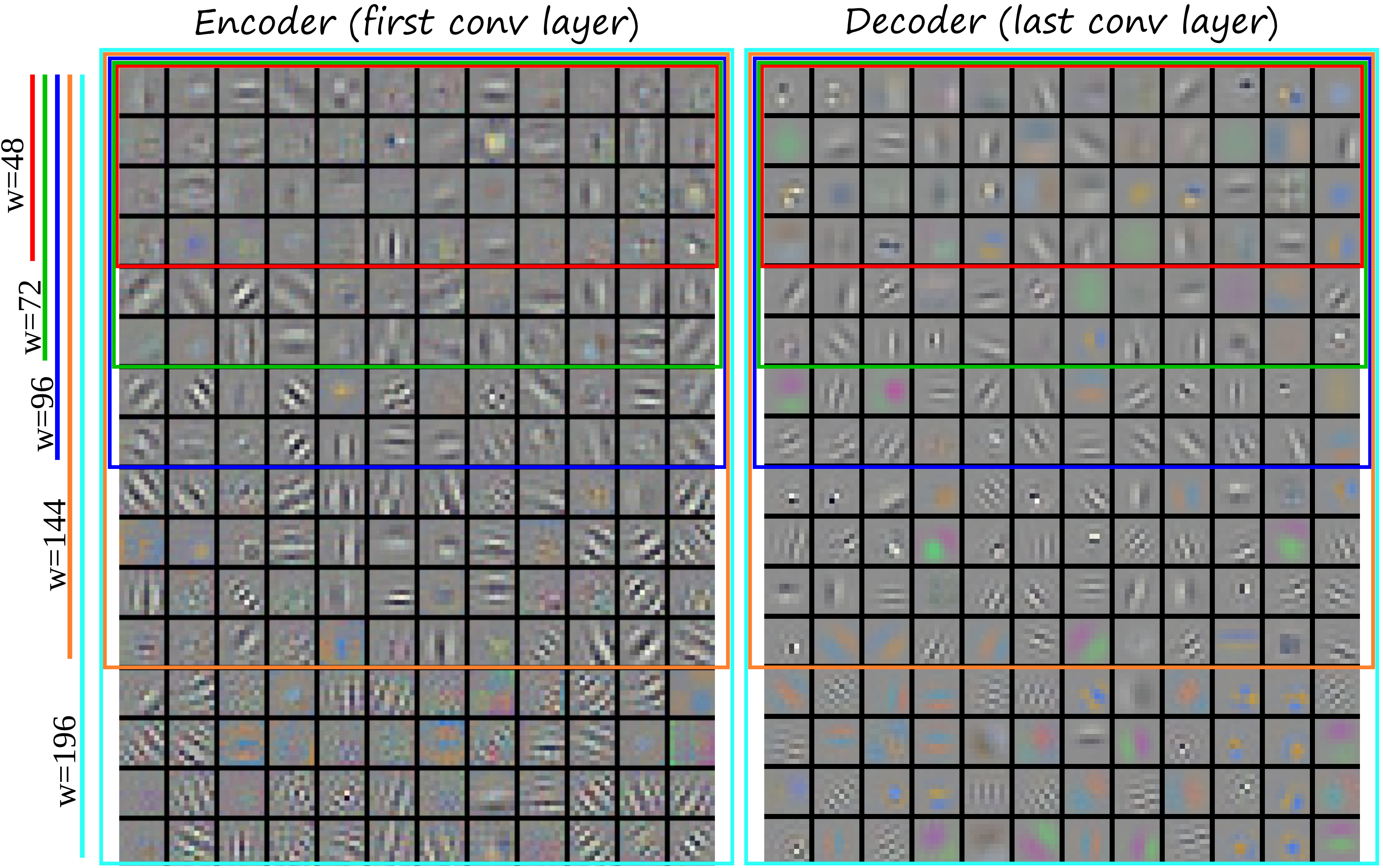
Performance comparison





[Slimmable compressive autoencoders for practical image compression](), CVPR 2021

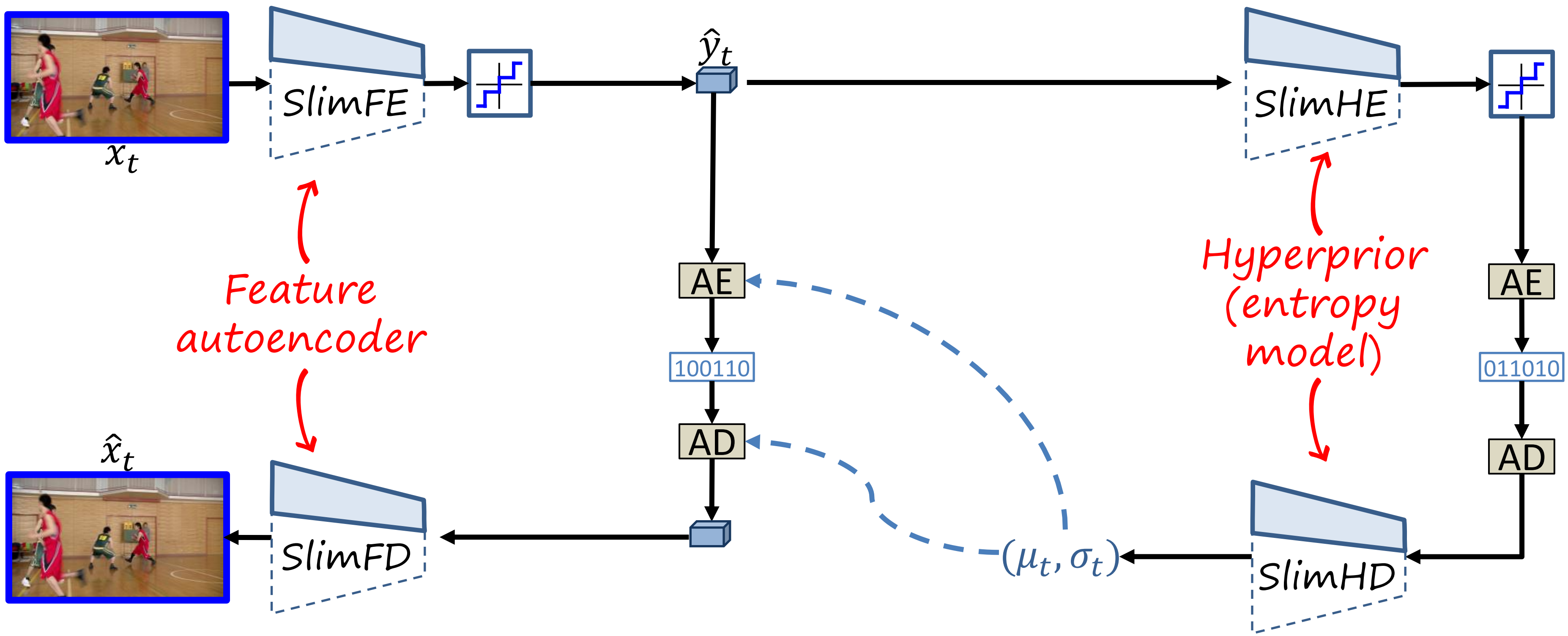
Visualizing some parameters



Slimmable compressive autoencoders for practical image compression, CVPR 2021

Slimmable video codec (SlimVC)

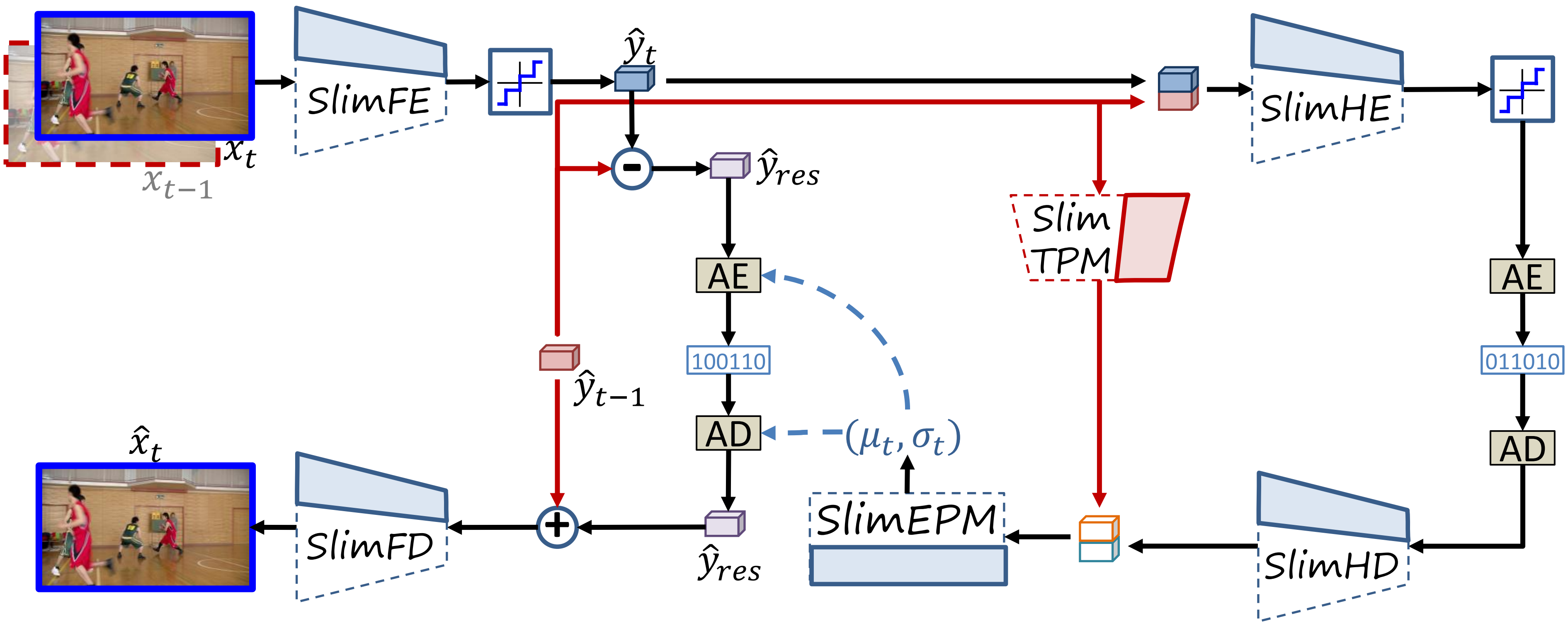
Extending SlimCAE to video



[Slimmable video codec](), CLIC 2022 at CVPR 2022

Slimmable video codec (SlimVC)

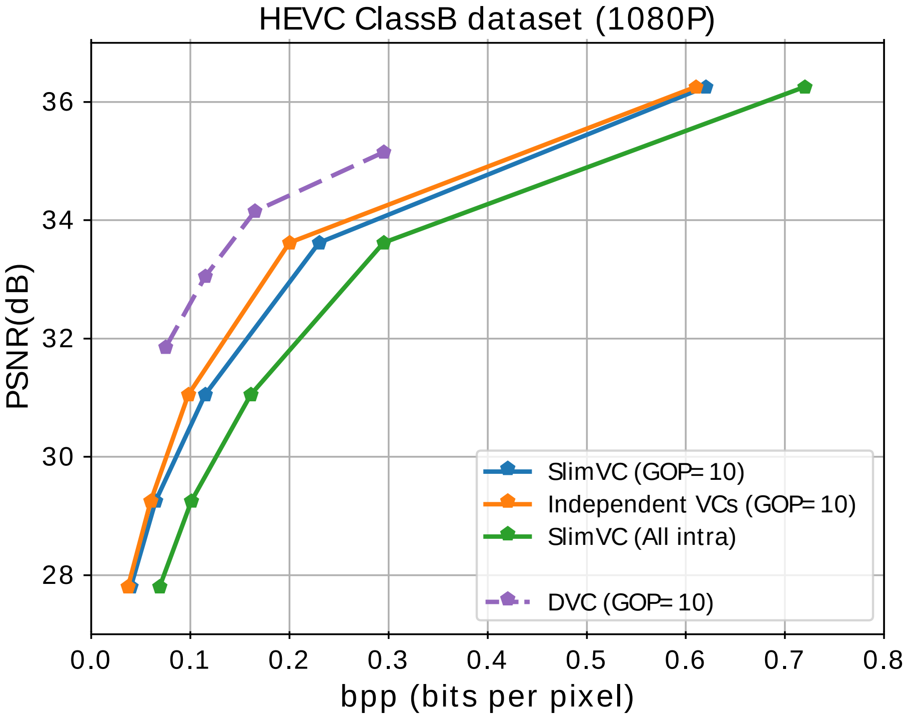
Extending SlimCAE to video



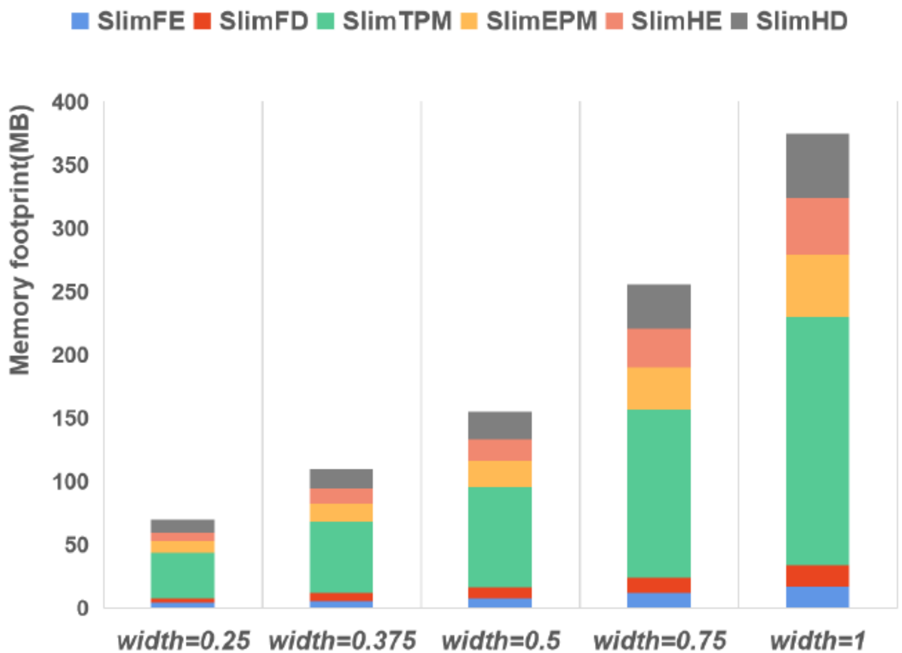
Slimmable video codec, CLIC 2022 at CVPR 2022

Slimmable video codec (SlimVC)

RD performance



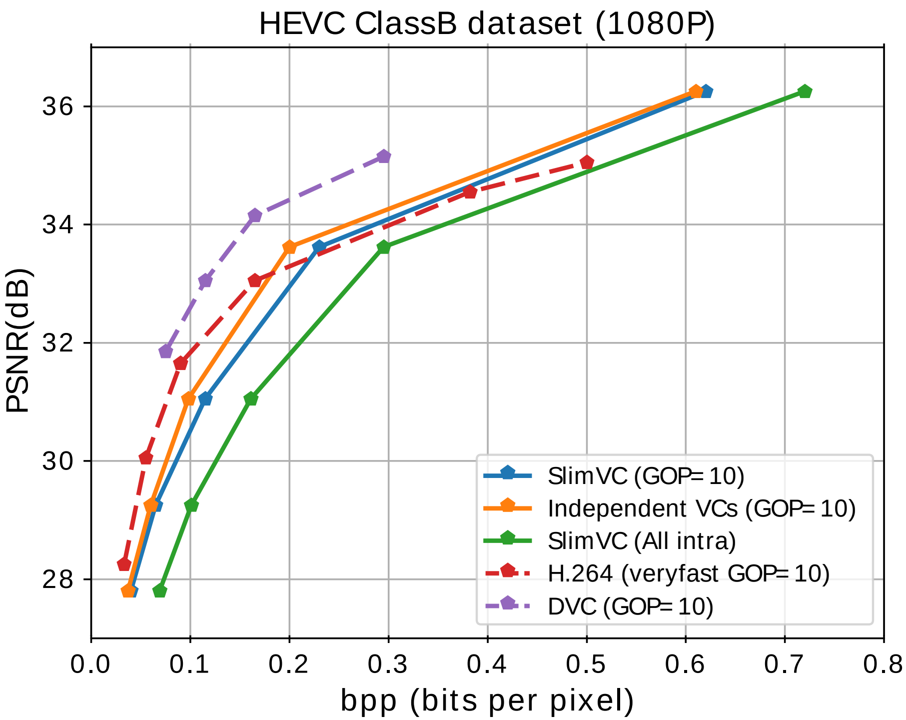
Memory footprint



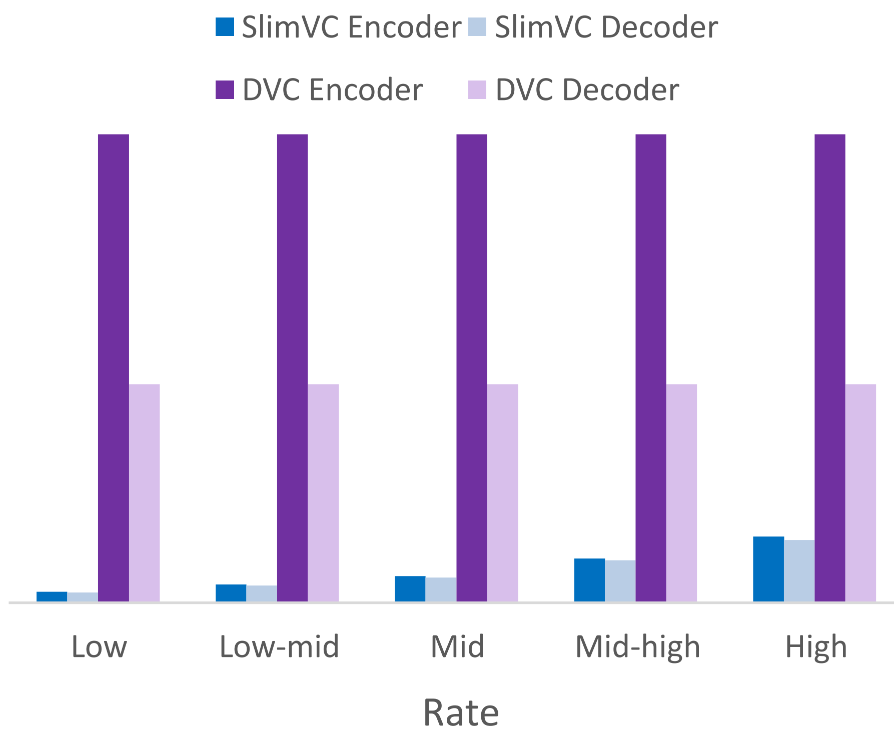
Slimmable video codec, CLIC 2022 at CVPR 2022

Slimmable video codec (SlimVC)

RD performance

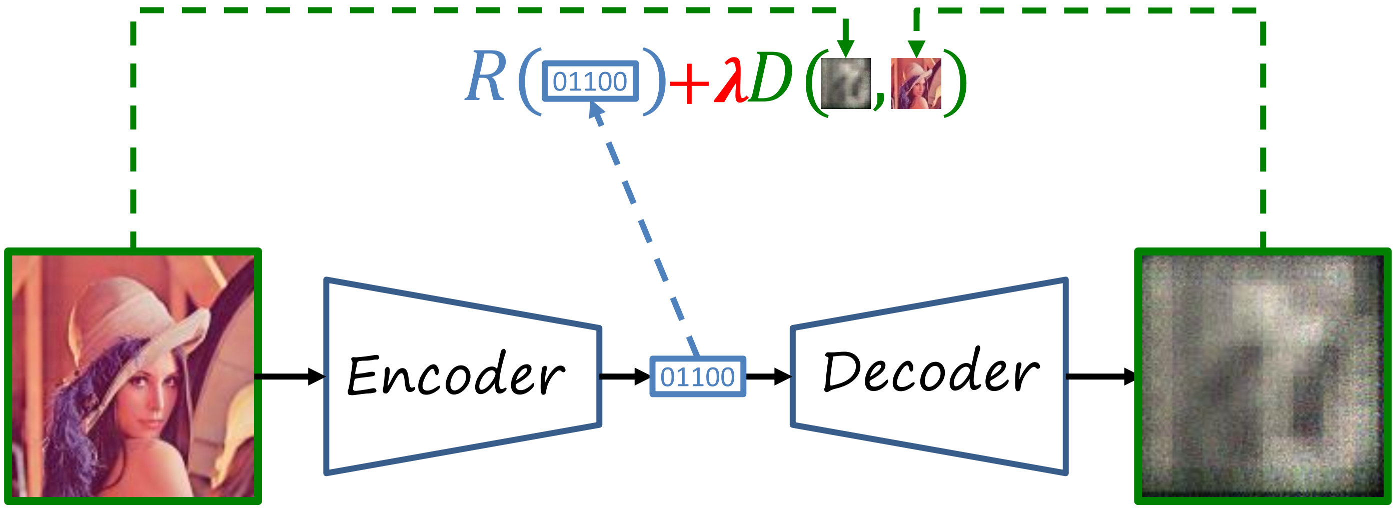


Computational cost (GFLOPs)



Slimmable video codec, CLIC 2022 at CVPR 2022

Is neural image compression practical?



Practical neural image compression?
- Minimize rate ✓
- Minimize distortion ✓
- Variable rate ✗
- Low memory ✗
- Low computation ✗
- Low latency ✗

MAE [SPL2020]

SlimCAE [CVPR2021]

Limitations
- λ is fixed
- Heavy encoders/decoders

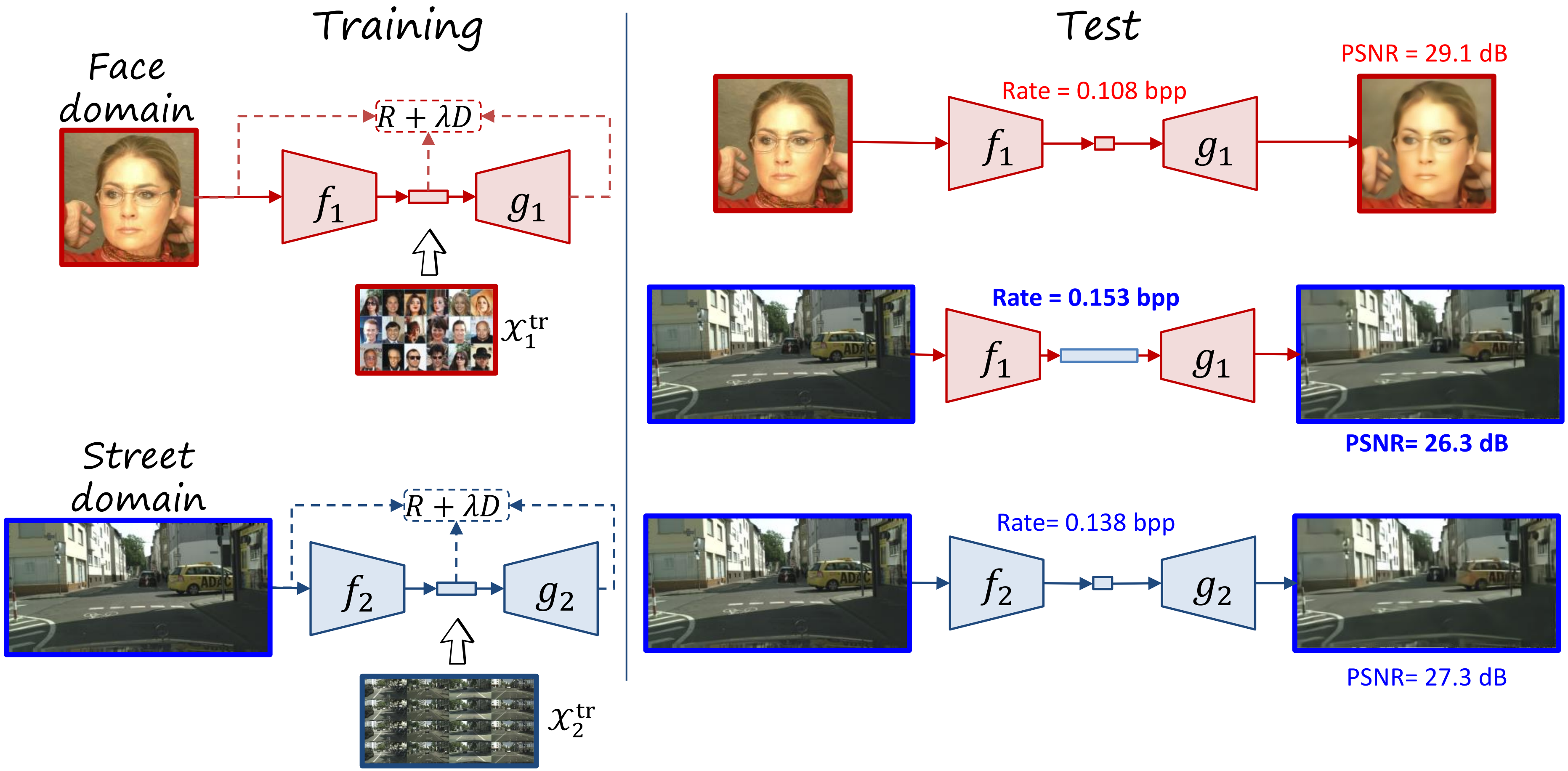
Other practical considerations
- **Domain-specific codecs** (e.g. videoconference, screencast)
- **Back./forw. compatibility** (with legacy encoders/decoders)

DANICE [CLIC2021]

[SPL2020] Variable Rate Deep Image Compression with Modulated Autoencoder, Signal Processing Letters 2020
[CVPR2021] Slimmable compressive autoencoders for practical imaga compression, CVPR 2021
[CLIC2021] DANICE: Domain adaptation without forgetting in neural image compression, CLIC 2021 at CVPR 2021

Rate-distortion optimality of learned codecs

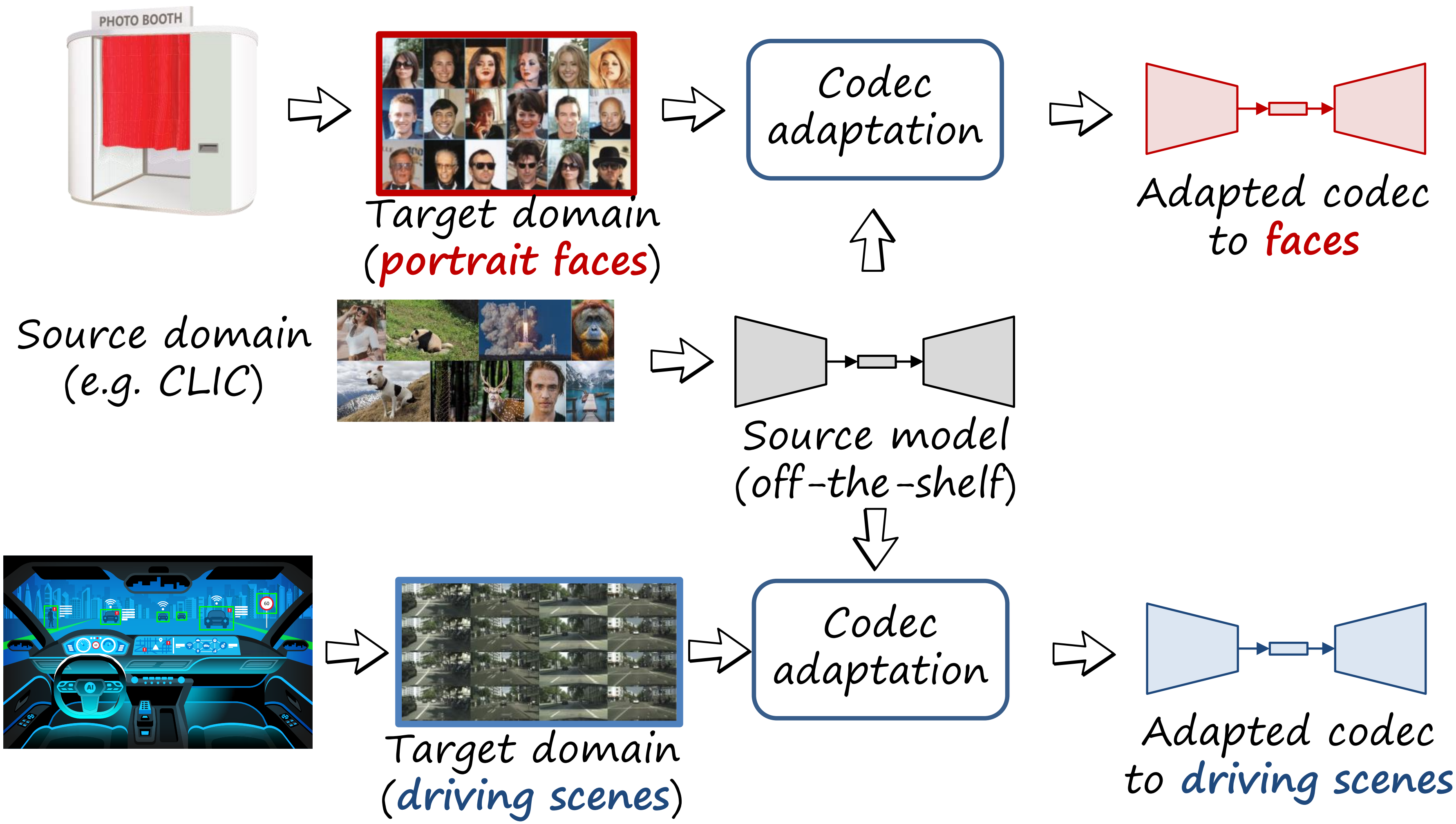
Learned codecs are only optimal in the domain of the training data



DANICE: Domain adaptation without forgetting in neural image compression, CLIC 2021 at CVPR 2021

Domain Adaptation in Neural Image ComprEssion (DANICE)

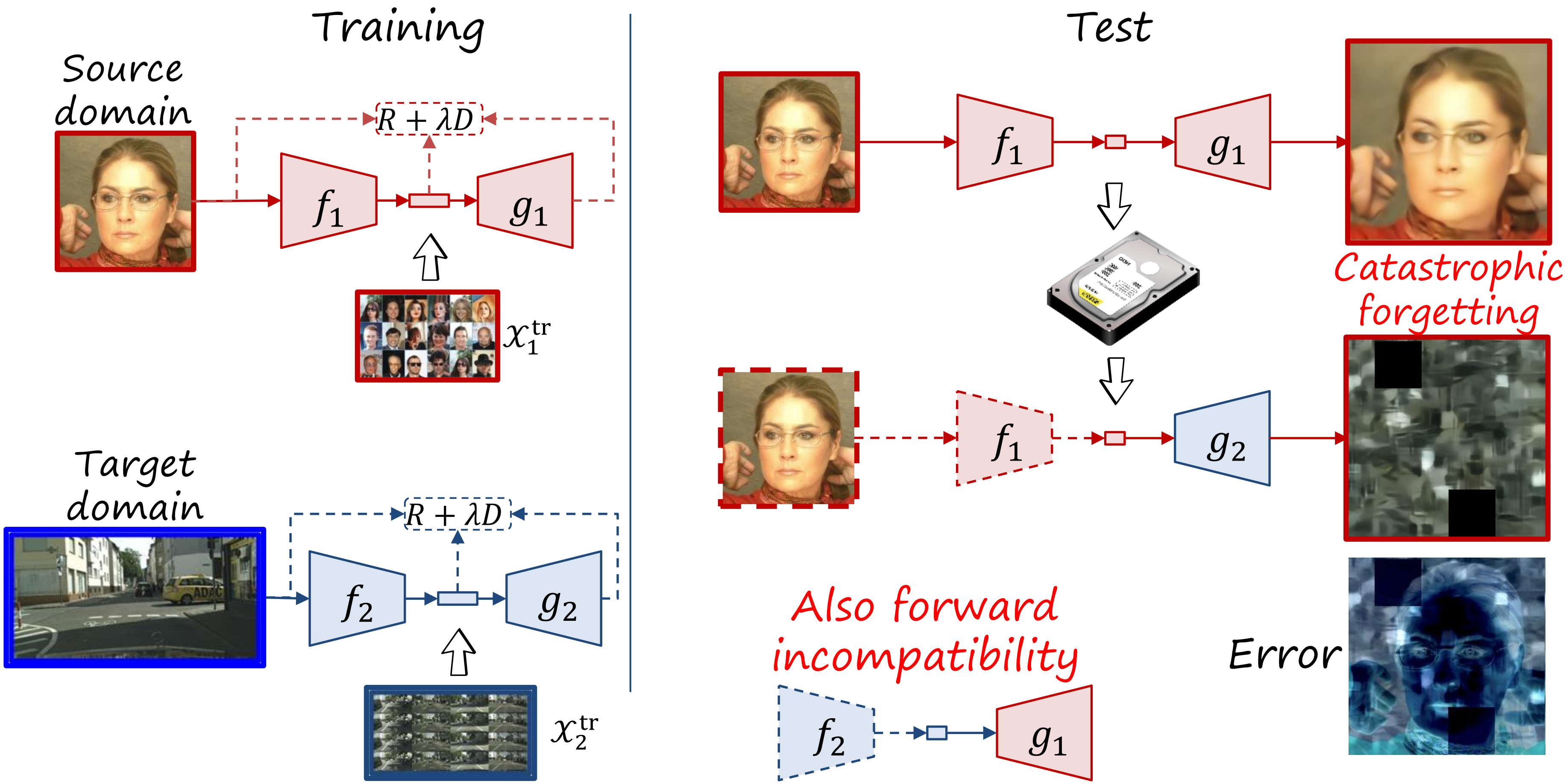
Learned codecs can be **customized with user content** to specific domains
Problem: usually **not enough custom data**; training is **expensive**
Solution: **transfer pre-trained codecs**



[DANICE: Domain adaptation without forgetting in neural image compression](), CLIC 2021 at CVPR 2021

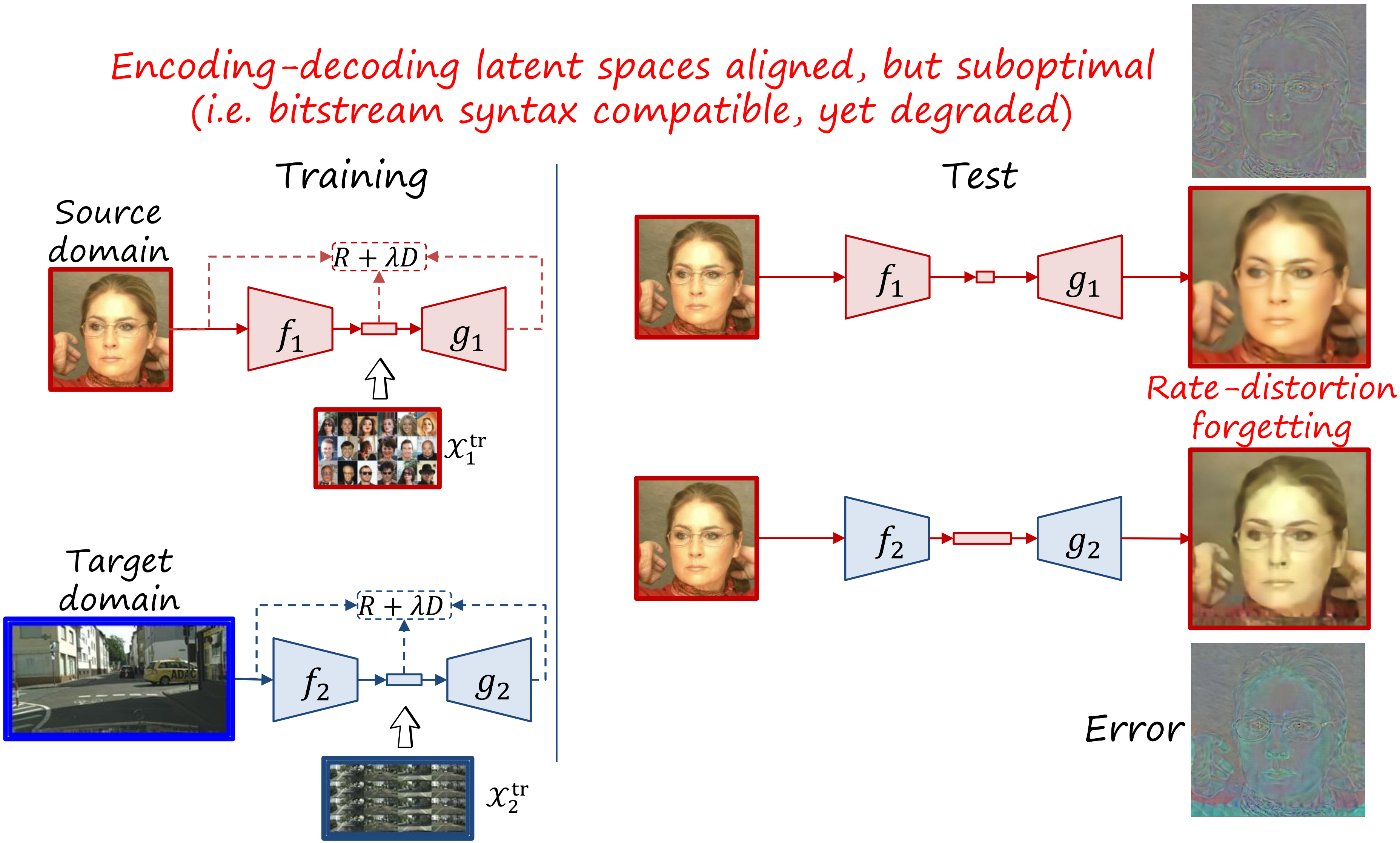
Backward incompatibility with legacy bitstreams: catastrophic forgetting

Misalignment between encoding-decoding latent spaces (i.e. bitstream syntax incompatible)



DANICE: Domain adaptation without forgetting in neural image compression, CLIC 2021 at CVPR 2021

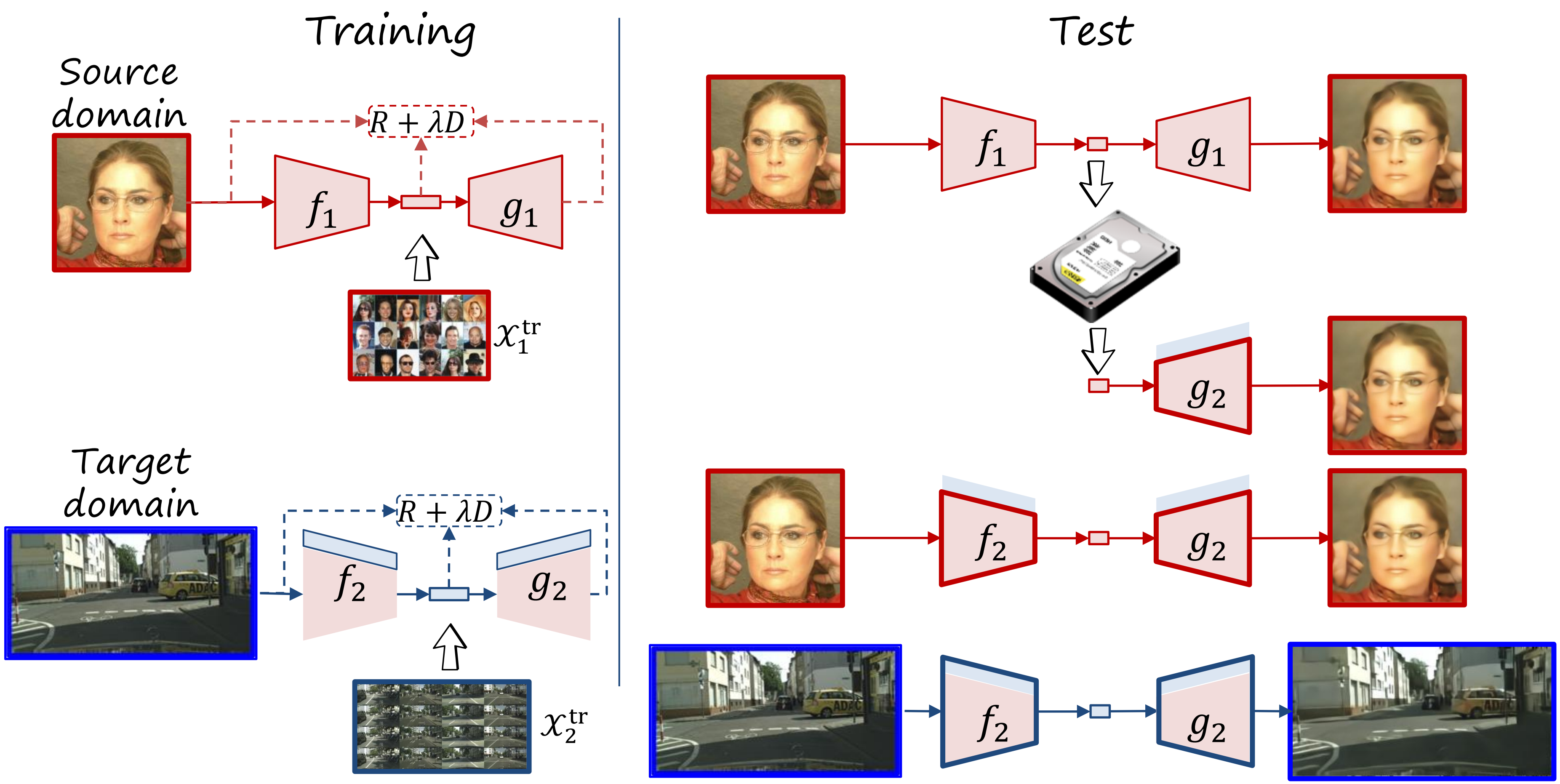
Rate-distortion forgetting



DANICE: Domain adaptation without forgetting in neural image compression, CLIC 2021 at CVPR 2021

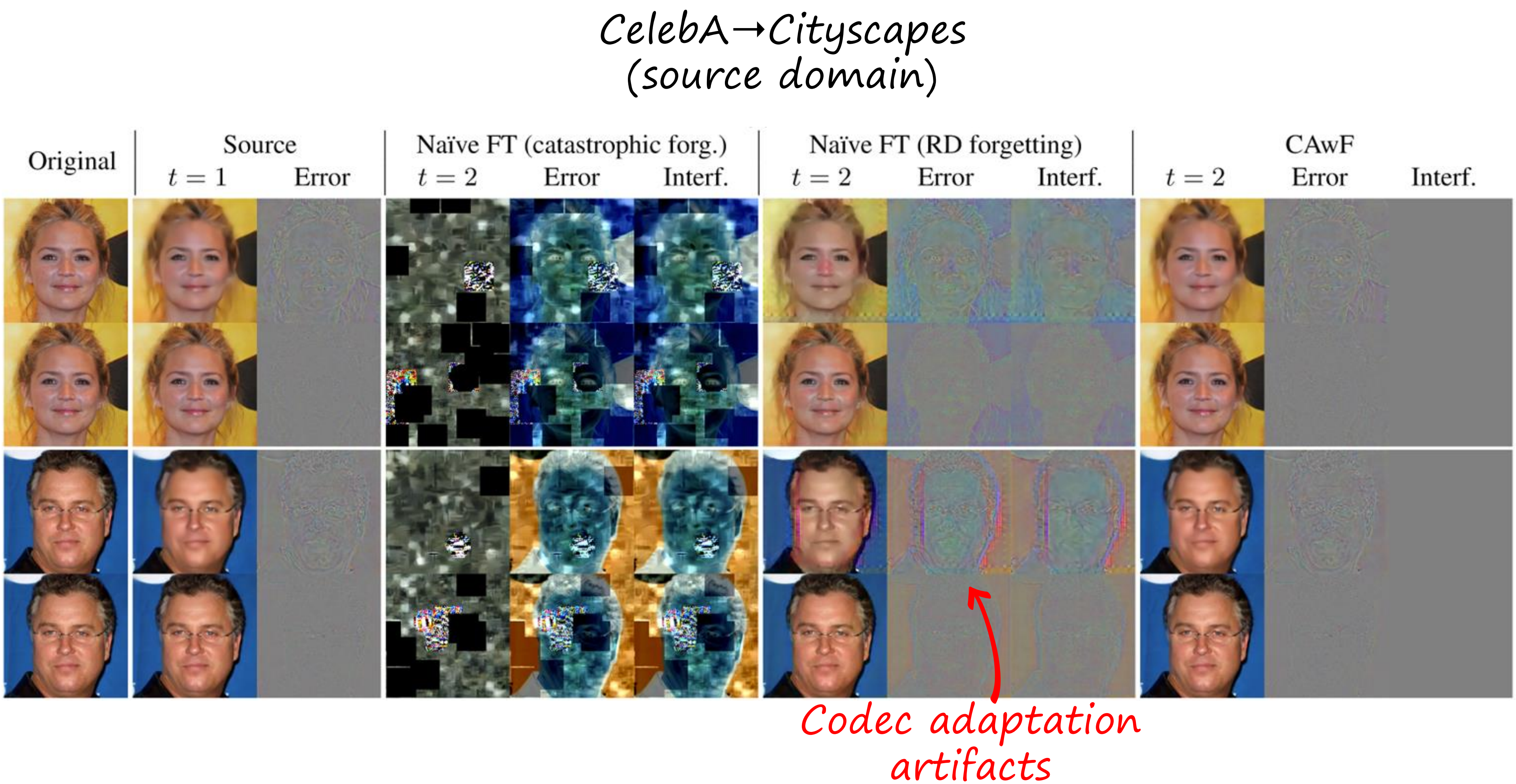
Codec adaptation without forgetting (CAwF)

Freeze source codec, and learn target codec as an enhancement layer
Drawback: adds additional parameters



DANICE: Domain adaptation without forgetting in neural image compression, CLIC 2021 at CVPR 2021

Codec adaptation without forgetting (CAwF)

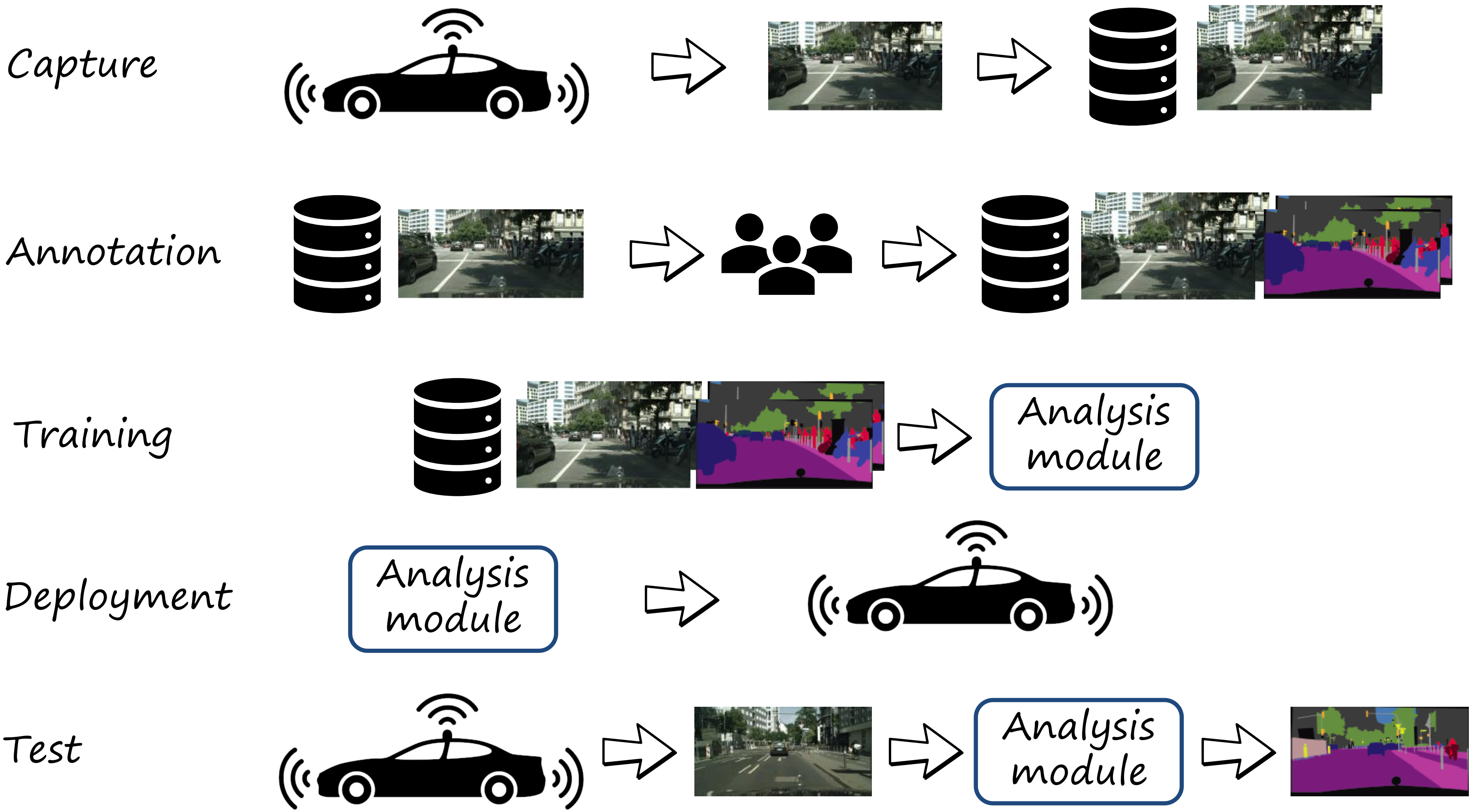


DANICE: Domain adaptation without forgetting in neural image compression, CLIC 2021 at CVPR 2021

Outline

- Neural image/video compression: a walkthrough
- Our work on neural image/video compression
  - Practical neural image compression
  - Neural image compression for machines
- Other works

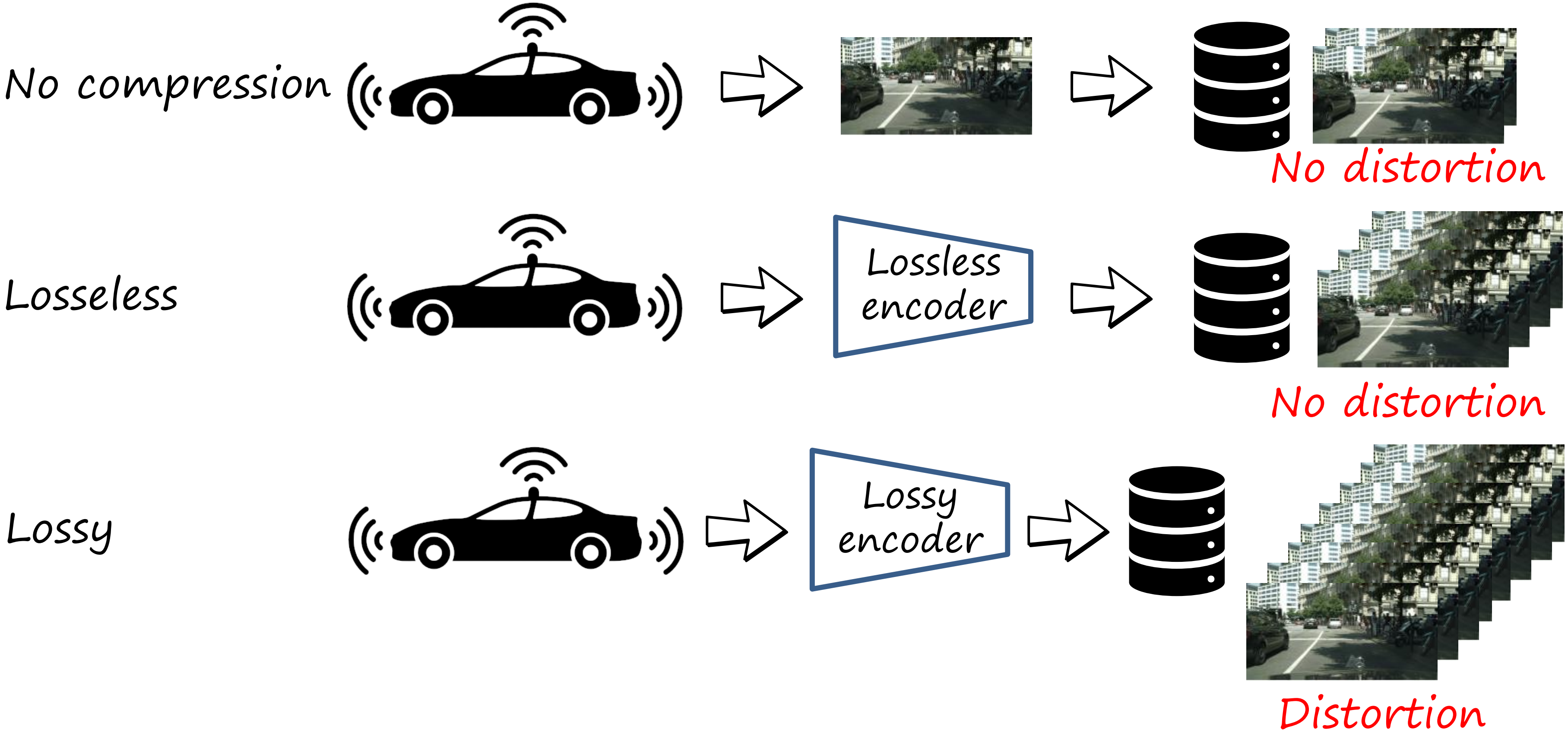
Data collection for onboard perception



The more images, the better model (in principle)

Distributed Learning and Inference with Compressed Images, IEEE Trans. Image Processing 2021

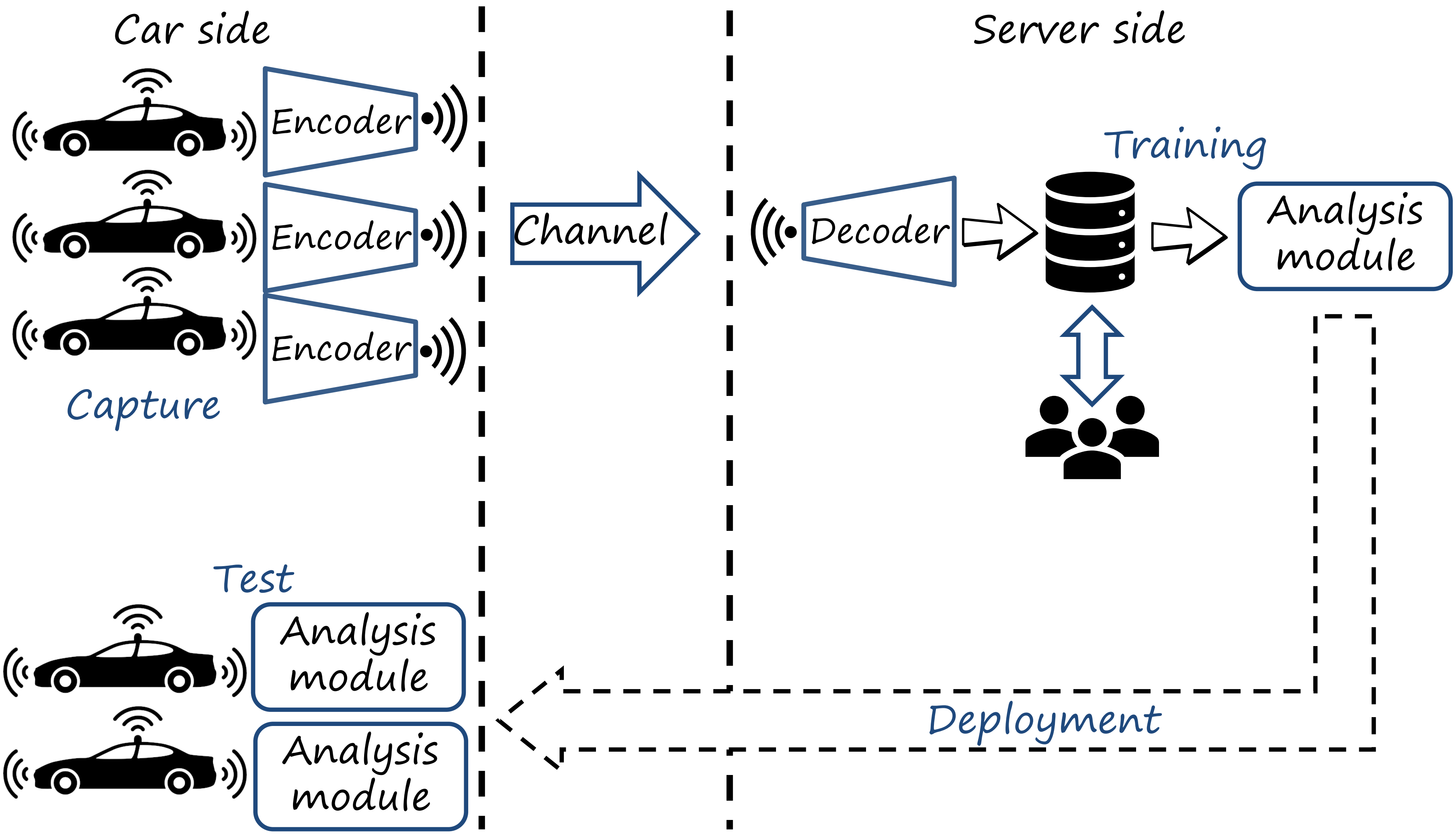
Data collection for onboard perception



The higher the compression rate the more images we can collect

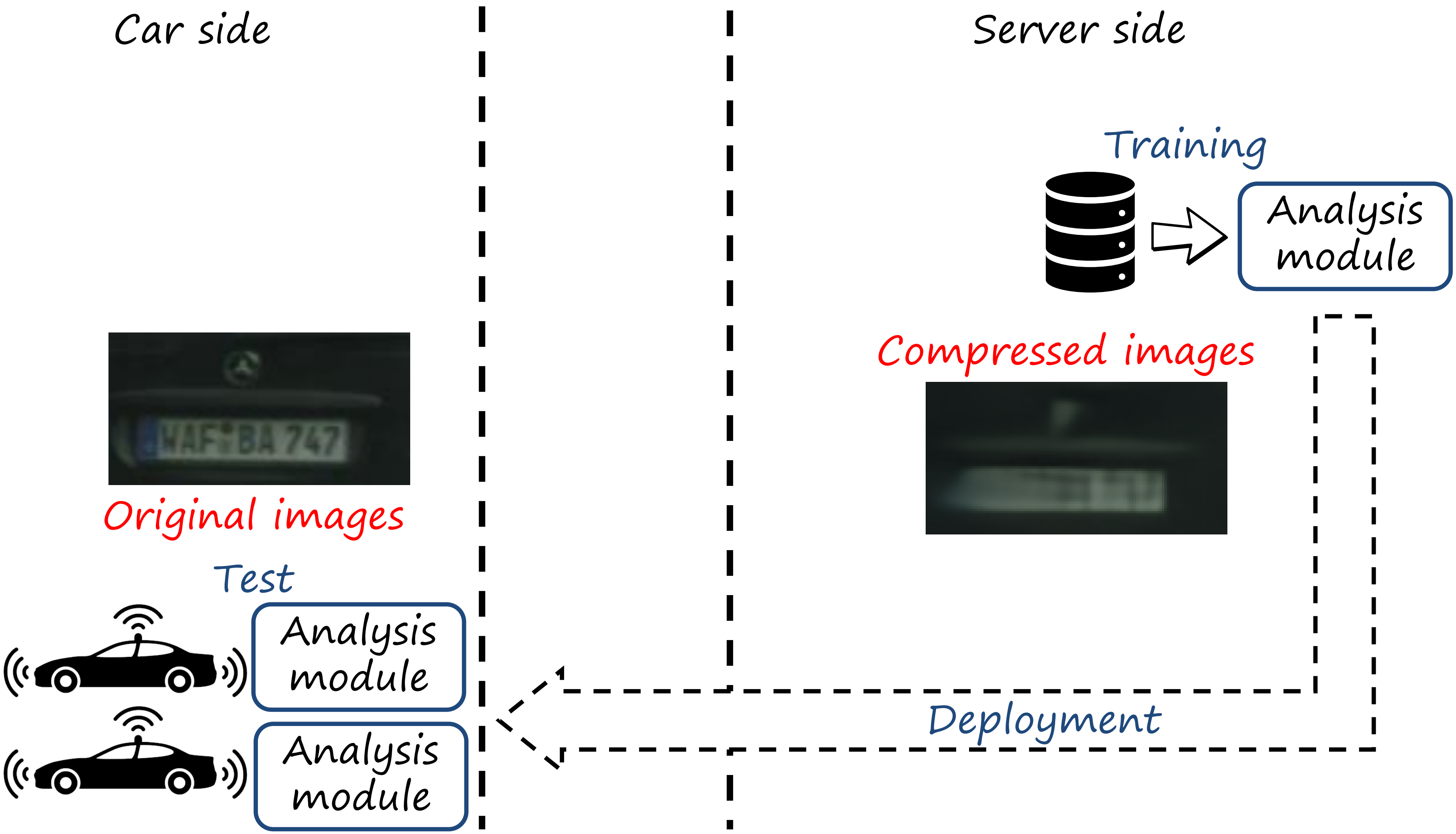
[Distributed Learning and Inference with Compressed Images](), IEEE Trans. Image Processing 2021

Distributed data collection



Distributed Learning and Inference with Compressed Images, IEEE Trans. Image Processing 2021

Distributed data collection



[Distributed Learning and Inference with Compressed Images](), IEEE Trans. Image Processing 2021

Training images vs test images

Training (compressed)

Test (original)

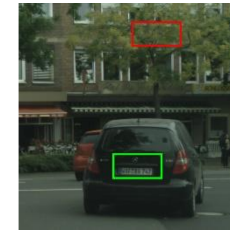
codec: mean-scale hyperprior

Distributed Learning and Inference with Compressed Images, IEEE Trans. Image Processing 2021

Training images vs test images

Training (compressed)

Test (original)



Distributed Learning and Inference with Compressed Images, IEEE Trans. Image Processing 2021

Training images vs test images

Training (compressed) Test (original)

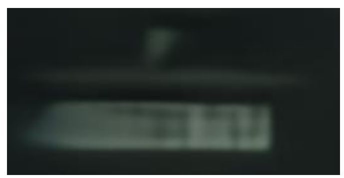
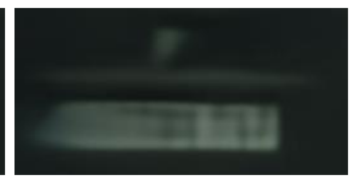
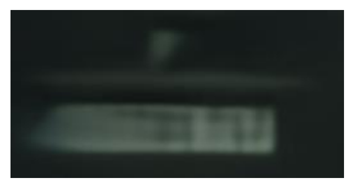
Configuration CO: compressed/original

Observation 1: training and test distributions are different (**covariate shift**)

Observation 2: training images have less information than test images (**loss of information**)

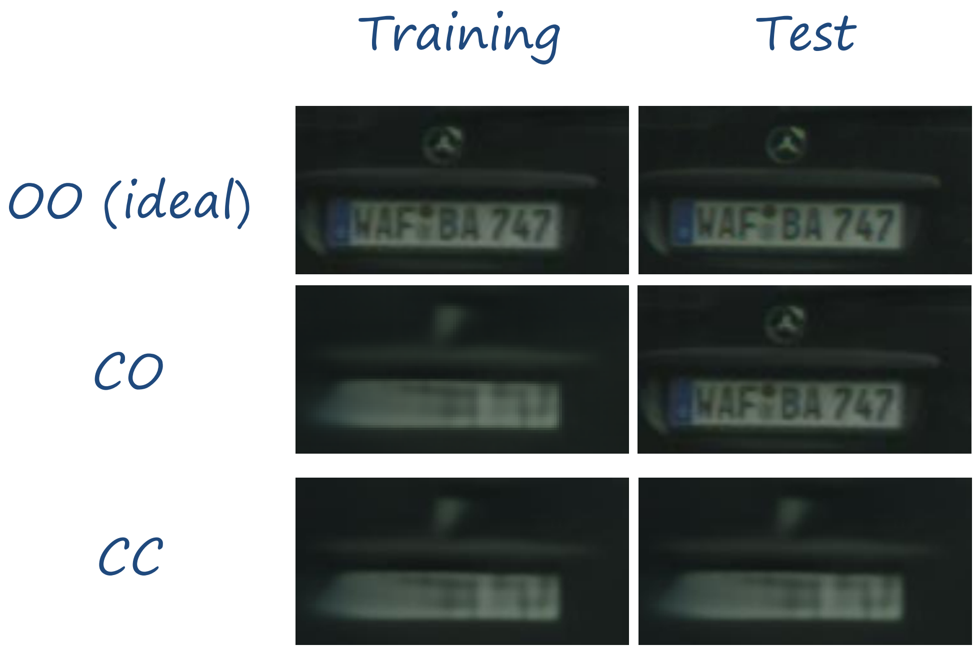
[Distributed Learning and Inference with Compressed Images](), IEEE Trans. Image Processing 2021

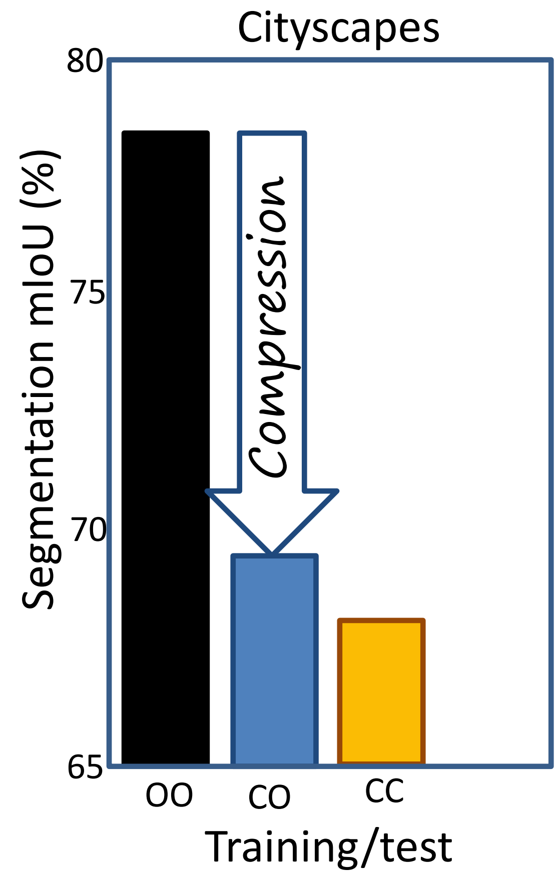
Training/test configurations

	Training	Test	Covariate shift	Information loss (training/test)
OO (ideal) original/original			No	No/No
CO compressed/ original			Yes	Yes/No
CC compressed/compressed			No	Yes/Yes
OC original/compressed			Yes	No/Yes

Distributed Learning and Inference with Compressed Images, IEEE Trans. Image Processing 2021

Effect on downstream task

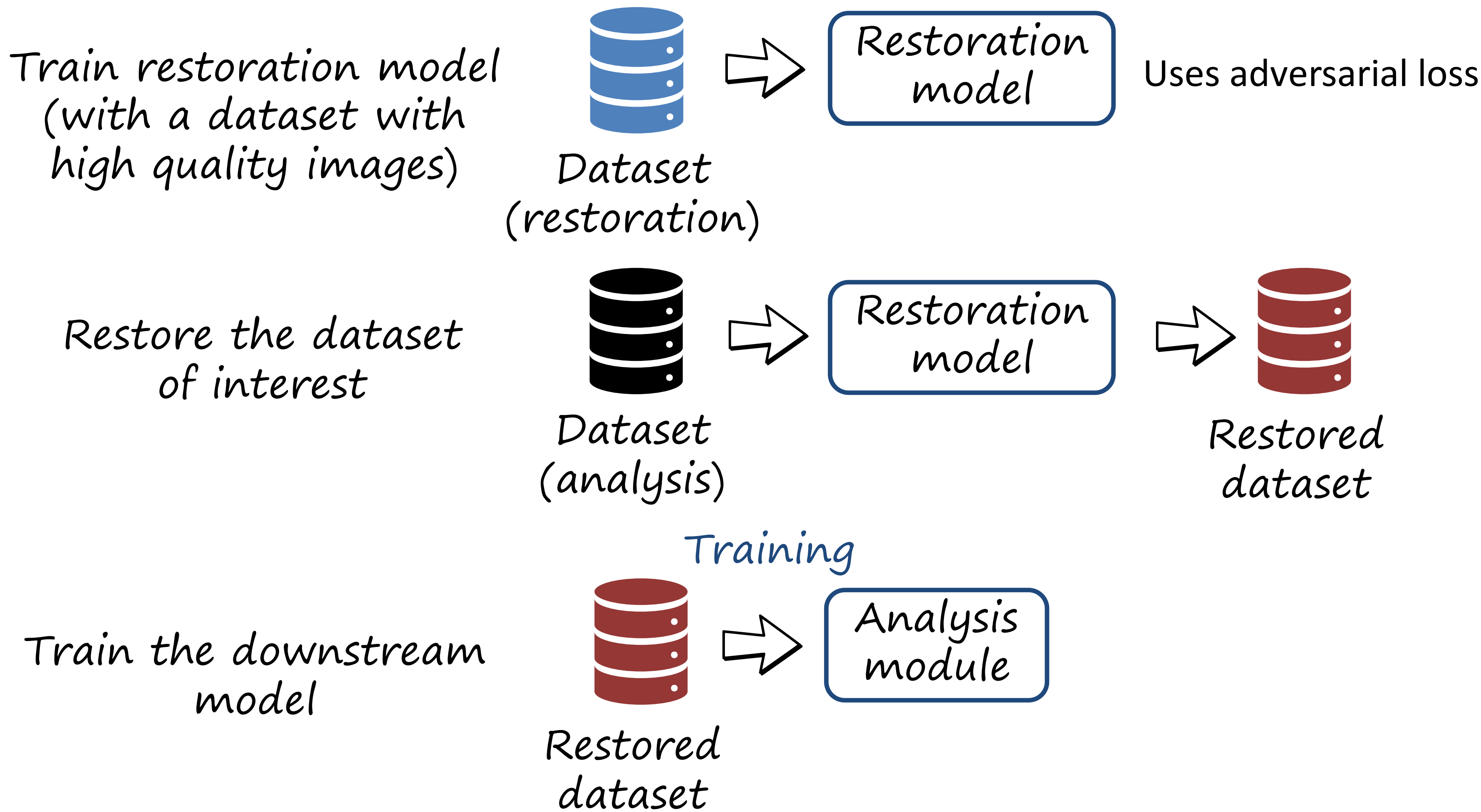




Conclusion (this dataset): better to keep more information in test than reduce the covariate shift

[Distributed Learning and Inference with Compressed Images](), IEEE Trans. Image Processing 2021

Proposed approach: dataset restoration

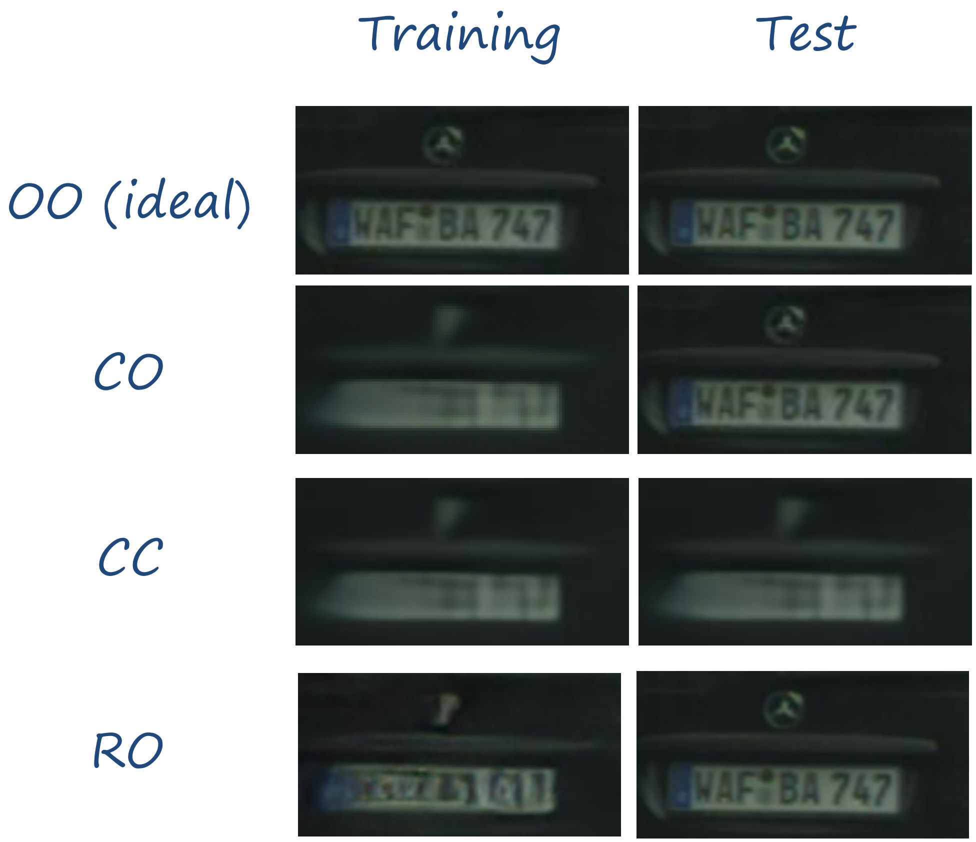


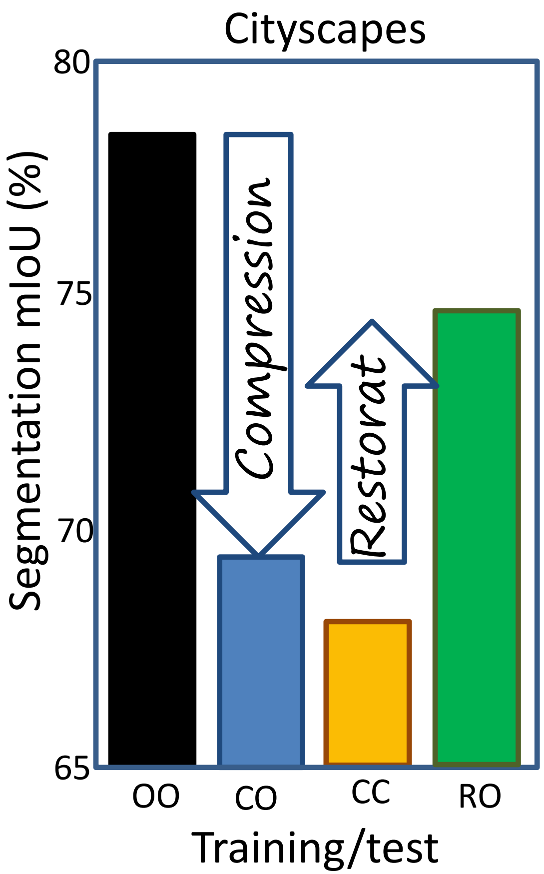
Distributed Learning and Inference with Compressed Images, IEEE Trans. Image Processing 2021

Training images vs test images

Distributed Learning and Inference with Compressed Images, IEEE Trans. Image Processing 2021

Effect on downstream task



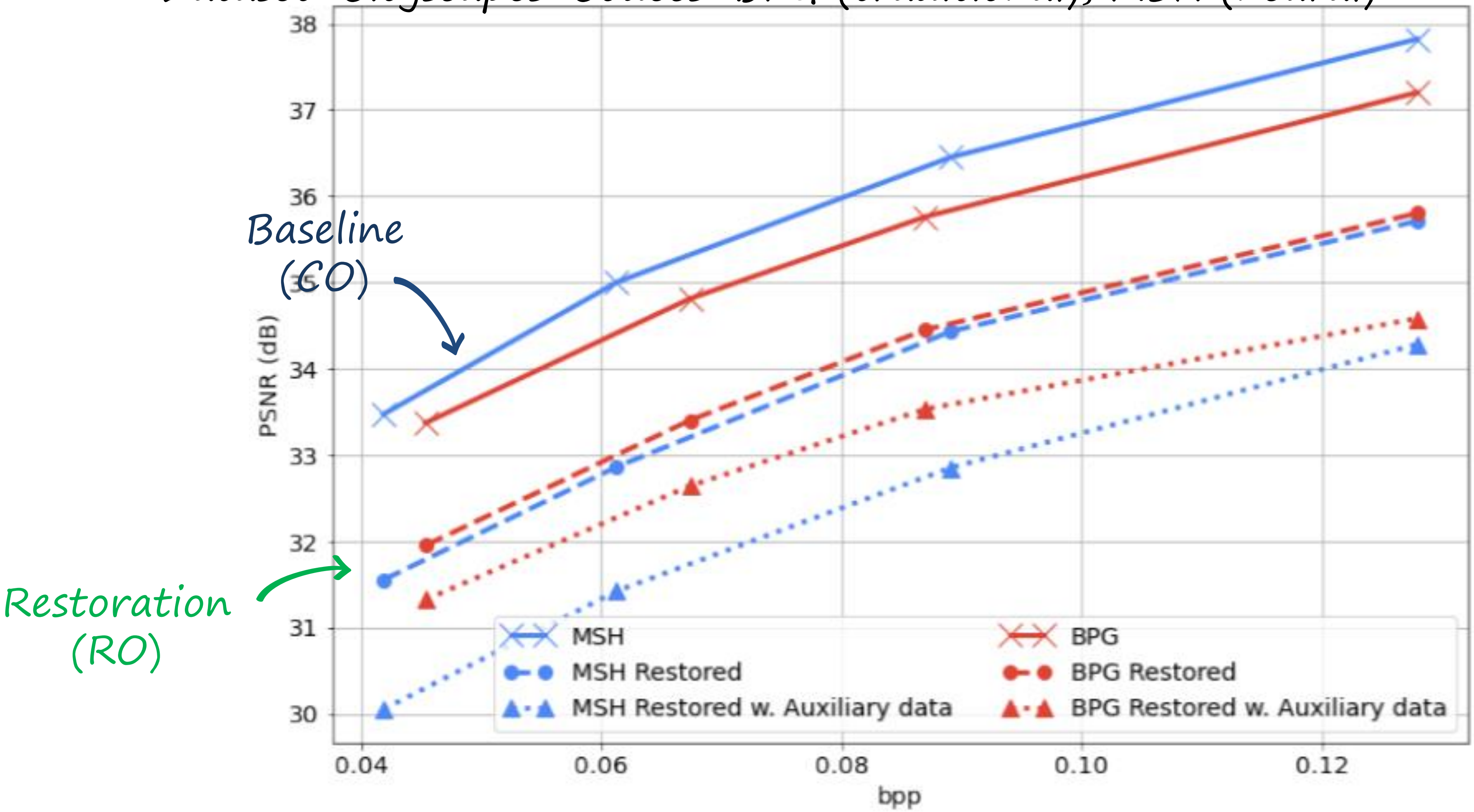


Why does it work?
- Alleviates the covariate shift
- Keeps useful information for segmentation (e.g. texture)

[Distributed Learning and Inference with Compressed Images](), IEEE Trans. Image Processing 2021

Experiments. Rate-distortion

Dataset: Cityscapes. Codecs: BPG (traditional), MSH (neural)

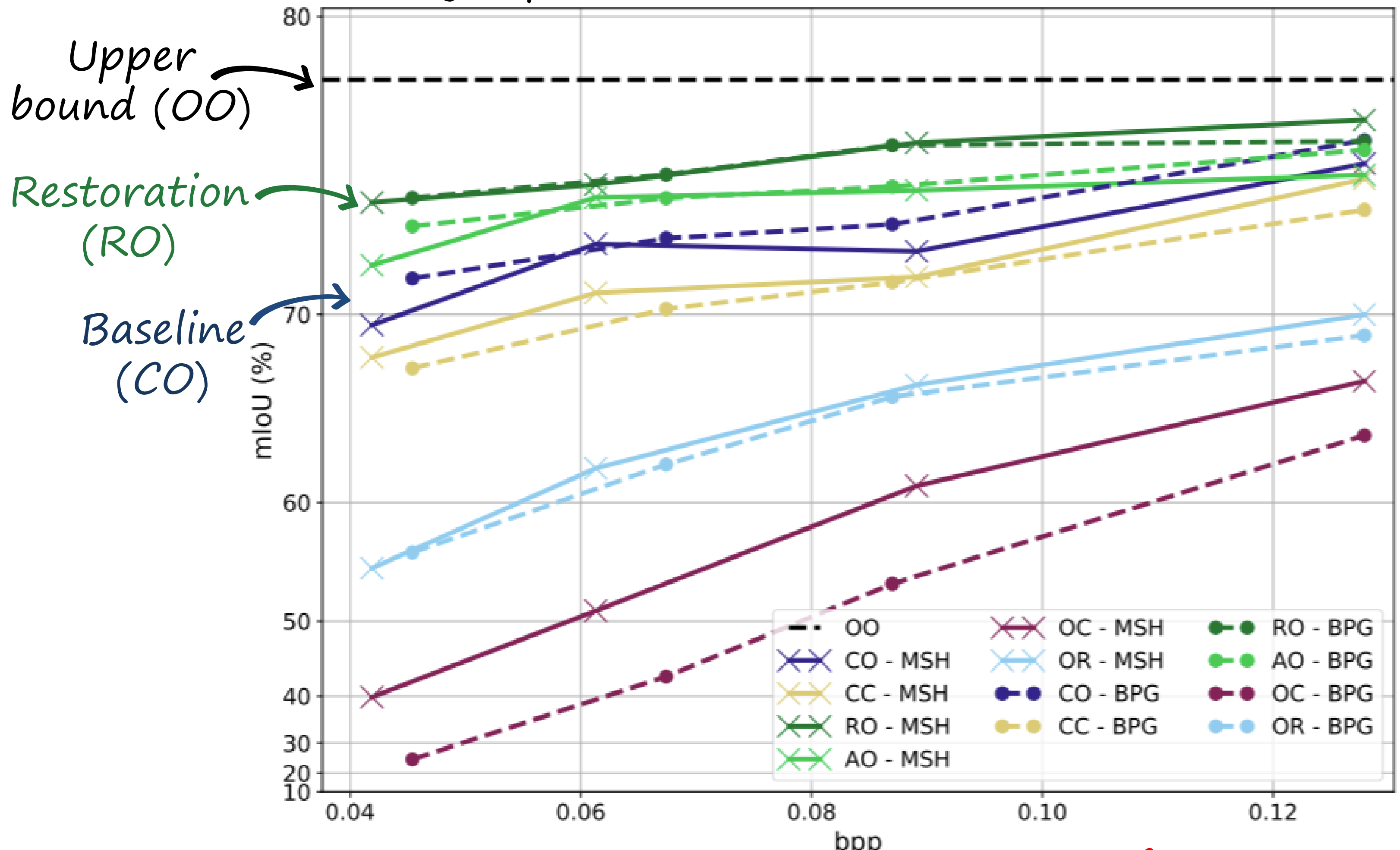


Restoration harms R-D performance

[Distributed Learning and Inference with Compressed Images](), IEEE Trans. Image Processing 2021

Experiments. Segmentation

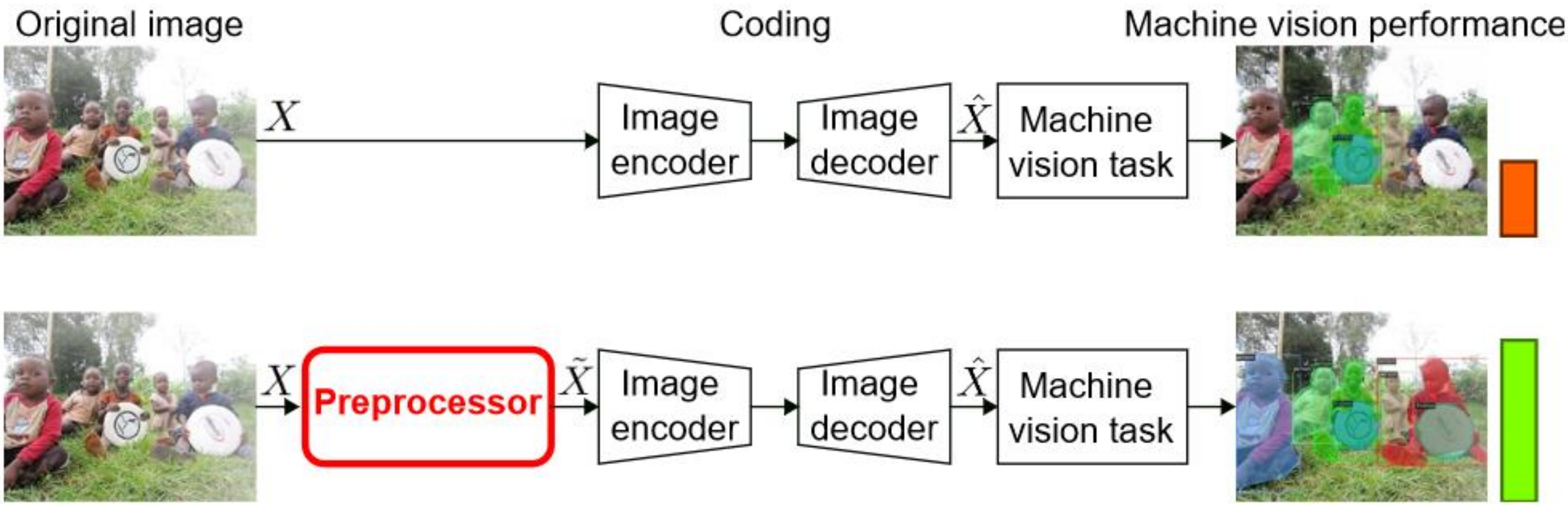
Dataset: Cityscapes. Codecs: BPG (traditional), MSH (neural)



Restoration improves segmentation performance

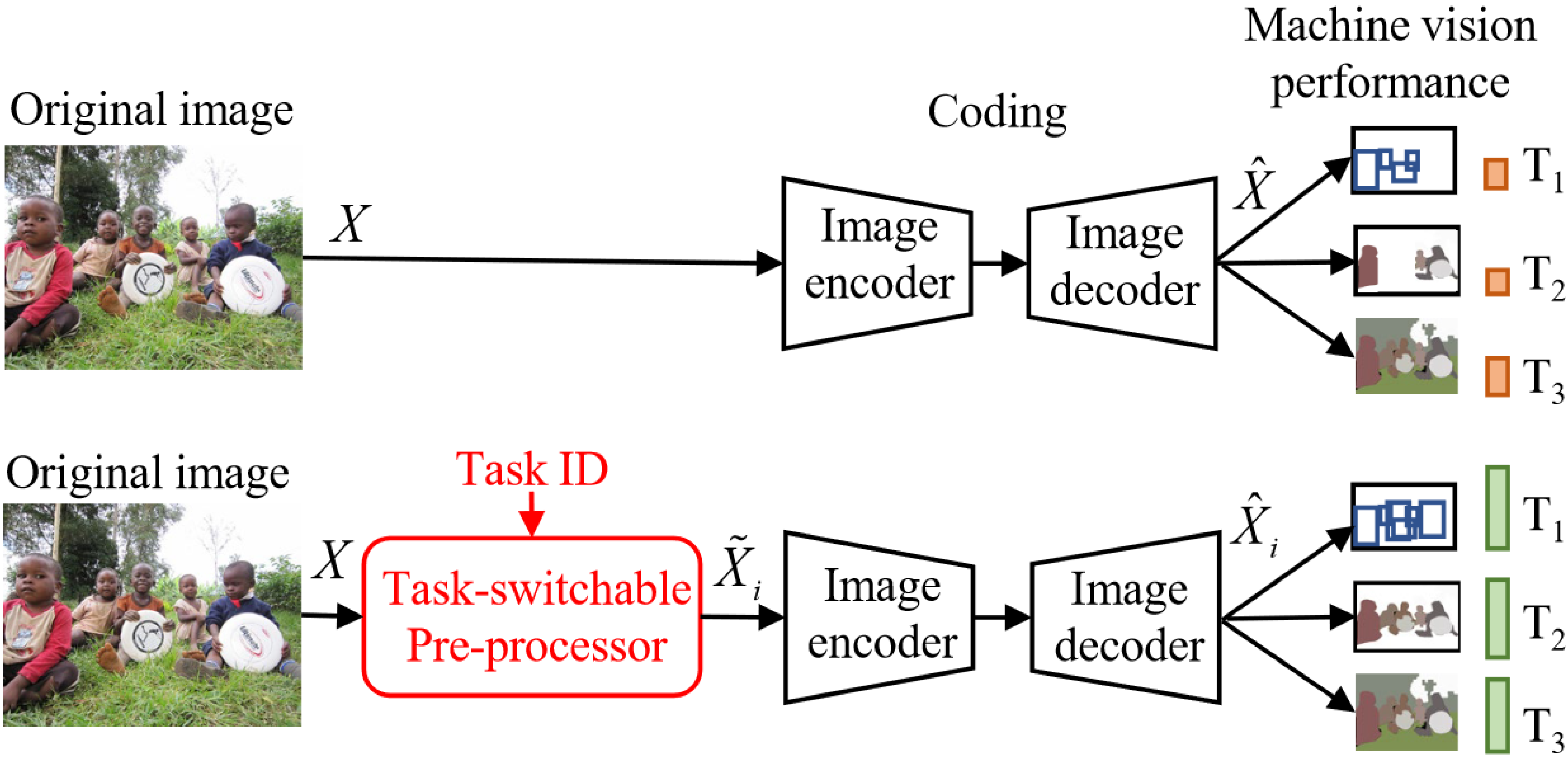
[Distributed Learning and Inference with Compressed Images](), IEEE Trans. Image Processing 2021

Semantic preprocessor for VCM

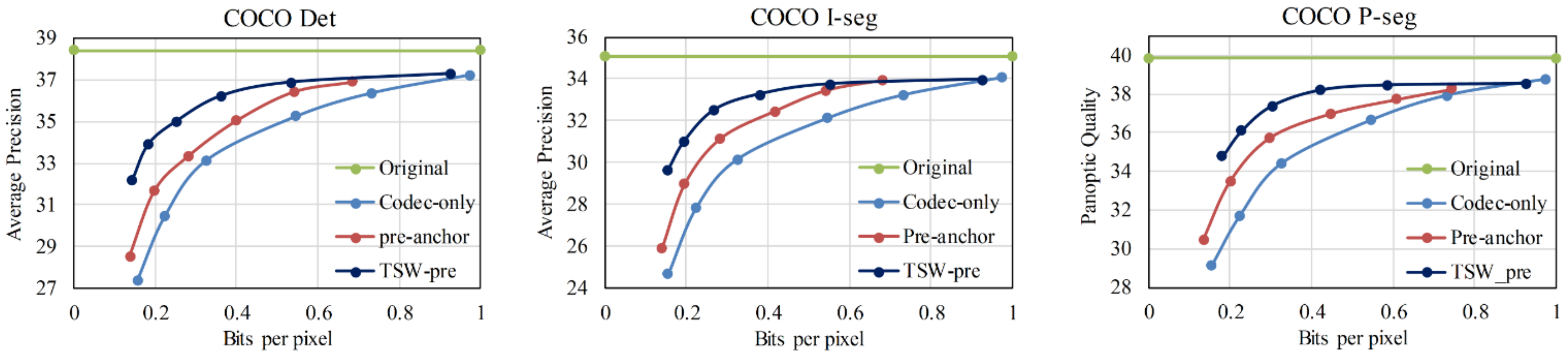


[Semantic preprocessor for image compression for machines, ICASSP 2023]

Task-switchable preprocessor for VCM



Task-switchable preprocessor for VCM



Method	COCO Dateset		
	Det	I-seg	P-seg
vs Codec-only	-48.98%	-50.07%	-50.54%
vs Pre-anchor	-36.54%	-28.75%	-35.19%

Method	BDD100k Dateset		
	Det	I-seg	P-seg
vs Codec-only	-30.30%	-74.46%	-29.47%
vs Pre-anchor	-25.37%	-60.69%	-17.37%

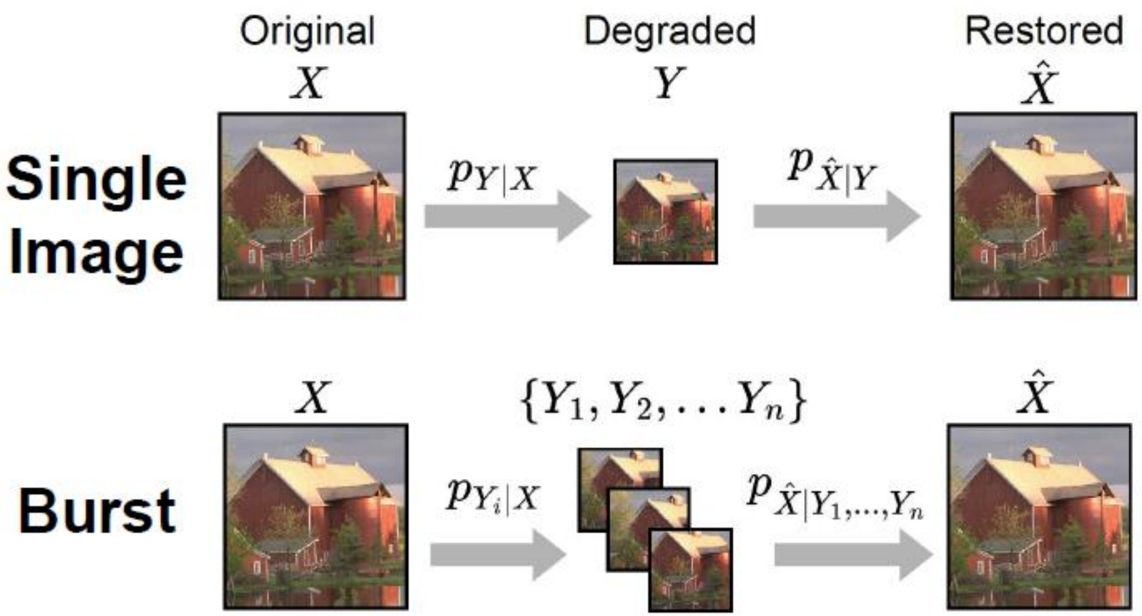
Submitted to CSVT

Outline

- Neural image/video compression: a walkthrough
- Our work on neural image/video compression
- Other works
  - Multi-image restoration
  - Semantic segmentation
  - Transfer and continual learning

Burst perception-distortion tradeoff

Scenario: burst image restoration
Motivation
- *How temporal information affects the restored image quality?*
- *How perception, distortion and their tradeoff change with multiple images*



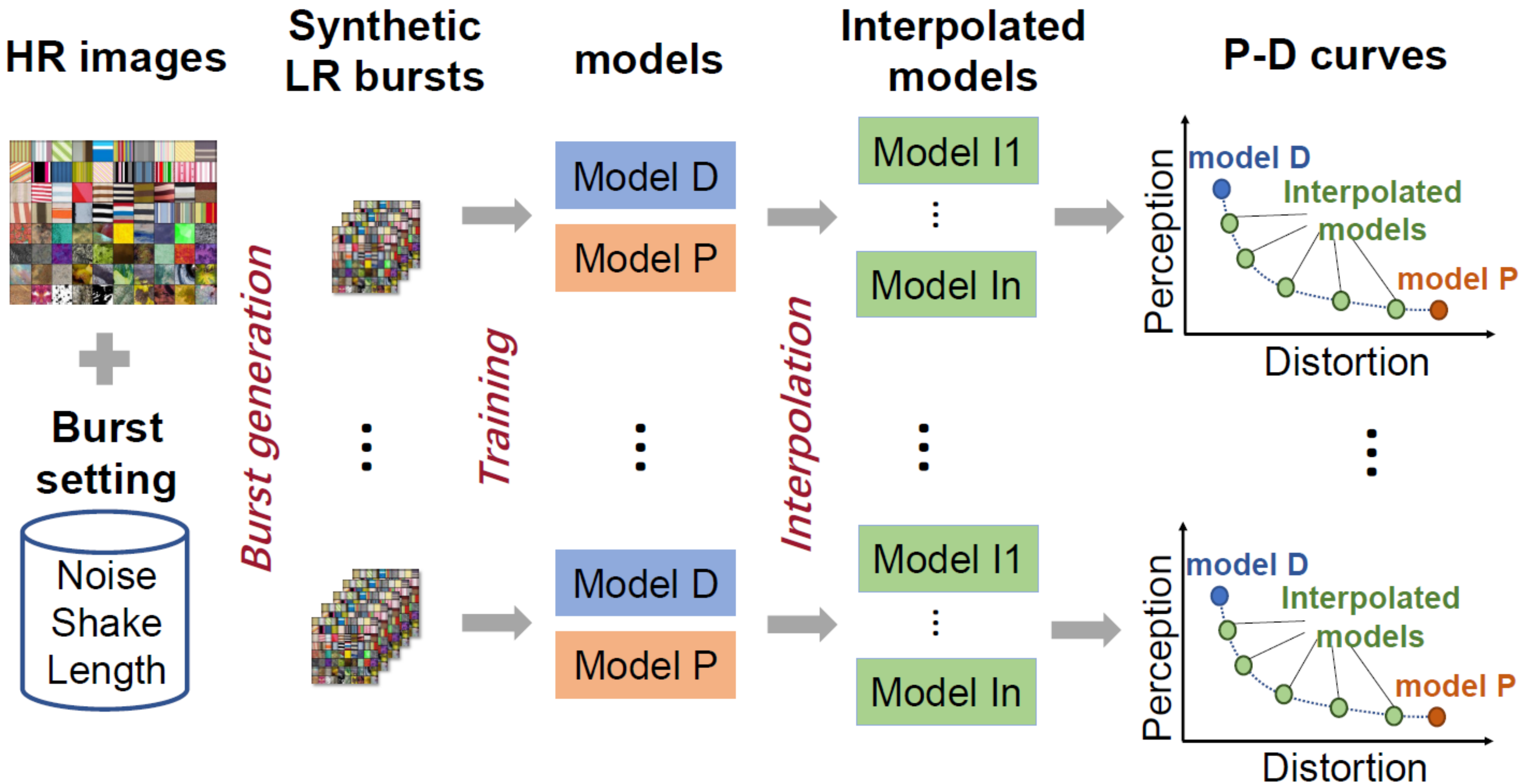
$$P(D) = \min_{p_{\hat{X}|Y}} d(p_X, p_{\hat{X}})$$
$$\text{s.t. } E[\Delta(X, \hat{X})] \le D$$

$$P(D) = \min_{\underline{p_{\hat{X}|Y_1,Y_2,\dots,Y_n}}} d(p_X, p_{\hat{X}})$$
$$\text{s.t. } E[\Delta(X, \hat{X})] \le D$$

Burst Perception-Distortion Tradeoff: Analysis and Evaluation, ICASSP 2023

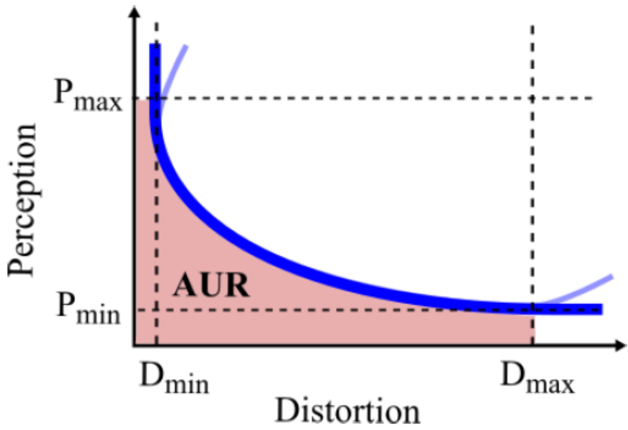
Burst perception-distortion tradeoff

Experimental setting (denoising+superresolution)

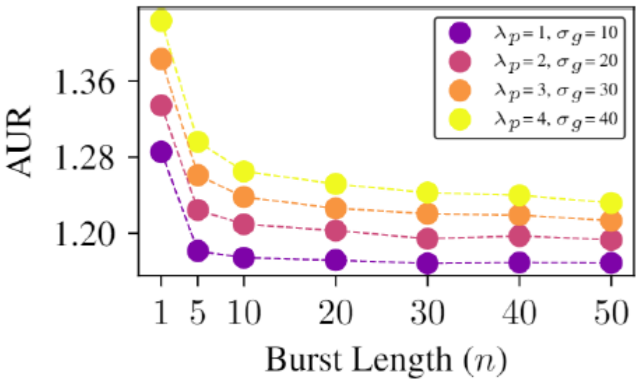


[Burst Perception-Distortion Tradeoff: Analysis and Evaluation](), ICASSP 2023

Burst perception-distortion tradeoff

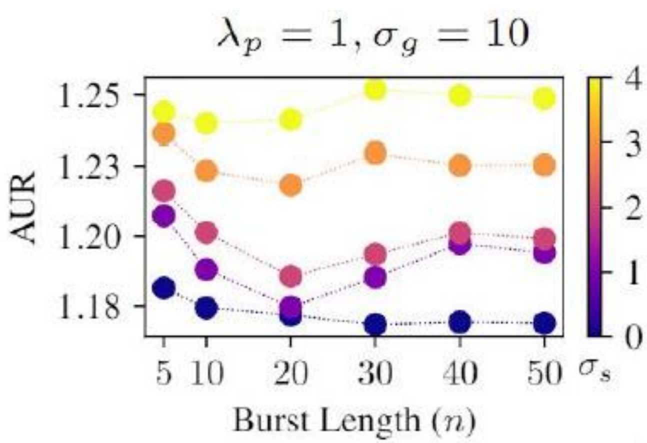


Case 1 (perfectly aligned bursts):
- *E.g. Stable shooting (no shaking or motion)*
- *E.g. Accurate flow estimation*



Perfect alignment: the more frames the better

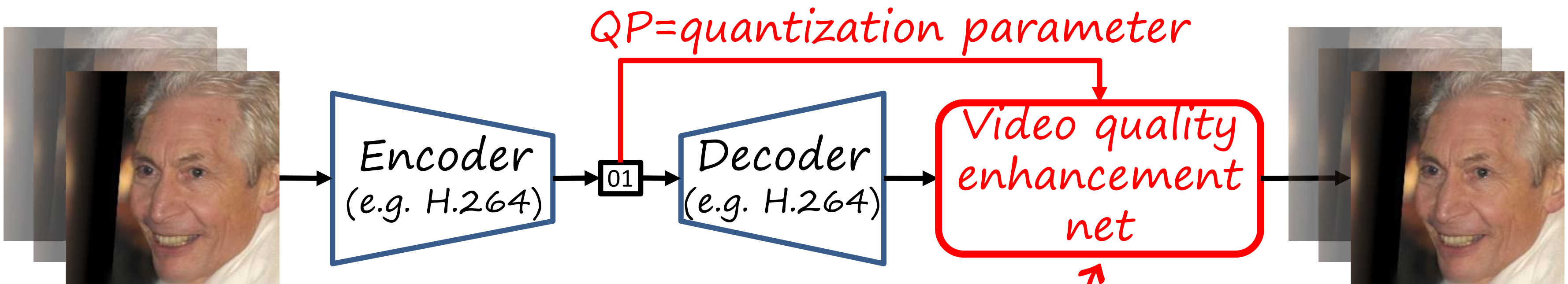
Case 2 (misaligned bursts):
- *E.g. Alignment errors, or errors in flow estimation*



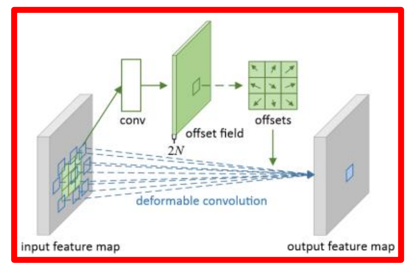
Imperfect alignment: more frames can be harmful (depending on the shake and noise levels)

Burst Perception-Distortion Tradeoff: Analysis and Evaluation, ICASSP 2023

Video quality enhancement and artifact removal



Deformable convolution



Typical approach:
- Align several frames
- Aggregate the aligned information to alleviate noise/artifacts

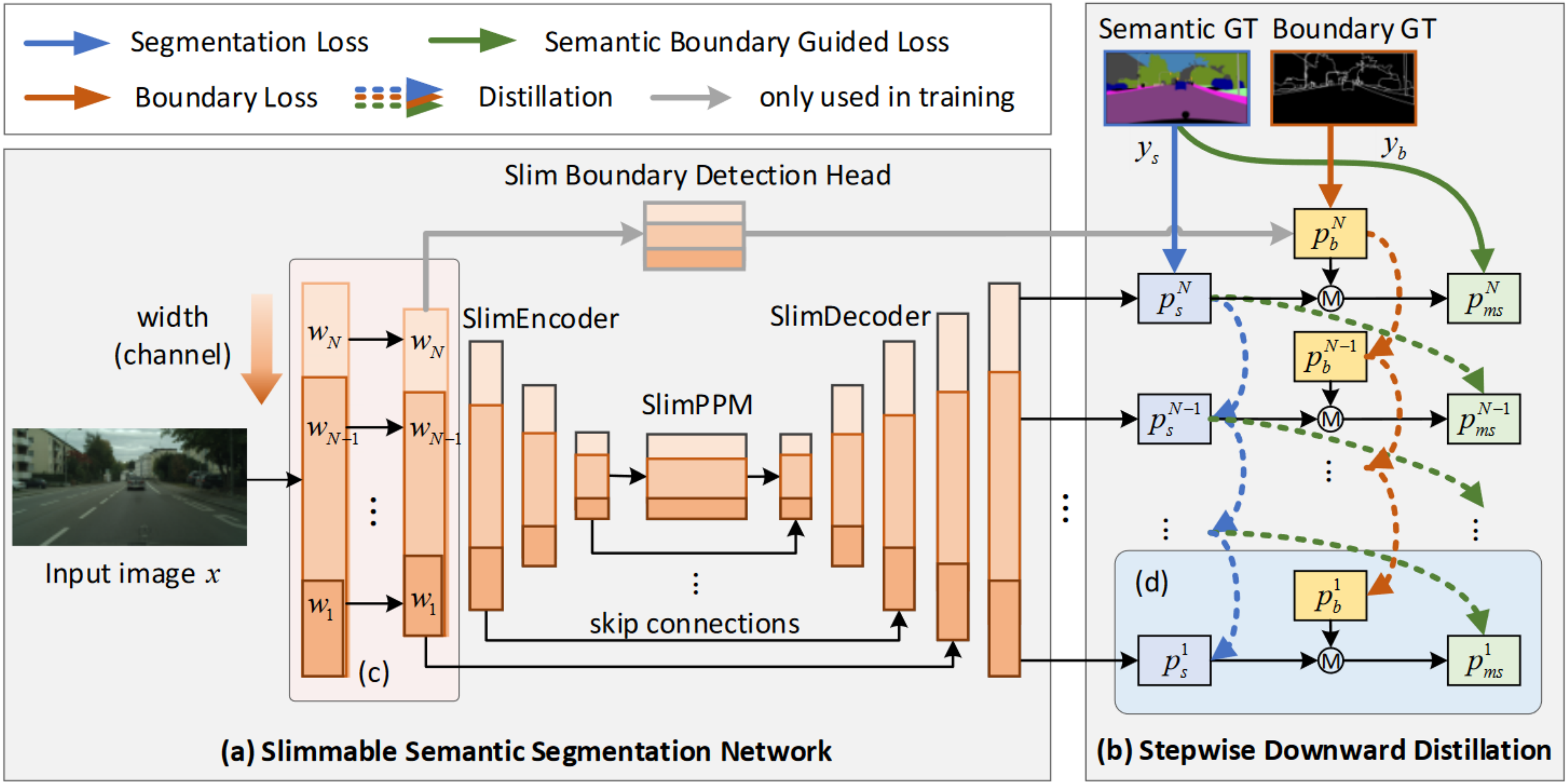
Our specific **contribution**:
- Use **deformable convolutions** for multiframe alignment
- **QP-conditional** quality enhancement network

Dai et al., Deformable Convolutional Networks, ICCV 2017
DCNGAN: A deformable convolution-based GAN with QP adaptation for perceptual quality enhancement of compressed video, ICASSP 2022

Outline

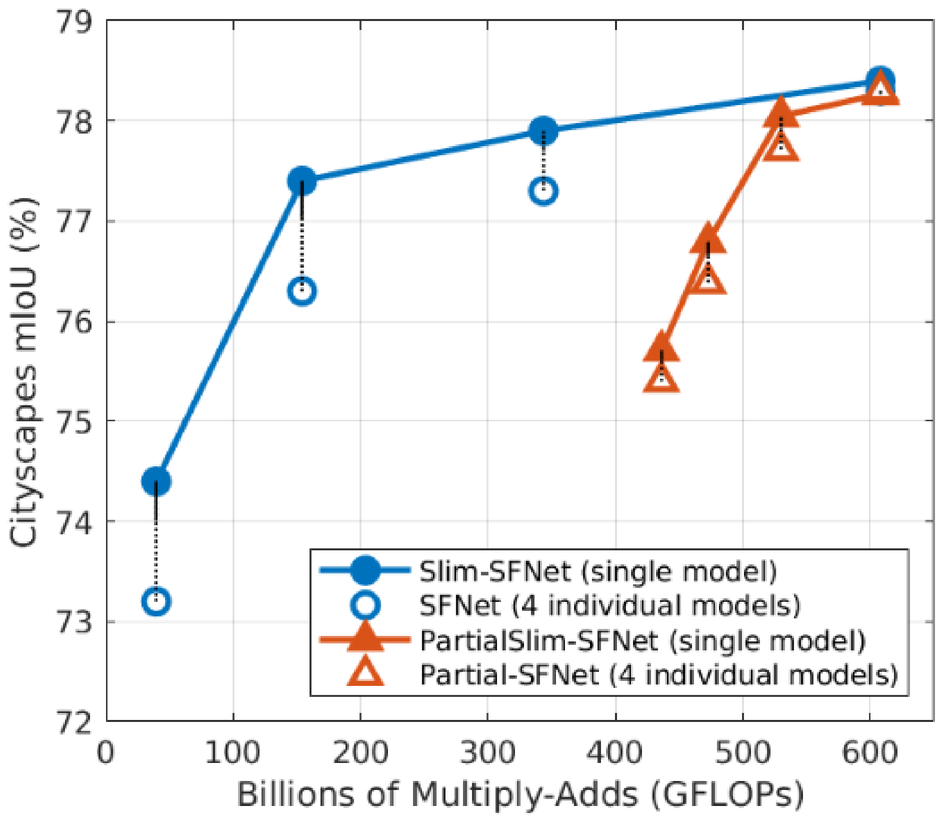
- Neural image/video compression: a walkthrough
- Our work on neural image/video compression
- Other works
  - Multi-image restoration
  - Semantic segmentation
  - Transfer and continual learning

Slimmable semantic segmentation



[SlimSeg: Slimmable Semantic Segmentation with Boundary Supervision]() ACM Multimedia 2022

Slimmable semantic segmentation



Network	Width	Independent mIoU	Independent Param	Slimmable mIoU	Slimmable Param	FLOPs
SFNet ResNet50	×1.0	78.3	31.20	**78.4** (0.1↑)	31.29	607.9
	×0.75	77.3	17.57	**77.9** (0.6↑)		343.4
	×0.5	76.3	7.82	**77.4** (1.1↑)		153.9
	×0.25	73.2	1.97	**74.4** (1.2↑)		39.4
SFNet ResNet18	×1.0	75.0	12.87	**75.6** (0.6↑)	12.89	243.4
	×0.75	74.0	7.24	**74.8** (0.8↑)		137.4
	×0.5	71.4	3.22	**72.5** (1.1↑)		61.5
	×0.25	65.5	0.79	**67.3** (1.8↑)		15.7

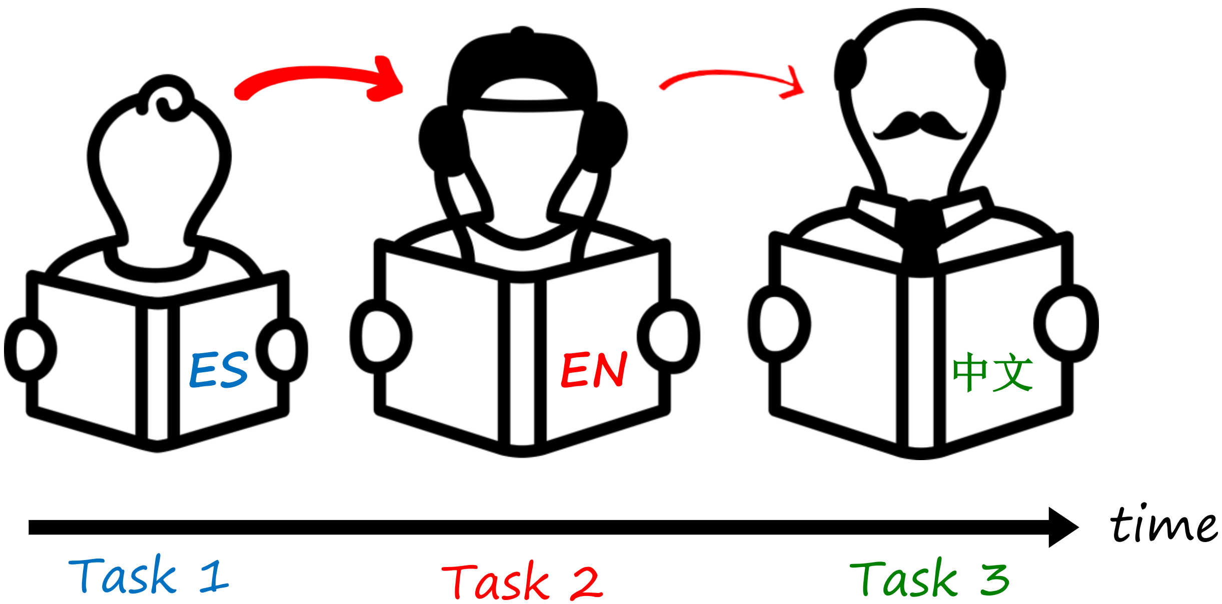
SlimSeg: Slimmable Semantic Segmentation with Boundary Supervision ACM Multimedia 2022

Outline

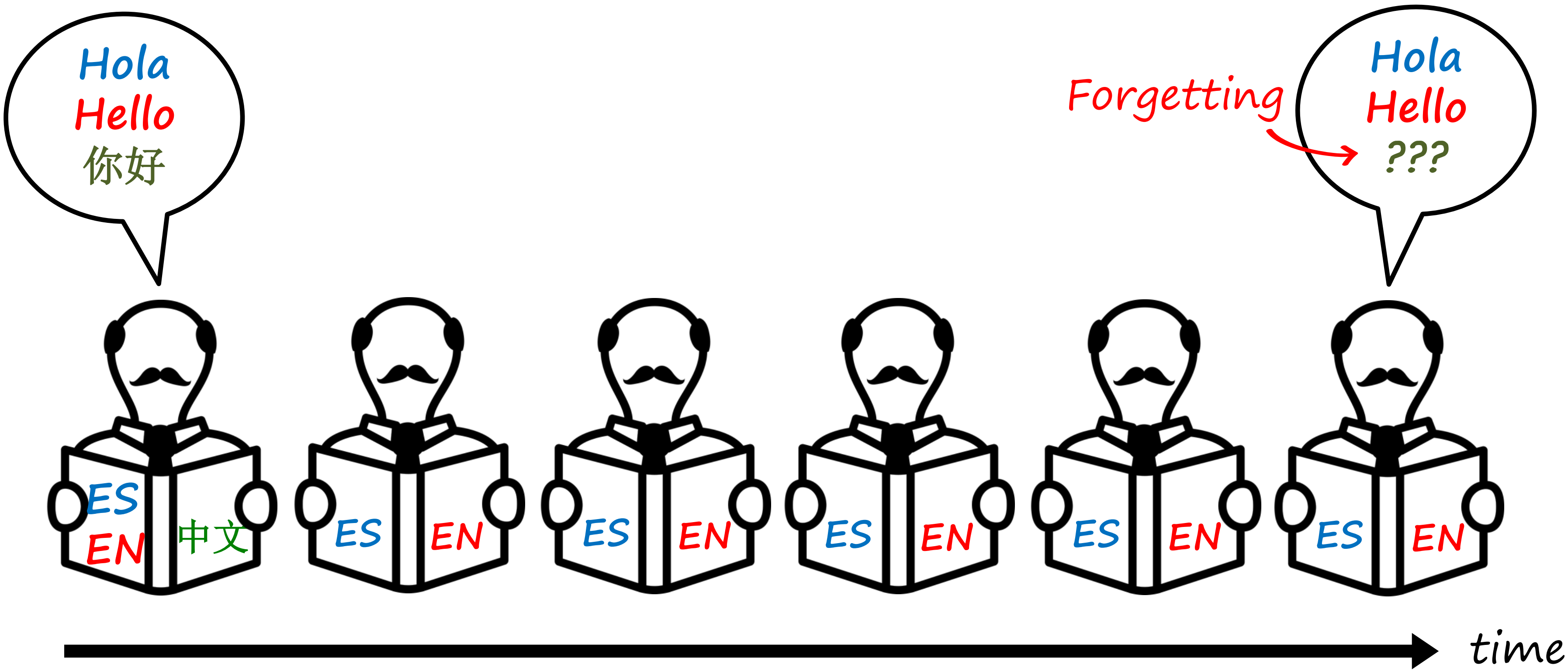
- Neural image/video compression: a walkthrough
- Our work on neural image/video compression
- Briefly: other works
  - Multi-image restoration
  - Semantic segmentation
  - Transfer learning, continual learning and domain adaptation

Continual learning in humans
(a.k.a. lifelong/sequential/incremental learning)

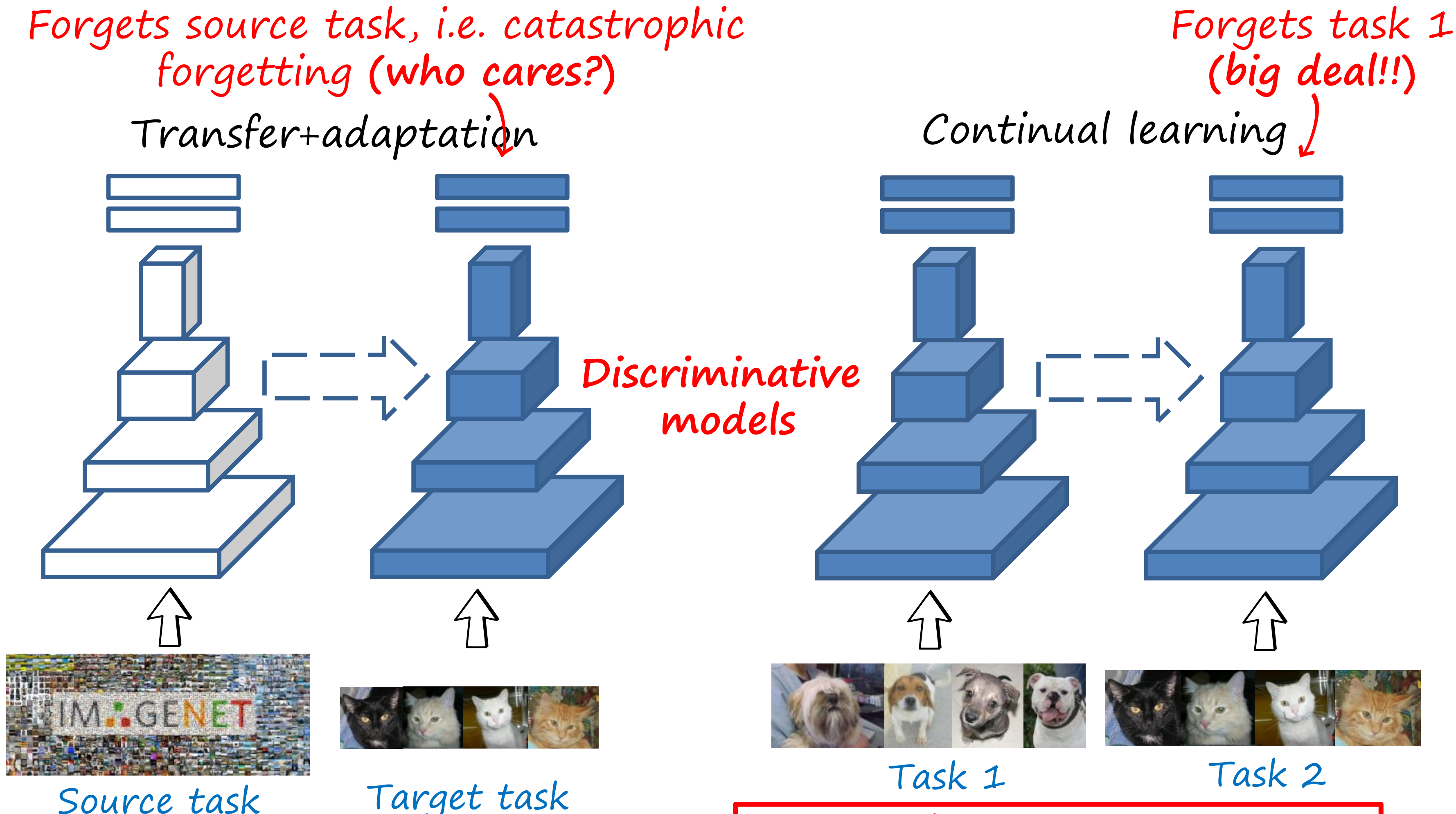
- *Reuse of past knowledge (i.e. knowledge transfer, transfer learning)*
- *Learn new skills for new tasks*



…and forgetting



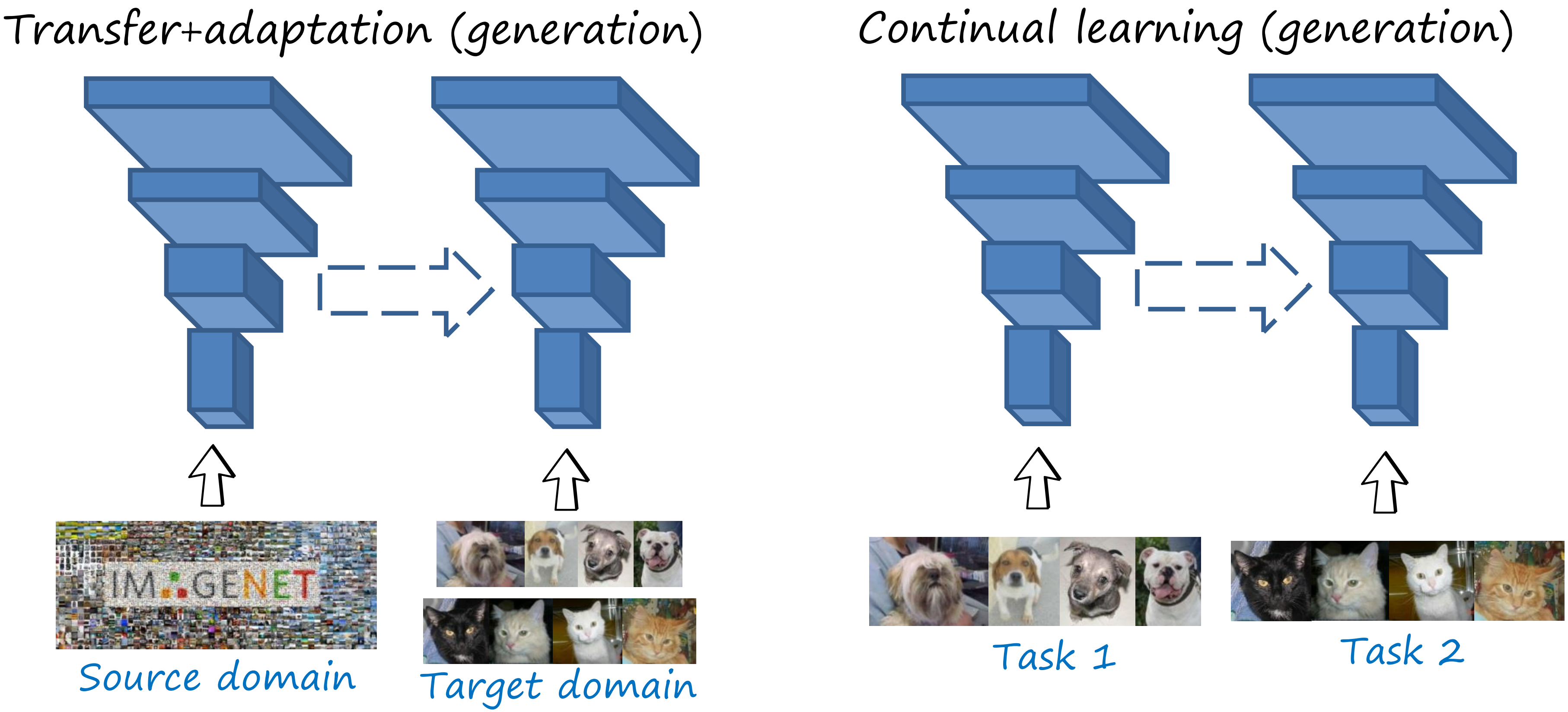
Transfer learning and continual learning



Rotated EWC (ICPR 2018), CVPR2020, …

Continual learning =
transfer learning – (catastrophic) forgetting

Transfer learning and continual learning (now with GANs for image generation)



Transferring GANs
(ECCV 2018)

Memory Replay GANs
(NeurIPS 2018)

Sequential learning for image generation

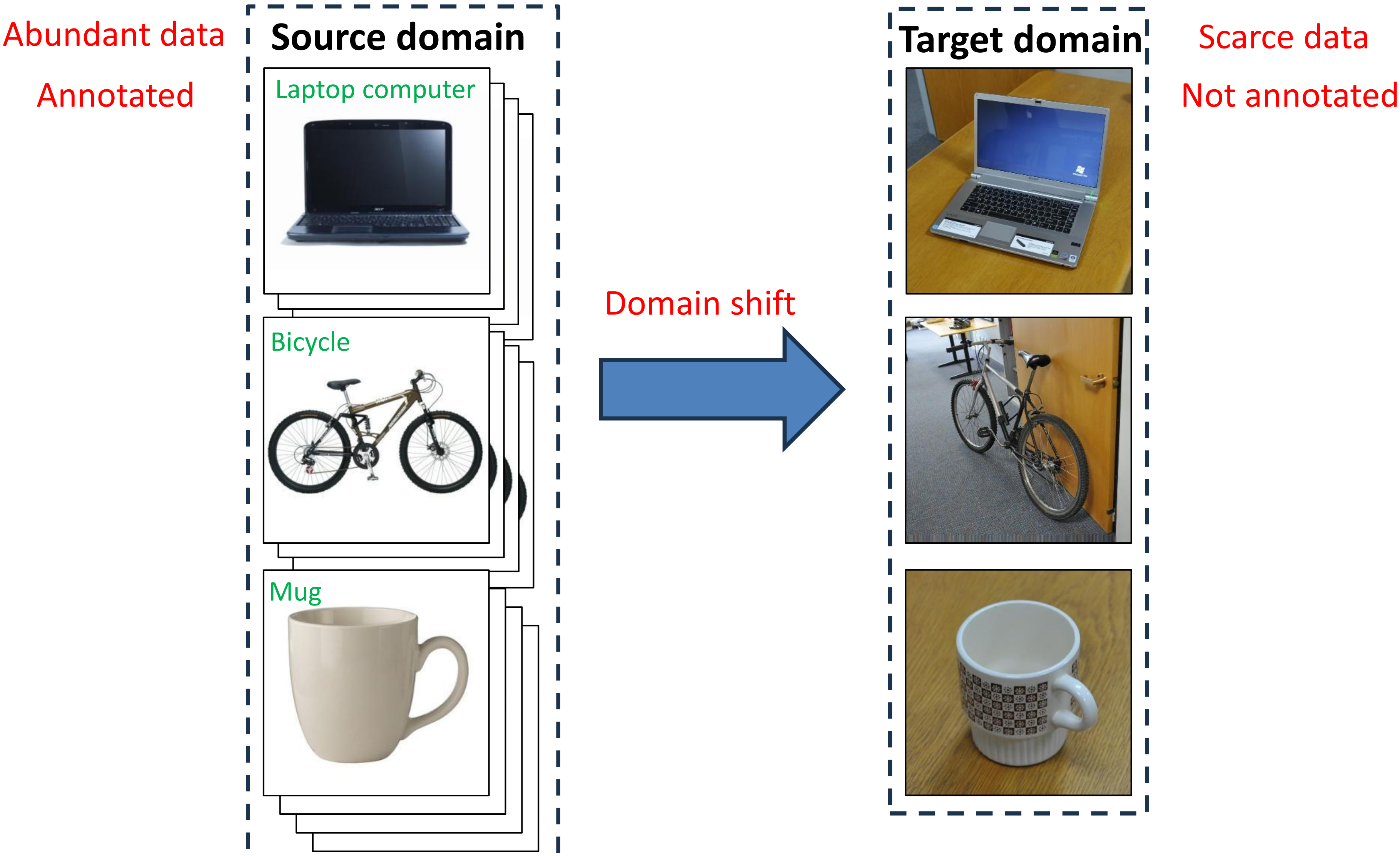
Check the videos at
https://www.lherranz.org/2018/10/29/mergans

Learning task 1 (bedroom) epoch 1

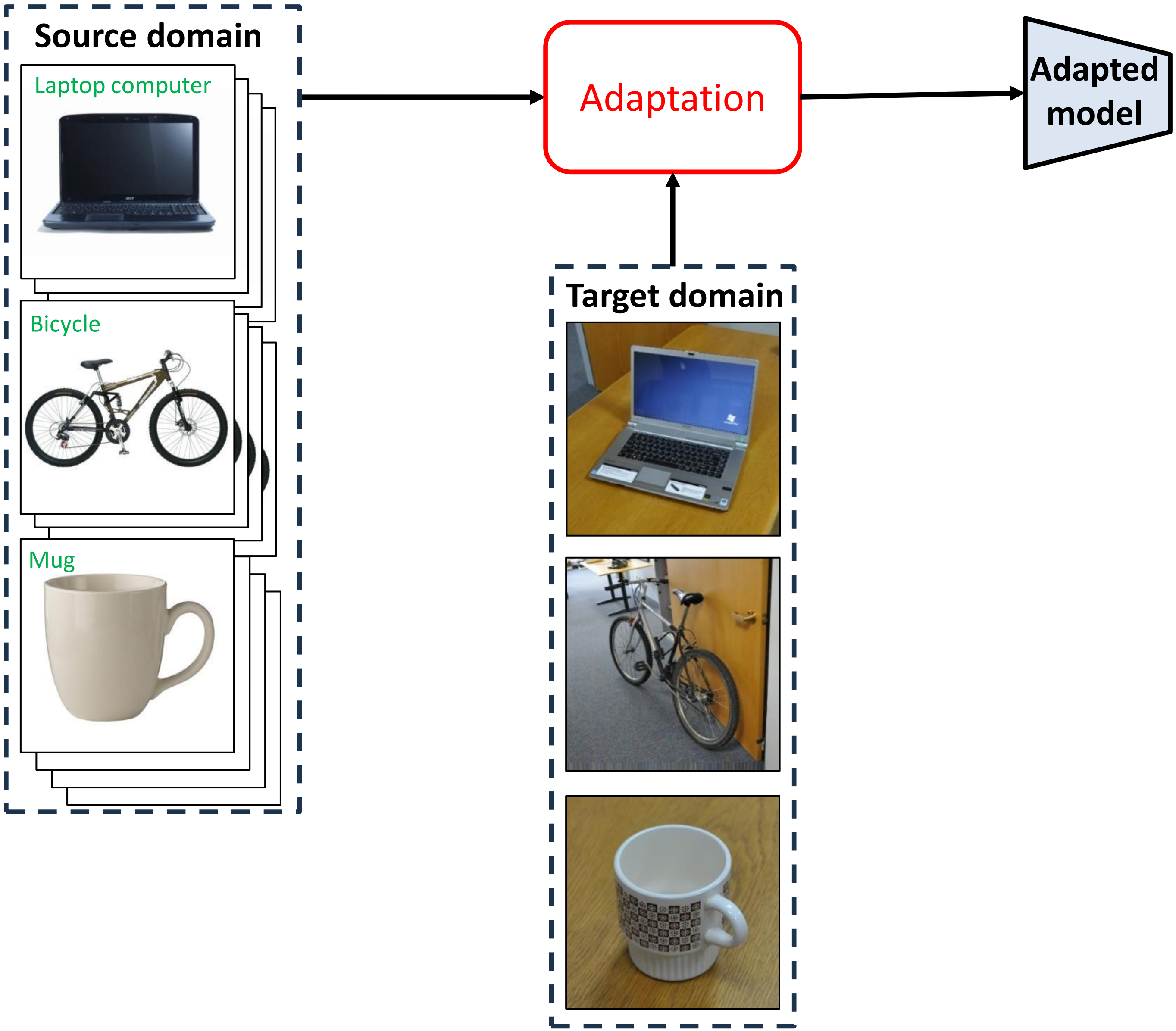
Check the video at

https://www.lherranz.org/2018/10/29/mergans

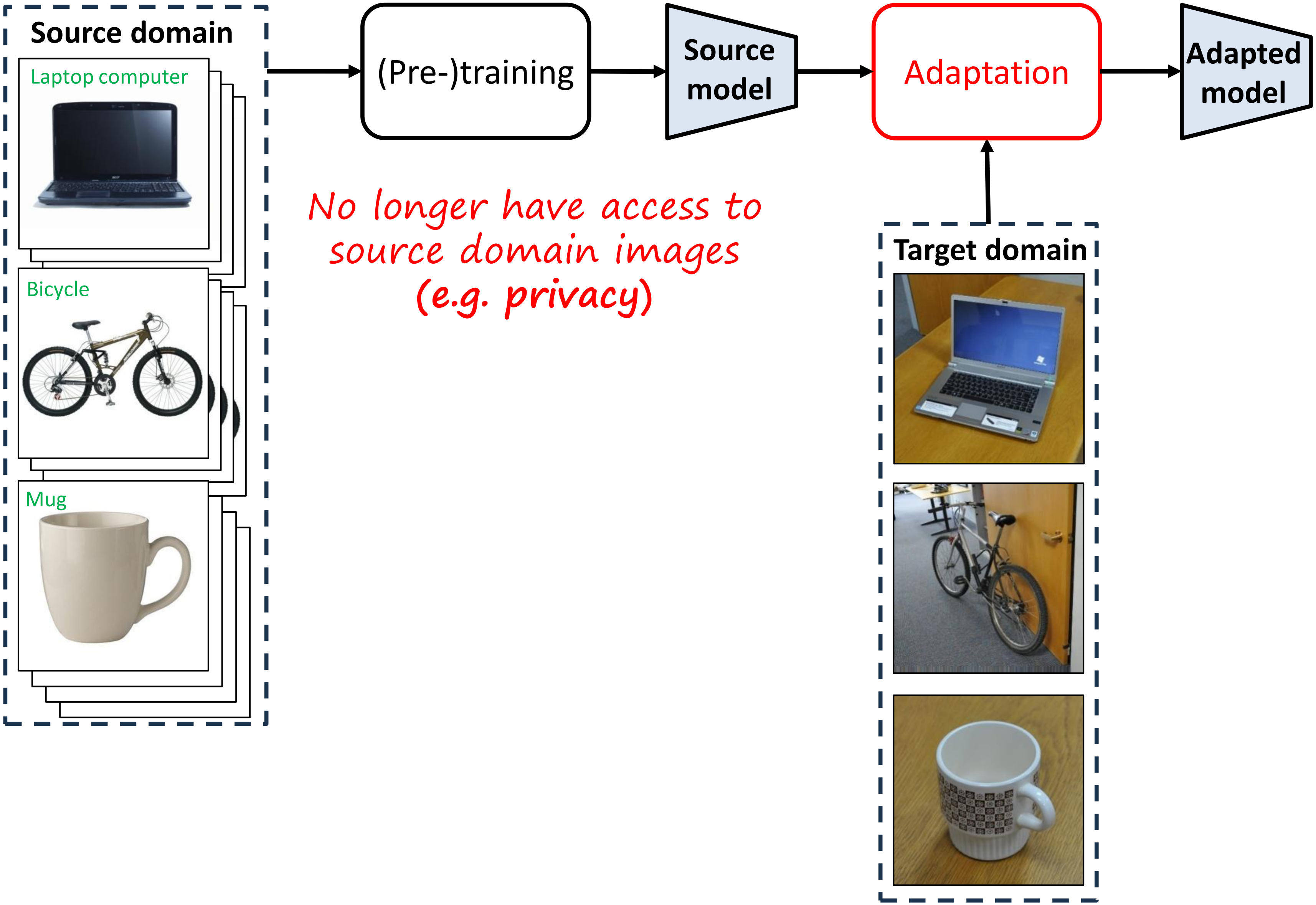
Unsupervised domain adaptation (UDA)



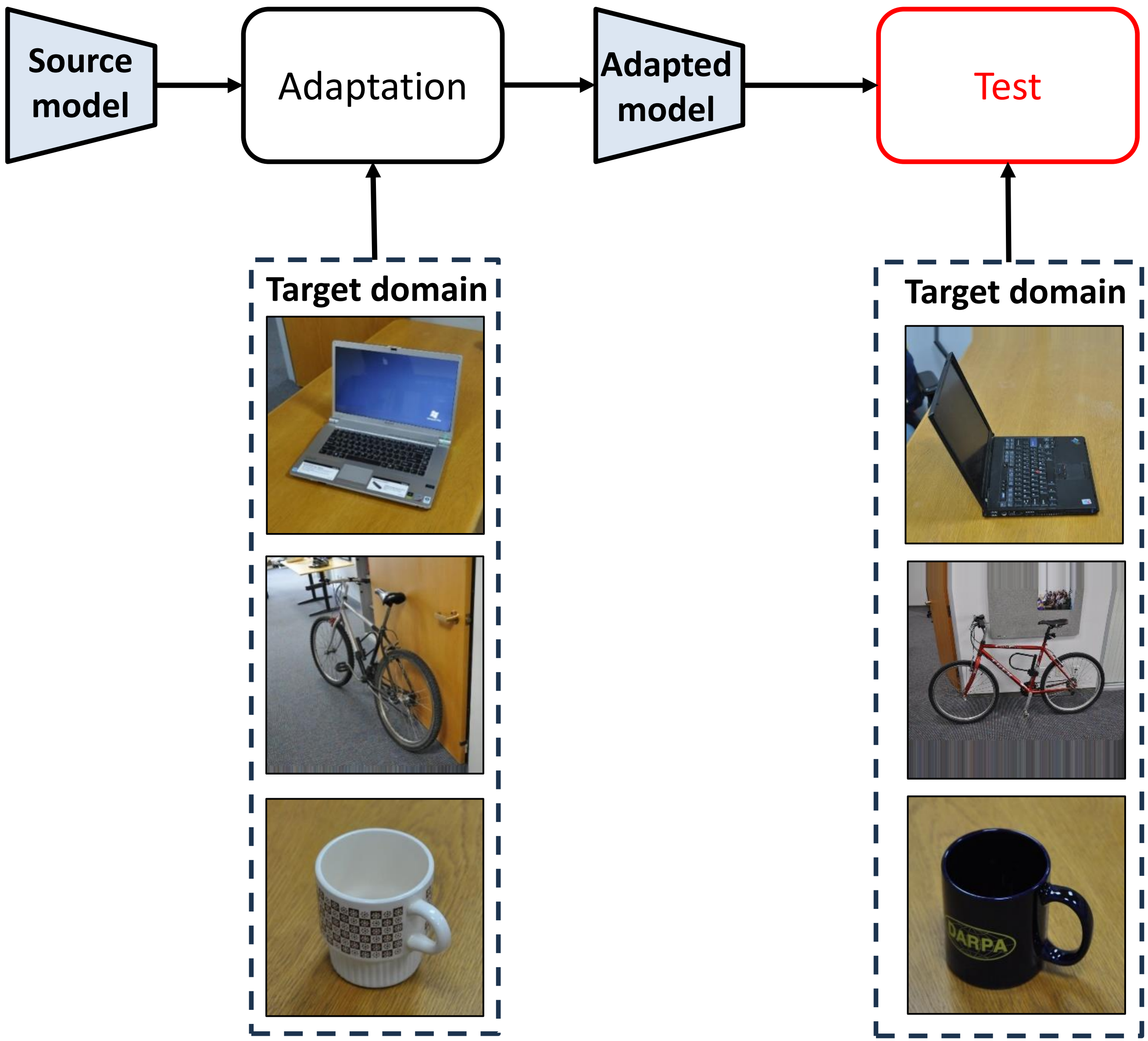
(Source-aware) UDA



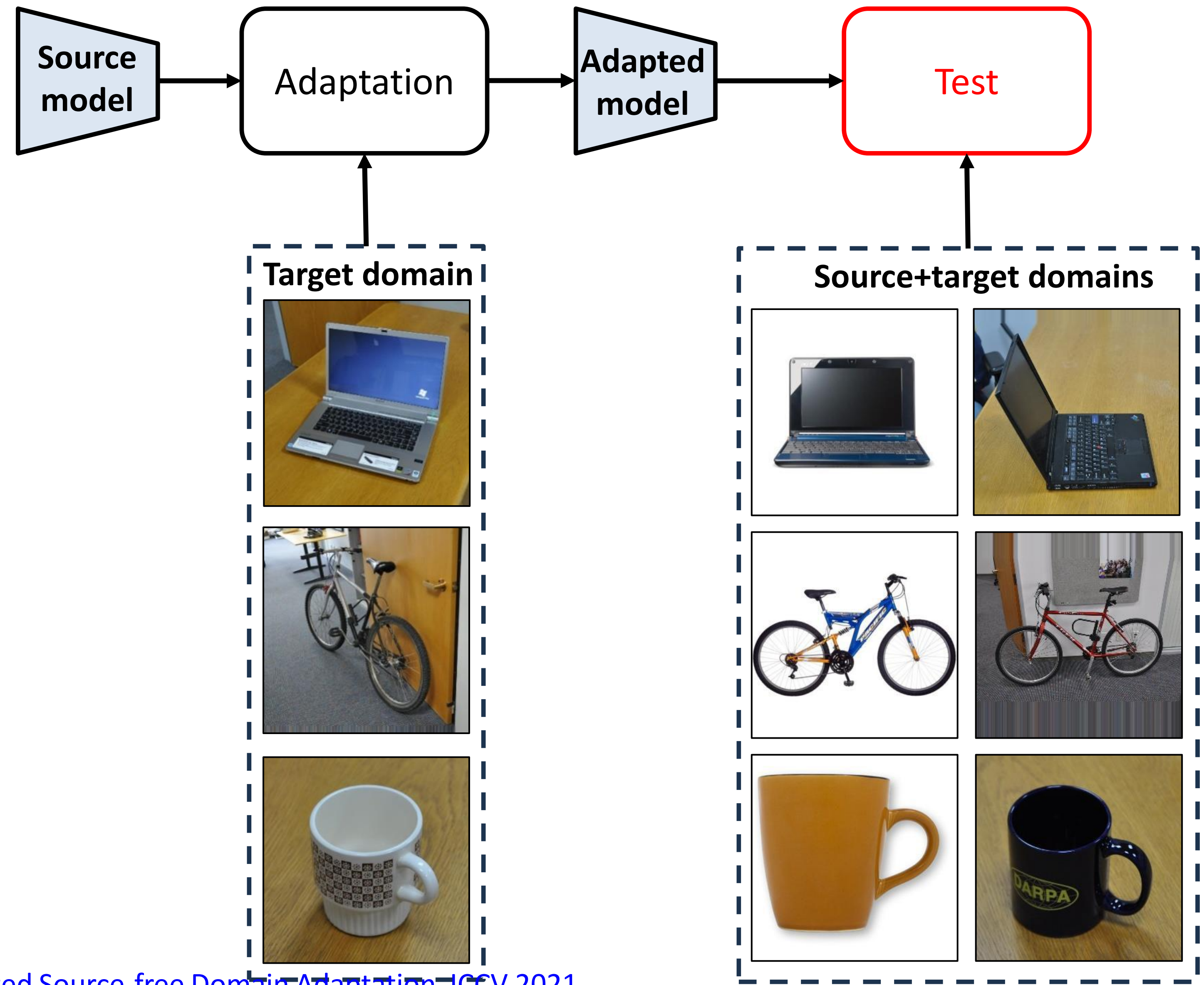
Source-free domain adaptation



Source-free domain adaptation



Generalized source-free domain adaptation



Generalized Source-free Domain Adaptation, ICCV 2021

Source-free domain adaptation

Results on **VisDA-C** with ResNet101 as backbone

Method (Synthesis → Real)	Source-free	Per-class
ResNet-101 [9]	×	52.4
ADR [30]	×	73.5
CDAN [22]	×	73.9
CDAN+BSP [5]	×	75.9
SWD [17]	×	76.4
MDD [49]	×	74.6
IA [11]	×	75.8
DMRL [42]	×	75.5
MCC [12]	×	78.8
DANCE [29]	×	70.4
DANCE [29]	✓	70.2
SHOT [20]	✓	<u>82.9</u>
3C-GAN [18]	✓	81.6
Ours	✓	**85.4**

- State-of-the-art target performance, compared to relative methods

Slide credit: Shiqi Yang

[Generalized Source-free Domain Adaptation, ICCV 2021]()

Generalized source-free domain adaptation

Results on **VisDA-C** under G-SFDA with ResNet101 as backbone

	Source-free	Avg. S /T	H
Source model		**99.6** /48.1	64.9
SHOT [20]	√	75.7/82.2	78.8
Ours	√	90.4/**85.0**	**87.6**

- State-of-the-art H over source/target performance compared to SHOT

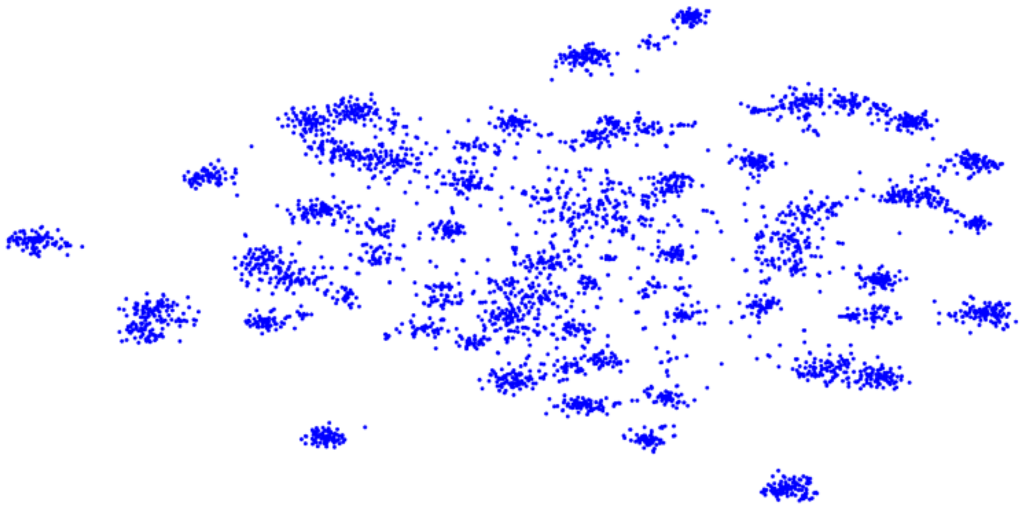
Slide credit: Shiqi Yang

[Generalized Source-free Domain Adaptation, ICCV 2021]

Exploiting neighborhood structure

Premise: We already have the source-pretrained model

t-SNE visualization



Observation 1: Target features from source pretrained model already form some clusters
Motivation 1: We can adopt neighborhood clustering for target adaptation

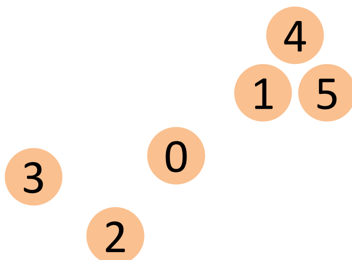
Slide credit: Shiqi Yang

[Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation, NeurIPS 2021]

Exploiting neighborhood structure

Reciprocal nearest neighbors (example K=2)

Feature space



Neighborhood structure

Query		NNs
0	⇒	1 2
1	⇒	4 5
2	⇒	0 3
...	⇒	... ...

Are they reciprocal?

0 ⇒ 1 2

2 ⇒ 0 3

0 and 2 are RNNs

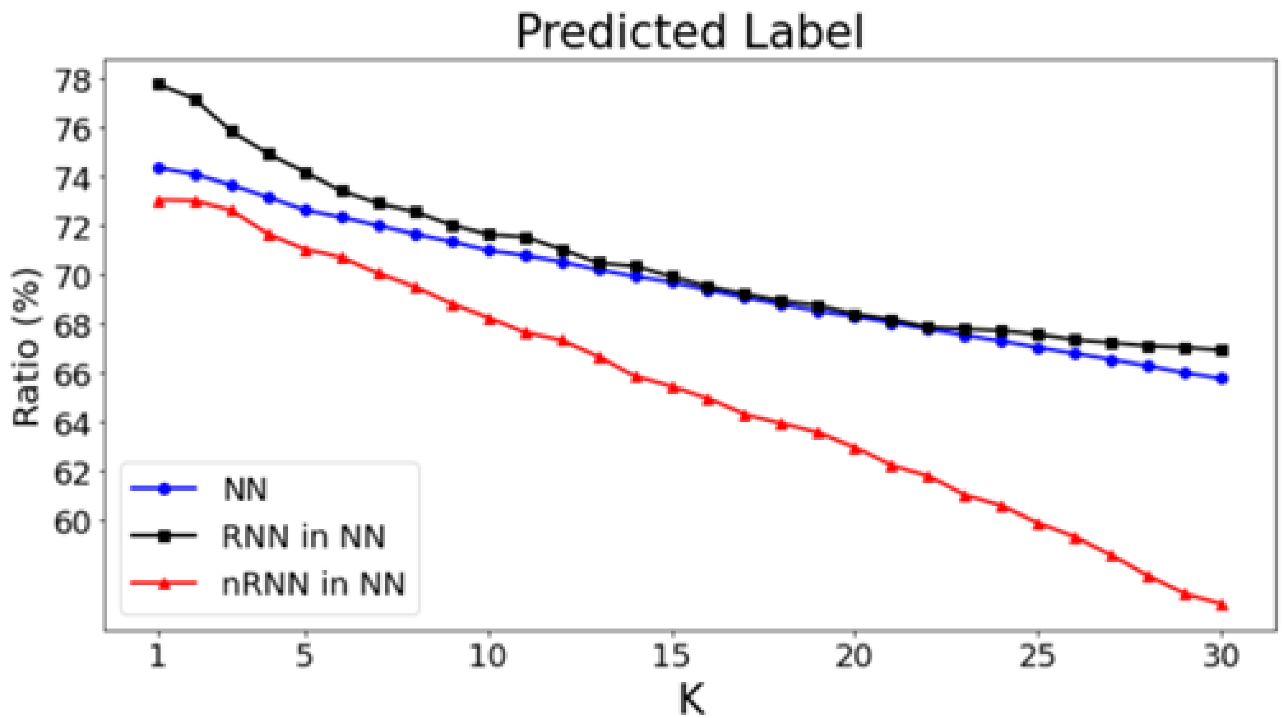
0 ⇒ 1 2

1 ⇒ 4 5

0 and 1 are not RNNs

[Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation, NeurIPS 2021]

Exploiting neighborhood structure



Observation 2: Reciprocal neighbors are more likely to have the **correct** predicted label

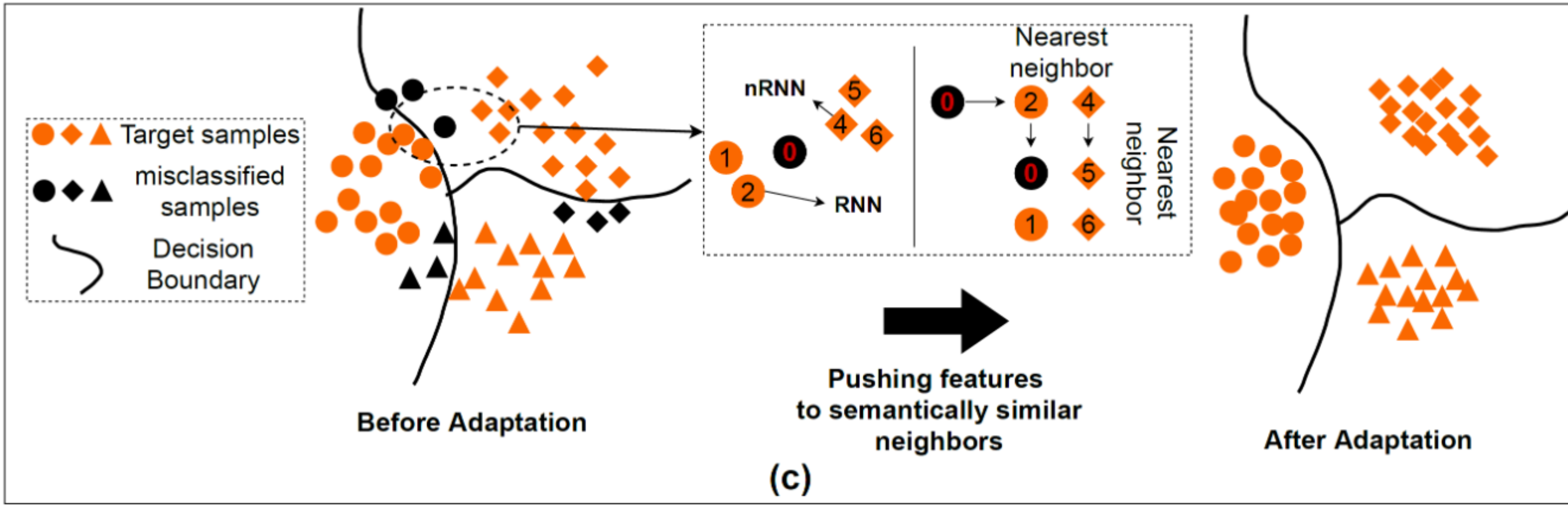
Motivation 2: We should assign higher credit to reciprocal neighbors.

Slide credit: Shiqi Yang

[Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation, NeurIPS 2021]()

Exploiting neighborhood structure

Method



Method overview: $$\mathcal{L} = -\frac{1}{n_t} \sum_{x_i \in \mathcal{D}_t} \sum_{x_j \in \text{Neigh}(x_i)} \frac{D_{sim}(p_i, p_j)}{D_{dis}(x_i, x_j)}$$

Slide credit: Shiqi Yang

[Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation, NeurIPS 2021]

Exploiting neighborhood structure. Results

- Results on Office-Home with ResNet50 as backbone

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	**Avg**
MCD [35]	✗	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [24]	✗	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SAFN [52]	✗	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
Symnets [58]	✗	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
MDD [59]	✗	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	**60.2**	82.3	68.1
TADA [47]	✗	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
BNM [4]	✗	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
BDG [53]	✗	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
SRDC [42]	✗	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	85.0	71.3
RSDA-MSTN [10]	✗	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
SHOT [21]	✓	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
NRC	✓	**57.7**	**80.3**	**82.0**	68.1	**79.8**	**78.6**	65.3	**56.4**	**83.0**	71.0	58.6	**85.6**	**72.2**

- Results on VisDA-C with ResNet101 as backbone

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
ADR [34]	✗	94.2	48.5	84.0	72.9	90.1	74.2	92.6	72.5	80.8	61.8	82.2	28.8	73.5
CDAN [24]	✗	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
CDAN+BSP [2]	✗	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SAFN [52]	✗	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
SWD [19]	✗	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MDD [59]	✗	-	-	-	-	-	-	-	-	-	-	-	-	74.6
DMRL [49]	✗	-	-	-	-	-	-	-	-	-	-	-	-	75.5
MCC [15]	✗	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
STAR [26]	✗	95.0	84.0	**84.6**	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
RWOT [51]	✗	95.1	80.3	83.7	**90.0**	92.4	68.0	**92.5**	82.2	87.9	78.4	**90.4**	**68.2**	84.0
3C-GAN [20]	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	**84.7**	89.1	84.7	83.5	48.1	81.6
SHOT [21]	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
NRC	✓	**96.8**	**91.3**	82.4	62.4	**96.2**	**95.9**	86.1	80.6	**94.8**	**94.1**	90.4	59.7	**85.9**

Slide credit: Shiqi Yang

[Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation, NeurIPS 2021]()

THANK YOU!

lherranz@cvc.uab.es
www.lherranz.org

www.lherranz.org/blog

CVC UAB Universitat Autònoma de Barcelona MINISTERIO DE CIENCIA E INNOVACIÓN UNIÓN EUROPEA FONDO SOCIAL EUROPEO El FSE invierte en tu futuro AGENCIA ESTATAL DE INVESTIGACIÓN