Towards practical neural image compression: SlimCAE and DANICE

Fei Yang, Luis Herranz

CVPR 2021/CLIC2021

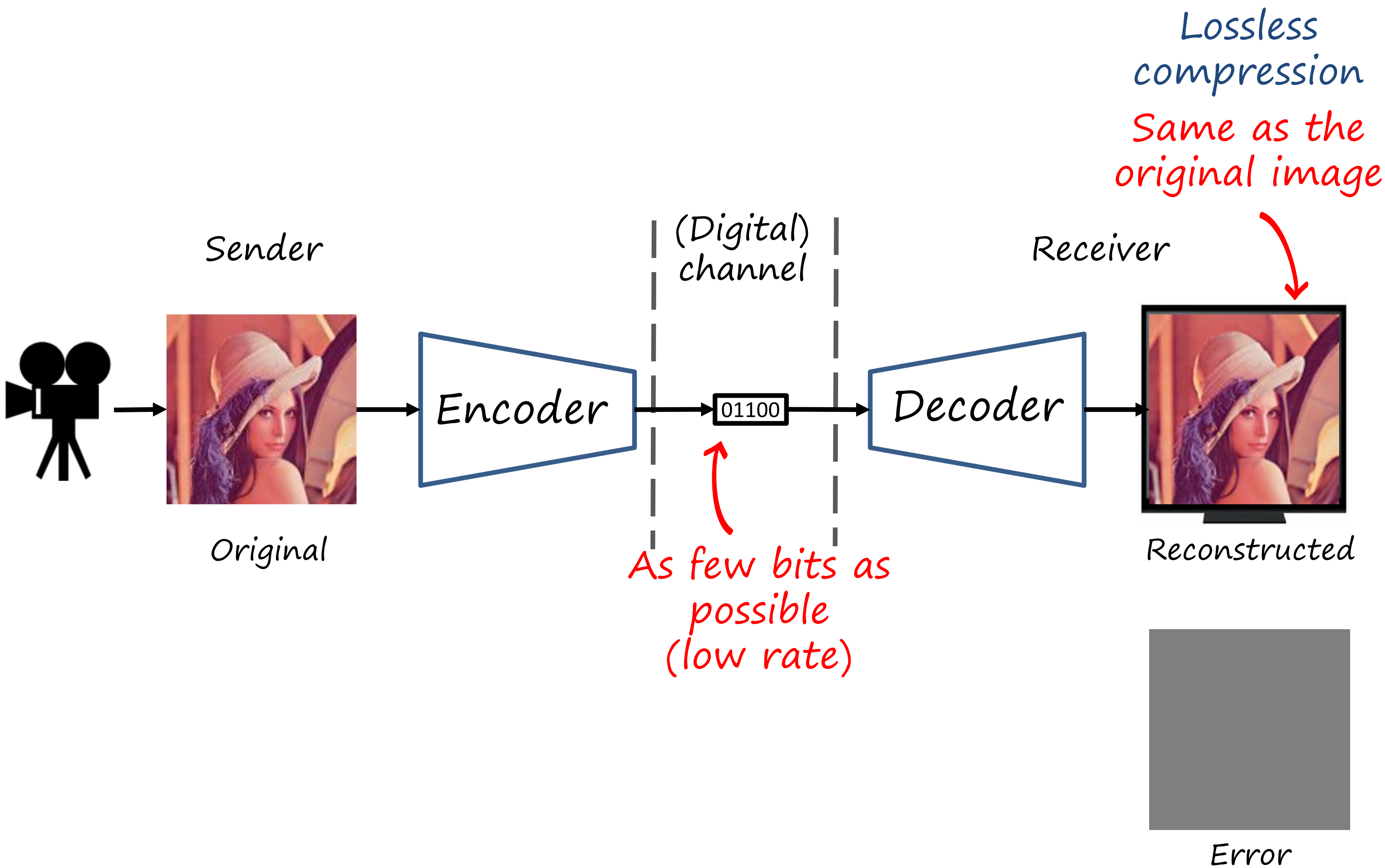
CVC

UAB Universitat Autònoma de Barcelona

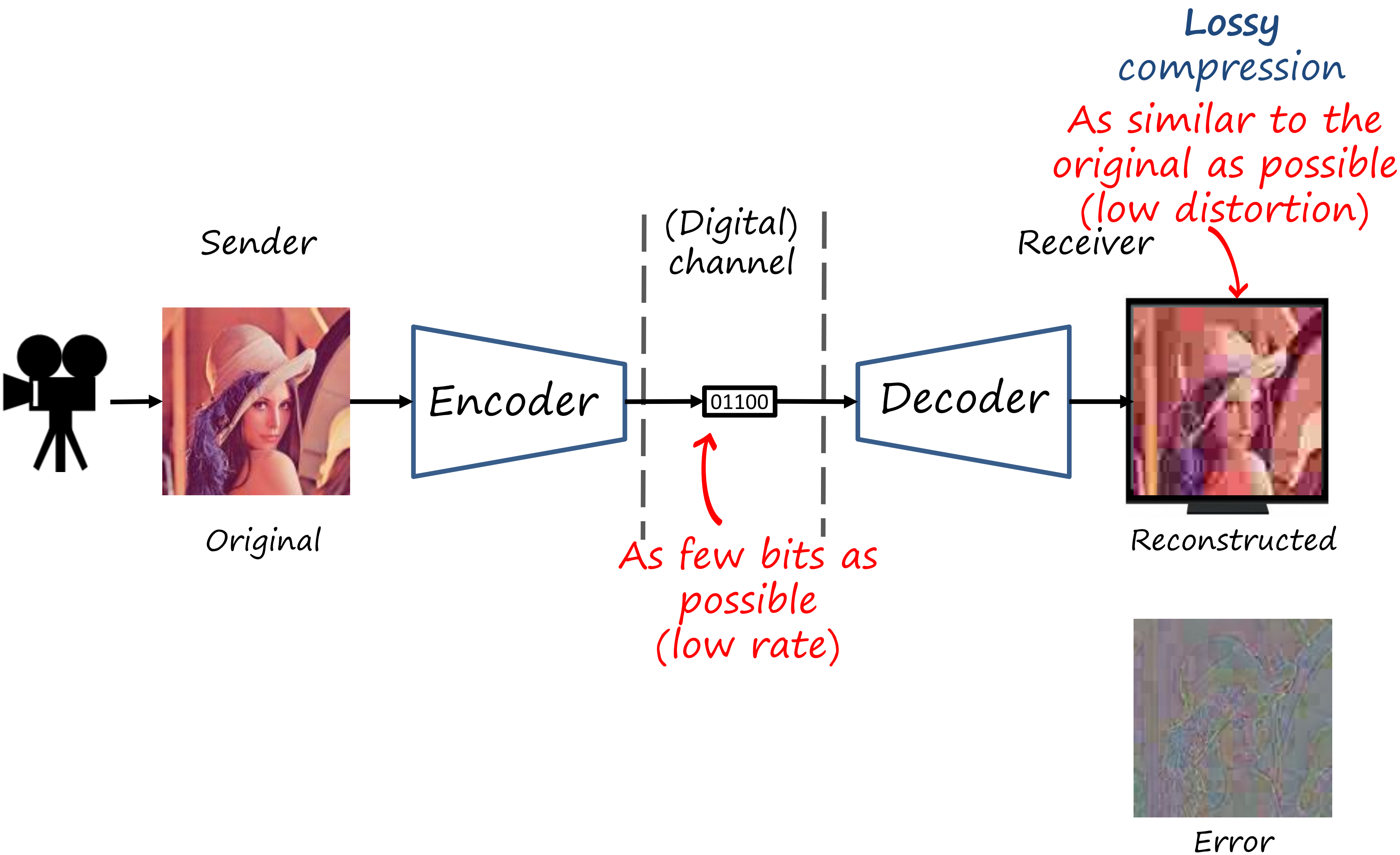
Outline

- Introduction and motivation
- SlimCAE (CVPR 2021)
- DANICE (CLIC workshop at CVPR 2021)

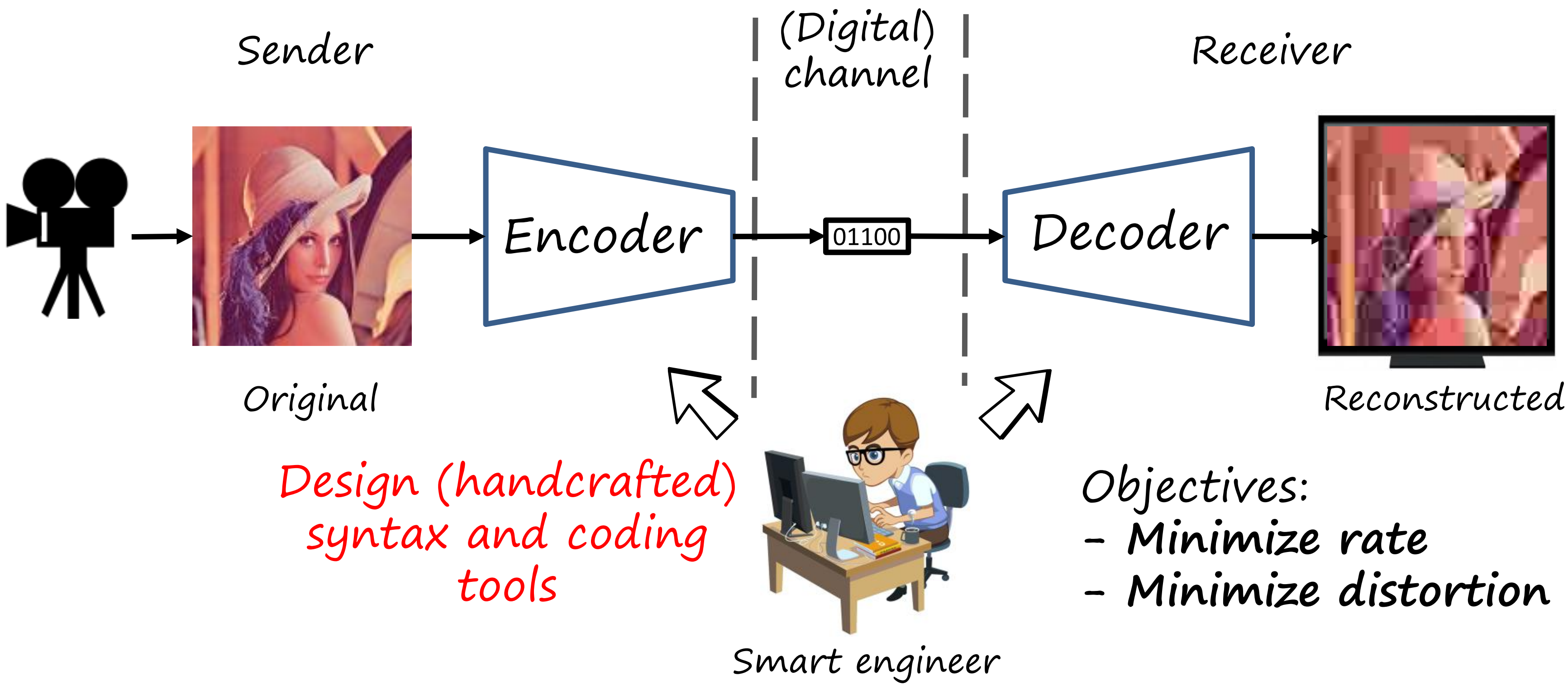
The visual communication problem



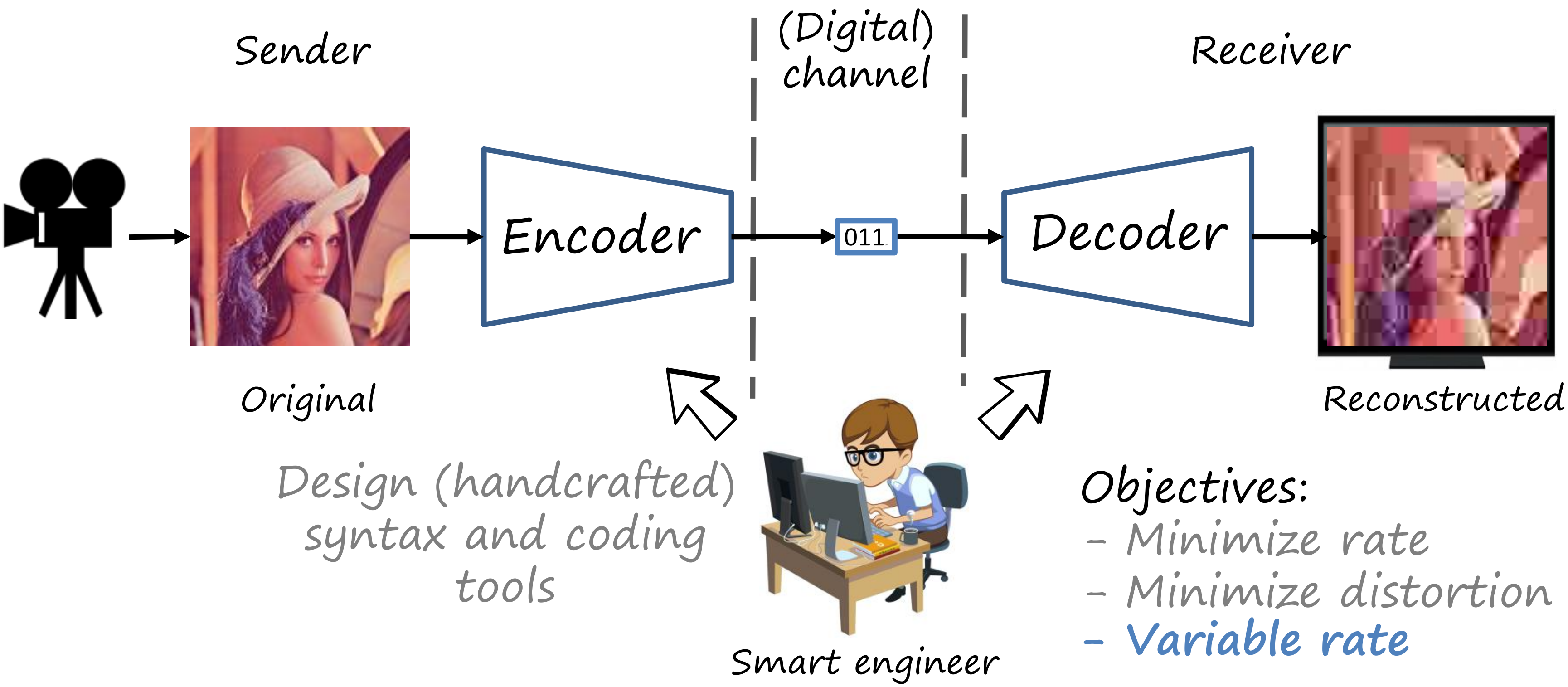
The visual communication problem



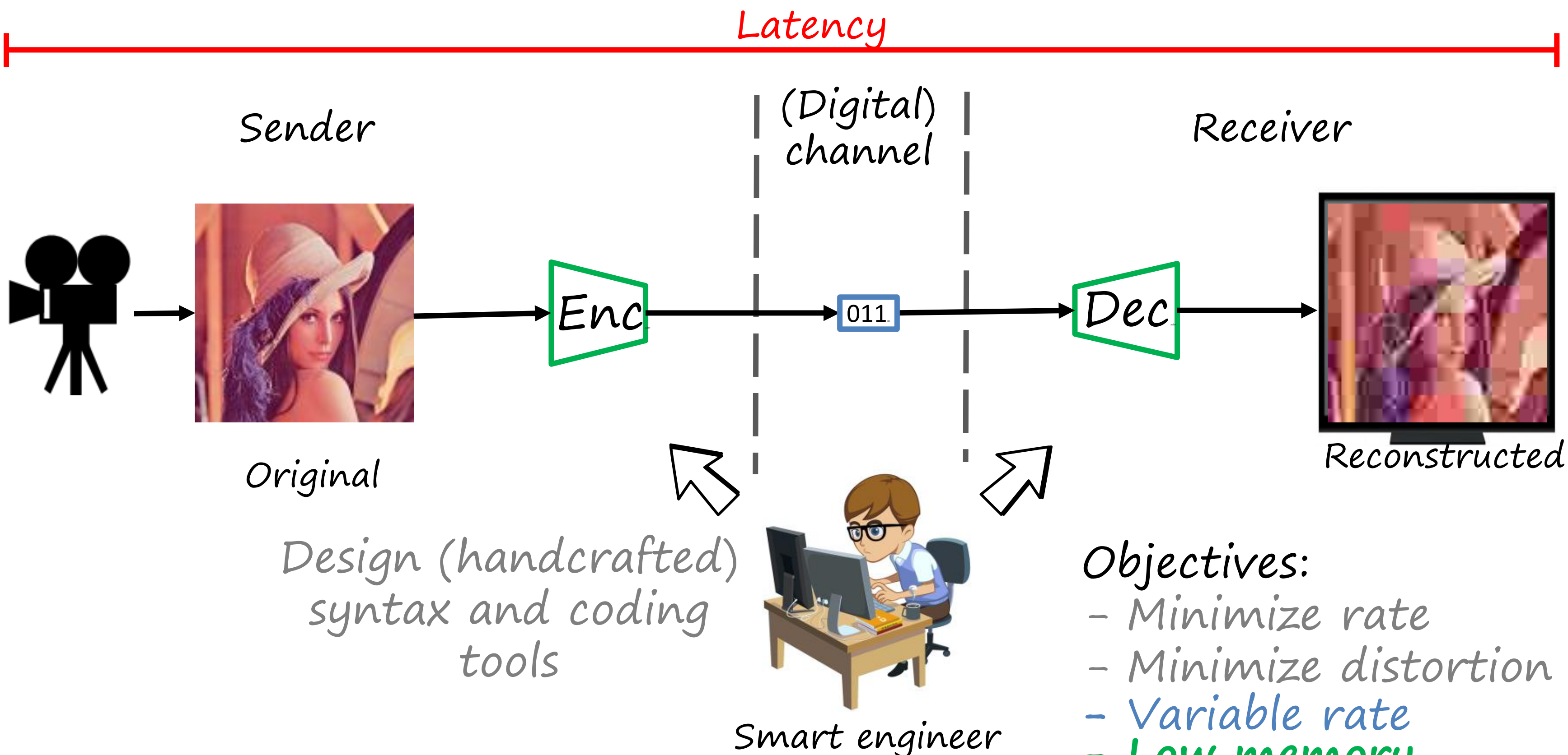
Developing traditional image/video codecs



… for practical applications

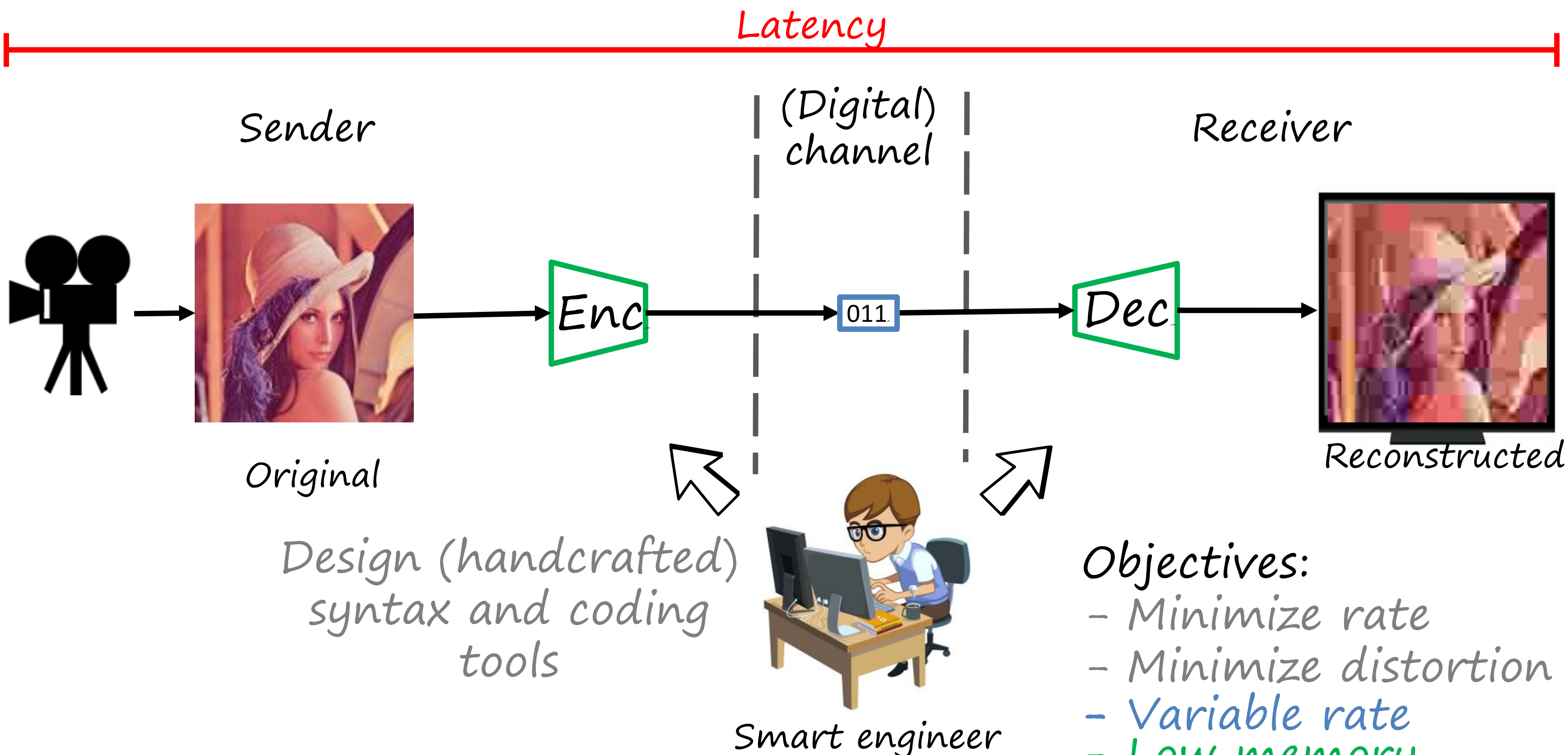


… for practical applications



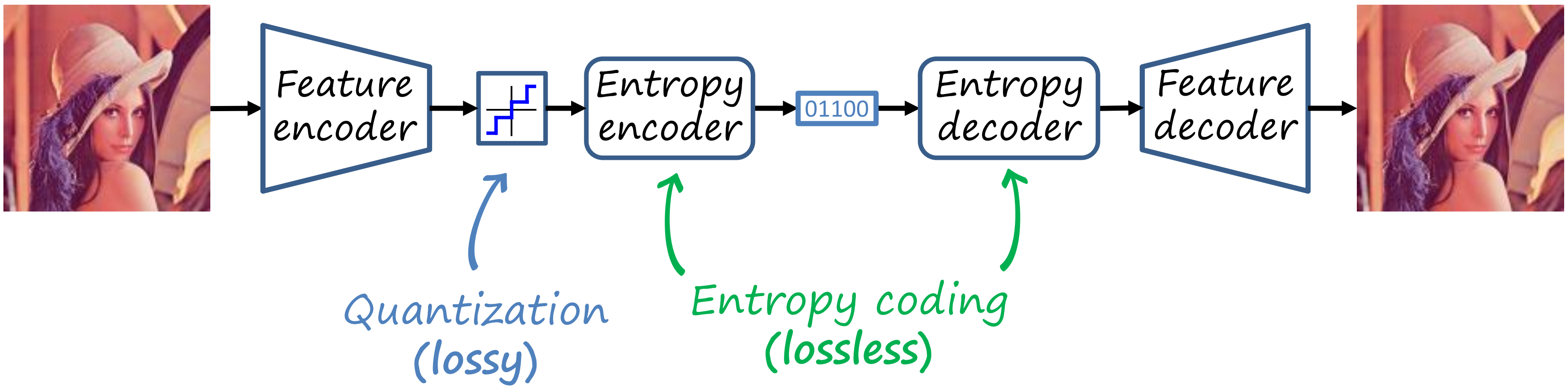
Objectives:
- Minimize rate
- Minimize distortion
- Variable rate
- Low memory
- Low computation
- Low latency

… for practical applications

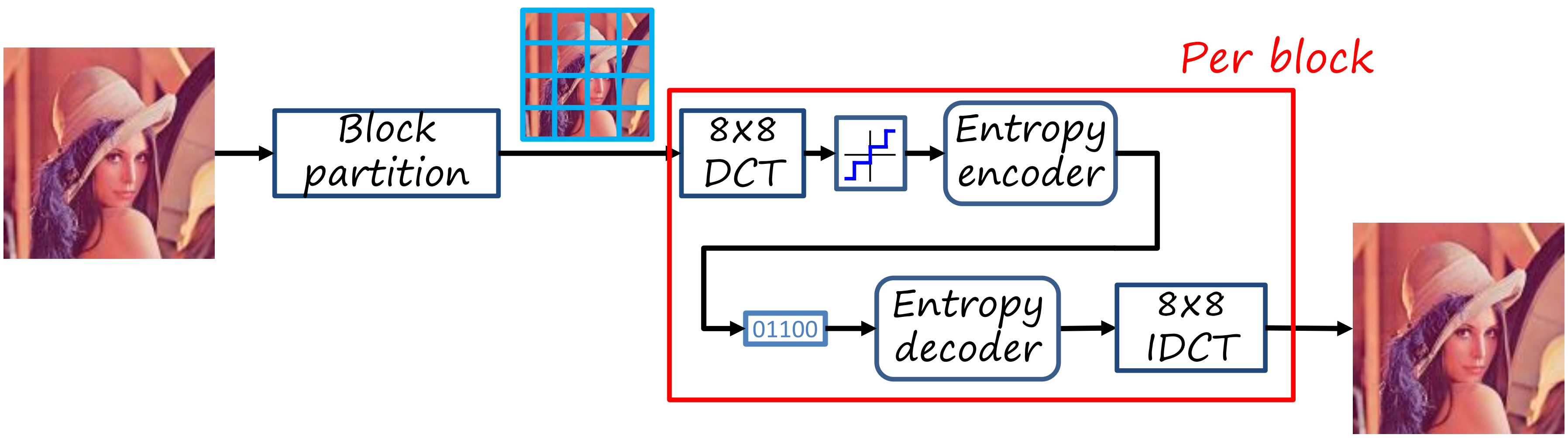


Objectives:
- Minimize rate
- Minimize distortion
- Variable rate
- Low memory
- Low computation
- Low latency
- Compatibility
- Domain-specific

Basic pipeline

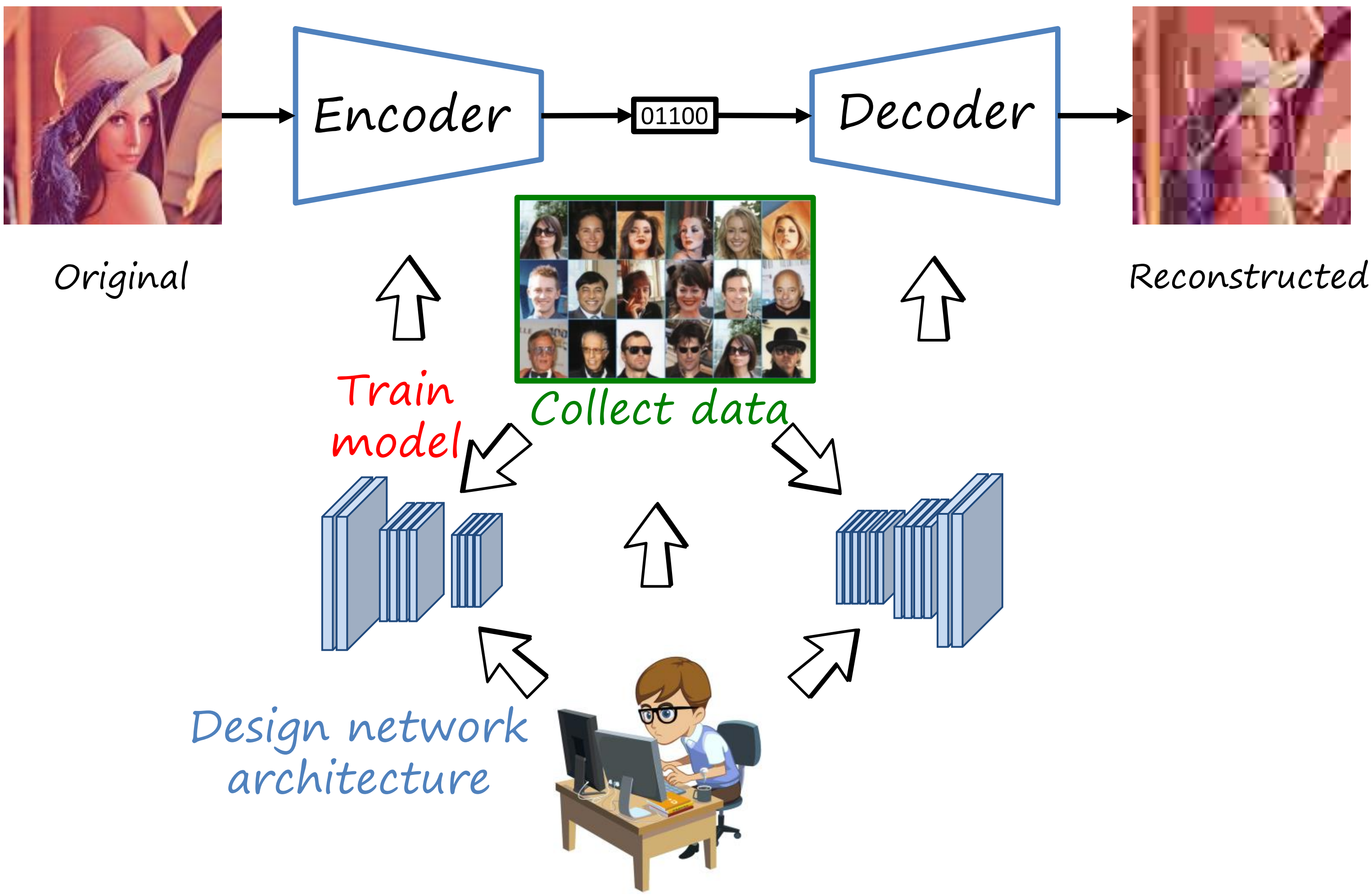


Example: block-based transform coding (e.g. JPEG, MPEG-4)



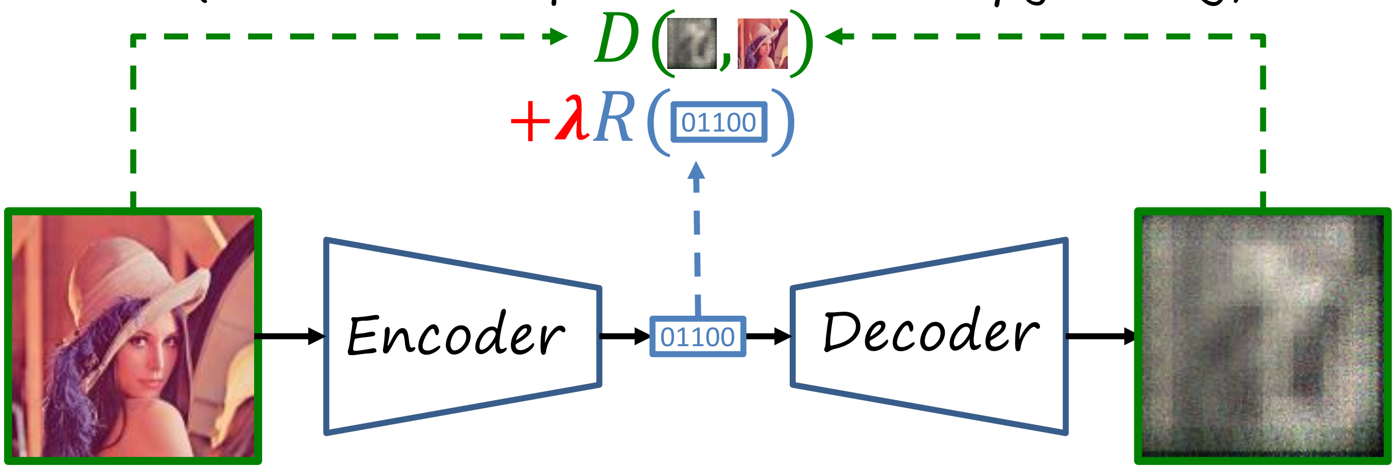
Neural image/video codecs

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



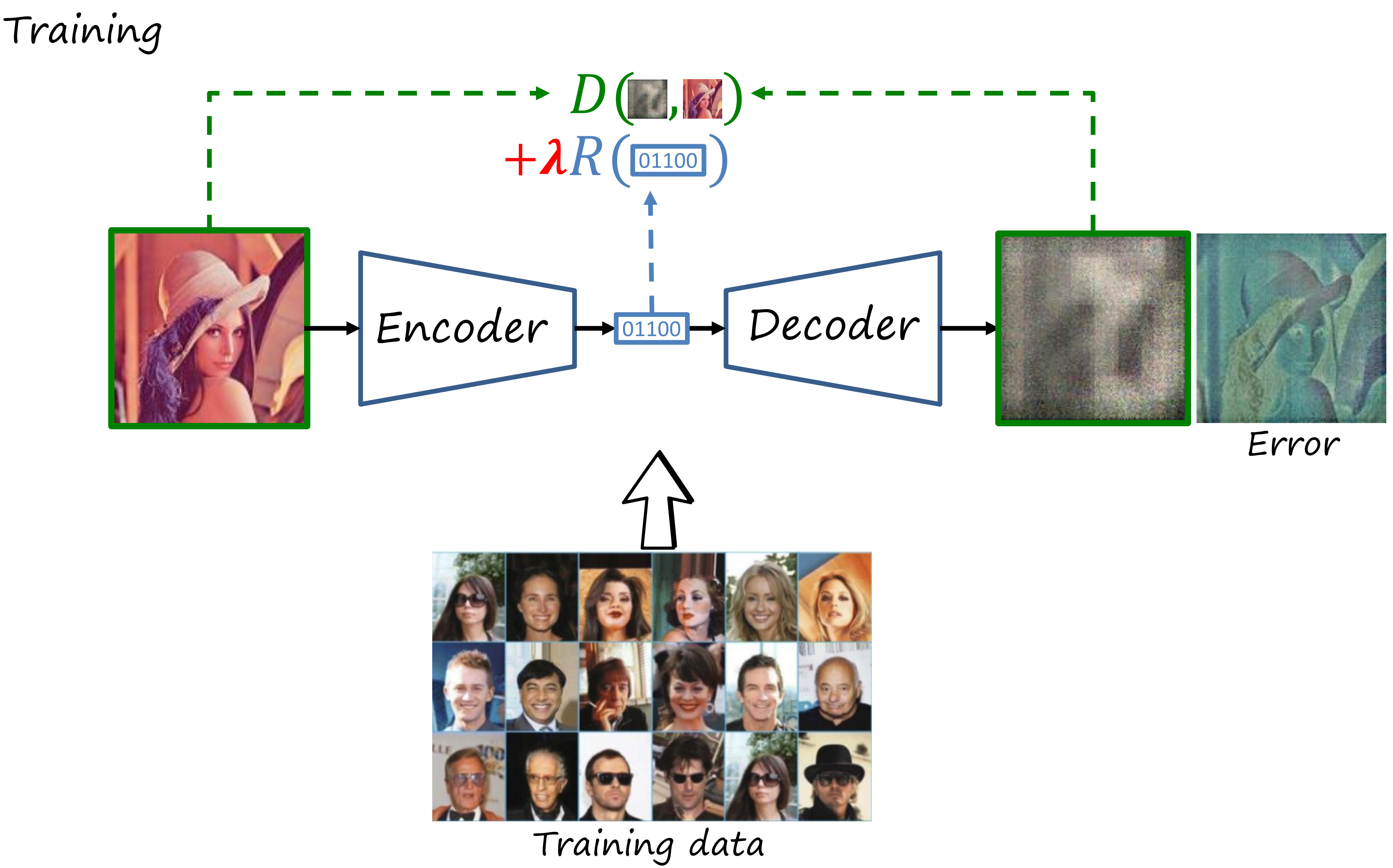
Neural image compression

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



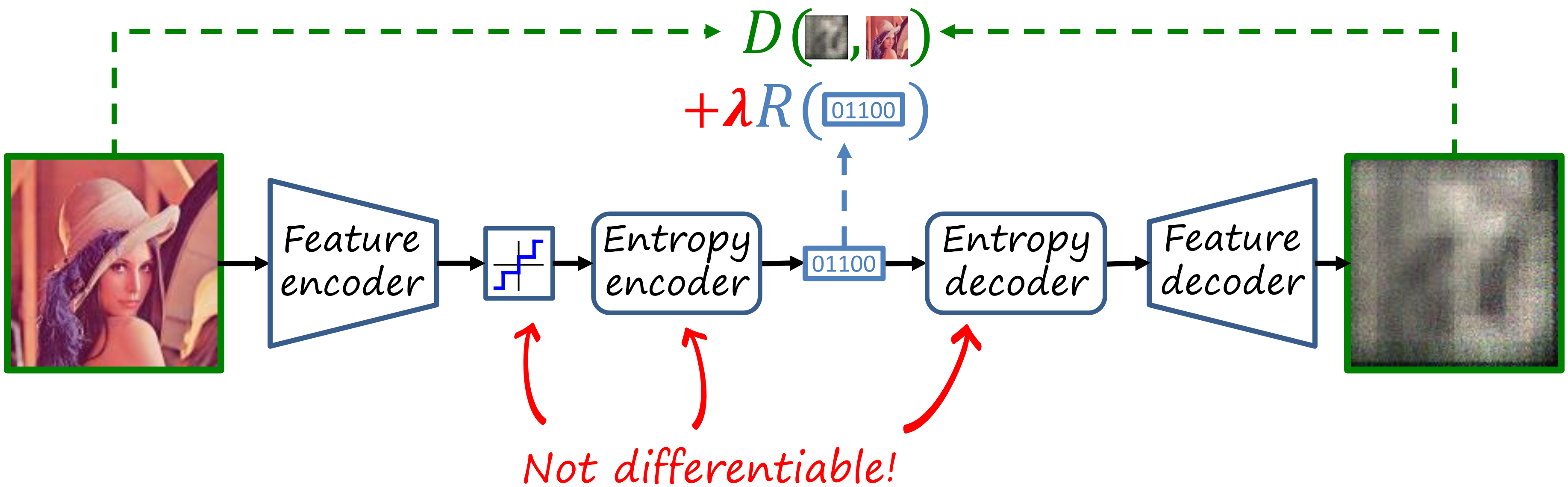
Optimize a weighted rate-distortion loss (λ controls the **tradeoff**)

Neural image compression



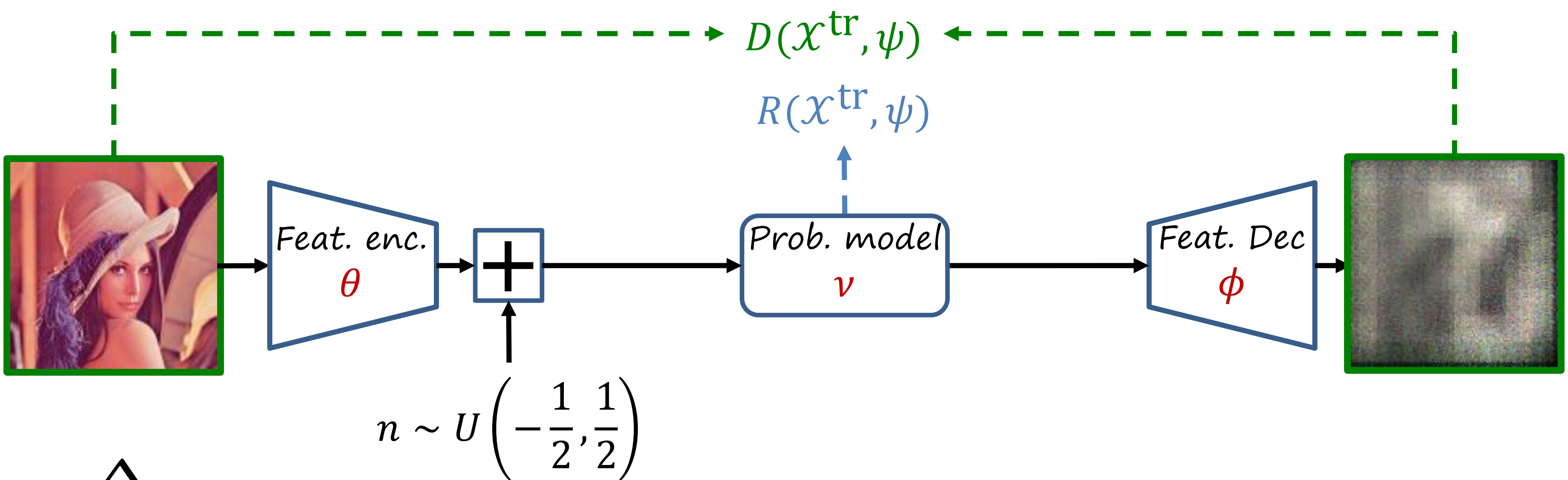
Architecture

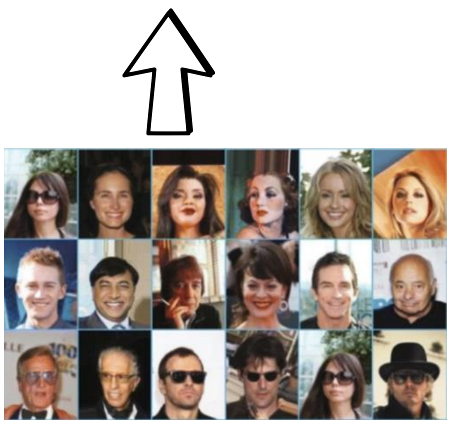
Compressive autoencoder (CAE) [Theis2017, Balle2017]
(autoencoder+quantization+entropy coding)



Architecture (training)

Use differentiable proxies for end-to-end training



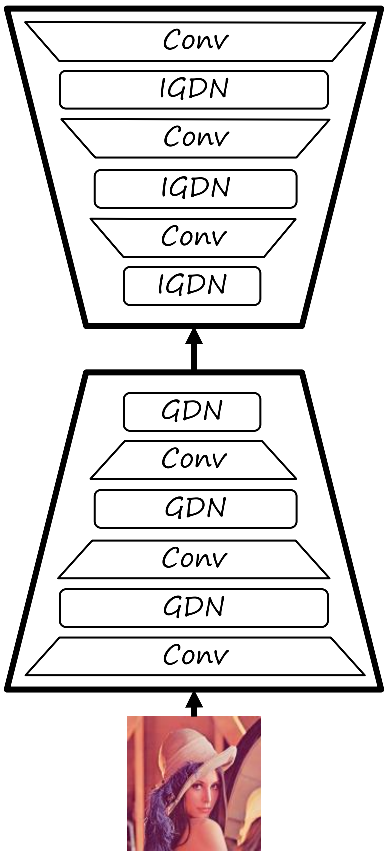


Training data $\mathcal{X}^{\mathrm{tr}}$

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	$\psi^* = \min_{\psi} J(\mathcal{X}^{\mathrm{tr}}, \psi; \lambda)$

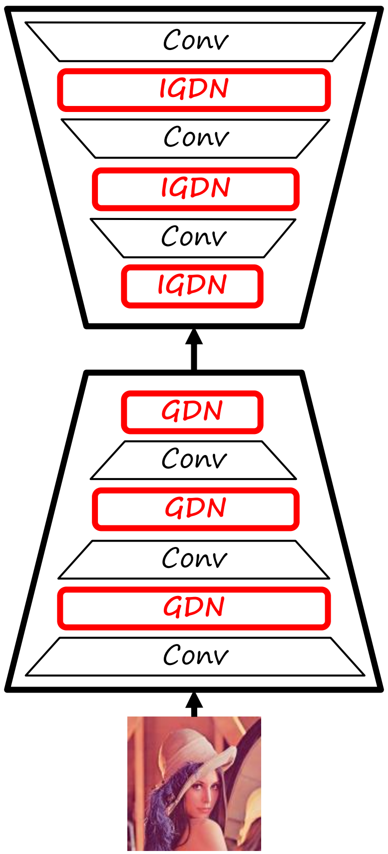
Autoencoder architecture

Balle et al.
[ICLR2017]



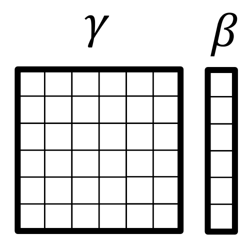
Autoencoder architecture

Balle et al.
[ICLR2017]



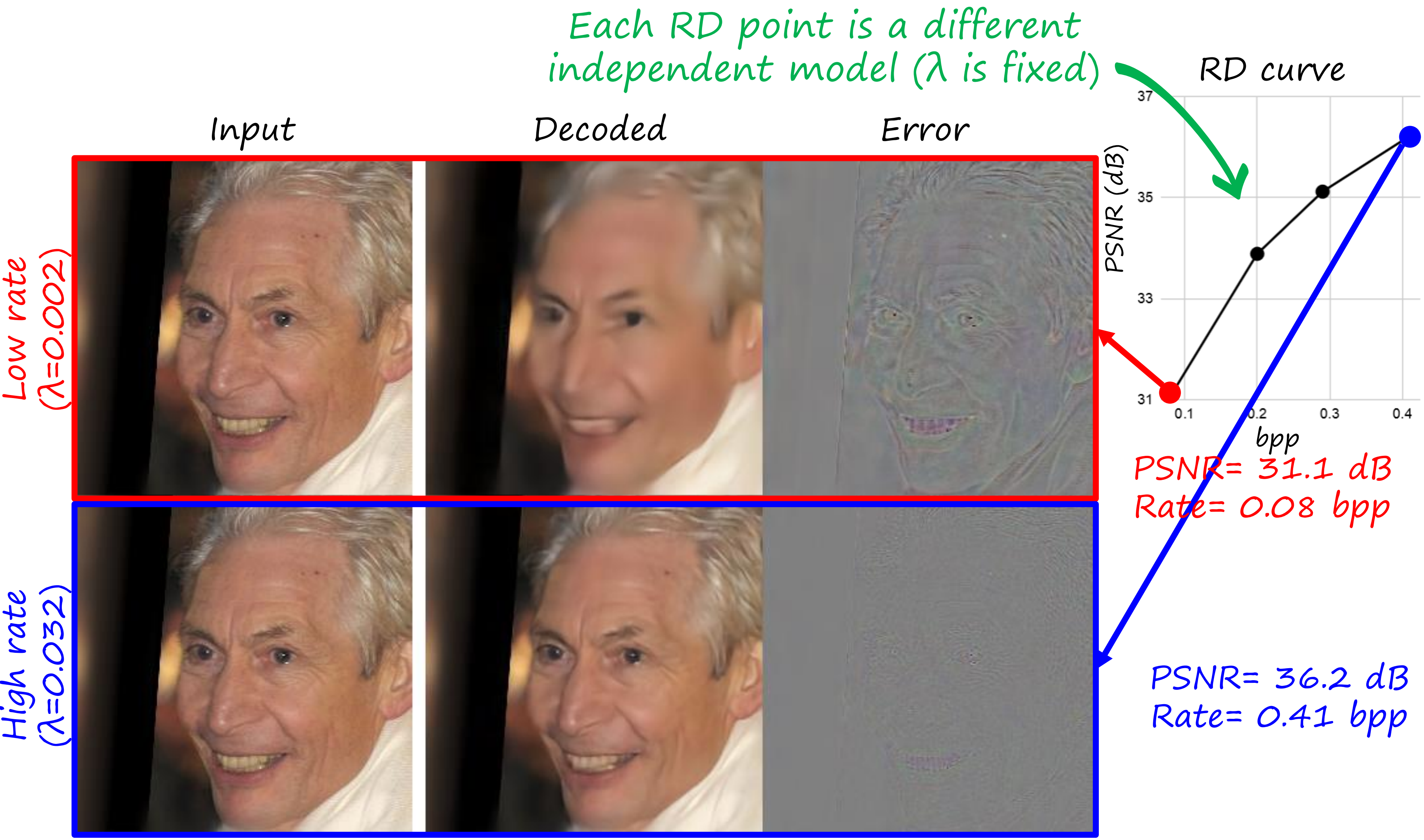
Generalized divisive normalization (GDN) [Balle2016]

$$\hat{y}_i = \frac{y_i}{\left(\beta_i + \sum_j \gamma_{ij}\, y_j^2\right)^{1/2}}$$

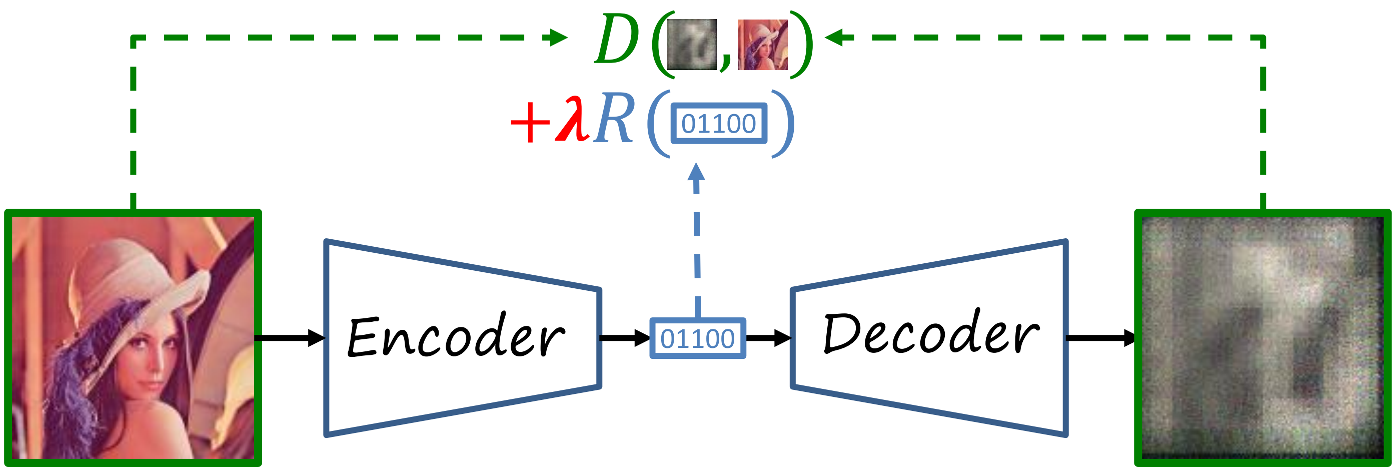


Learnable
parameters

Rate-distortion tradeoff λ



Is neural image compression practical?



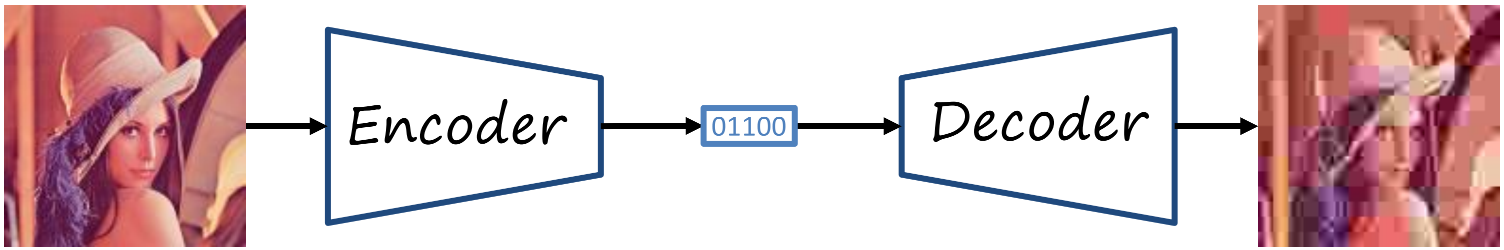
Limitations

- λ is fixed
- Heavy encoders/decoders

Practical neural image compression?

- Minimize rate ✓
- Minimize distortion ✓
- Variable rate ✗
- Low memory ✗
- Low computation ✗
- Low latency ✗

Towards **practical** neural image compression



Main objectives
- Minimize rate
- Minimize distortion

Practical objectives
- **Variable rate**
- **Low memory**
- **Low computation**
- **Low latency**

MAE [SPL2020]

SlimCAE [CVPR2021]

Other practical considerations
- **Domain-specific codecs** (e.g. videoconference, screencast)
- **Backward/forward compatibility** (with legacy formats and 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

Slimmable Compressive Autoencoders for Practical Neural Image Compression

Fei Yang[1,2,3], Luis Herranz[1,2], Yongmei Cheng[3], Mikhail Mozerov[1,2]

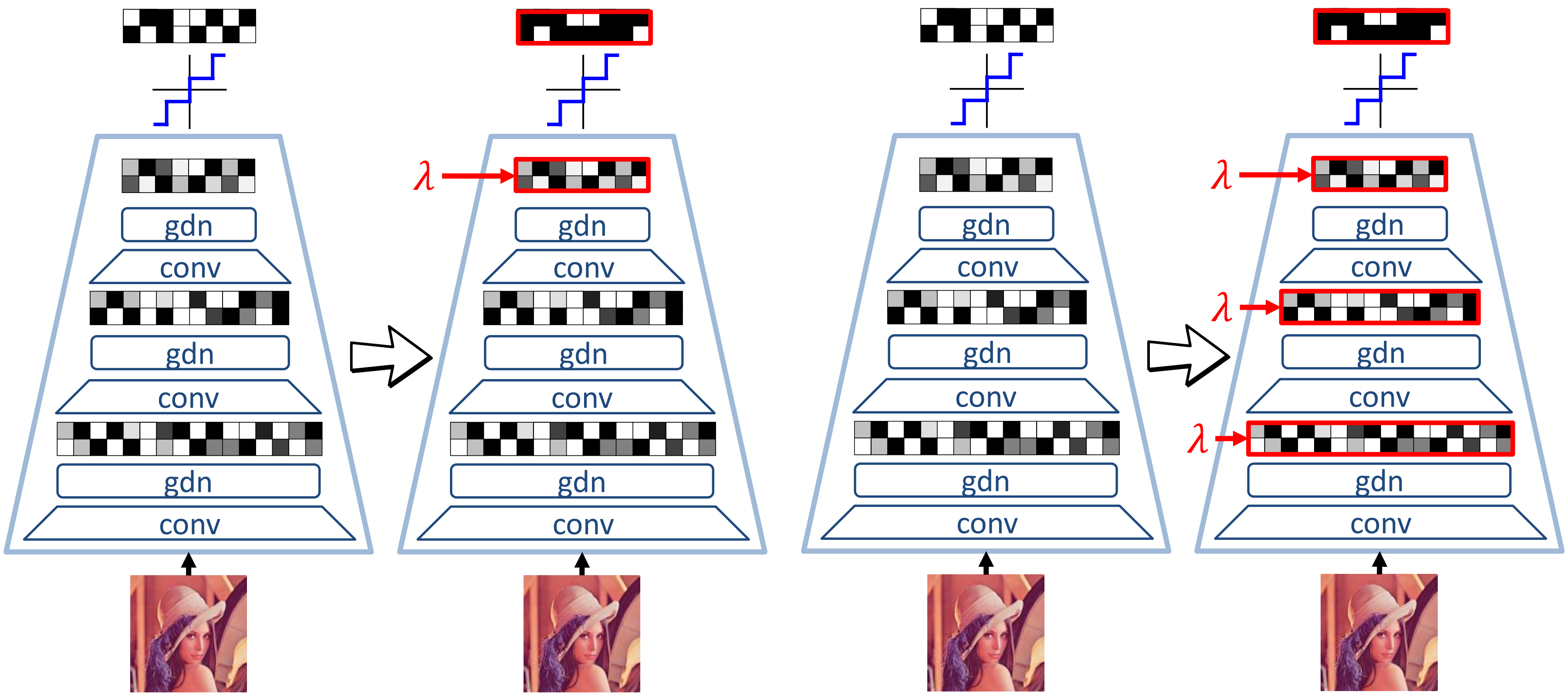
[1]Computer Vision Center
[2]Universitat Autònoma de Barcelona
[3]Northwestern Politechnical University

CVPR 2021

Variable rate neural image compression

Objective: one single model for multiple λ

Bottleneck scaling [Theis2017] | **Feature modulation [MAE, cAE]**

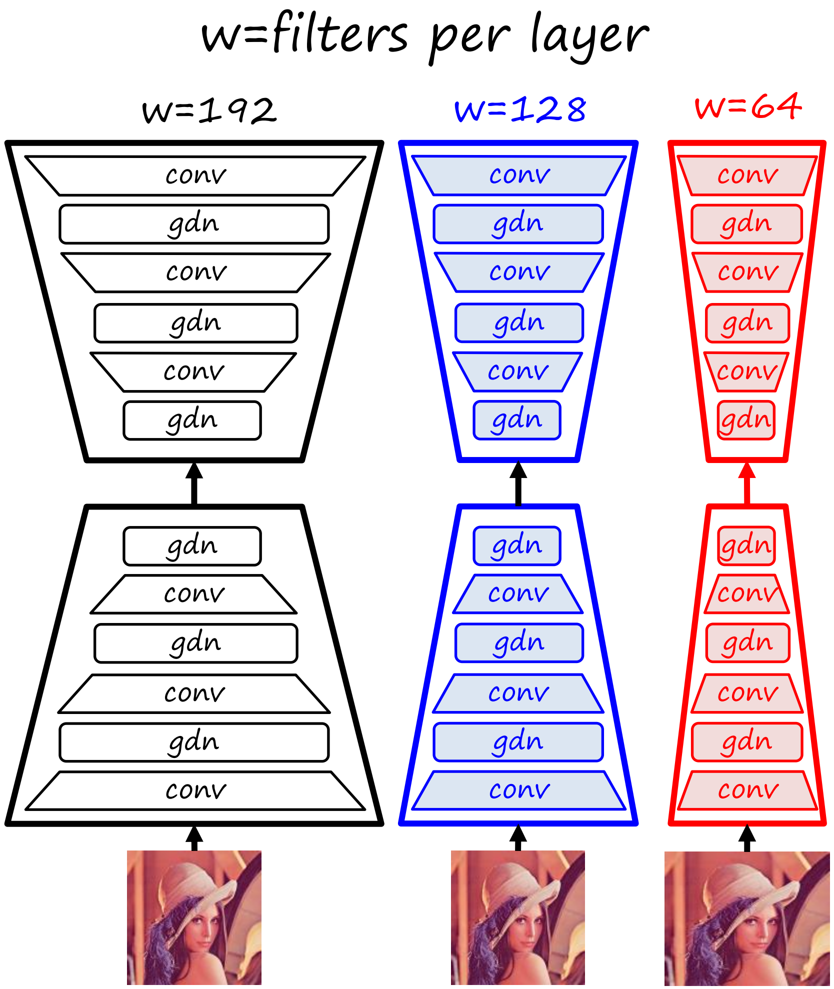


- Minimize rate ✓
- Minimize distortion ✓
- Variable rate ✓

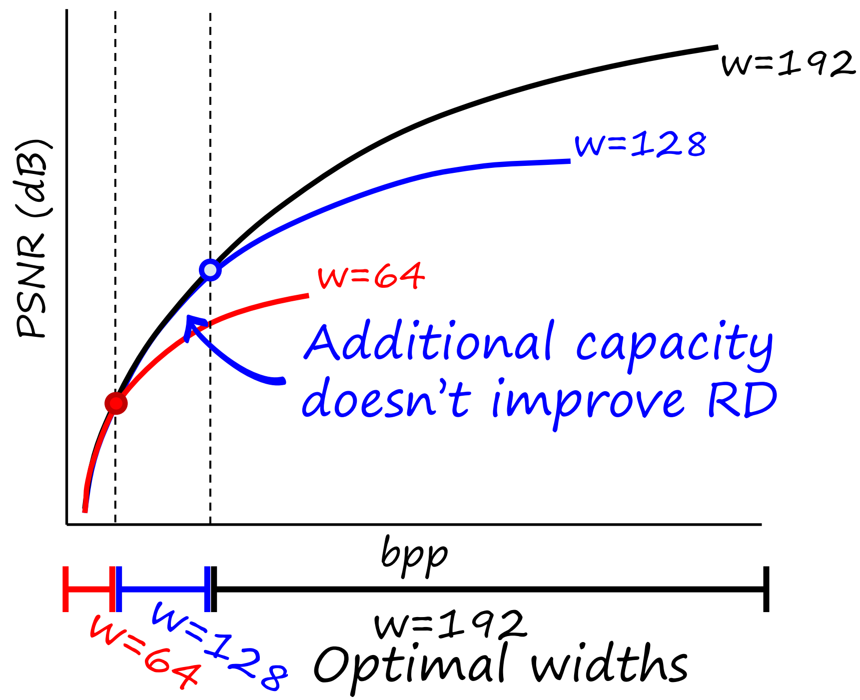
- Low memory ✗
- Low computation ✗
- Low latency ✗

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

Model capacity and rate-distortion



There is a **minimal capacity** for every RD tradeoff

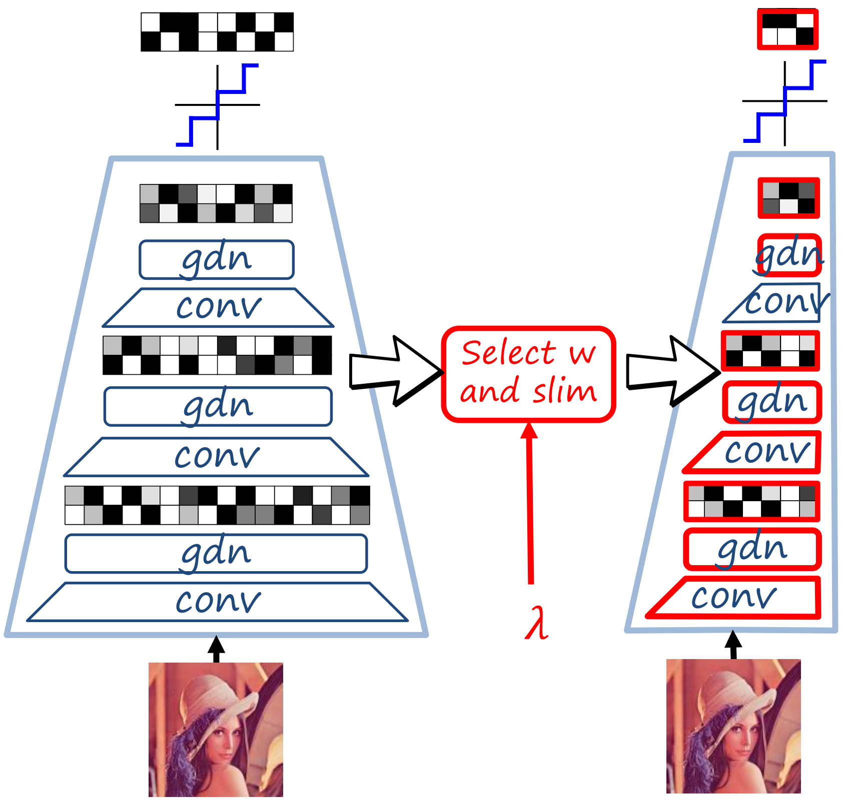


Lower w results in less memory and computation!!

Slimmable compressive autoencoder

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

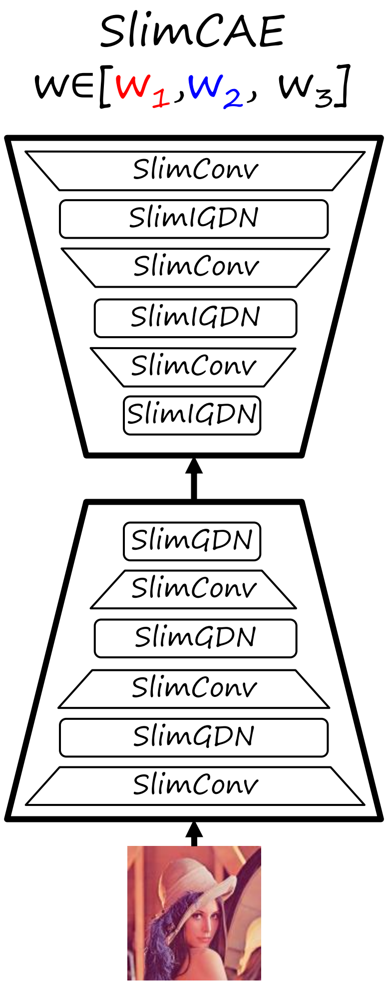
Slimming [SlimCAE]



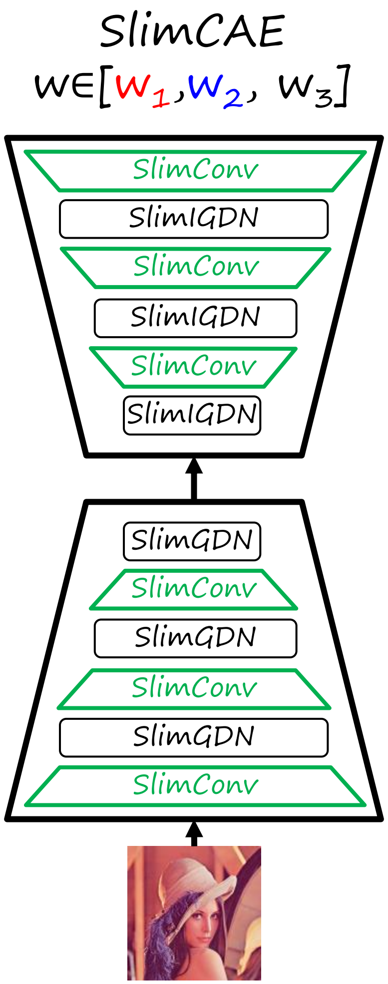
- Minimize rate ✓
- Minimize distortion ✓
- Variable rate ✓
- Lower memory ✓
- Lower computation ✓
- Lower latency ✓

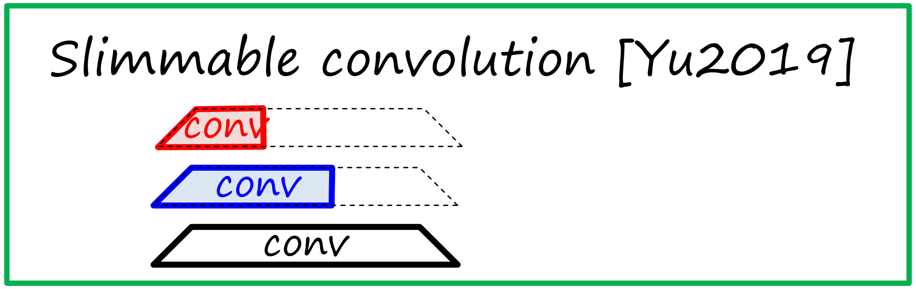
(for low-mid rates)

Slimmable layers in SlimCAE

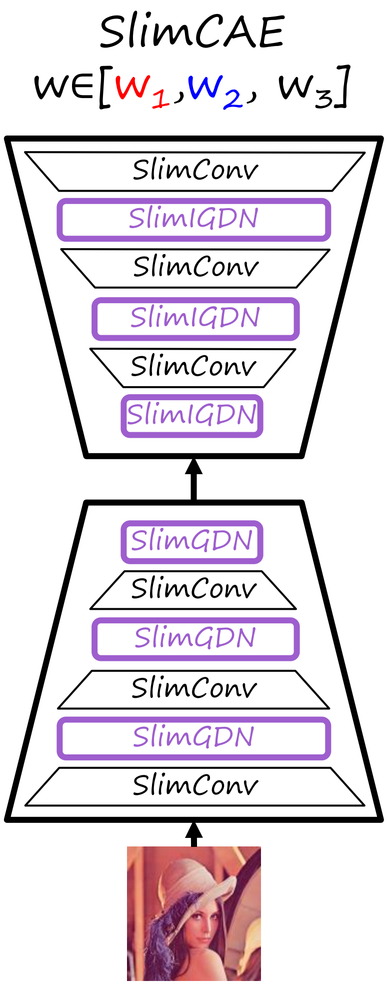


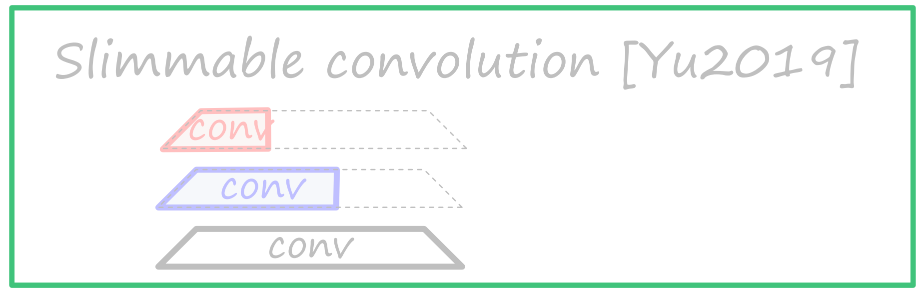
Slimmable layers in SlimCAE

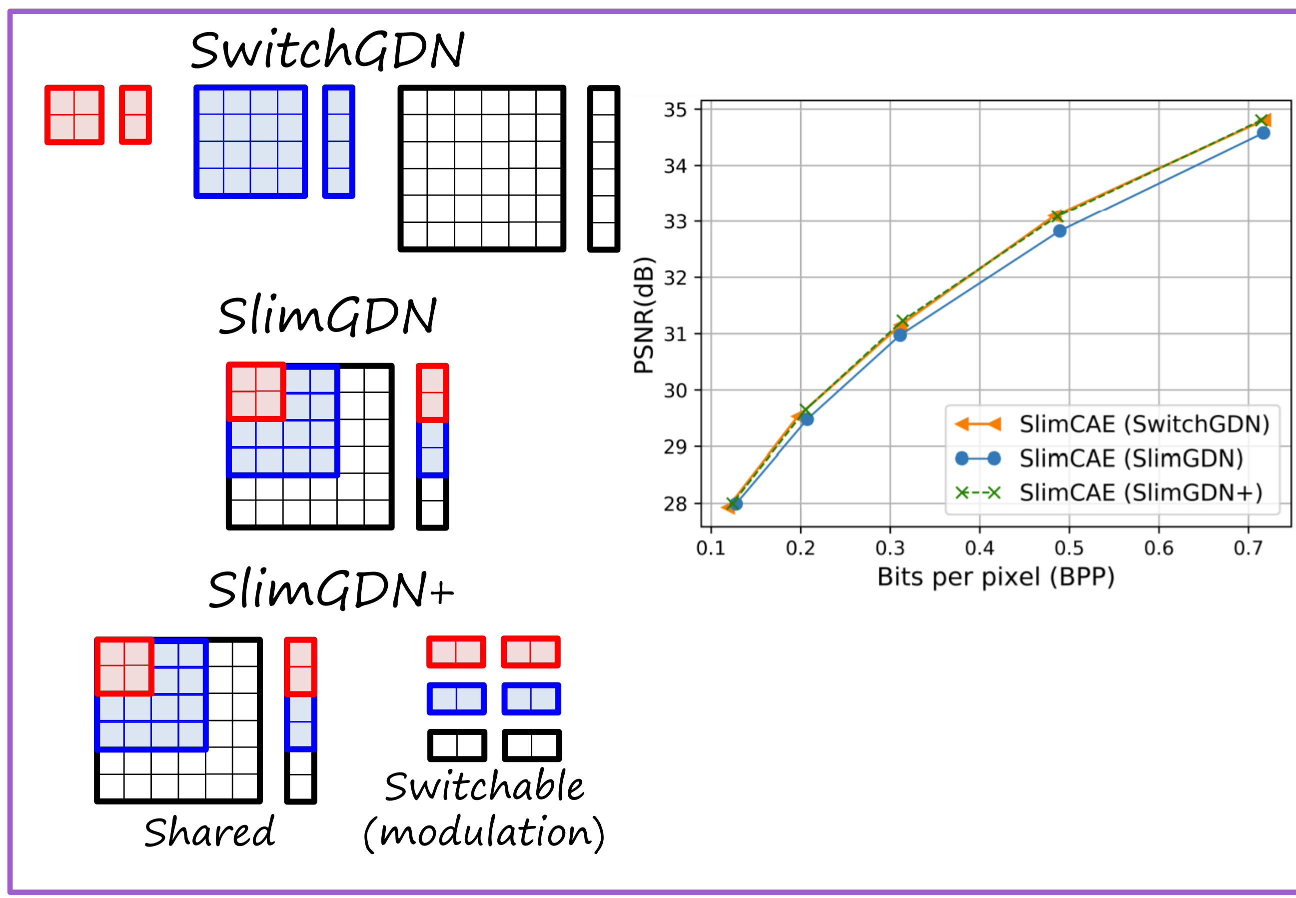




Slimmable layers in SlimCAE



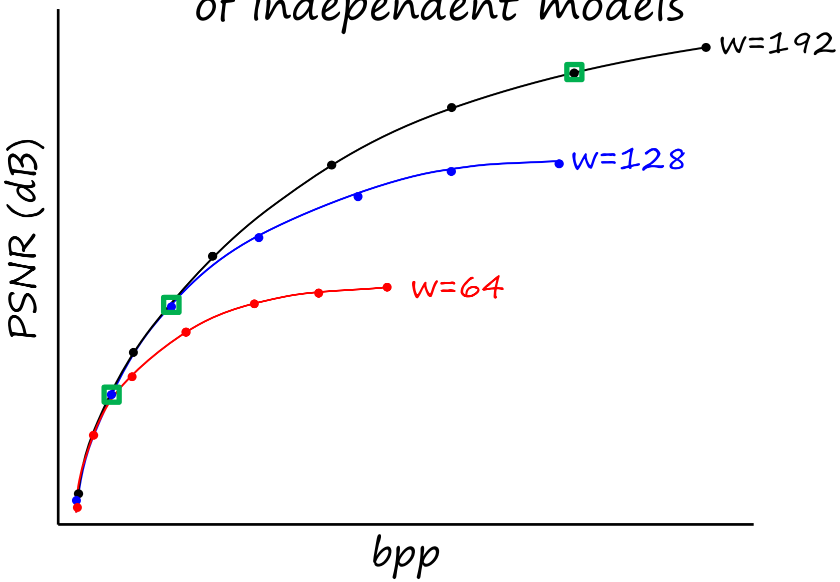




Training SlimCAE

Problem: we need the optimal λs to train the SlimCAE

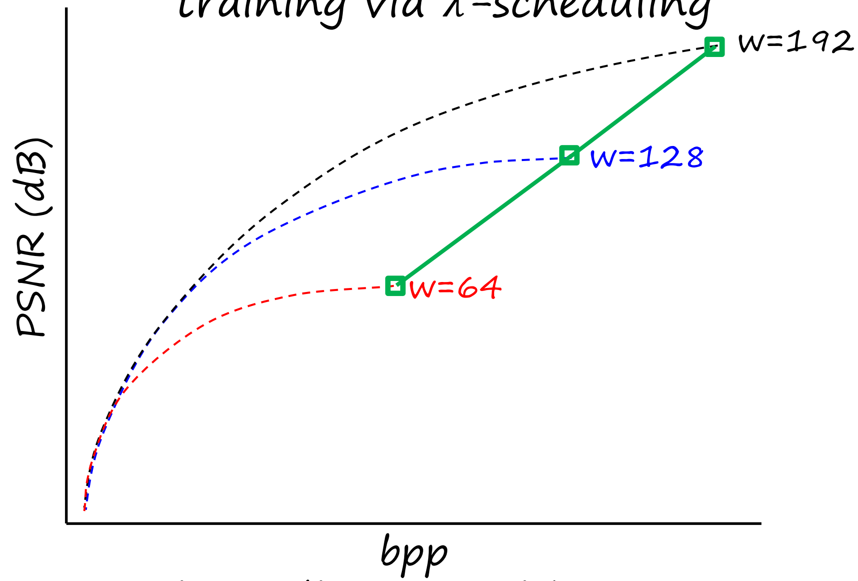
Estimate from RD curves of independent models



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

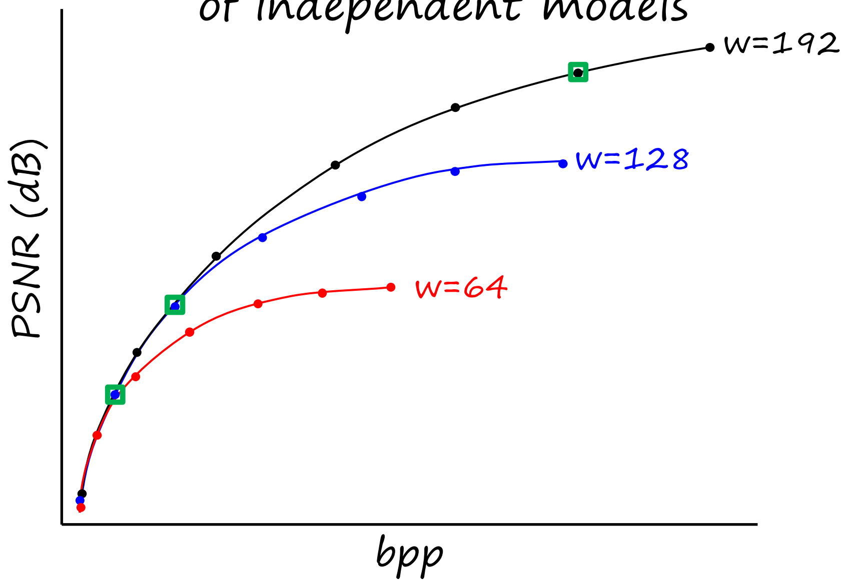


1. Train a SlimCAE with $\lambda_1 = \lambda_2 = \lambda_3$
2. While not converged do
   - Update λ s according to schedule
   - Optimize CAE

Training SlimCAE

Problem: we need the optimal λs to train the SlimCAE

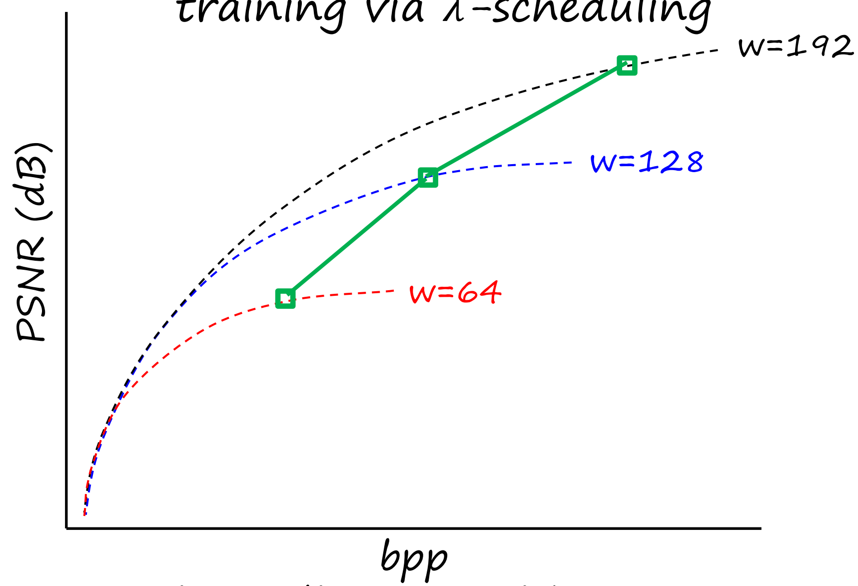
Estimate from RD curves of independent models



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

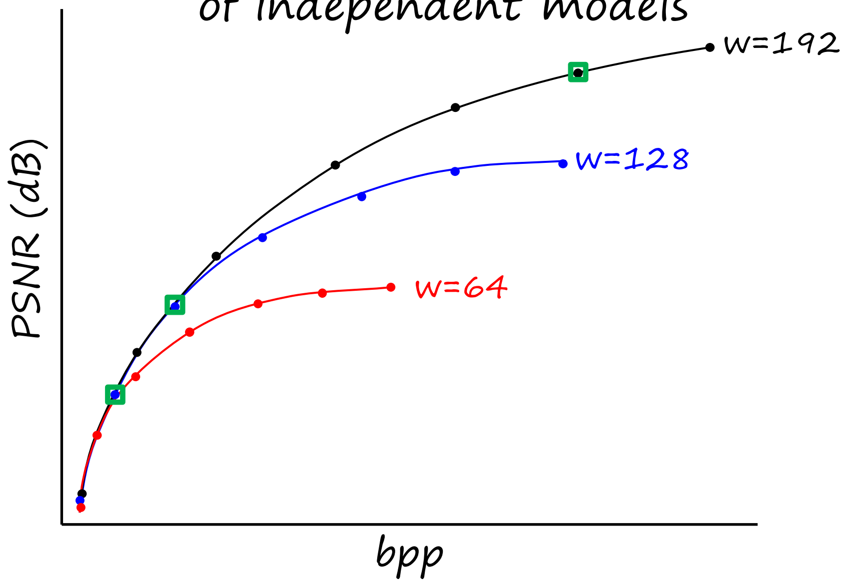


1. Train a SlimCAE with $\lambda_1 = \lambda_2 = \lambda_3$
2. While not converged do
   - Update λ s according to schedule
   - Optimize CAE

Training SlimCAE

Problem: we need the optimal λs to train the SlimCAE

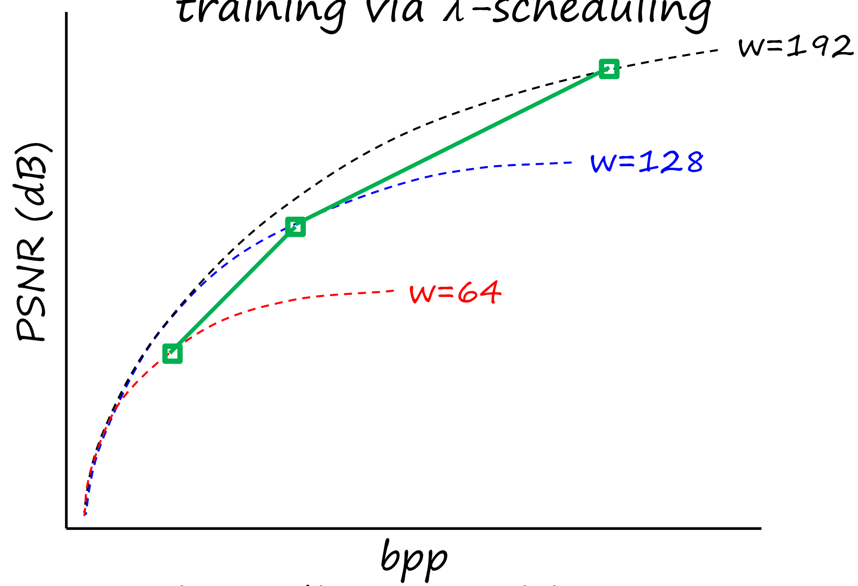
Estimate from RD curves of independent models



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

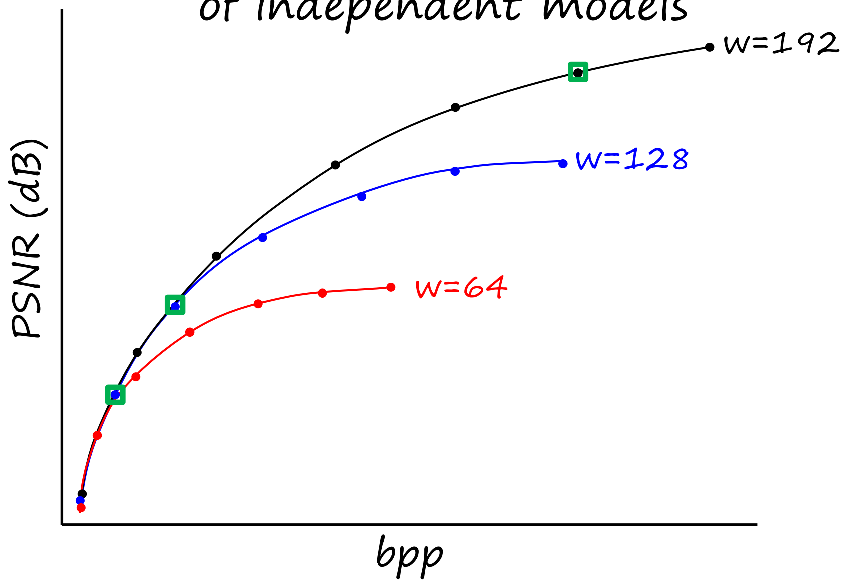


1. Train a SlimCAE with $\lambda_1 = \lambda_2 = \lambda_3$
2. While not converged do
   - Update λs according to schedule
   - Optimize CAE

Training SlimCAE

Problem: we need the optimal λs to train the SlimCAE

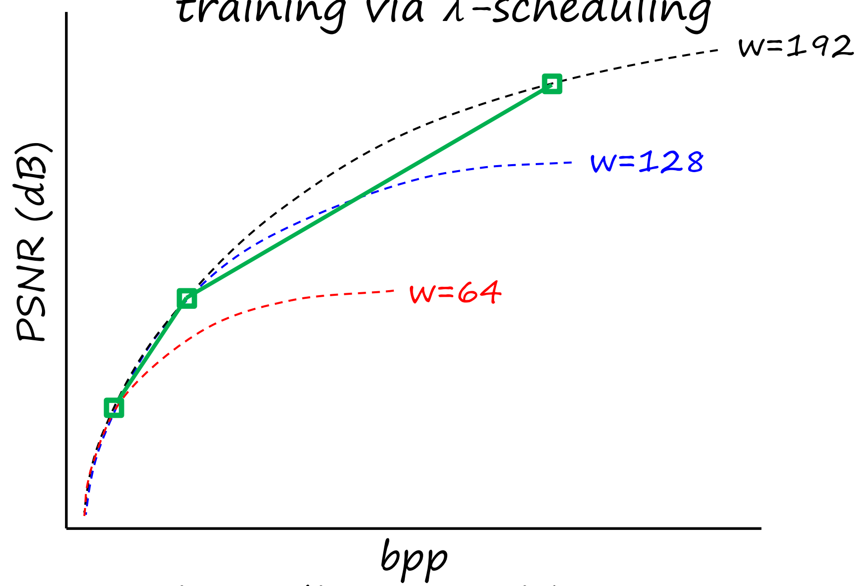
Estimate from RD curves of independent models



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!

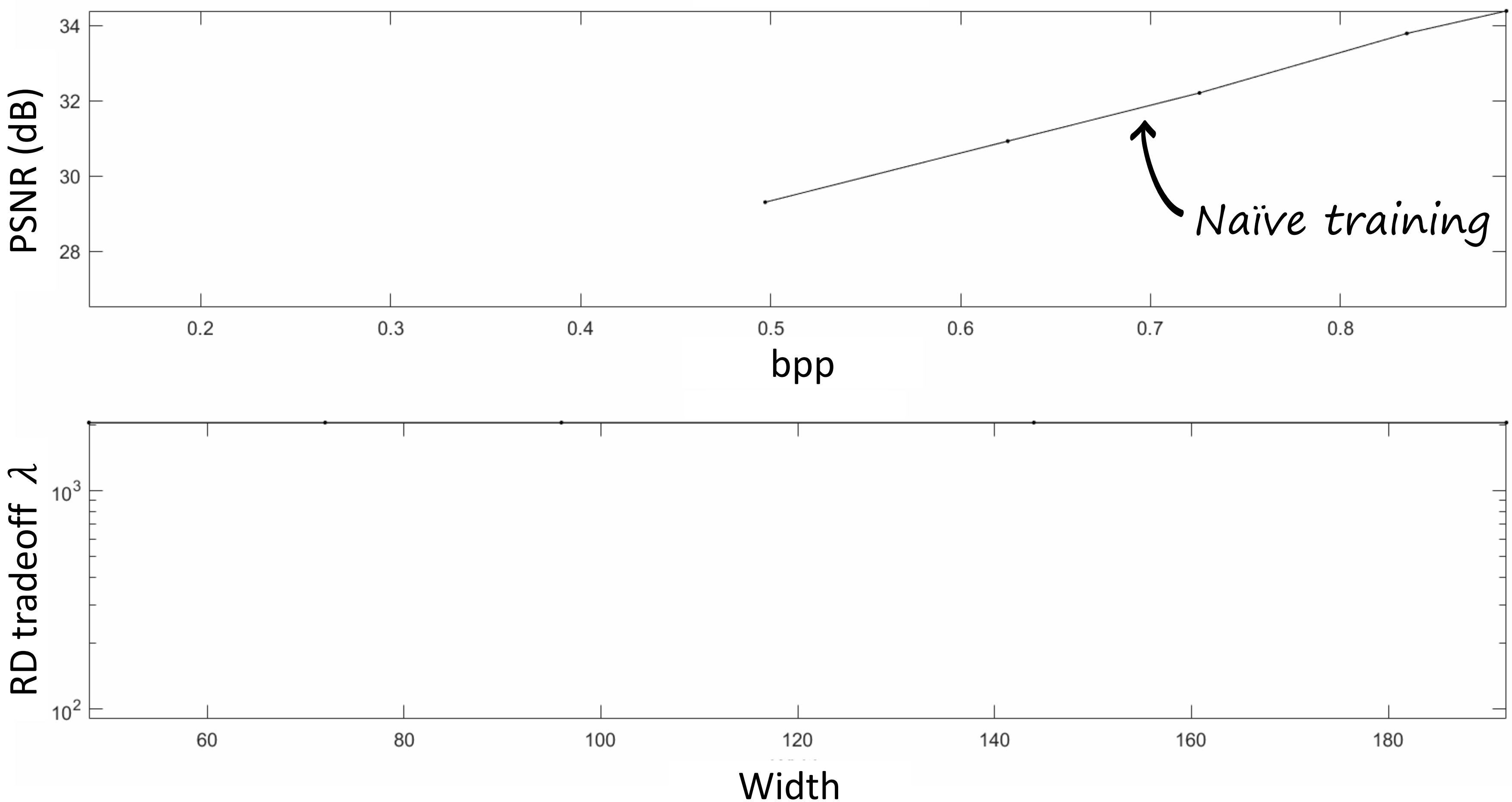
Automatically estimate during training via λ-scheduling



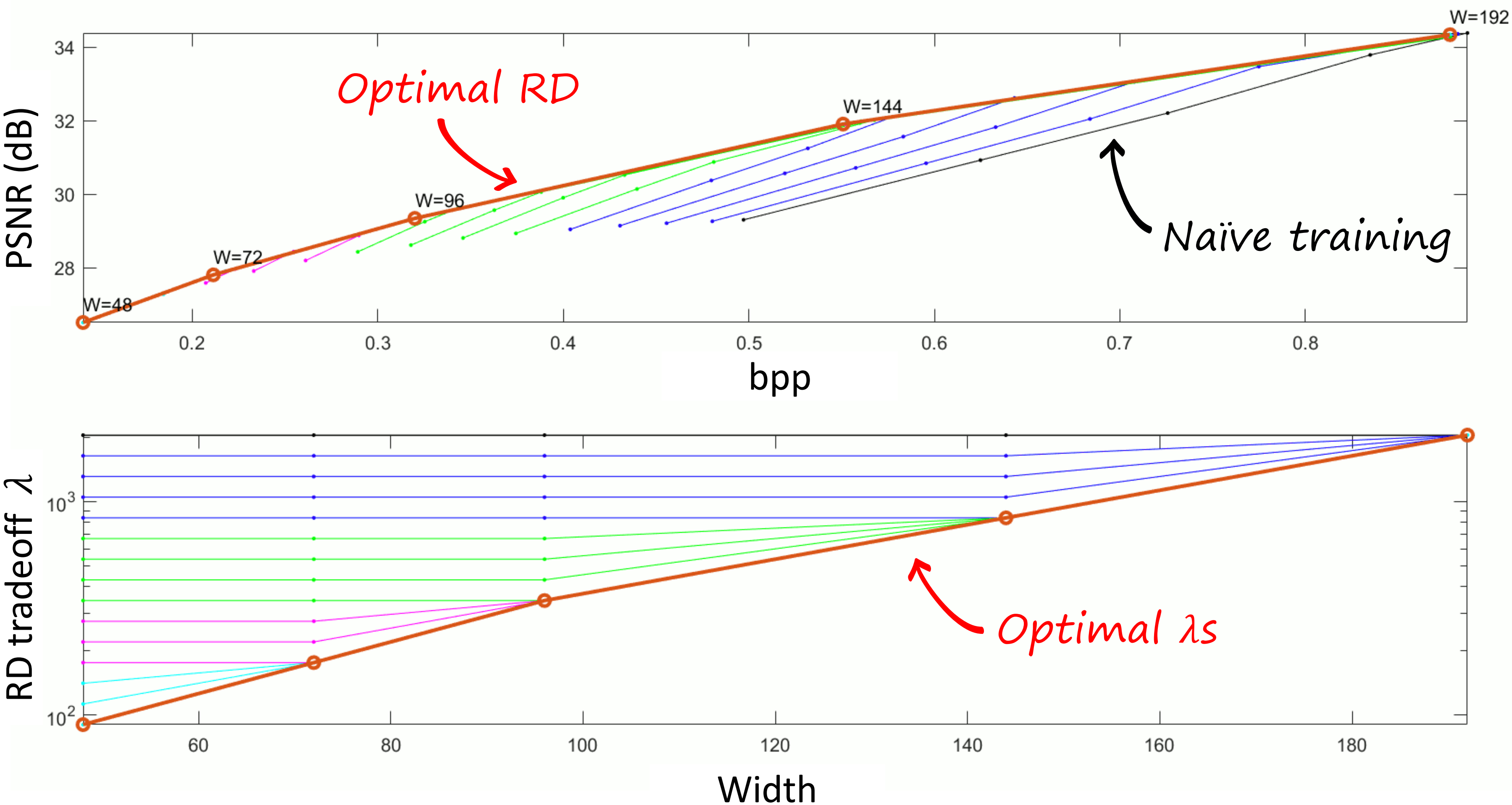
1. Train a SlimCAE with $\lambda_1 = \lambda_2 = \lambda_3$
2. While not converged do
   - Update λ s according to schedule
   - Optimize CAE

Directly train one model!

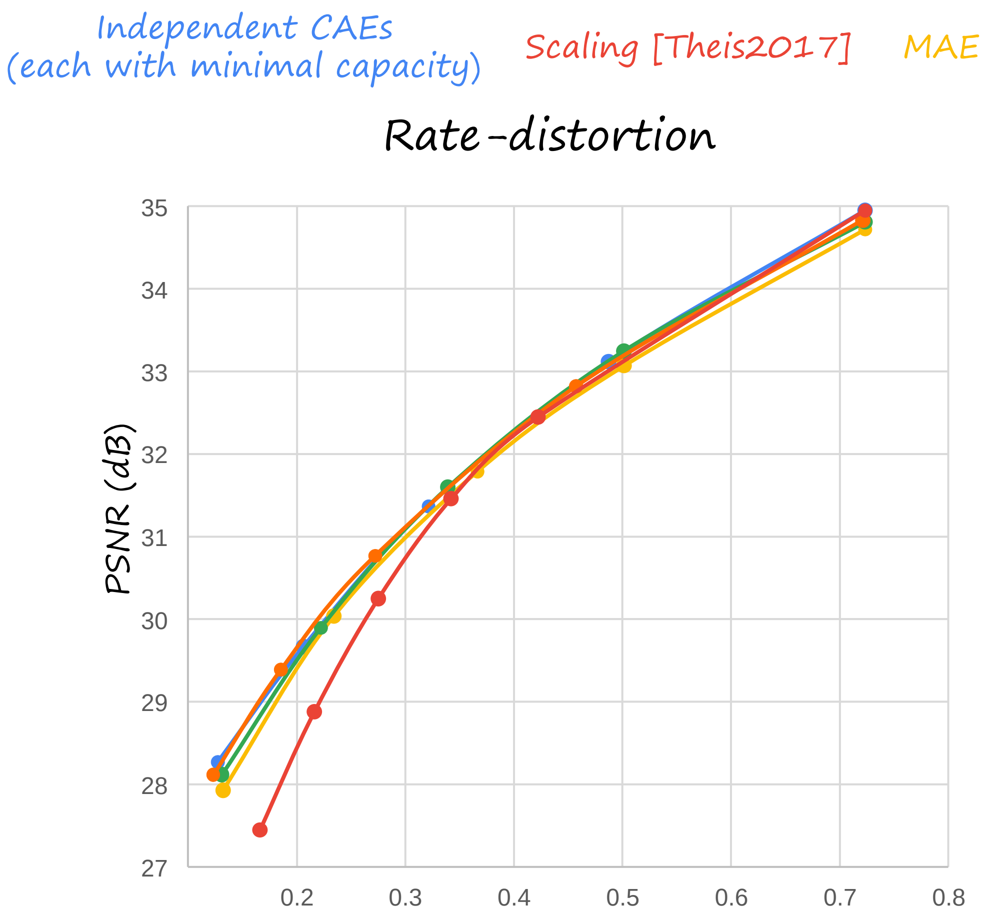
λ-scheduling. Example

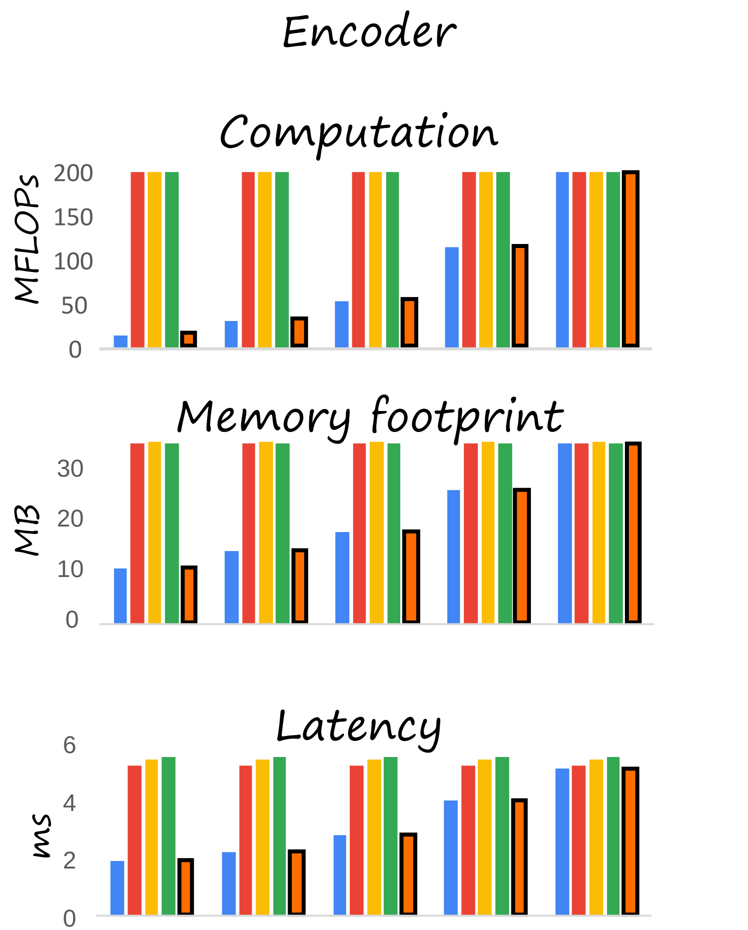


λ-scheduling



Performance comparison





Thanks!

https://arxiv.org/abs/2103.15726

https://github.com/FireFYF/SlimCAE

Fei Yang

Luis Herranz

Yongmei Cheng

Mikhail Mozerov

CVC

UAB Universitat Autònoma de Barcelona

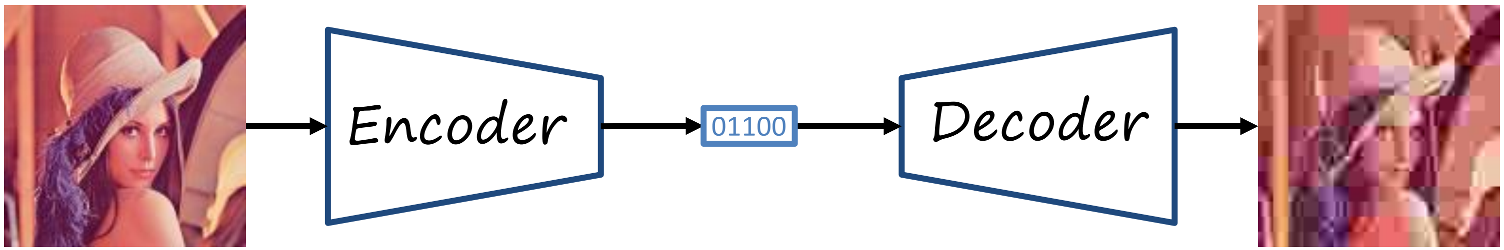
DANICE: Domain adaptation without forgetting in neural image compression

Sudeep Katakol[1,(2)], Luis Herranz[2,3], Fei Yang[2,3,4], Marta Mrak[5]

[1]University of Michigan, Ann Arbor, [2]Computer Vision Center, [3]Universitat Autònoma de Barcelona, [4]Northwestern Politechnical University, [5]BBC R&D

CLIC 2021 (@CVPR 2021)

Towards **practical** neural image compression



Main objectives
- Minimize rate
- Minimize distortion

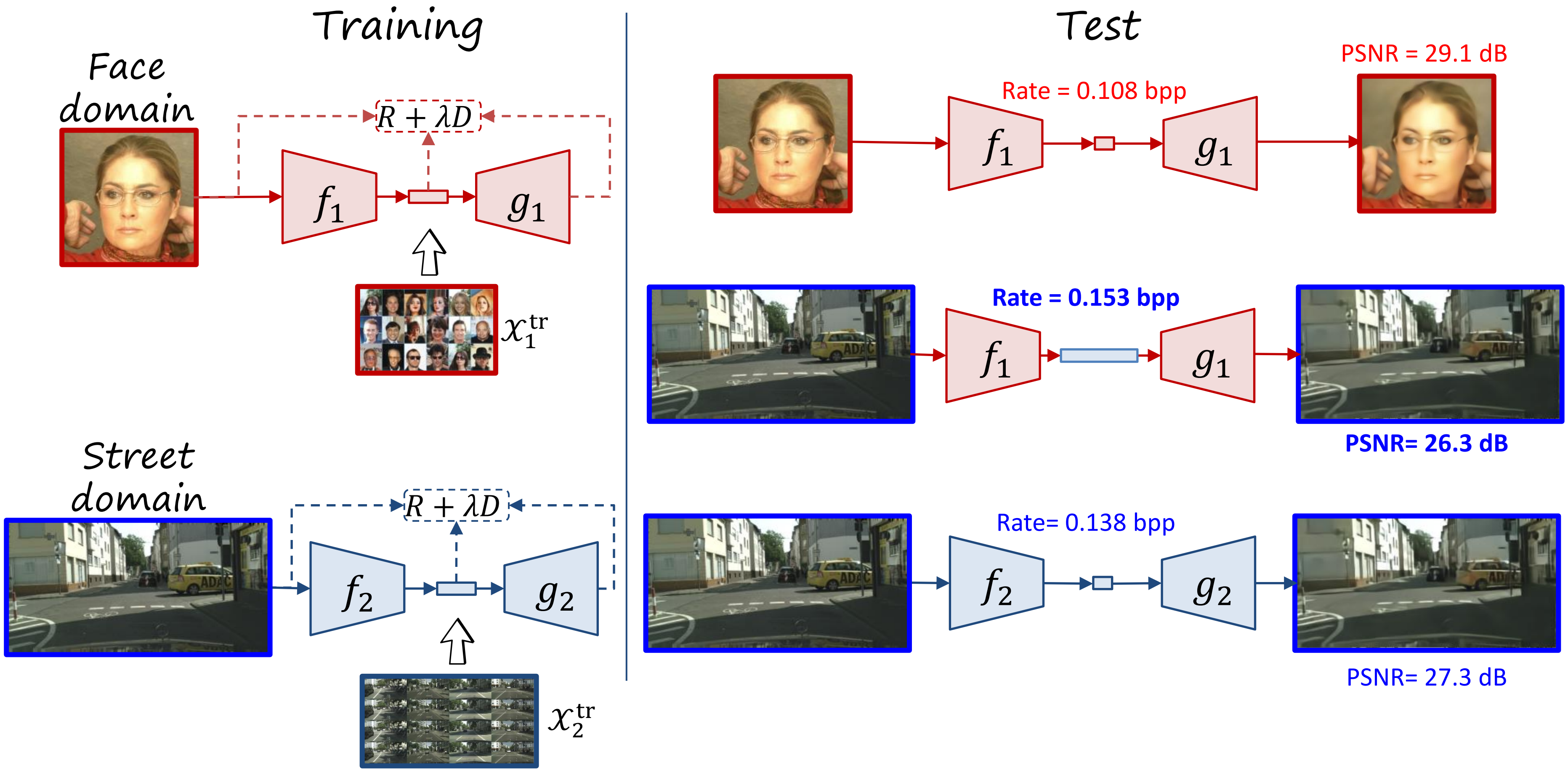
Practical objectives
- Variable rate
- Low memory
- Low computation
- Low latency

Other practical considerations
- **Domain-specific codecs**
  (e.g. videoconference, screencast)
- **Backward/forward compatibility**
  (with legacy formats and encoders/decoders)

DANICE
[CLIC2021]

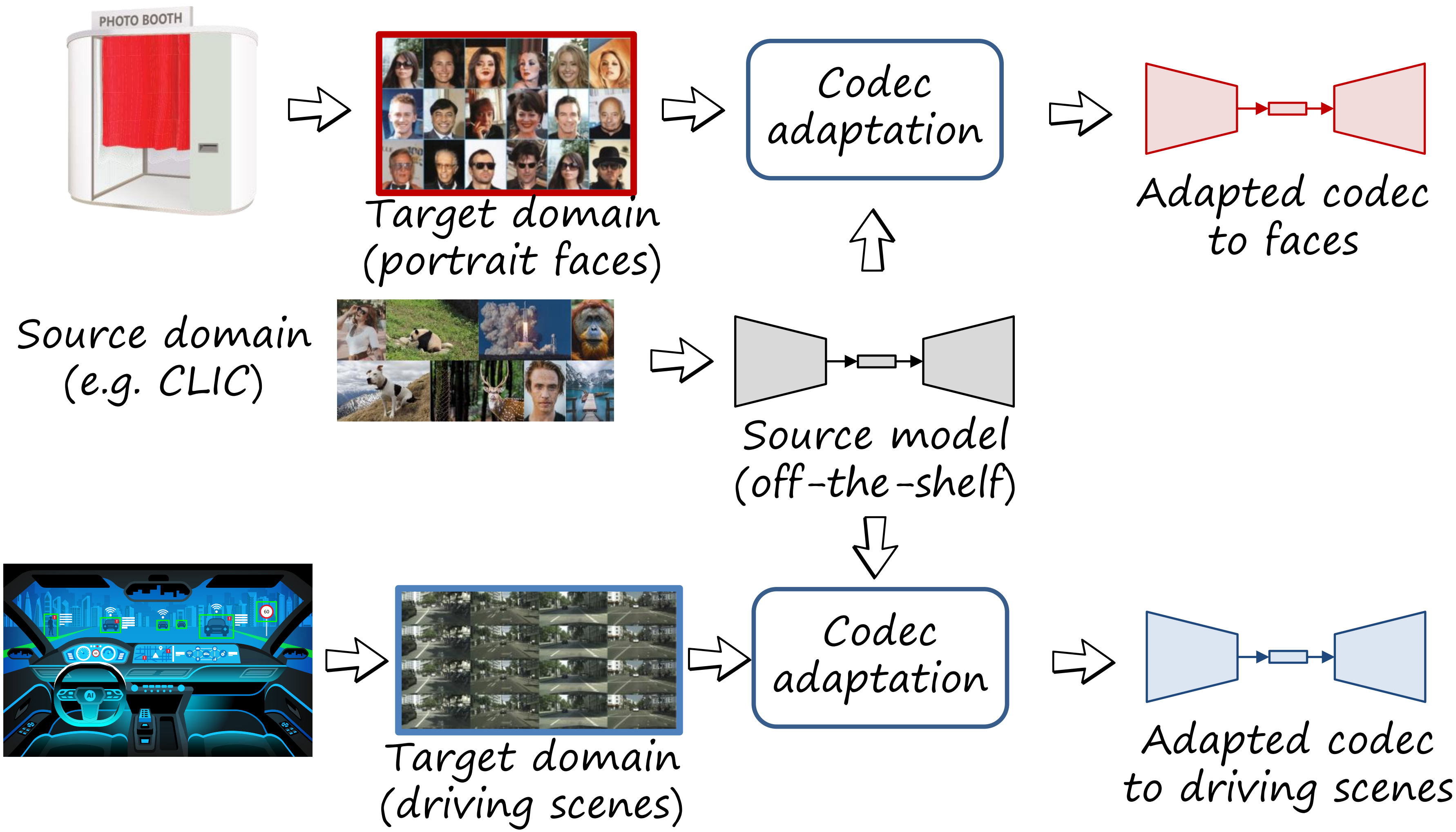
Rate-distortion optimality of learned codecs

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

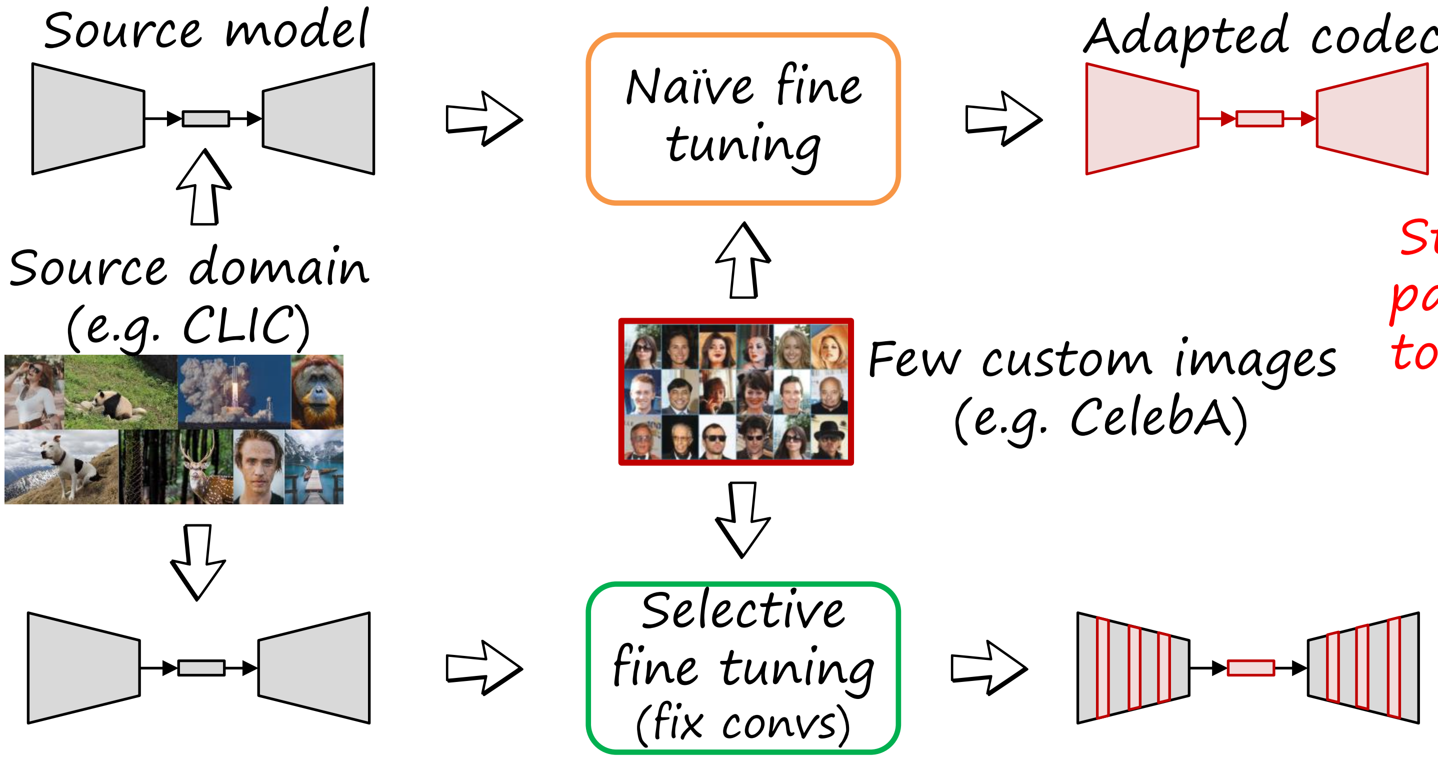


Domain Adaptation in Neural ImageomprEssion (DANICE)

Learned codecs can be customized with user content to specific domains
Problem: usually we don't have enough custom data; training is expensive
Solution: transfer pre-trained codecs



Domain adaptation via fine tuning



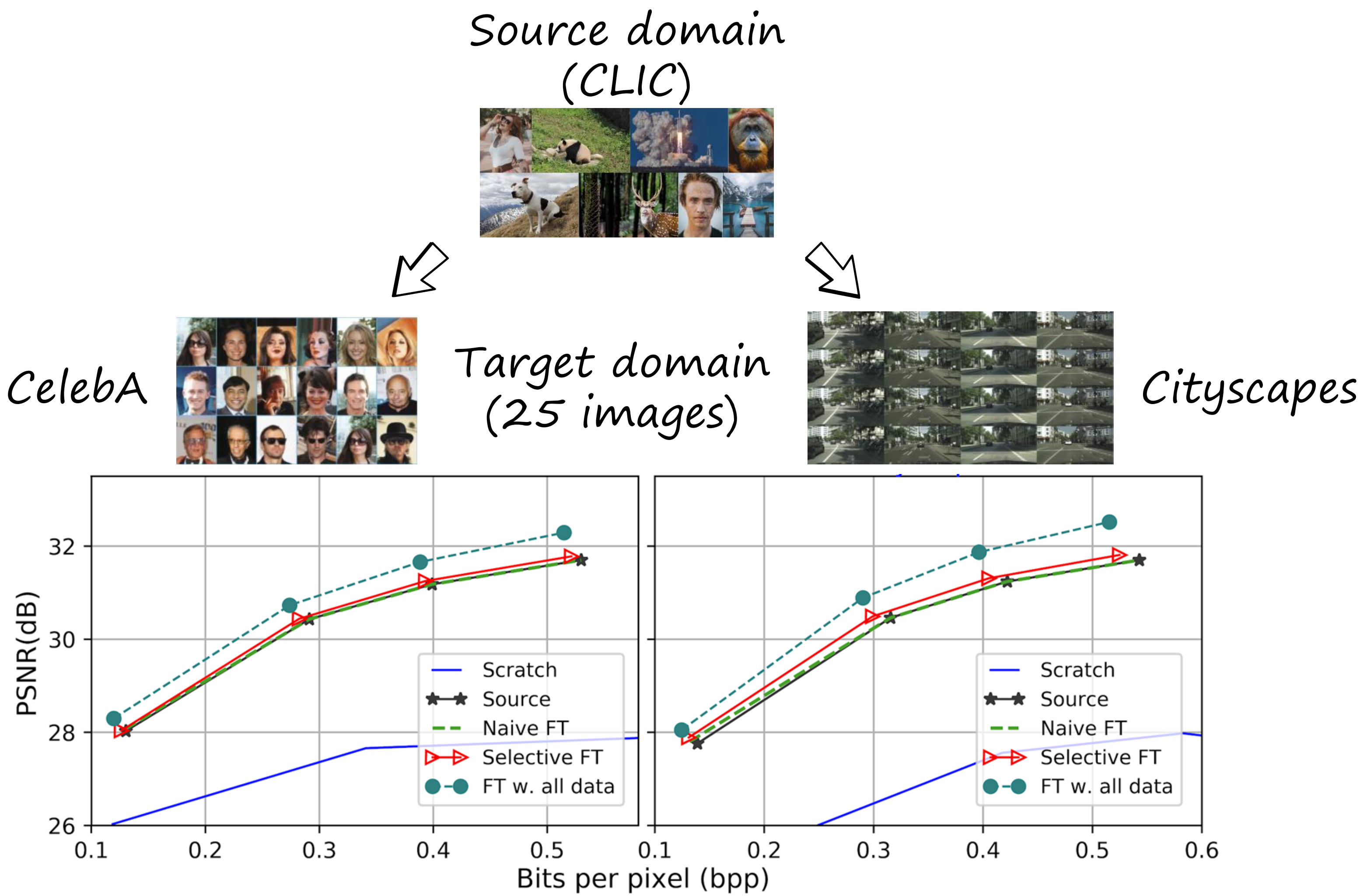
Still too many parameters for too few images

Experiments

	CLIC → CelebA		CLIC → Cityscapes	
Source model	19.24		23.93	
Number of target images	Naïve fine tuning	Selective fine tuning	Naïve fine tuning	Selective fine tuning
10	19.24	**16.46**	22.96	**17.54**
25	18.76	**14.93**	18.44	**15.79**
50	15.59	**13.73**	16.29	**15.33**

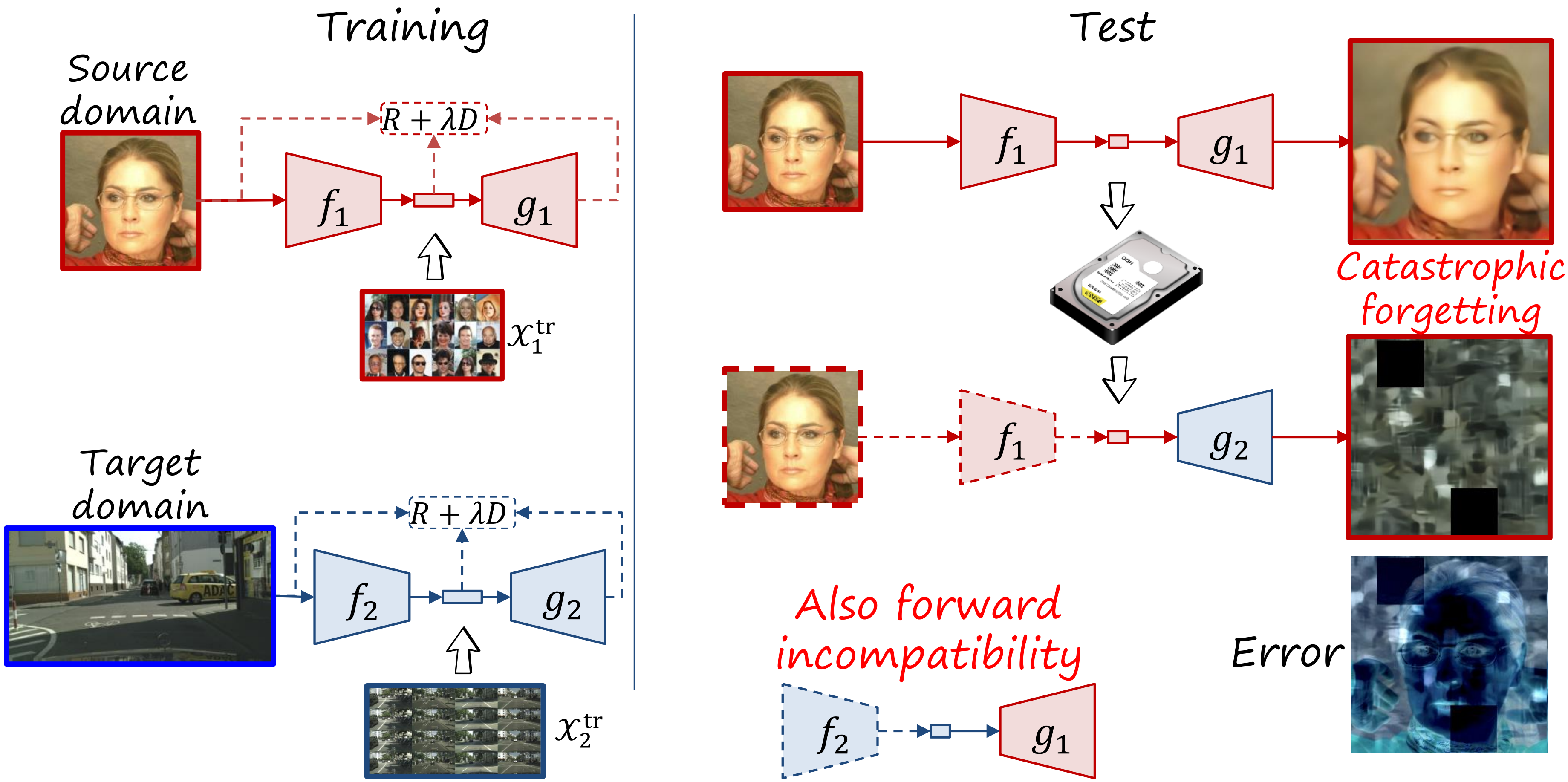
BD-rate (reference: training with all target data)

Domain adaptation via fine tuning



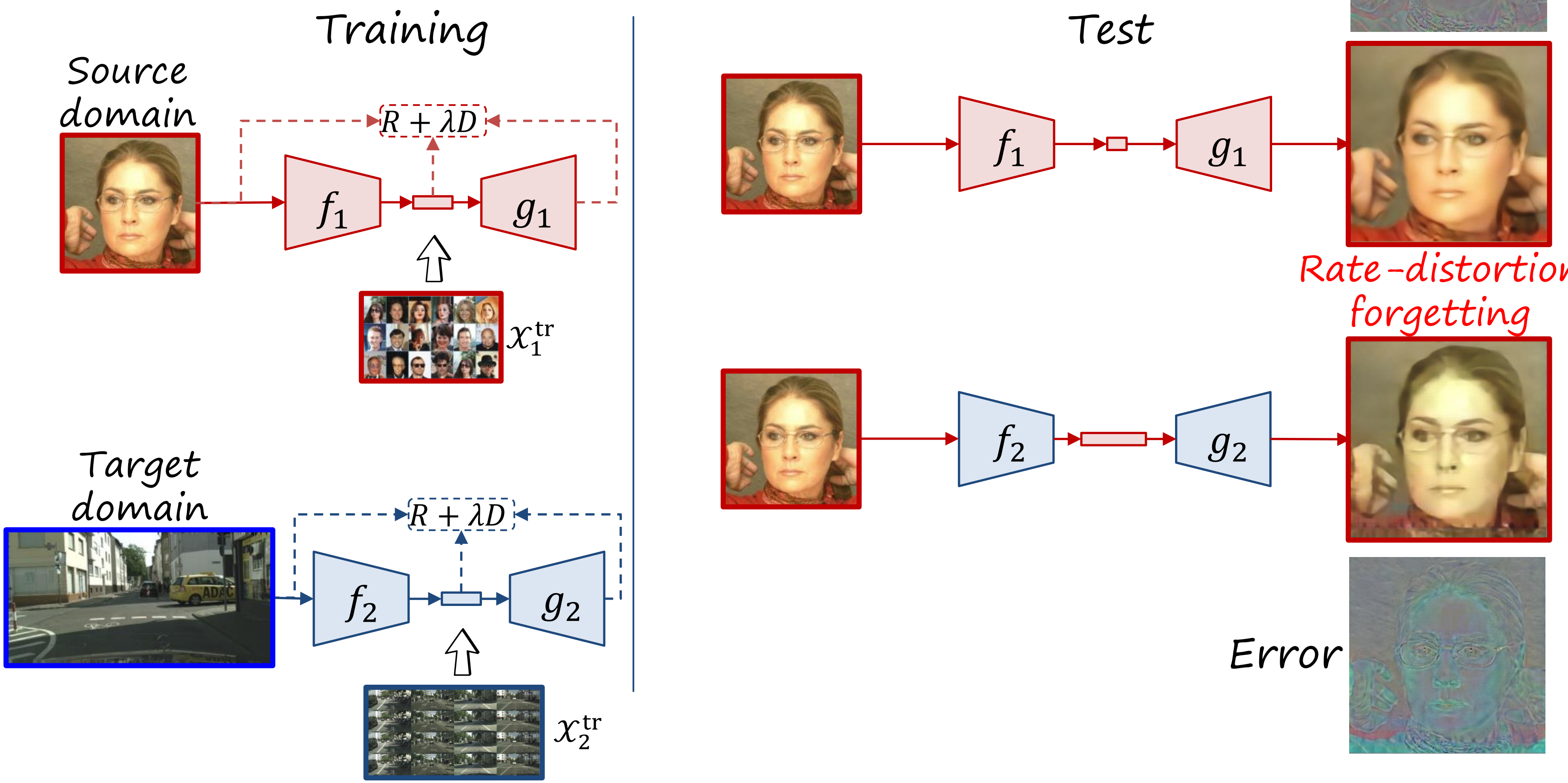
Backward incompatibility with legacy bitstreams: catastrophic forgetting

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



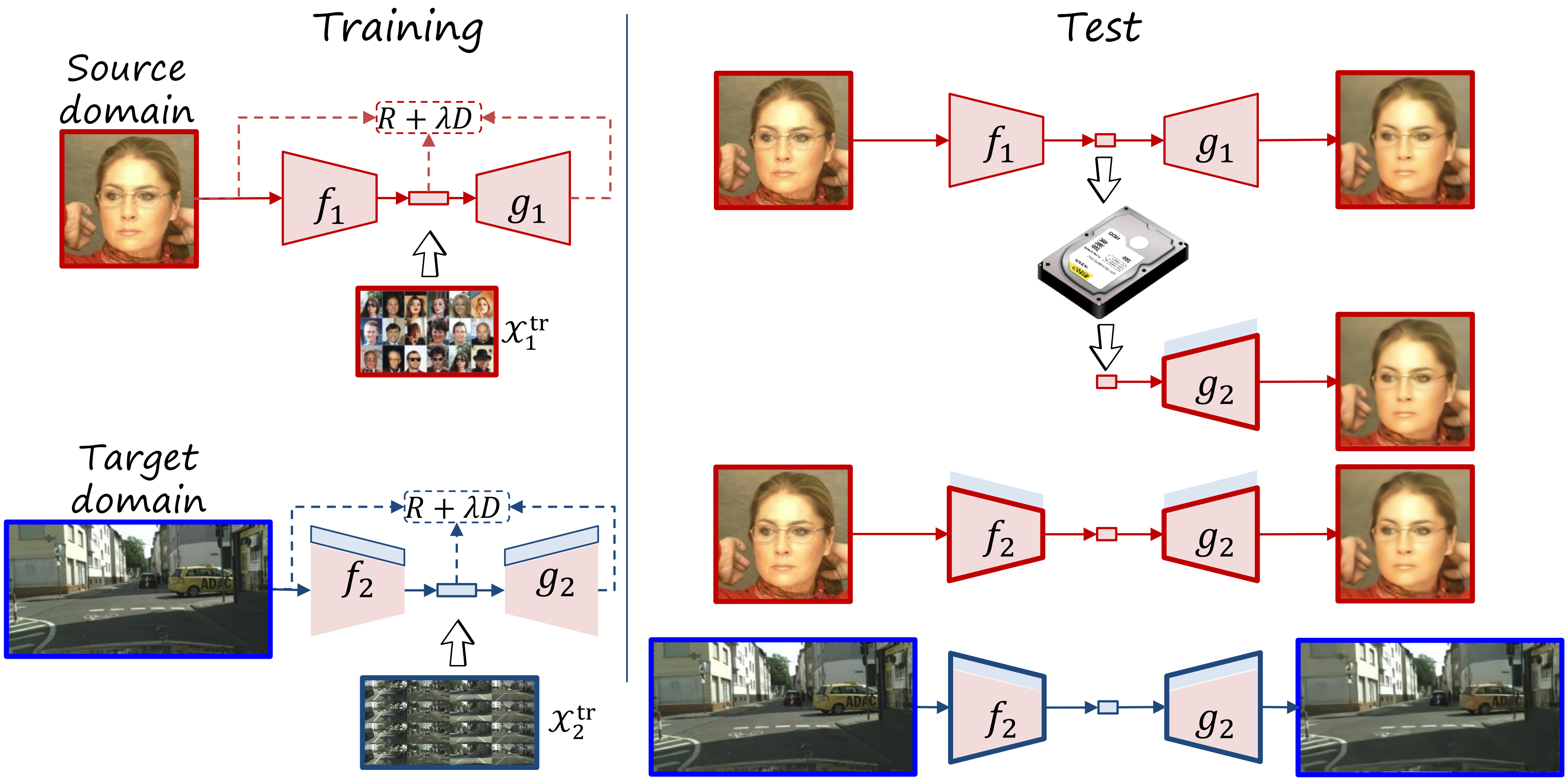
Rate-distortion forgetting

Encoding-decoding latent spaces aligned, but suboptimal
(i.e. bitstream syntax compatible, yet degraded)



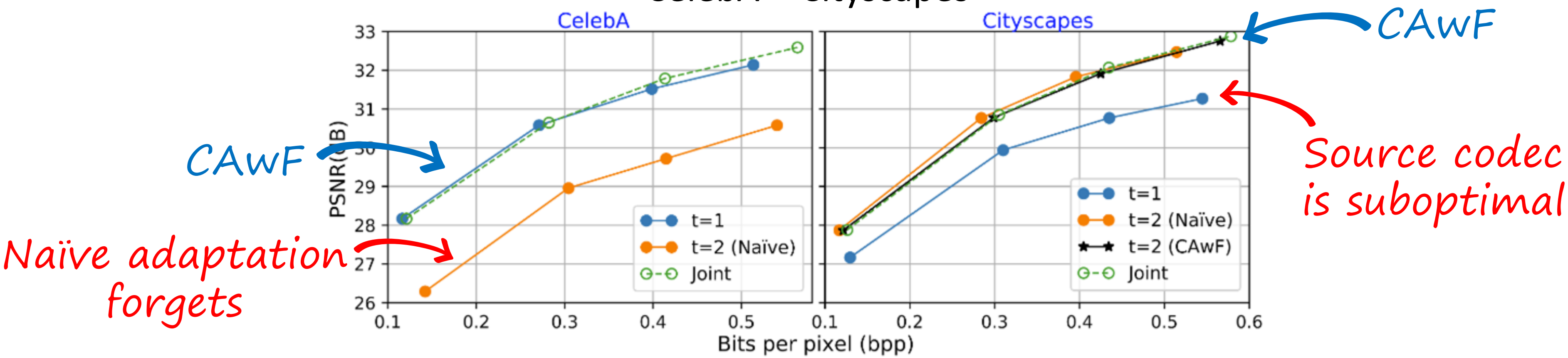
Codec adaptation without forgetting (CAwF)

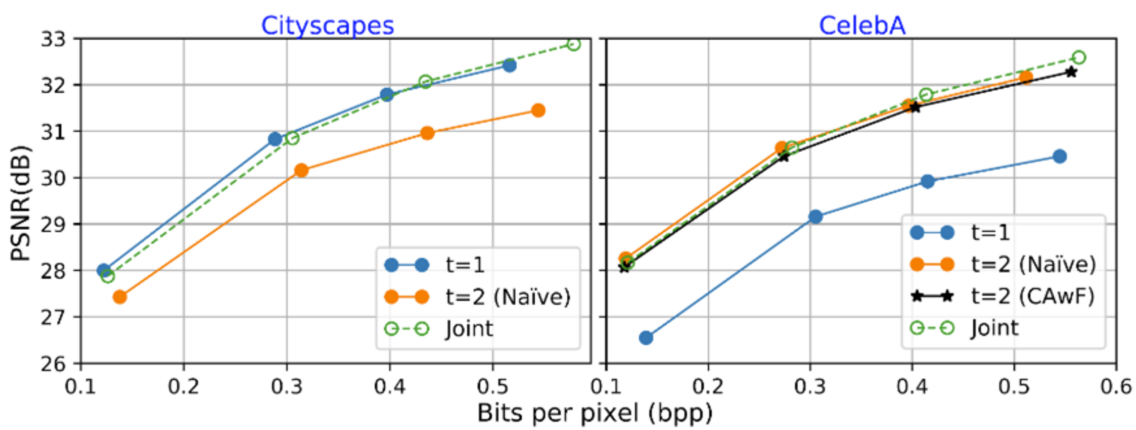
Freeze source codec, and learn target codec as an enhancement layer
Drawback: adds additional parameters



Codec adaptation without forgetting (CAwF)

CelebA→Cityscapes

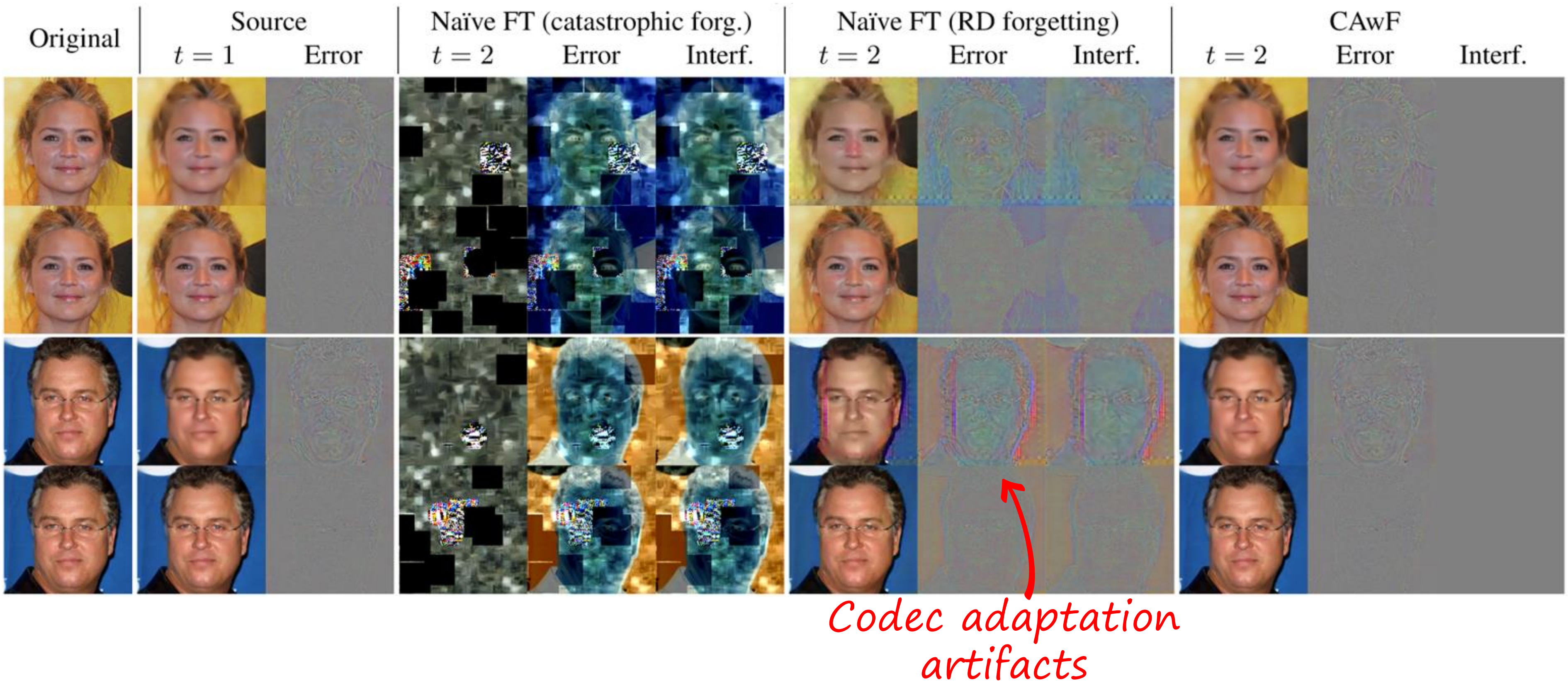




Cityscapes→CelebA

Codec adaptation without forgetting (CAwF)

CelebA→Cityscapes
(source domain)



Thanks!

https://arxiv.org/abs/2103.15726


Sudeep Katakol


Luis Herranz


Fei Yang


Marta Mrak

MICHIGAN CVC UAB Universitat Autònoma de Barcelona BBC | Research & Development