Investigations on NN-based Intra Prediction

Maria Meyer, Friedhelm Hamann, Christian Rohlfing

Problem Statement: Neural Networks for Intra Prediction

Neural network-based intra mode(s) for hybrid video codecs:

Network related:

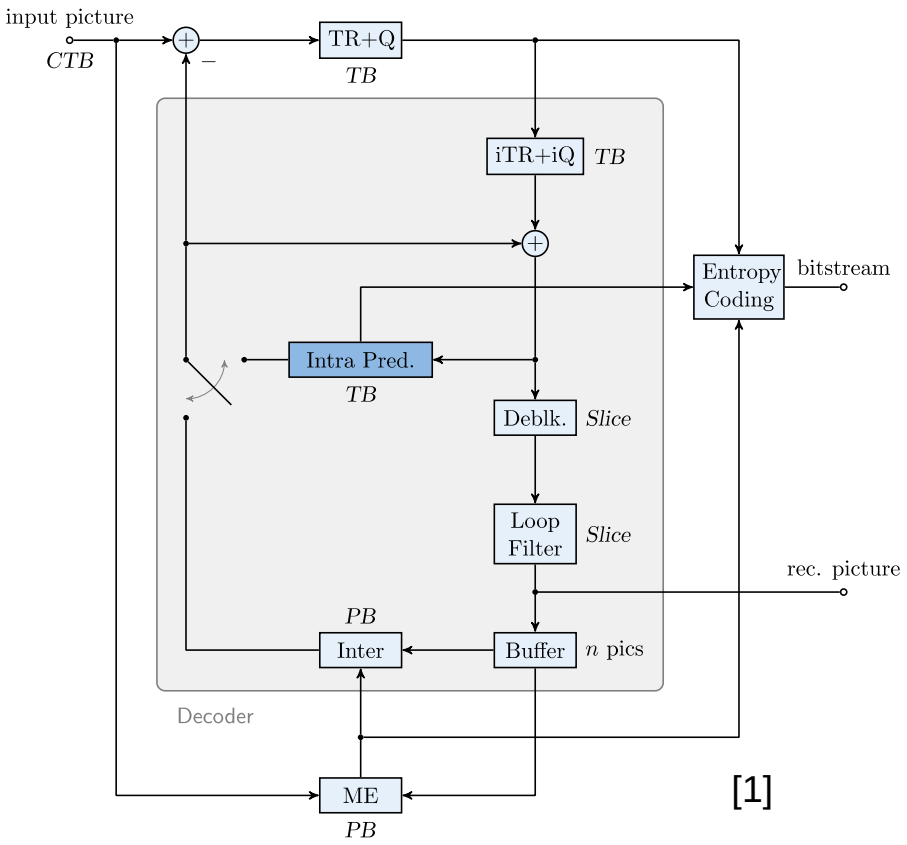
- Architecture
- Optimization Function
- Training methods

Integration related:

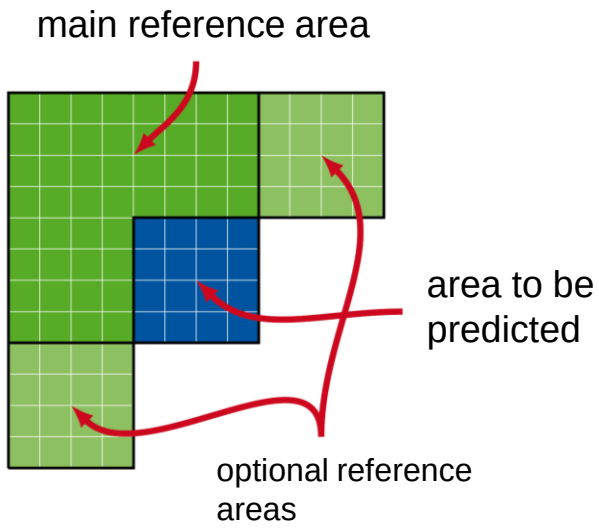
- Signaling
- Rate-distortion decisions

Real-time application constraint:

- Low complexity



[1]



[1] M. Wien, High Efficiency Video Coding – Coding Tools and Specification. Berlin, Heidelberg: Springer, Sept. 2014

Previous Work – Overview

Previous work:

- [2] – [5]: First successful NN-based approaches
- [6], [7]: Matrix-based intra prediction for VVC
- [8] – [10]: New architectures and training methods

Remaining problems:

- Training Data Engineering
- Additional side information
- Reference area size

  and Hyperparameters
- Pruning

[2] M. Meyer, J. Wiesner, J. Schneider and C. Rohlfing, "Convolutional Neural Networks for Video Intra Prediction Using Cross-component Adaptation," *ICASSP 2019, May 2019*
[3] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, “Fully connected network-based intra prediction for image coding,” IEEE Transactions on Image Processing, July 2018
[4] T. Dumas, A. Roumy, and C. Guillemot, “Context-adaptive neural network based prediction for image compression,” CoRR 2018
[5] Y. Hu, W. Yang, S. Xia, and J. Liu, “Optimized spatial recurrent network for intra prediction in video coding,” in VCIP 2018

[6] J. Pfaff, P. Helle, D. Maniry, S. Kaltenstadler, W. Samek, H. Schwarz, D. Marpe, and T. Wiegand: “Neural Network based Intra Prediction for Video Coding”, SPIE 2018
[7] P. Helle, J. Pfaff, M. Schäfer, R. Rischke, H. Schwarz, D. Marpe, and T. Wiegand, “Intra picture prediction for video coding with neural networks,” DCC 2019

[8] F. Brand, J. Seiler, and A. Kaup, “Intra frame prediction for video coding using a conditional autoencoder approach,” PCS 2019
[9] L. Zhu, S. Kwong, Y. Zhang, S. Wang, and X. Wang, “Generative adversarial network based intra prediction for video coding,” IEEE Transactions on Multimedia, 2019
[10] Y. Wang, X. Fan, S. Liu, D. Zhao, and W. Gao, “Multi-scale convolutional neural network based intra prediction for video coding,” IEEE Trans. on Circuits and Systems for Video Technology, 2019

Outline

- Intro

- Part 1 – CNNs
  - Coded Training Data
  - Loss function evaluation
  - Architecture Optimization

- Part 2 – Autoencoders
  - Optimizing conditonal Autoencoders
  - Variational Autoencoders
  - Vector-Quantized-VAEs

- Conclusion and Outlook

Outline

- Intro

- Part 1 – CNNs
  - Coded Training Data
  - Loss function evaluation
  - Architecture Optimization

- Part 2 – Autoencoders
  - Optimizing conditonal Autoencoders
  - Variational Autoencoders
  - Vector-Quantized-VAEs

- Conclusion and Outlook

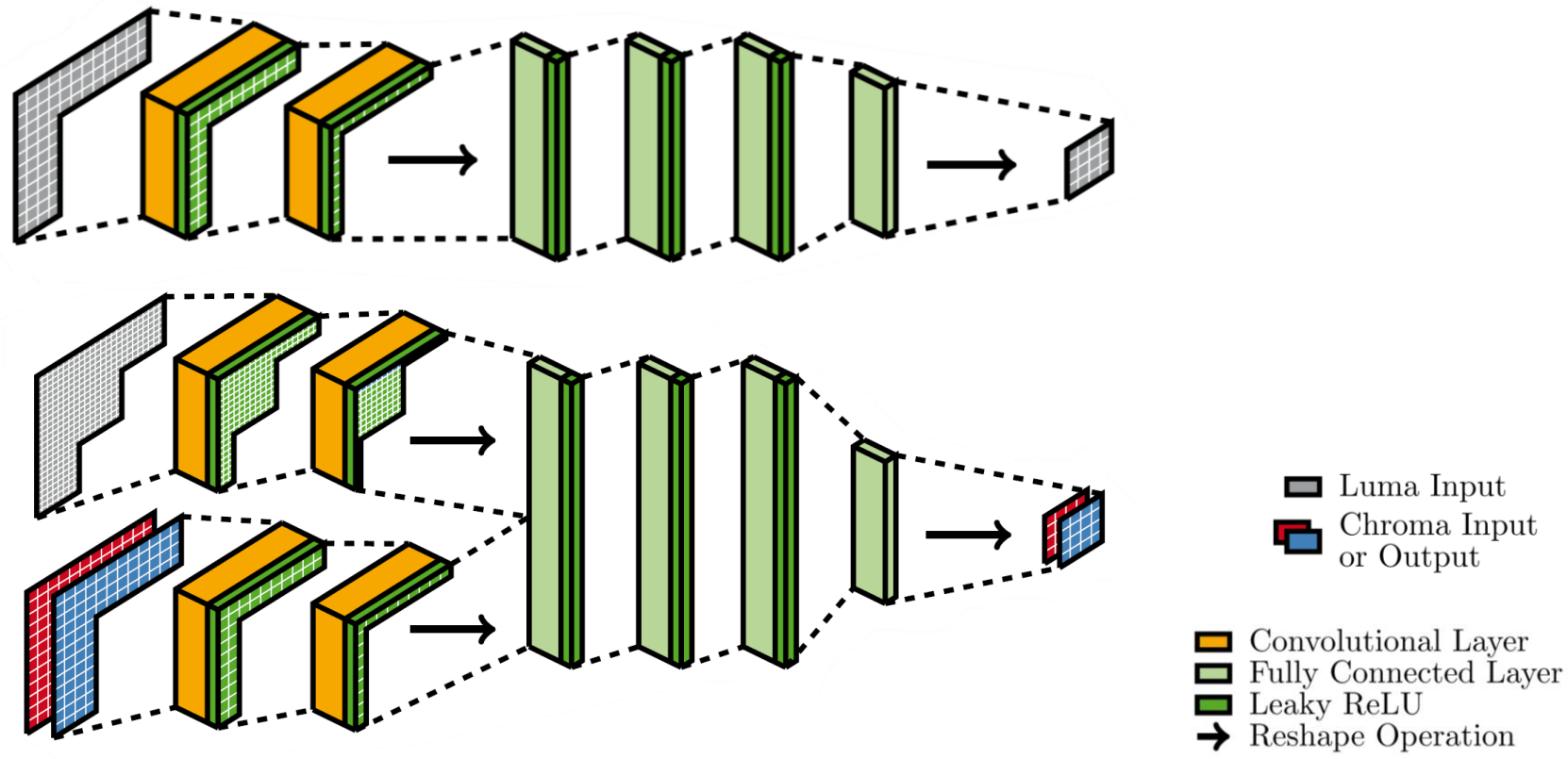
Previous Work – Prediction Network

Architecture:

- Convolutional & dense layers
- For chroma with luma input
- Four reference lines

Training:

- SATD Loss function
- Variance-based filtering of training set



[2] M. Meyer, J. Wiesner, J. Schneider and C. Rohlfing, "Convolutional Neural Networks for Video Intra Prediction Using Cross-component Adaptation," *ICASSP 2019, May 2019*

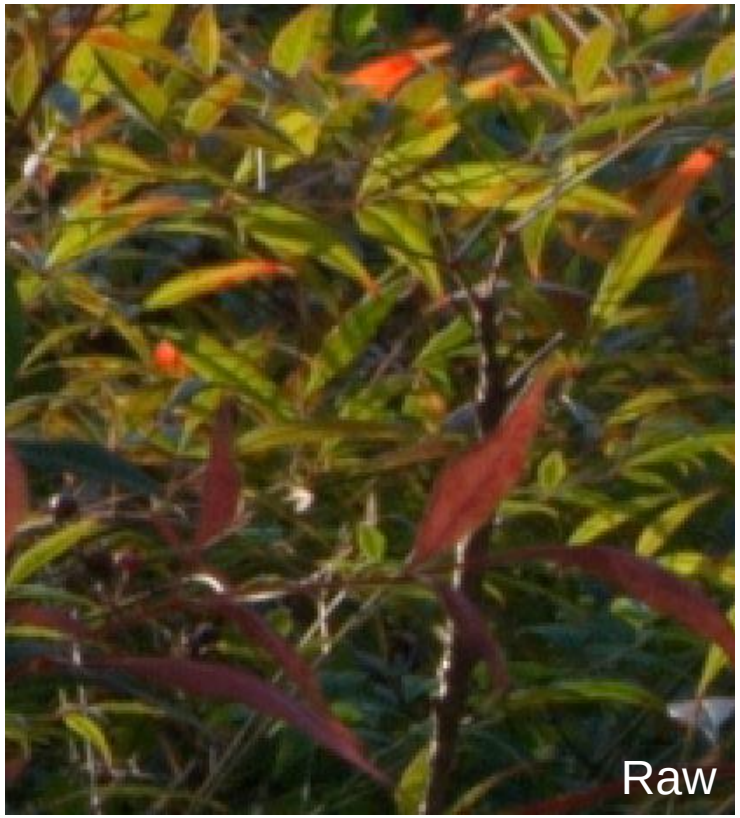
Training Improvements – Coding Artefacts

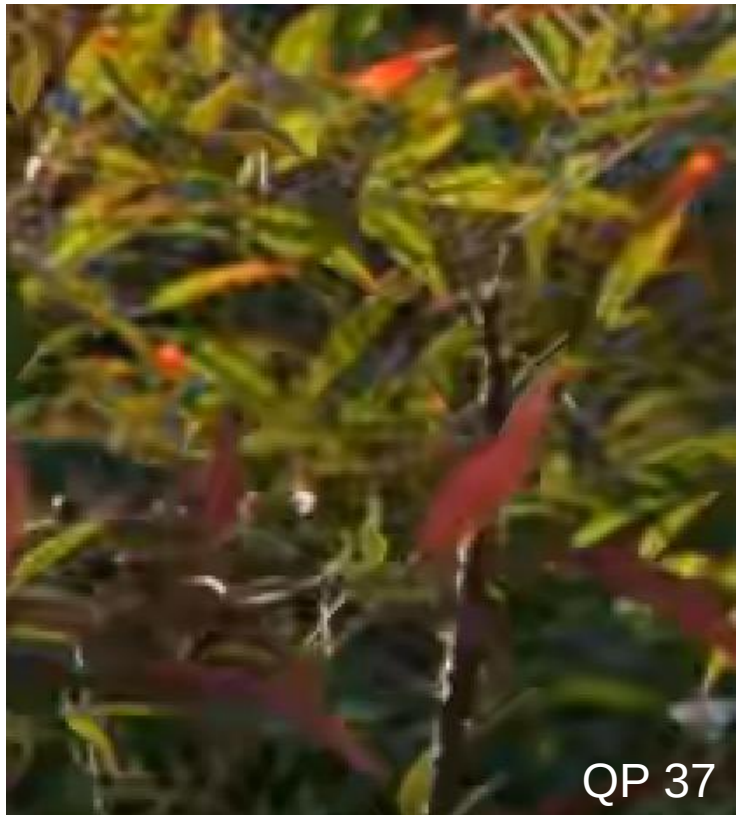
So far:

- no coding artefacts in training set, but always when inferring

Test:

- training with different features
  - all of the samples coded
  - half of the samples coded
  - none of the samples coded
- Label always uncoded
- Coded parts of the samples mixture of QPs 22, 27, 32 and 37





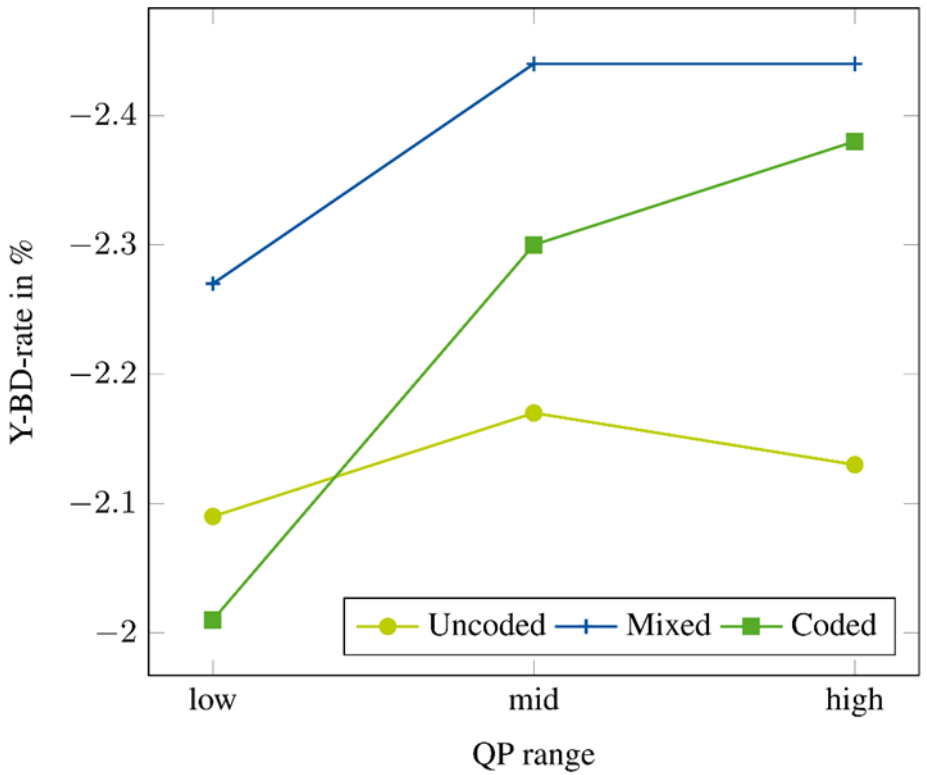
Training Improvements – Coding Artefacts

So far:

- no coding artefacts in training set, but always when inferring

Test:

- training with different features
  - all of the samples coded
  - half of the samples coded
  - none of the samples coded
- Label always uncoded
- Coded parts of the samples mixture of QPs 22, 27, 32 and 37



Training Procedure

Loss function

- Comparison: SATD, DCT and Log

$$L_{\mathrm{SATD}} = b^{-2} \sum |\mathbf{T}_{HAD} \cdot \mathbf{R}|$$

$$L_{\mathrm{DCT}} = b^{-2} \sum |\mathbf{T}_{\mathrm{DCT}} \cdot \mathbf{R}|$$

$$L_{\mathrm{LOG}} = b^{-2} \sum f(\mathbf{T}_{\mathrm{DCT}} \cdot \mathbf{R})$$

$$\text{with } f(x) = |x| + \frac{\alpha}{1+e^{-\beta|x|-\gamma}}$$

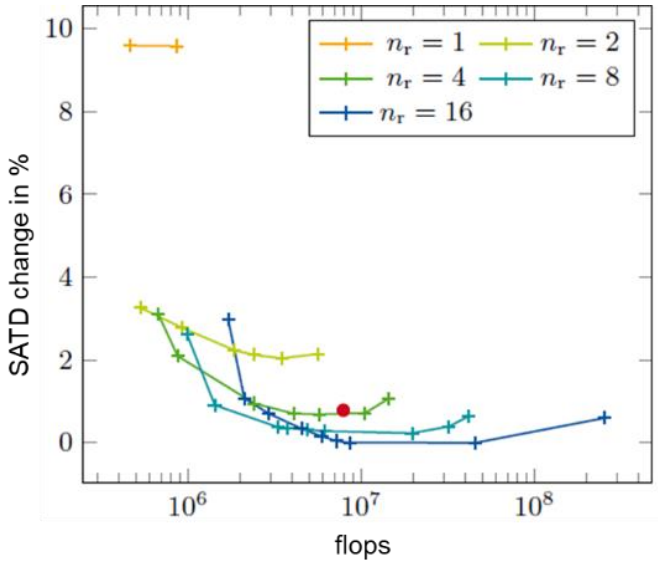
L = loss, $\mathbf{R}$ = residual, $\mathbf{T}$ = transform, b = block size

Table 1 Average Y BD-rate gains for different loss functions

Loss function	L_{SATD}	L_{DCT}	L_{LOG}
Class B	-2.62%	-2.62%	-2.54%
Class C	-2.14%	-2.14%	-2.10%
Class D	-2.09%	-2.06%	-2.01%
AVG All	-2.31%	-2.30%	-2.24%

Reference Area

- Optionally available parts masked
- Larger reference area
  - Better prediction, higher complexity
  - Improvement steadily decreasing



Architecture Optimization – Deconvolution and Pruning

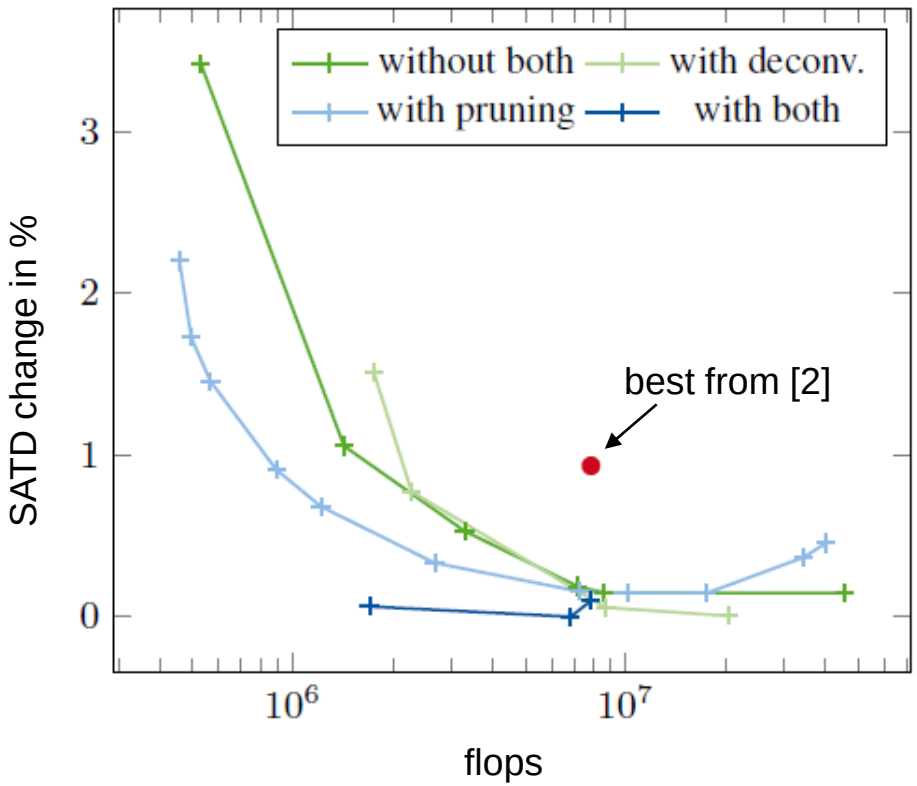
Tested:

- Adding final deconvolutional Layers
- Pruning the Network

Results:

- Deconvolution improves overall best result
- Only improvement on already complex networks

- Pruning lowers complexity significantly
- Slightly improved results



Integration and Results

Integration

- In HM 16.9 as additional intra mode
- Extended MPM-list to 4 elements

Settings

- 4 reference lines
- 2 conv, 4 dense layer
- Kernel sizes: 3 and 2

Results

- Gains most likely lower in VTM
- Architecture and training findings mostly independent of codec

	All-Intra		
Channel	**Y**	**U**	**V**
BQ Terrace	-1.77	-0.27	0.29
Basketball Drive	-2.79	-1.75	-2.09
Cactus	-2.98	-2.08	-2.20
Kimono	-2.82	-2.46	-2.78
Park Scene	-2.71	-1.95	-2.56
AVG Class B	-2.61	-1.70	-1.87
BQ Mall	-3.00	-2.60	-2.92
Basketball Drill	-2.62	-3.18	-3.21
Party Scene	-1.75	-1.07	-1.21
Race Horses	-2.19	-1.70	-1.90
AVG Class C	-2.39	-2.14	-2.31
BQ Square	-1.21	-0.38	-0.26
Basketball Pass	-2.69	-2.56	-2.58
Blowing Bubbles	-1.81	-1.38	-1.34
Race Horses	-2.70	-2.55	-2.32
AVG Class D	-2.10	-1.72	-1.63
AVG All Classes	**-2.39**	**-1.84**	**-1.93**

Summary- Part 1

Architecture

- Luma
  - CNN outperforms FCN at same complexity
  - FCN better for noisy content
  - Increased number of reference lines

- Chroma
  - Additional conv. branch for luma
  - Predicting both chroma channels at once

Pruning

- Better than unpruned networks at same complexity
- Even at 87.5% pruning only 0.33% less BD-gain

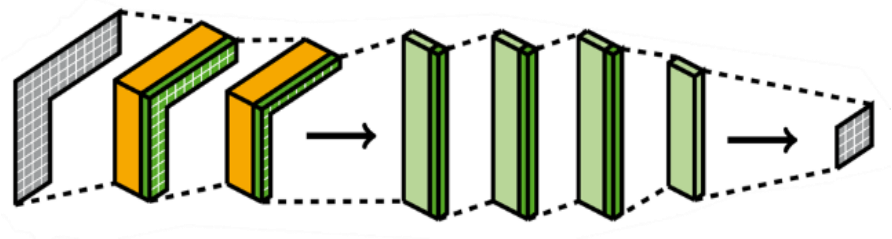


Figure 4: Example of a network architecture for the prediction of a 4x4 luma block using 2 convolutional and 4 fully connected layers

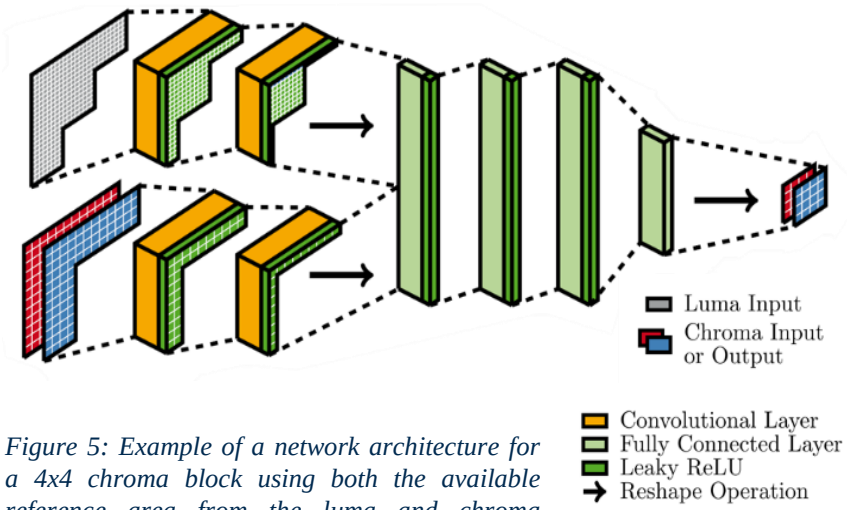
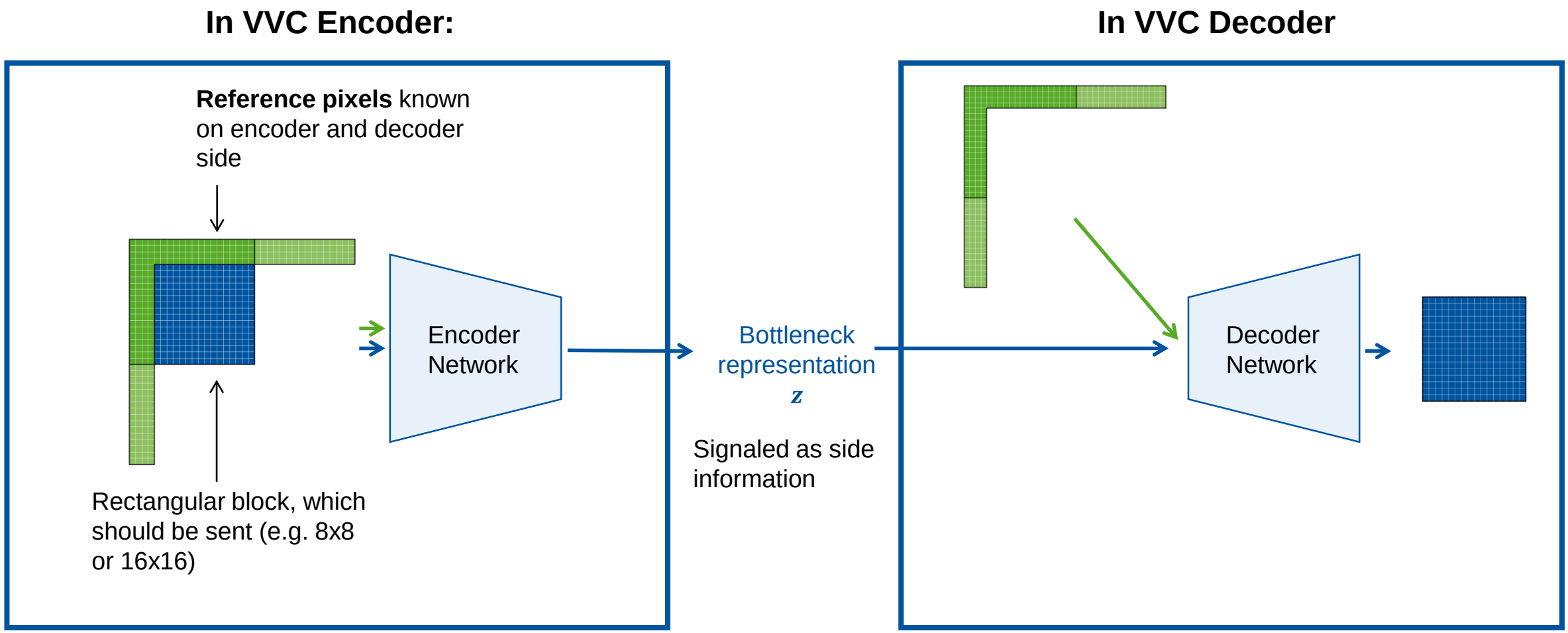


Figure 5: Example of a network architecture for a 4x4 chroma block using both the available reference area from the luma and chroma surrounding

Outline

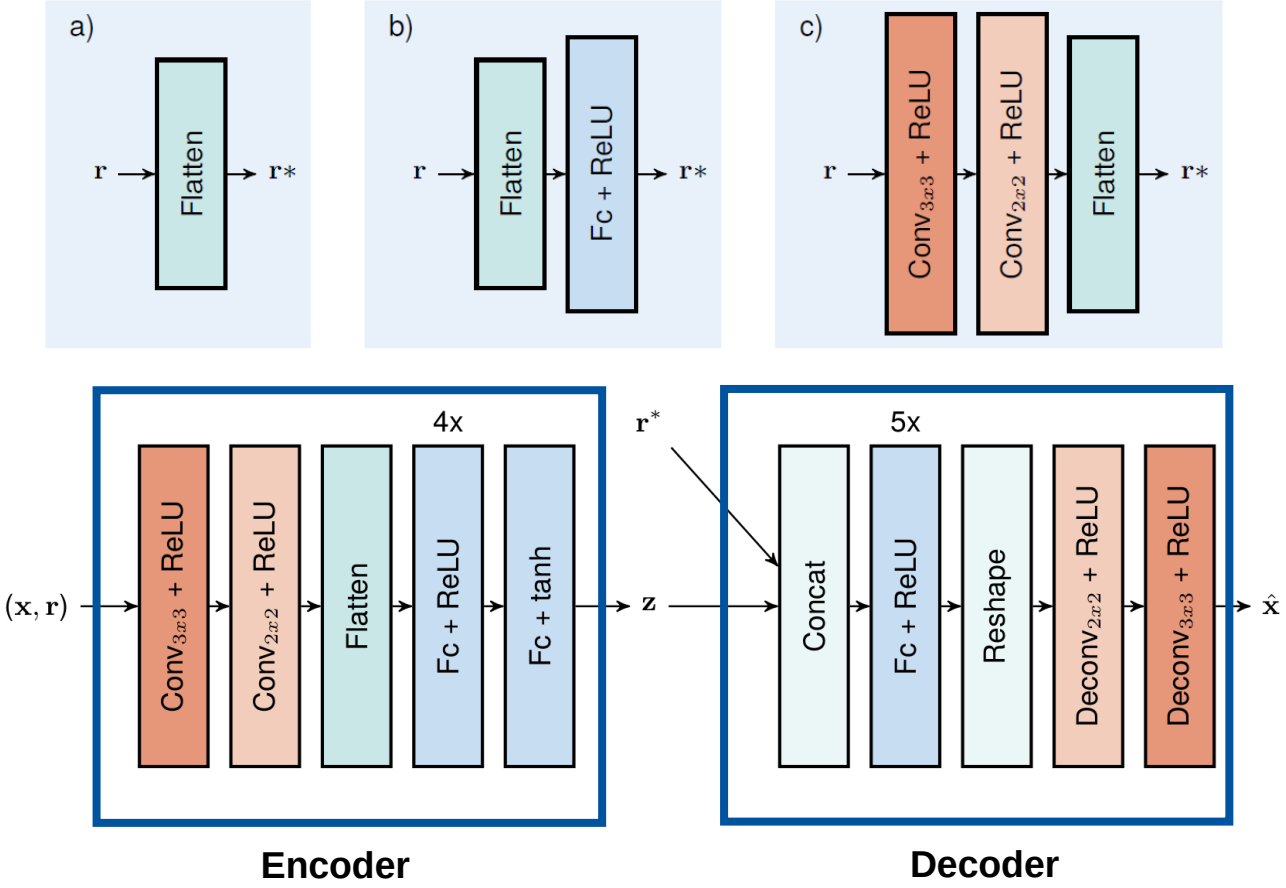
- Intro
- Part 1 – CNNs
  - Coded Training Data
  - Loss function evaluation
  - Architecture Optimization
- Part 2 – Autoencoders
  - Optimizing conditional Autoencoders
  - Variational Autoencoders
  - Vector-Quantized-VAEs
- Conclusion and Outlook

Conditional Autoencoders in Intra Prediction



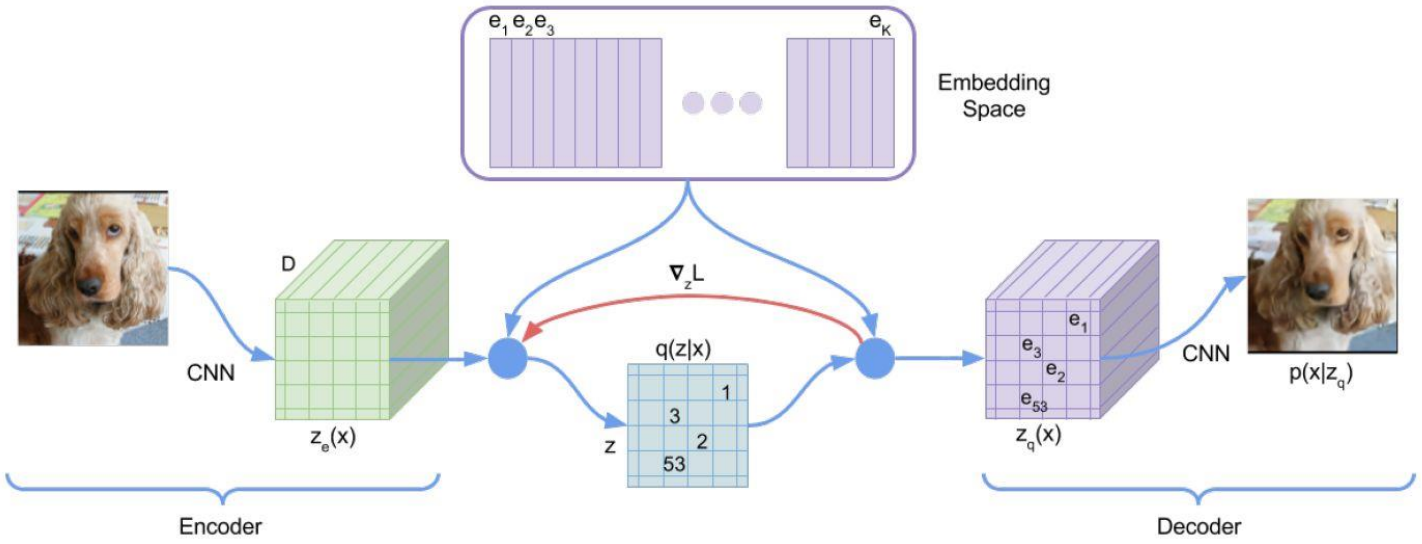
[8] Brand, Fabian, Jürgen Seiler, and André Kaup. "Intra frame prediction for video coding using a conditional autoencoder approach." *2019 Picture Coding Symposium (PCS)*. IEEE, 2019.

Architectural Adaptations



Adaptation	Result
Convolutional Layers	On average 7.62% better SATD compared to Fully connected
Additional layers between Decoder reference input and concatenation with code	Raw input (a) performs best
Evaluation of different reference line numbers	~ 3.5% SATD improvement for 8 instead of 1 reference line

The Vector Quantization VAE



Number of embeddings:	K
Dimenson bottleneck:	D
Embedding vectors:	$\boldsymbol{e} \in \mathbb{R}^{K \times D}$
Continuous output of encoder:	$\boldsymbol{z}_e(\boldsymbol{x})$
Stop gradient:	$sg(\cdot)$

Reconstruction loss — Regularization terms

$$L = \log p\left(\boldsymbol{x} \middle| \boldsymbol{z}_q(\boldsymbol{x})\right) + \| sg[\boldsymbol{z}_e(\boldsymbol{x})] - \boldsymbol{e} \|_2^2 + \beta \| \boldsymbol{z}_e(\boldsymbol{x}) - sg[\boldsymbol{e}] \|_2^2$$

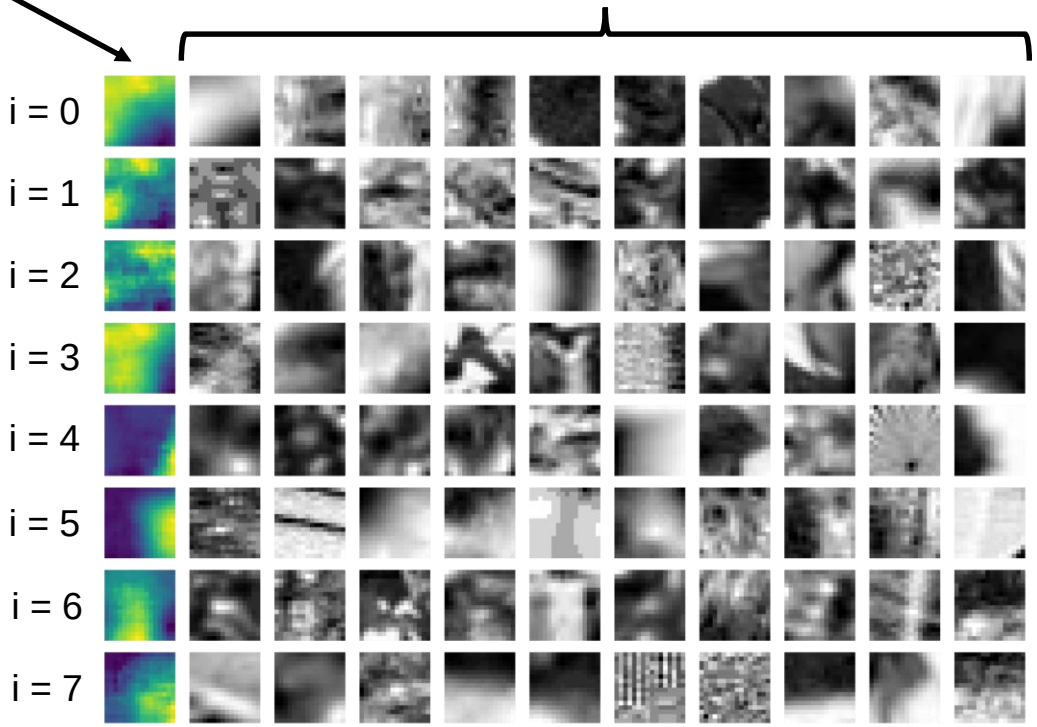
[11]: Oord, Aaron van den, Oriol Vinyals, and Koray Kavukcuoglu. "Neural discrete representation learning." *arXiv preprint arXiv:1711.00937* (2017).

The Vector Quantization VAE

Visualization for a model with blocksize $bs = 16$ and codebook size $K = 8$

Decoder Output for each embedding vector $\boldsymbol{e}_i$
While reference input set to zero:
$\boldsymbol{r} = \boldsymbol{0}$

Random patches from dataset
→ The row indicates the embedding vector $\boldsymbol{e}_i$ the patch is mapped to by the encoder



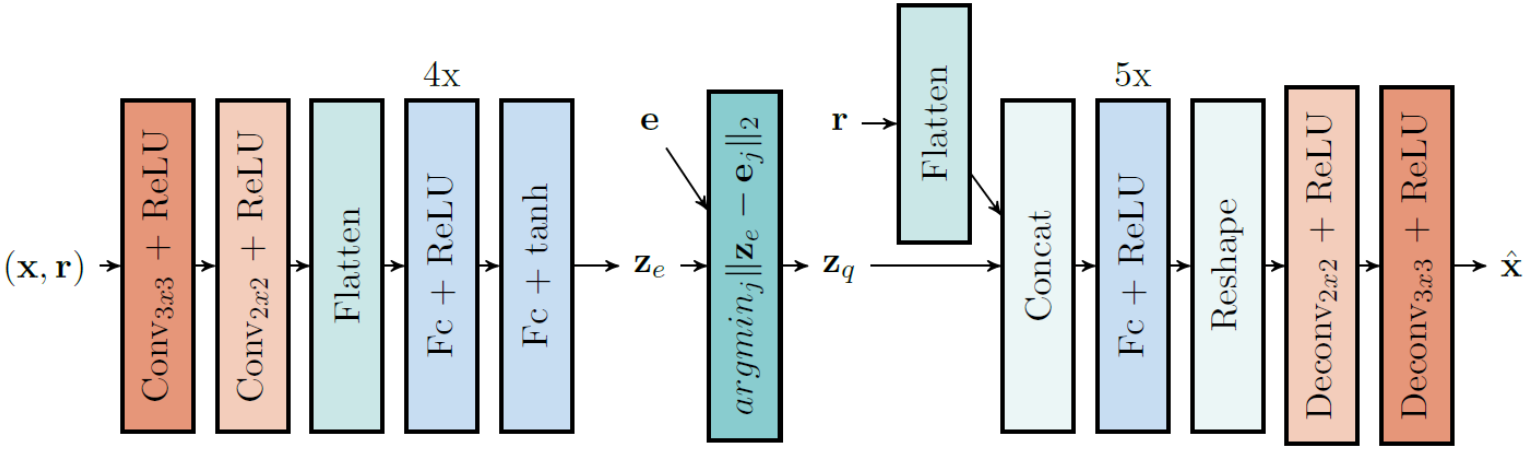
Summary Autoencoders

Architectural changes:

- Convolutional Architecture
  7.62 % SATD improvement
- Increased number of reference lines
  ~3.5% SATD improvement

Vector-Quantized Variational Autoencoder

→Joint learning of Embedding Codebook and Autoencoder



Conclusion and Outlook

Still much potential in …

- Architecture optimization
  - Both for architectures with and without side information
  - Pruning

- Optimized training methods
  - Loss adaption
  - Training Data statistics
  - Joint quantization optimization

Current work:

- VTM integration
- Side information signaling

Thank you
for your attention

Questions?