PORT: Perception-Oriented Image Compression with Fast Decoding

Wei Jiang[1], Bohao Feng[2], Wenqiang Wang[2], Bo Huang[2], Lin Ding[2], Ronggang Wang[1,3*]

[1]Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology
Shenzhen Graduate School, Peking University
[2]Alibaba Cloud Computing [3]Pengcheng Laboratory
wei.jiang1999@outlook.com rgwang@pkusz.edu.cn

Abstract

This white paper presents PORT, a Perception-ORiented image compression framework with fasT decoding. PORT is our approach for the image compression track at CLIC 2025. To enhance perceptual quality, we incorporate both semantic and patch-wise adversarial losses to generate realistic textures, and employ a region-of-interest (ROI) mask to guide bit allocation across different regions. To accelerate decoding, PORT builds upon the DCVC-RT architecture, while introducing more advanced entropy models to capture long-range correlations. Our team is PKUSZ-AliMerlin.

1 Introduction

Learned image compression [1] has become an active research area in recent years. Several models [2–7] have already outperformed the latest non-learned codec, VVC. However, most existing methods primarily focus on optimizing non-perceptual distortion metrics. To improve perceptual quality, some approaches [8–10] employ Generative Adversarial Networks (GANs) [11] to synthesize perceptually convincing textures. In addition, learned perceptual metrics such as VGG-based [12] or LPIPS [13] losses are often used to stabilize training and improve convergence.

Since subjective quality standards vary across users, content-sensitive images (e.g., faces and documents) require authenticity preservation rather than the generation of vivid yet unrealistic details. To this end, building upon the generative compression framework of [9], we propose PORT: Perception-Oriented Image Compression with fast Decoding, developed for the CLIC 2025 Image Track. PORT employs semantic- and patch-wise discriminators to guide texture synthesis, thereby enhancing semantic and spatial consistency between generated and reference images. Furthermore, a region-of-interest (ROI) mask is used to adaptively control pixel-level distortion weights, enabling flexible rate allocation.

For efficient decoding, PORT adopts the architecture of DCVC-RT, while incorporating a more advanced entropy model to capture long-range correlations. Finally, to meet specific bit-rate constraints, our model is trained with a variable-rate compression strategy inspired by [14]. Our team is PKUSZ-AliMerlin.

* Ronggang Wang is the corresponding author. This work is financially supported by Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology(Grant No. 2024B1212010006), this work is also financially supported for Outstanding Talents Training Fund in Shenzhen, Shenzhen Science and Technology Program(Grant No. SYSPG20241211173440004 and Grant No. RCJC20200714114435057) and the Alibaba Innovative Research Program.

2 Method

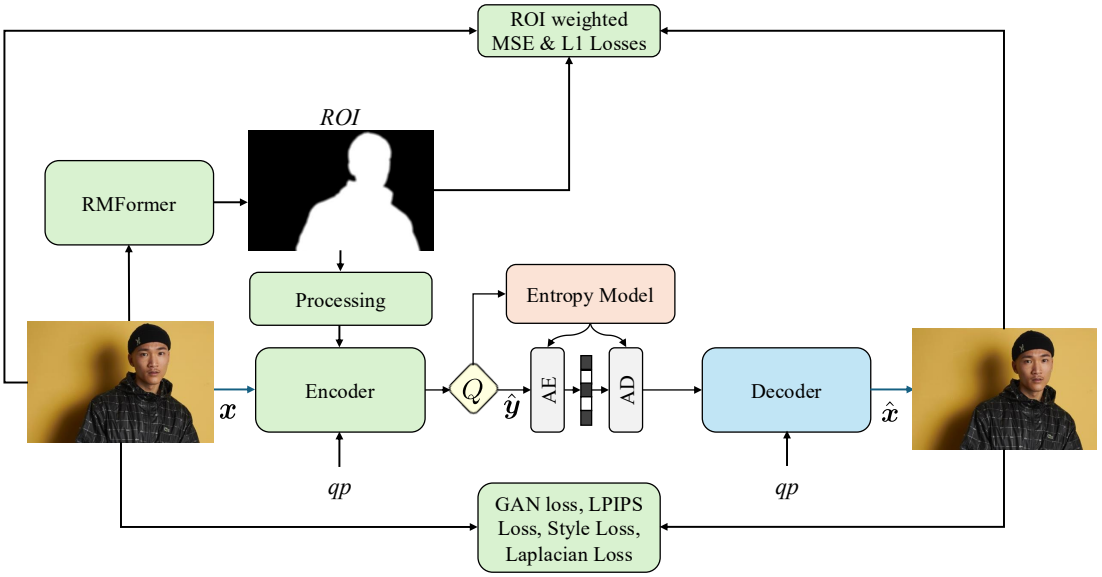

Figure 1: The overall architecture of PORT. *qp* denotes the quantization parameter for variable rate control. ROI represents the region of interest map for bit allocation. $\boldsymbol{x}$ is the input image, $\hat{\boldsymbol{x}}$ is the reconstructed image, $\hat{\boldsymbol{y}}$ is the quantized latent representation.

2.1 Overview

In this section, we briefly introduce our solution PORT to the CLIC 2025 Image Track. The overall architecture of PORT follows DCVC-RT [15] and Ma et al. [16], while the gain and inverse gain units [14] are adopted for variable-rate control. The overall architecture of PORT is illustrated in Figure 1.

The optimization process consists of two stages. In the first stage, the model is trained with a mean squared error (MSE) objective, together with the gain and inverse gain units. In the second stage, the model is further optimized for perceptual quality.

For perceptual optimization, we employ both patch-wise and semantic adversarial losses [11] to guide the decoder in generating realistic textures at low bitrates. To preserve sharpness, we additionally use the Charbonnier loss. The LPIPS loss [13] and style loss [17] are incorporated to enhance perceptual fidelity, while the Laplacian loss [18] is used to mitigate color variations. Following [16], a region-of-interest (ROI) mask is introduced to allocate more bits to semantically important regions.

The overall loss function is formulated as

$$\begin{aligned}\mathfrak{L} = \lambda_r \times \mathcal{R} + \lambda_{mse} \times (\delta \odot \mathfrak{D}_{mse}) + \\ \lambda_{L1} \times [\delta \odot \mathfrak{D}_{L1}] + \lambda_{lpips} \times \mathfrak{D}_{lpips} + \\ \lambda_{sty} \times \mathfrak{D}_{sty} + \lambda_{lap} \times \mathfrak{D}_{lap} + \lambda_{adv} \times \mathfrak{D}_{adv},\end{aligned} \tag{1}$$

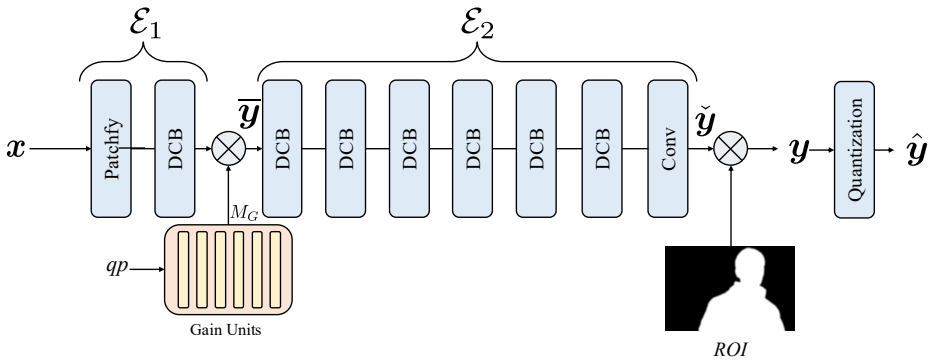

Figure 2: The architecture of the encoder of PORT. "DCB" is the depth convolutional block in DCVC-RT [15].

where $\mathcal{R}$ is the rate, $\mathfrak{D}_{mse}$ is the MSE distortion, $\mathfrak{D}_{L1}$ is the L1 distortion, $\mathfrak{D}_{lpips}$ is the VGG-16 [12] Lpips [13] distortion, $\mathfrak{D}_{sty}$ is the style loss [17], $\mathfrak{D}_{lap}$ is the Laplacian distortion [18], $\mathfrak{D}_{adv}$ is the BCE adversarial distortion, δ is the normalized ROI map. We use $\{\lambda_r, \lambda_{mse}, \lambda_{L1}, \lambda_{lpips}, \lambda_{sty}, \lambda_{lap}, \lambda_{adv}\}$ to adjust the weight of each loss.

2.2 Variable Rate Adaptation

Gain units and inverse gain units are employed to enable continuous rate adaptation across multiple target bitrates. Specifically, the gain units, denoted as $M_G \in \mathbb{R}^{c \times n}$, modulate the quantization step size of each channel, where c represents the number of channels and n is the number of supported bitrates. The inverse gain units are implemented as $1/M_G$ to reverse this modulation during decoding. These gain modules are applied to the intermediate features within both the encoder and decoder.

The gain adaptation process operates are illustrated in Figure 2 and Figure 3. At the encoder side, the intermediate latent representation is scaled by the gain units:

$$\overline{\boldsymbol{y}} = \mathcal{E}_1(\boldsymbol{x}) \odot M_G(qp), \tag{2}$$

where $\mathcal{E}_1$ denotes the first stage of the encoder that processes the input image $\boldsymbol{x}$. At the decoder side, the inverse gain units rescale the quantized features:

$$\tilde{\boldsymbol{y}} = \mathcal{D}_1(\hat{\boldsymbol{y}}) \odot \frac{1}{M_G(qp)}, \tag{3}$$

where $\mathcal{D}_1$ represents the first stage of the decoder, and $\odot$ denotes element-wise multiplication. This mechanism allows M_G to adaptively scale the latent representation according to the quantization parameter qp before quantization, while the inverse gain compensates for this scaling during synthesis, enabling flexible bitrate control without requiring multiple models.

2.3 ROI-Weighted Distortion and Bit Allocation

Following Ma et al. [16], we employ a saliency map as the ROI mask, as saliency detection naturally separates an image into focused regions and background, which

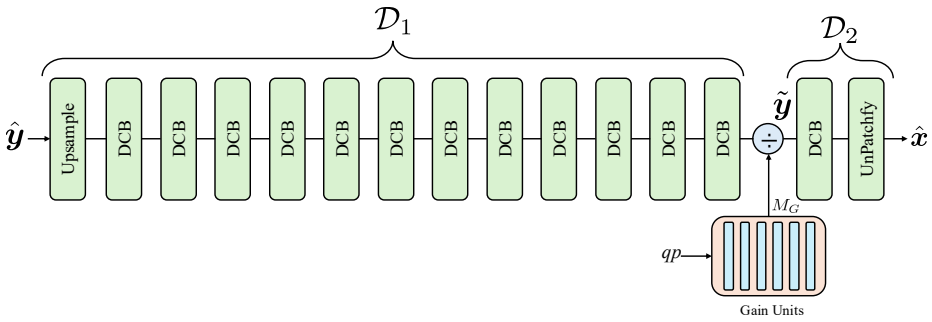

Figure 3: The architecture of the decoder of PORT.

aligns well with our bit allocation strategy. Specifically, we adopt RMformer [19] to generate ROI maps, which remains fixed during both training and inference. The ROI detection process is formulated as:

$$\text{Mask}_{2D} = \text{sigmoid}(\text{RMformer}(\boldsymbol{x})), \tag{4}$$

where $\boldsymbol{x}$ is the input image and Mask_{2D} is the detected ROI map.

To smooth region boundaries and facilitate gradual bit allocation, the ROI map is processed with average pooling:

$$\text{RM}_{2D} = \text{AvgPool}(\text{Mask}_{2D}), \tag{5}$$

where RM_{2D} denotes the smoothed ROI map. A scaling factor α is introduced to control the relative importance of ROI regions during rate allocation. To prevent background texture degradation, a fixed number of channels (64 in our implementation) are reserved to retain sufficient information for non-ROI regions.

The bit allocation is performed at the encoder side as follows:

$$\tilde{\boldsymbol{y}} = \mathcal{E}_2(\mathcal{E}_1(\boldsymbol{x}) \odot M_G(qp)), \boldsymbol{y} = (\boldsymbol{y}_{0:63} \,||\, \boldsymbol{y}_{64:320} \odot \frac{\text{RM}_{2D} + \alpha}{1 + \alpha}), \tag{6}$$

where $\mathcal{E}_1$ and $\mathcal{E}_2$ denote the first and second stages of the encoder, respectively. The first 64 channels $\boldsymbol{y}_{0:63}$ are preserved without ROI weighting, while the remaining channels $\boldsymbol{y}_{64:320}$ are modulated by the scaled ROI map. The operator $||$ represents channel-wise concatenation. At the decoder side, the inverse ROI mapping is not required, and reconstruction proceeds as:

$$\hat{\boldsymbol{x}} = \mathcal{D}_2(\mathcal{D}_1(\hat{\boldsymbol{y}}) \odot \frac{1}{M_G(qp)}), \tag{7}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ denote the first and second stages of the decoder, respectively. For pixel-wise losses such as L1 and MSE, the ROI map is directly applied to weight the distortions, guiding the model to allocate more bits to salient regions while maintaining acceptable quality in background areas.

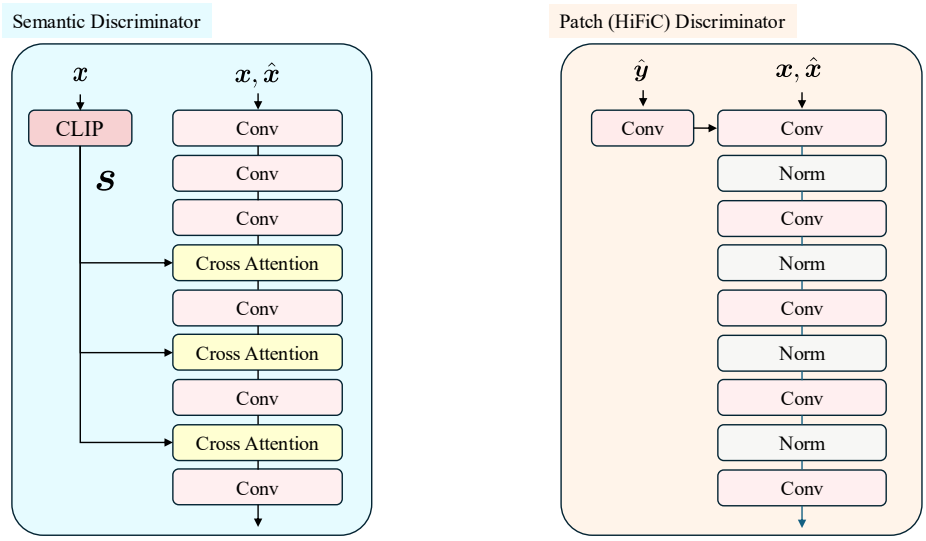

Figure 4: The architecture of the semantic and patch discriminators. $\boldsymbol{s}$ denotes the semantic features extracted by CLIP.

2.4 Adversarial Training

To generate realistic textures, we optimize the model with adversarial losses that include both semantic-level and patch-wise components. The architectures of the semantic discriminator and patch discriminator are illustrated in Figure 4. Following prior works [9, 16], we employ the binary cross-entropy (BCE) loss for adversarial training.

Semantic-level adversarial loss. We leverage CLIP [20] to extract semantic features $\boldsymbol{s}$ from the input image, which serve as conditional inputs for semantic-aware discrimination:

$$\begin{aligned} \mathfrak{L}^s_{adv} &= \mathbb{E}\big[-\log(D^s(\hat{\boldsymbol{x}} \mid \boldsymbol{s}))\big], \\ \mathfrak{L}^s_{disc} &= \mathbb{E}\big[-\log(1 - D^s(\hat{\boldsymbol{x}} \mid \boldsymbol{s}))\big] + \mathbb{E}\big[-\log(D^s(\boldsymbol{x} \mid \boldsymbol{s}))\big], \end{aligned} \tag{8}$$

where $\hat{\boldsymbol{x}}$ denotes the reconstructed image and $\boldsymbol{x}$ the original image. Inspired by latent diffusion models, we employ a cross-attention mechanism for semantic-guided conditional discrimination, where the semantic features $\boldsymbol{s}$ are used as queries to modulate the discriminator's attention over image features.

Patch-wise adversarial loss. For fine-grained texture discrimination, the patch-wise loss is defined as:

$$\begin{aligned} \mathfrak{L}^p_{adv} &= \mathbb{E}\big[-\log(D^p(\hat{\boldsymbol{x}} \mid \hat{\boldsymbol{y}}))\big], \\ \mathfrak{L}^p_{disc} &= \mathbb{E}\big[-\log(1 - D^p(\hat{\boldsymbol{x}} \mid \hat{\boldsymbol{y}}))\big] + \mathbb{E}\big[-\log(D^p(\boldsymbol{x} \mid \hat{\boldsymbol{y}}))\big], \end{aligned} \tag{9}$$

where $\hat{\boldsymbol{y}}$ denotes the quantized latent representation, which provides rate-dependent conditioning for the discriminator.

Here, D^s and D^p denote the semantic and patch-wise discriminators, respectively. The discriminators are trained to minimize $\mathfrak{L}^s_{disc}$ and $\mathfrak{L}^p_{disc}$, while the generator

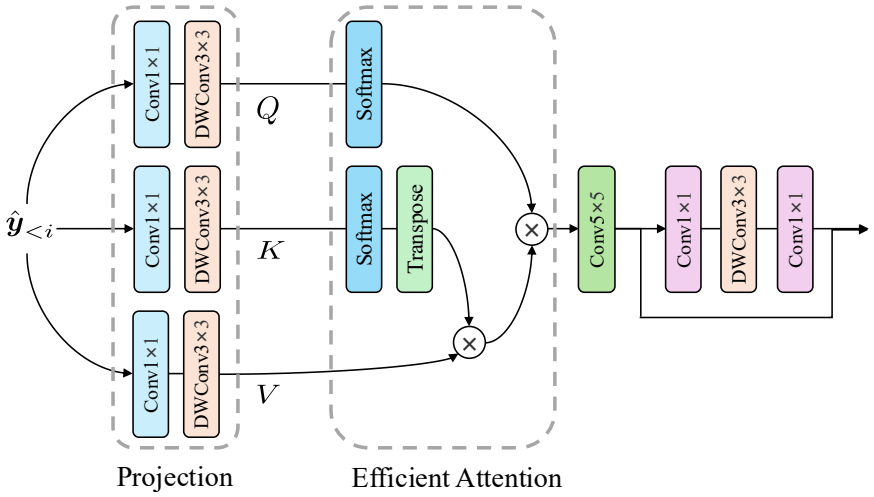

Figure 5: The process of the efficient attention-based global context modeling.

(encoder-decoder) is optimized with the adversarial losses $\mathfrak{L}^s_{adv}$ and $\mathfrak{L}^p_{adv}$. The overall adversarial loss for the generator is defined as:

$$\mathfrak{L}_{adv} = \lambda^s_{adv}\mathfrak{L}^s_{adv} + \lambda^p_{adv}\mathfrak{L}^p_{adv}, \tag{10}$$

where λ^s_{adv} and λ^p_{adv} are hyperparameters that balance the contributions of semantic and patch-wise adversarial objectives.

2.5 Entropy Model

The entropy model integrates three complementary modules for accurate probability estimation: a quadtree context module from DCVC-DC [21], a global context module, and a hyperprior module. These modules collectively capture multi-scale spatial dependencies to enable efficient entropy coding. **Quadtree context module.** In the quadtree-based partitioning, the quantized latent representation $\hat{\boldsymbol{y}}$ is hierarchically divided into four parts $\{\hat{\boldsymbol{y}}_0, \hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \hat{\boldsymbol{y}}_3\}$. During encoding/decoding of $\hat{\boldsymbol{y}}_i$, all previously processed parts $\hat{\boldsymbol{y}}_{<i} = \{\hat{\boldsymbol{y}}_0, \ldots, \hat{\boldsymbol{y}}_{i-1}\}$ are employed as autoregressive context. For each partition, we employ a depth-wise convolutional block [15] to efficiently capture local spatial context within neighboring regions.

Efficient attention for global context. To extract long-range dependencies across the latent space, we apply an efficient self-attention mechanism [5, 22]. Unlike standard attention with $O(N^2)$ complexity where N is the number of spatial locations, the efficient attention approximates the attention matrix using two sequential softmax operations, reducing complexity to $O(N)$. Specifically, given the previous context $\hat{\boldsymbol{y}}_{<i} \in \mathbb{R}^{H\times W\times C}$ with $N = H \times W$ spatial locations and C channels, the query, key, and value are computed as:

$$Q = W_Q\hat{\boldsymbol{y}}_{<i}, \quad K = W_K\hat{\boldsymbol{y}}_{<i}, \quad V = W_V\hat{\boldsymbol{y}}_{<i}, \tag{11}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are learnable projection matrices. The efficient attention

is then computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}_2(Q) \cdot (\text{softmax}_1(K)^\top \cdot V), \tag{12}$$

This factorization allows the model to capture global context efficiently by first aggregating value features weighted by keys ($O(NC^2)$), then reweighting by queries ($O(NC)$), achieving overall $O(NC^2)$ complexity instead of $O(N^2C)$ for standard attention.

The global context module processes the entire latent map to capture non-local correlations, complementing the local quadtree context. The hyperprior module [1] encodes additional side information $\hat{\boldsymbol{z}}$ about the latent distribution through an auxiliary autoencoder, providing additional priors that improve entropy estimation. The outputs from all three modules are fused to obtain the final probability distribution $p(\hat{\boldsymbol{y}}_i \mid \hat{\boldsymbol{y}}_{<i}, \hat{\boldsymbol{z}}, \text{global context})$ for arithmetic coding.

3 Experiments

3.1 Training

PORT is trained on 10^5 images sampled from ImageNet [23], COCO2017 [24], DIV2K [25], and Flickr2K [26]. The training process consists of two stages and runs for 1M steps total with a batch size of 8, using the Adam optimizer [27] with a fixed learning rate of 10^{-4} on 8 Tesla V100 GPUs. In the first stage, we optimize PORT using only the MSE loss to establish a solid baseline. In the second stage, we introduce the unified loss defined in Equation 1 to enhance perceptual quality.

3.2 Evaluation

Following [28], we use PSNR, LPIPS (VGG)[13], and DISTS[29] to assess perceptual quality. All metrics are computed on the reconstructed images saved in PNG format (**8-bit integer**).

3.3 Hyperparameters

To meet the three target bit rates of $\{0.075, 0.15, 0.3\}$ bpp specified by CLIC, we train three separate variable-rate models by sampling the rate parameter λ_r from different ranges. The complete hyperparameter configuration is summarized in Table 1.

3.4 Main Results

The rate-distortion and rate-perception performance of PORT and baseline methods are shown in Figure 6. Compared to MS-ILLM and CRDR, PORT achieves significant performance improvements in terms of LPIPS and DISTS metrics, demonstrating superior perceptual quality at comparable bit rates.

As illustrated in Figure 7, PORT achieves superior perceptual quality compared to MS-ILLM and CRDR while using lower bitrates.

Table 1: Hyperparameter configuration for variable-rate training and loss weights.

Hyperparameter	Value
λ_r (low rate)	$(7, 64)$
λ_r (medium rate)	$(2, 25)$
λ_r (high rate)	$(1, 9)$
λ_{mse}	448
λ_{L1}	32
λ_{lpips}	5.12
λ_{sty}	1.28
λ_{lap}	1
λ_{adv}	3.84

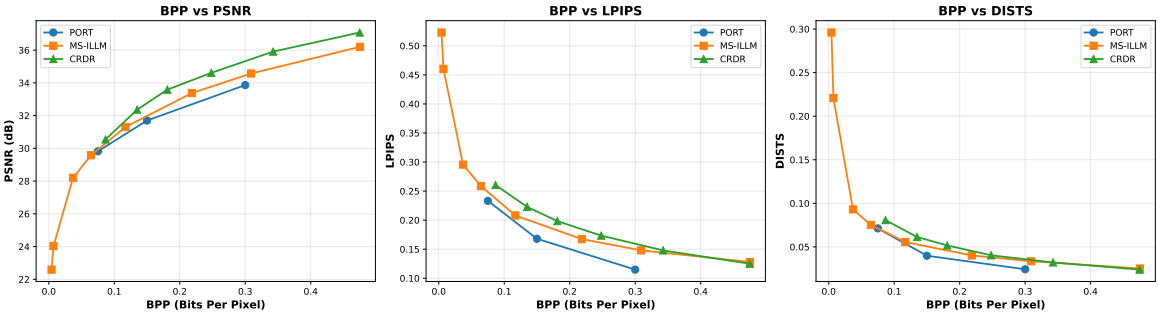

Figure 6: Rate-Distortion-Perception curves of PORT and the baselines [28, 33] on CLIC 2025 Valid.

3.5 Complexity Analysis

We compare the model size, encoding time, decoding time, and VRAM consumption on the CLIC 2020 validation set [30]. The results are presented in Table 2. PORT exhibits significantly faster decoding speeds and lower memory consumption compared to previous approaches, particularly those based on diffusion models such as CDC [31] and DiffEIC [32]. Notably, PORT achieves the fastest decoding speed and lowest memory footprint among all compared methods.

Method	Params (M)	CLIC 2025 Test		
		Enc Time (s)	Dec Time (s)	VRAM (GB)
MS-ILLM (ICML'23)	181.5	0.58	0.57	5.10
CRDR (WACV'24)	127.7	0.51	0.58	4.31
PORT (Ours)	89.6	**0.13**	**0.10**	**1.62**

① All evaluations are conducted on a Tesla A100-40G GPU with a Xeon(R) Platinum 8336C CPU.

Table 2: Encoding times (s), decoding times (s) and Peak GPU VRAM consumptions (GB) during encoding / decoding on CLIC 2020 [30].

(a) GT (b) MS-ILLM(0.0439bpp)

(c) CRDR(0.0448bpp) (d) PORT(0.0420bpp)

(e) GT (f) MS-ILLM (g) CRDR (h) PORT

Figure 7: Visual comparison of reconstructions from our PORT and state-of-the-art perception-oriented methods MS-ILLM and CRDR. **Please zoom in for better view.**

4 Conclusion

In this white paper, we have presented PORT, a Perception-Oriented Image Compression framework with fast Decoding for the CLIC 2025 Image Track. PORT integrates semantic- and patch-wise adversarial learning, ROI-guided bit allocation, variable-rate gain units, and an advanced entropy model with quadtree and global context modules. Extensive design choices, including model architecture, perceptual and reconstruction losses, ensure both high perceptual quality and efficient decoding. Our

approach demonstrates the effectiveness of combining perceptual optimization with fast performance, providing a practical solution for content-aware image compression.

References

[1] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston, "Variational image compression with a scale hyperprior," in *ICLR*, 2018.

[2] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *CVPR*, June 2020.

[3] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *CVPR*, June 2022.

[4] Wei Jiang, Jiayu Yang, Yongqi Zhai, Peirong Ning, Feng Gao, and Ronggang Wang, "Mlic: Multi-reference entropy model for learned image compression," in *ACM MM*, 2023.

[5] Wei Jiang, Jiayu Yang, Yongqi Zhai, Feng Gao, and Ronggang Wang, "Mlic++: Linear complexity multi-reference entropy modeling for learned image compression," *ACM TOMM*, vol. 21, no. 5, pp. 1–25, 2025.

[6] Wei Jiang, Peirong Ning, Jiayu Yang, Yongqi Zhai, Feng Gao, and Ronggang Wang, "Llic: Large receptive field transform coding with adaptive weights for learned image compression," *TMM*, 2024.

[7] Wei Jiang, Yongqi Zhai, Jiayu Yang, Feng Gao, and Ronggang Wang, "Mlicv2: Enhanced multi-reference entropy modeling for learned image compression," *arXiv:2504.19119*, 2025.

[8] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool, "Generative adversarial networks for extreme learned image compression," in *ICCV*, 2019, pp. 221–231.

[9] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson, "High-fidelity generative image compression," *NeurIPS*, vol. 33, pp. 11913–11924, 2020.

[10] Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer, "Multi-realism image compression with a conditional generator," in *CVPR*, 2023, pp. 22324–22333.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *NeurIPS*, vol. 27, 2014.

[12] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[13] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.

[14] Ze Cui, Jing Wang, Shangyin Gao, Tiansheng Guo, Yihui Feng, and Bo Bai, "Asymmetric gained deep image compression with continuous rate adaptation," in *CVPR*, 2021, pp. 10532–10541.

[15] Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu, "Towards practical real-time neural video compression," in *CVPR*, 2025, pp. 12543–12552.

[16] Yi Ma, Yongqi Zhai, Chunhui Yang, Jiayu Yang, Ruofan Wang, Jing Zhou, Kai Li, Ying Chen, and Ronggang Wang, "Variable rate roi image compression optimized for visual quality," in *CVPR Workshops*, 2021, pp. 1936–1940.

[17] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "Image style transfer using convolutional neural networks," in *CVPR*, 2016, pp. 2414–2423.

[18] Simon Niklaus and Feng Liu, "Context-aware synthesis for video frame interpolation," in *CVPR*, 2018, pp. 1701–1710.

[19] Xinhao Deng, Pingping Zhang, Wei Liu, and Huchuan Lu, "Recurrent multi-scale transformer for high-resolution salient object detection," in *ACM MM*, 2023, pp. 7413–7423.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., "Learning transferable visual models from natural language supervision," in *ICML*. PmLR, 2021, pp. 8748–8763.

[21] Jiahao Li, Bin Li, and Yan Lu, "Neural video compression with diverse contexts," in *CVPR*, 2023, pp. 22616–22626.

[22] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li, "Efficient attention: Attention with linear complexities," in *IEEE WACV*, 2021.

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*. IEEE, 2009, pp. 248–255.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[25] Eirikur Agustsson and Radu Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPR Workshops*, 2017, pp. 1122–1131.

[26] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *CVPR Workshops*, 2017.

[27] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[28] Matthew J Muckley, Alaaeldin El-Nouby, Karen Ullrich, Hervé Jégou, and Jakob Verbeek, "Improving statistical fidelity for neural image compression with implicit local likelihood models," in *ICML*. PMLR, 2023, pp. 25426–25443.

[29] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *TPAMI*, vol. 44, no. 5, pp. 2567–2581, 2020.

[30] George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Ballé, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer, "Workshop and challenge on learned image compression," 2020.

[31] Ruihan Yang and Stephan Mandt, "Lossy image compression with conditional diffusion models," *NeurIPS*, vol. 36, 2024.

[32] Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Jingwen Jiang, "Towards extreme image compression with latent feature guidance and diffusion prior," *TCSVT*, 2024.

[33] Shoma Iwai, Tomo Miyazaki, and Shinichiro Omachi, "Controlling rate, distortion, and realism: Towards a single comprehensive neural image compression model," in *WACV*, 2024, pp. 2900–2909.