All-in-One Image Coding for Joint Human-Machine Vision with Multi-Path Aggregation

Xu Zhang Peiyao Guo Ming Lu* Zhan Ma
School of Electronic Science and Engineering
Nanjing University
{xu.zhang,peiyao}@smail.nju.edu.cn {minglu,mazhan}@nju.edu.cn

Abstract

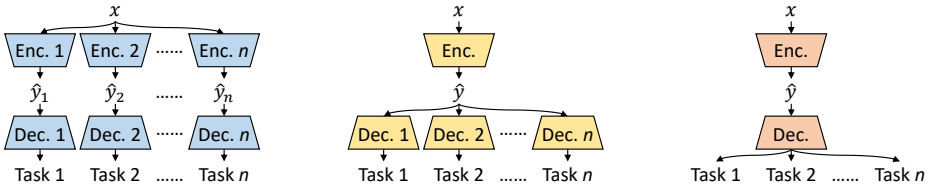
Image coding for multi-task applications, catering to both human perception and machine vision, has been extensively investigated. Existing methods often rely on multiple task-specific encoder-decoder pairs, leading to high overhead of parameter and bitrate usage, or face challenges in multi-objective optimization under a unified representation, failing to achieve both performance and efficiency. To this end, we propose Multi-Path Aggregation (MPA) integrated into existing coding models for joint human-machine vision, unifying the feature representation with an all-in-one architecture. MPA employs a predictor to allocate latent features among task-specific paths based on feature importance varied across tasks, maximizing the utility of shared features while preserving task-specific features for subsequent refinement. Leveraging feature correlations, we develop a two-stage optimization strategy to alleviate multi-task performance degradation. Upon the reuse of shared features, as low as 1.89% parameters are further augmented and fine-tuned for a specific task, which completely avoids extensive optimization of the entire model. Experimental results show that MPA achieves performance comparable to state-of-the-art methods in both task-specific and multi-objective optimization across human viewing and machine analysis tasks. Moreover, our all-in-one design supports seamless transitions between human- and machine-oriented reconstruction, enabling task-controllable interpretation without altering the unified model. Code is available at https://github.com/NJUVISION/MPA.

1 Introduction

Coding for multi-task that satisfies both human perception and machine vision has been extensively explored over the past few years [69, 11, 80, 12, 45]. The intuitive and simple solution to achieve optimal performance for various tasks involves defining distinct encoder-decoder pairs tailored to specific instances with multiple bitstreams (separate [49, 69, 11, 42, 7, 75] or scalable [51, 80, 30, 74, 12, 26]), which however incurs significant parameter overhead and inefficient bitrate consumption. As a result, alternative efforts [8, 19, 22, 45] attempt to enhance the compression performance across multiple tasks by deriving *a unified and compact representation* of input images. Typically, they rely on a generalized encoder for feature extraction and deploy diverse decoding models to support corresponding vision tasks. However, they still suffer from the parametric inefficiency of multiple dedicated decoders. Recent work has made some progress in designing *a unified compression model* for multi-task, where the encoder generates a unified representation, and the decoder focuses on task-oriented reconstruction. Popular techniques include conditional generation [2, 32] and residual prediction [33, 24, 41], enabling user-controllable modulation of decoded results through task-driven guidance. While achieving performance comparable to decoder-specific approaches, they solely

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Separate pairs (b) Unified representation (c) Unified model

Figure 1: Paradigm comparison for multi-task coding.

focus on human-centric requirements, such as perception-oriented and fidelity-oriented aspects. More critically, the challenge of multi-objective optimization [62, 20], particularly in the context of multi-task learning [66, 63], persists in the unified paradigm for joint human-machine vision, offering the performance largely inferior to that optimized for each task independently [42, 7, 8]. The crux of the problem lies in the indiscriminate treatment of the unified representation across diverse tasks, which necessitates searching for an accurate Pareto front for transition between tasks. This requires the use of massive and complicated techniques to fit it [47, 58, 57, 75] and multi-objective optimization with variable weights [2, 32, 24, 41]. Its failure can lead to significant performance degradation. And as the number of tasks increases, the difficulty of optimal search grows dramatically [15, 31, 23]. So far, few approaches consider the correlation of features across tasks to ensure efficient multi-task collaboration while also optimizing for the specificity of distinct tasks.

To overcome the challenges above, we propose a unified image coding method with Multi-Path Aggregation (MPA) for joint human-machine vision tasks. Specifically, a set of Multi-Layer Perceptron (MLP) branches is inserted to form multiple aggregation paths, replacing the single-path MLP in the feature transform blocks [53, 54] which currently dominate compression models [56, 18, 55, 84, 50]. Each path is customized and tailored to different tasks with varying complexities. Considering diverse importance of features across different tasks, we devise a predictor for importance scores to allocate latent features among task-specific paths based on their importance. Leveraging feature correlations across tasks, we develop an efficient two-stage optimization strategy with fine-tuning partial parameters on generalized features to alleviate multi-task performance degradation, avoiding extensive optimization of the entire model. This strategy significantly eases the optimization of multi-task coding while maintaining performance comparable to other fully optimized models. Considering any viewing task for human vision (e.g., low distortion or high realism) or analysis task for machine vision (e.g., high-level or low-level vision task), MPA can switch flexibly between them with seamless transitions, enabling a single representation to be interpreted in different ways within a unified model.

Our contributions are summarized as follows:

- Considering different importance of features for different tasks, we propose MPA to non-uniformly treat features. By developing an importance score predictor, MPA can allocate generalized latent features among task-specific paths based on their varying importance. This enables a *unified model* to support multi-task coding with seamless transitions in an all-in-one manner.
- Leveraging feature correlations across tasks, we propose a two-stage optimization strategy with fine-tuning partial parameters on generalized features to overcome the challenge of multi-objective optimization in multi-task coding. This strategy allows MPA to be easily extended to support new tasks without independent optimization of separate task-specific models.
- Extensive experiments demonstrate that using the unified model, MPA achieves rate-distortion (R-D) and rate-perception (R-P) performance for human vision comparable to other state-of-the-art (SOTA) models and significantly improves accuracy for analysis tasks close to the fully fine-tuned ones, outperforming other separately optimized methods.

2 Related work

Separate pairs. Developing multiple task-specific encoder-decoder pairs, as shown in Fig. 1a, is easy to optimize and beneficial to high performance but has significant parameter and bitrate

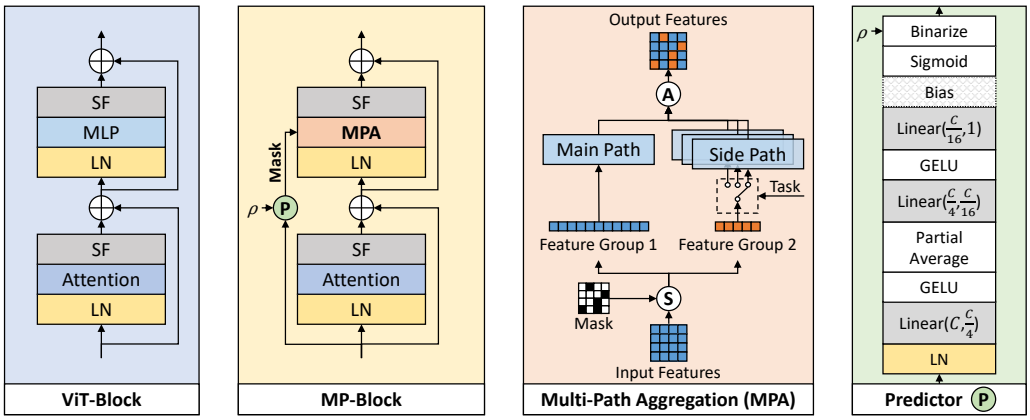


Figure 2: The proposed Multi-Path Aggregation (MPA). Compared to typical Vision Transformer (ViT) block [73, 16], original MLP is replaced with MPA. LN and SF are Layer Normalization and Scaling Factors [10]. Ⓟ, Ⓢ and Ⓐ denote predictor, split and aggregation respectively. C represents the number of input channels. ρ is the ratio ρ_{enc} in the encoder or the ratio ρ_{dec} in the decoder.

overhead [49, 69, 11, 42, 7]. Song et al. [69] introduced Spatially-adaptive Feature Transform (SFT) for tuning bit allocation, incurring significant latency for optimizing quality map for each image to achieve best task performance. Chen et al. [11] leveraged visual prompt tuning (VPT) [36] for task-specific optimization, resulting in quadratic complexity due to the self-attention mechanism with respect to the number of VPT tokens. Scalable coding [51, 80, 30, 74, 12, 26] improved bitstream efficiency by embedding multiple bitstreams in a scalable manner. However, organizing representations for various tasks in a layered manner without introducing redundancy is challenging.

Unified representation. Some efforts [8, 19, 22, 45] aimed to derive a unified and compact representation of input images but developed separate decoders (Fig. 1b) to ease optimization challenges. Feng et al. [22] compressed intermediate features from a vision backbone as a generic representation, while Duan et al. [19] proposed a plug-and-play adapter to bridge compressed representations with existing vision backbones, both requiring optimization of the backbone for optimal performance. Li and Zhang [45] used a semantic enhancement network to improve analysis accuracy without jointly optimizing the vision backbone but still faced parameter overhead due to multiple decoders.

Unified model. The paradigm in Fig. 1c uses a single encoder to generate a unified representation and a single decoder for task-specific reconstructions [23, 2, 24, 41, 32]. Gao et al. [23] optimized the unified model for multiple analysis tasks jointly without transitions, resulting in trade-offs and suboptimal performance. Ghouse et al. [24] and Korber et al. [41] reconstructed images with a basic model and used additional modules to predict and add residuals for task-specific goals, while Agustsson et al. [2] and Iwai et al. [32] developed hyper-parameter conditioning modules to modulate the transition process. These methods require variable weighted objectives during optimization processes, making it challenging to extend to more tasks.

3 A unified framework: multi-path aggregation

3.1 Preliminaries

Typically, the LIC model involves an encoder network $E(\boldsymbol{x})$ that maps the input image $\boldsymbol{x}$ to a compact latent representation $\hat{\boldsymbol{y}}$, and a decoder network $G(\hat{\boldsymbol{y}})$ that reconstructs $\hat{\boldsymbol{x}}$ as an approximation to $\boldsymbol{x}$. For the best compression performance, the optimization objective minimizes the distortion between the original and reconstructed images while reducing the expected bitrate $r(\hat{\boldsymbol{y}})$, using $\mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[-\log_2 p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}})]$ characterized by entropy model P. To achieve perceptually pleasing reconstruction, a Generative Adversarial Network (GAN) strategy is often incorporated, where a conditional discriminator $D(\text{cond.}, \text{input})$ is used to assist in model optimization. In our work, the MPA is implemented in TinyLIC [55] to validate its functionality for joint human-machine vision coding, while the GAN strategy in HiFiC [61] is used for guiding perceptual optimization. Furthermore, to support continuously variable-rate coding, we introduce the scaling factor (SF) modulation $\boldsymbol{s}_q$ [10]

and non-linear interpolation [13, 43] in each Multi-Path Block (MP-Block in Fig. 2). The compression quality $q \in [1, Q_{\text{max}}]$ matching in the encoder and decoder determines the bitrate consumption, while the task orientation $\alpha \in [0, 1]$ in the decoder controls the continuously transitions between tasks.

3.2 Multi-path aggregation

To support various task optimizations flexibly with the unified model, we develop an efficient architecture named MPA. As illustrated in Fig. 2, the proposed method contains two key modules, multiple paths with MLP layers of varying complexity, and a predictor for binarized mask $\boldsymbol{M}$. Different from recent advances in Mixture-of-Experts (MoE) [17, 21], the paths are divided into two groups: the main and the side paths. The main path captures generalized features for various tasks, while the side paths are fine-tuned based on the generalized features. The input features are allocated into different paths on the basis of $\boldsymbol{M}$. The features corresponding to "1" in $\boldsymbol{M}$ will enter the main path, while the remaining features will enter the side paths. Unlike existing methods, which apply consistent transformations [24, 41, 2, 32] to all features, MPA only exploits a few parameters in the side path to refine the generalized features for task transitions.

On the encoder side, we set the original MLP in ViT block as the main (high-quality) path ϕ_{hq} adapted to high bitrate features and add a bottleneck MLP as the side (low-quality) path ϕ_{lq} specific to low bitrate coding. For task-controllable LIC, the features optimized for perceptual loss are the most generalized ones [41]. Therefore, on the decoder side, the main path is the Perc. (Perceptual) Path $\phi_{\text{perc}}(\cdot)$, while the side paths contain the MSE path $\phi_{\text{MSE}}(\cdot)$, the Cls. (Classification) Path $\phi_{\text{cls}}(\cdot)$, and the Seg. (Segmentation) Path $\phi_{\text{seg}}(\cdot)$, standing for four representative tasks to showcase the versatility of MPA, i.e., high-realism, low-distortion, high-level visual, and low-level visual image reconstructions. Setting the task index i_{task} by the user, MPA can realize the aggregation of the main path and one of the side path. To be efficient, we configure each path for a different complexity depending on the characteristics of tasks. Specifically, the paths for human vision (i.e., $\phi_{\text{perc}}(\cdot)$ and $\phi_{\text{MSE}}(\cdot)$) are designed as an inverted (inv.) bottleneck MLP [54] to achieve higher realism and lower distortion, while the paths for machine vision (i.e., $\phi_{\text{cls}}(\cdot)$ and $\phi_{\text{seg}}(\cdot)$) was designed as a bottleneck MLP to reduce inference latency. Having an input feature $\boldsymbol{x} \in \mathbb{R}^{H\times W\times C}$, i_{task} and a mask $\boldsymbol{M} \in \{0, 1\}^{H\times W\times 1}$, MPA processes as Alg. 1. In our implementation, we integrate MPA into the 1st, 2nd and 3rd stages in both the encoder and decoder, without the 4th stage since its spatial size is too small to produce a smooth transition.

Algorithm 1 Multi-Path Aggregation

Require: feature $\boldsymbol{x}$, mask $\boldsymbol{M}$, task index i_{task}
if encoding **then**
  $\phi_{\text{main}} \leftarrow \phi_{\text{hq}}, \phi_{\text{side}} \leftarrow \phi_{\text{lq}}$
else if decoding **then**
  $\phi_{\text{main}} \leftarrow \phi_{\text{perc}}$
  $\phi_{\text{side}} \leftarrow \{\phi_{\text{MSE}}, \phi_{\text{cls}}, \phi_{\text{seg}}\}[i_{\text{task}}]$
end if
$\{\boldsymbol{x}_1, \boldsymbol{x}_2\} \leftarrow \text{Split}(\boldsymbol{x}, \boldsymbol{M})$
$\boldsymbol{x}_1 \leftarrow \phi_{\text{main}}(\boldsymbol{x}_1), \boldsymbol{x}_2 \leftarrow \phi_{\text{side}}(\boldsymbol{x}_2)$
$\boldsymbol{x} \leftarrow \text{Aggregate}(\boldsymbol{x}_1, \boldsymbol{x}_2)$
return $\boldsymbol{x}$

3.3 Importance score predictor

For the portability of MPA, we introduce a lightweight importance score predictor as illustrated in Fig. 2 (denoted as Ⓟ) to generate mask and allocate features to each path. In addition to only three parametric linear layers for fast point-wise computation, we use a non-parametric partial average layer $\mathcal{A}(\cdot)$ to capture multi-scale information. Given intermediate feature $\boldsymbol{u} \in \mathbb{R}^{H\times W\times C'}$, $\mathcal{A}(\cdot)$ performs global average pooling on the latter $C'/2$ channels of intermediate feature followed by expansion to the spatial size of feature, which aggregates information from all features as global information. The other $C'/2$ channels are left unchanged as local information. Formally, the partial average operation can be formulated as

$$\mathcal{A}\left(\boldsymbol{u}(h, w, c)\right) = \begin{cases} \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{u}(i,j,c)}{H\times W}, & \text{if } c > C'/2, \\ \boldsymbol{u}(h, w, c), & \text{otherwise.} \end{cases} \tag{1}$$

A hyper parameter (q or α) is leveraged to control the feature allocation between the main and side paths, which indicates the performance transition between tasks. In the encoder, the aggregation of ϕ_{hq} and ϕ_{lq} is associated with bitrate, thus we approximate a linear relationship between bitrate and aggregation ratio ρ_{enc} following [43] by an inverse log transformation

$$\rho_{\text{enc}} = \mathcal{F}(q) = \frac{\beta^{(q-1)/(Q_{\text{max}}-1)} - 1}{\beta - 1} \in [0, 1], \quad q \in [1, Q_{\text{max}}], \tag{2}$$

where β is the base of the log transformation and Q_{max} represents the maximum value of the range of q. Thus, as q increases, more features will enter $\phi_{\text{hq}}(\cdot)$ for supporting high-quality reconstruction. The aggregation in the decoder is irrelevant to bitrate, and $\rho_{\text{dec}} = 1 - \alpha$ so that as α increases, more features will enter the side paths. For efficiency, the mask is only generated once at the beginning of a stage and shared by all MP-blocks throughout the stage.

During training, since the direct sampling of $\boldsymbol{M}$ based on ρ from the importance score is non-differentiable, we first bias the output $\boldsymbol{u}'$ of the last Linear$(\cdot)$ in the predictor using a set of learnable parameters b to get shifted logits corresponding to each discrete q or α, and then use Gumbel-Sigmoid [35, 60] with threshold $\tau = 0.5$ to soften the sampling process as Eq. (3). After training, the predictor can binarize the score map according to ρ, and the bias layer in Fig. 2 will be discarded.

$$\boldsymbol{M} = \text{Gumbel-Sigmoid}(\boldsymbol{u}' + b, \tau). \tag{3}$$

4 Optimization strategy

4.1 Stage 1: training a generalized basic model

Since the perceptually optimized LIC model is well aligned with both human and machine perception, we first train a variable-rate model based on TinyLIC [55] and GAN method from HiFiC [61] for subsequent extension. We add SF in all ViT blocks and implement MPA in the encoder. Following the common practice in literatures [61, 2, 32], the optimized losses are Eqs. (6) and (7) for the joint optimization of the encoder E, decoder G, entropy model P and discriminator D:

$$\mathcal{L}_{\text{ratio}} = \frac{1}{S}\sum_{s=1}^{S}\left(\rho_{\text{enc}} - \frac{1}{H^{(s)}W^{(s)}}\sum_{h=1}^{H^{(s)}}\sum_{w=1}^{W^{(s)}}\boldsymbol{M}^{(s)}(h,w)\right)^2, \tag{4}$$

$$\mathcal{L}_G = \mathbb{E}_{\hat{\boldsymbol{y}}\sim p_{\boldsymbol{y}}}[-\log(D(\hat{\boldsymbol{y}}, G(\hat{\boldsymbol{y}})))], \tag{5}$$

$$\mathcal{L}_D = \mathbb{E}_{\hat{\boldsymbol{y}}\sim p_{\boldsymbol{y}}}[-\log(1 - D(\hat{\boldsymbol{y}}, G(\hat{\boldsymbol{y}})))] + \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[-\log D(E(\boldsymbol{x}), \boldsymbol{x})], \tag{6}$$

$$\mathcal{L}_{EGP} = \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[\lambda_r^{(q)} r(\hat{\boldsymbol{y}}) + d(\boldsymbol{x}, \hat{\boldsymbol{x}})] + \lambda_G\mathcal{L}_G + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}} + \lambda_{\text{ratio}}\mathcal{L}_{\text{ratio}}, \tag{7}$$

where $\boldsymbol{x}$, $\hat{\boldsymbol{x}}$, $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are the input image, reconstructed image, compressed latents before and after quantization, respectively. $\mathcal{L}_{\text{ratio}}$ is introduced to optimize the predictors in MP-Blocks which constrains the ratio of "1"s in $\boldsymbol{M}$ at all encoder stages to the encoder ratio target ρ_{enc}. S is the number of stages applying MPA, while $H^{(s)}$ and $W^{(s)}$ represent the spatial size at Stage s. $r(\cdot)$ and $d(\cdot,\cdot)$ represent the bitrate estimated by the entropy model P and $0.01\text{MSE}(\cdot,\cdot)$ (pixel values are scaled to $[0, 255]$). $\lambda_r^{(q)}$ in Eq. (7) varies according to the sampling of quality level q during training. Learned Perceptual Image Patch Similarity (LPIPS) [81] is chosen for $\mathcal{L}_{\text{perc}}$.

4.2 Stage 2: extending and optimizing side paths for multi-task

Obtaining a generalized model, we can add side paths to the decoder to achieve task-controllable image coding. In this training stage, there is no need to train the model entirely, but only to optimize the added parameters, i.e., the added side path ϕ_{side} and the corresponding predictor ϕ_{pred}. By sampling α during training, the main path and the added side path are randomly aggregated to fit the transition between different tasks. The training objective is simplified to

$$(\phi^*_{\text{side}}, \phi^*_{\text{pred}}) = \underset{\phi_{\text{side}}, \phi_{\text{pred}}}{\arg\min}\, \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[\lambda_r^{(q)} r(\hat{\boldsymbol{y}})] + \lambda_{\text{task}}\mathcal{L}_{\text{task}} + \lambda_{\text{ratio}}\mathcal{L}_{\text{ratio}}, \tag{8}$$

where ϕ^*_{side} and ϕ^*_{pred} are the optimal parameters of the added side path and predictor, $\mathcal{L}_{\text{task}}$ represents the task loss, and the target ratio in $\mathcal{L}_{\text{ratio}}$ is changed to decoder ratio ρ_{dec}. The options for task loss are varied. When optimizing for MSE, $\mathcal{L}_{\text{task}}$ is just a simple MSE loss measured between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$. When optimizing for a visual analysis task, $\mathcal{L}_{\text{task}}$ is a compound loss containing $d(\boldsymbol{x}, \hat{\boldsymbol{x}})$, $\mathcal{L}_{\text{perc}}$ and the full loss function of the task. Normalization should be applied if needed for augmentation. The task model is frozen during optimization. For instance, using the classification model ClsModel$(\cdot)$ with Norm$(\cdot)$, cross-entropy is used to compute loss between the classification result of $\hat{\boldsymbol{x}}$ and the ground truth GT:

$$\mathcal{L}_{\text{task}} = \text{CrossEntropy}(\text{ClsModel}(\text{Norm}(\hat{\boldsymbol{x}})), GT) + \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[d(\boldsymbol{x}, \hat{\boldsymbol{x}})] + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}}. \tag{9}$$

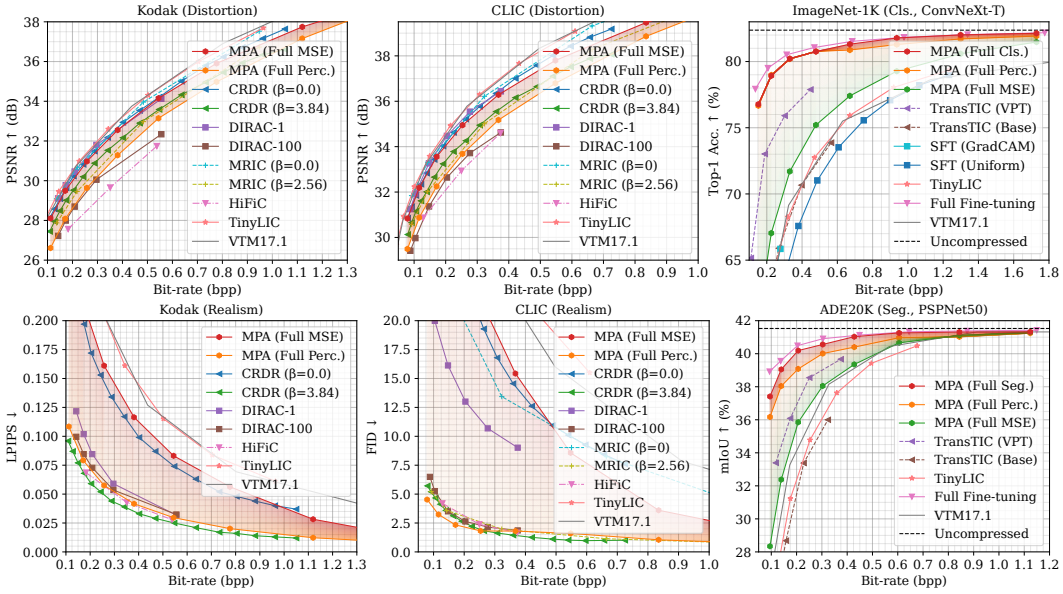


Figure 3: Multi-task performance. The curves of variable-rate models are plotted as solid lines, while dashed lines are for single-rate models. Colored areas represent the adjustable range of MPA.

5 Experiments

5.1 Experimental settings

Dataset. When training for perceptual quality and MSE, we use a combined dataset including Flicker2W [48], DIV2K [3], and CLIC [71] training sets, about 23K images in total. ImageNet-1K [14] and ADE20K [83] are used to train ϕ_{cls} and ϕ_{seg}, respectively. We evaluate the model on Kodak [39] and CLIC test set [71] for image compression, ImageNet validation set [14] for classification, and ADE20K validation set [83] for semantic segmentation.

Training. We set $\beta = 5$ in Eq. (2), $Q_{\text{max}} = 8$, $\lambda_r^{(q)} = \{18, 9.32, 4.83, 2.5, 1.3, 0.67, 0.35, 0.18\}$, $\lambda_G = 2.56$, $\lambda_{\text{perc}} = 4.26$, $\lambda_{\text{task}} = 1$, and $\lambda_{\text{ratio}} = 10$ for all training stages. q and α are uniformly sampled from $\{1, 2, 3, 4, 5, 6, 7, 8\}$ and $\{0, 1, 2, 3, 4, 5, 6, 7\}/7$ respectively. The training steps of Stage 1 in Sec. 4.1 are set to 3M, the first half without $\mathcal{L}_G$ and the last half with $\mathcal{L}_G$. The training steps of Stage 2 are set to 500K. In each training stage, the initial learning rate is set to 10^{-4}, decayed to 10^{-5} for the last 25% steps. When training ϕ_{perc} and ϕ_{MSE}, images are randomly cropped to 256×256×3, and a random horizontal/vertical flip is applied. ConvNeXt-Tiny [54] is used for training ϕ_{cls}, with images randomly resized and cropped to 256×256×3 followed by a random horizontal flip. PSPNet50 [82] is used for training ϕ_{seg}, with images randomly resized, flipped and cropped to 256×256×3. The batch size is set to 8 for all tasks. Adam [38] is used for optimization.

Evaluation. For human vision evaluation, we use Peak Signal-to-Noise Ratio (PSNR) to measure distortion, and use Fréchet Inception Distance (FID) [28] and LPIPS [81] to measure realism. We use the same protocol in [61, 2, 24, 32] to calculate FID, i.e., cropping images to overlapped patches of size 256×256×3. Note that FID is not calculated on Kodak because it yields only 192 patches, which is not sufficient for measuring FID. For the classification task, we use top-1 accuracy (acc.) to present the performance, with images first resized to a short edge of 292 and then center-cropped to 256×256×3. For the segmentation task, we use mean Intersection over Union (mIoU) to present the performance, with images first resized to a short edge of 512 and then center-cropped to 512×512×3.

5.2 Results of multi-task performance

For human vision, we compare the proposed MPA to the SOTA baselines of the unified model to evaluate R-D and R-P performance, i.e., MRIC [2], DIRAC [24], CRDR [32]. In addition, we

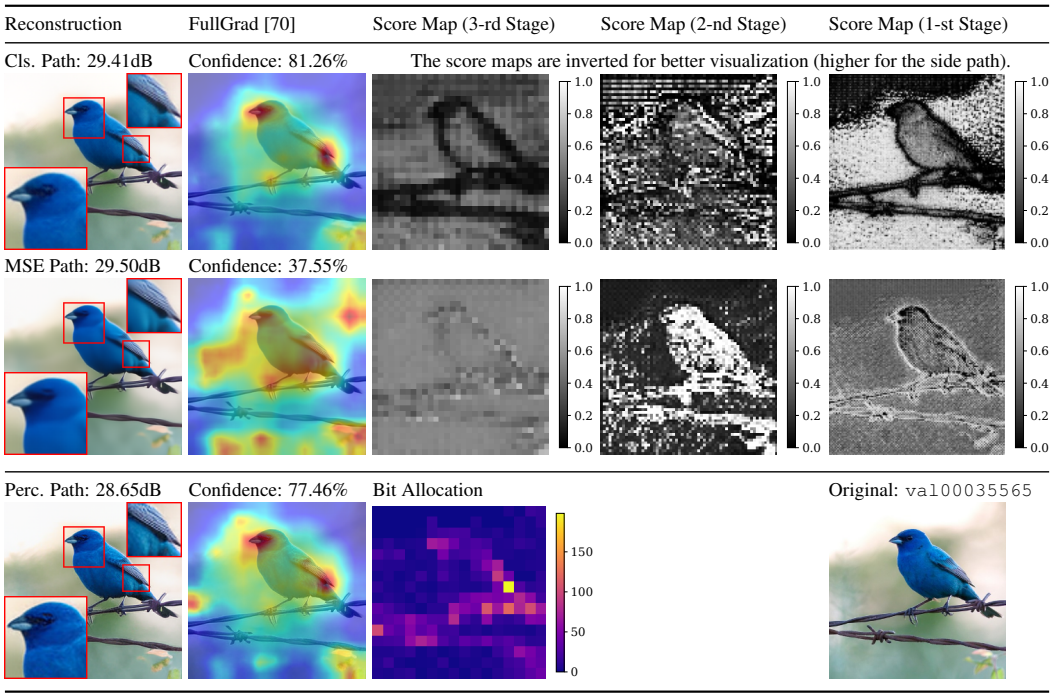


Figure 4: Visualization of the reconstructed images, FullGrad [70] and score maps. The image is from ImageNet [14] and resized to 256×256×3. The regions with warmer colors in FullGrad have larger gradients, indicating a stronger impact on the classification decision. The bitrate is 0.0888bpp.

add HiFiC [61] and VTM17.1 [1] for comparison as the anchors of fully perceptual optimization and traditional coding method, respectively. For machine vision, the classification task is the understanding of the global semantic information of an image and represents a high-level vision task. In contrast, the segmentation task challenges the model's ability to understand pixel-level semantics and represents a low-level vision task. Thus, we test the reconstructed images on classification and semantic segmentation tasks to comprehensively evaluate the performance of MPA for machine vision. We compare MPA to SOTA baselines with task-specific optimization, SFT [69] and TransTIC [11], along with VTM17.1 [1].

As shown in Fig. 3, MPA can achieve performance comparable to existing SOTA models. For human vision, the FID of MPA is lower than that of the best-performing model, CRDR, at a low bitrate. Since perceptual performance is more reflective of the human visual system at a low bitrate, this part of the gain is significantly beneficial. Considering the distortion, even with only partial parameters optimized for MSE, MPA still has a wide adjustable range comparable to other fully fine-tuned models, especially at a low bitrate. Note that we achieve such performance with a smaller decoder (9.295M for MPA vs. 13.38M for CRDR), fewer training steps (3.5M steps for MPA vs. 5M steps for CRDR), and partial fine-tuning (7.27% for MPA and 100% for CRDR). In terms of machine vision, ϕ_{perc}, ϕ_{cls} and ϕ_{seg} perform significantly better than models optimized for MSE, outperforming SFT and TransTIC which are specially optimized for vision tasks. With only 1.89% parameters fine-tuned for machine vision, MPA can even achieve accuracy comparable to the fully fine-tuned models, showcasing its powerful task transition capability. Note that any point in the colored areas in Fig. 3 can be achieved by adjusting q and α, as MPA supports continuously variable-rate coding and seamless transitions between all tasks using a unified model.

Discussion. We choose the perceptually optimized path to serve as the main path due to its ability to preserve high-level semantic features that are useful across various tasks. This generalization is achieved by using pre-trained classification models for measurement (VGG [68] for training and AlexNet [40] for testing). This method aligns well with both human and machine perceptions in the latent space [81]. The pre-trained VGG implicitly incorporates label information, making the distribution of the reconstructed image semantically closer to the original one, thus reducing

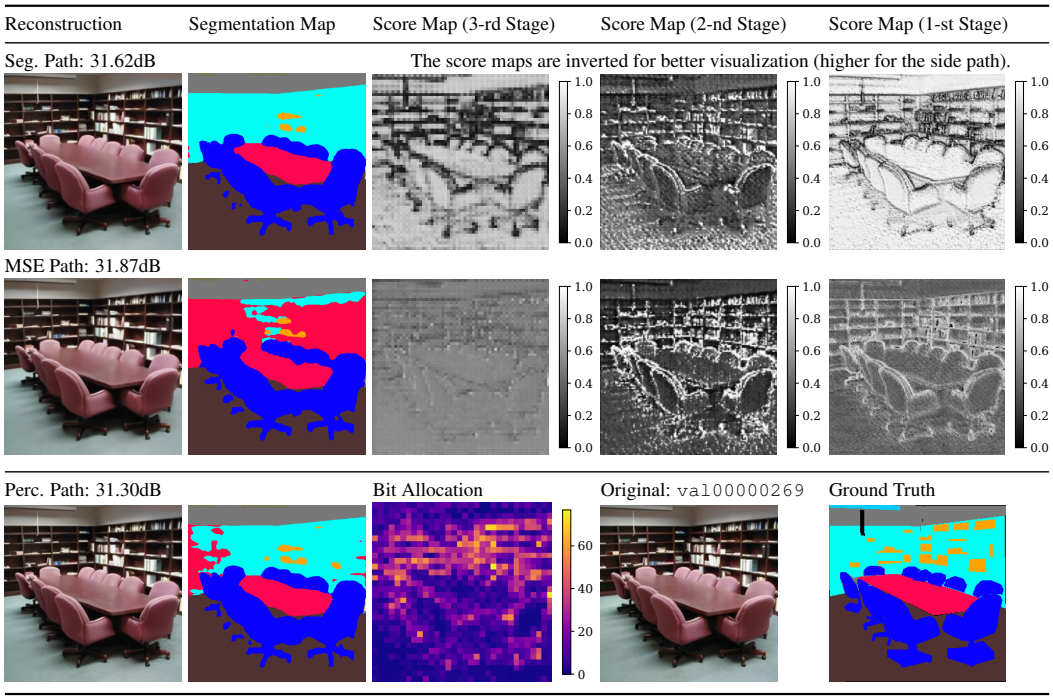


Figure 5: Visualization of the reconstructed images, segmentation maps and score maps. The image is from ADE20K [83] and resized to 512×512×3. The bitrate is 0.0718bpp.

perceptual distance and enhancing accuracy. Additionally, ϕ_{cls} and ϕ_{seg} use task-specific pre-trained models to compute cross-entropy loss (image-wise for classification and pixel-wise for segmentation), directly targeting task-specific accuracy. These constraints, along with semantic information from the first training stage, allow ϕ_{cls} and ϕ_{seg} to achieve higher accuracy than ϕ_{perc} with lower computational complexity. Moreover, perceptually optimized features can be easily adapted for MSE optimization, achieving higher PSNR. Perceptual loss preserves high-level semantic features, which supports the minimization of pixel-wise distortion when fine-tuned with MSE loss especially at a low bitrate [18]. Thus, perceptual optimization not only unify human and machine vision tasks but also enhance performance in traditional metrics like PSNR, making it a suitable method for the generalized model.

5.3 Visualization

To investigate the effects of different paths, we visualize the qualitative results in Figs. 4 and 5. MSE optimization prioritizes image textures, with fine-grained textures scoring higher, while analysis tasks focus on semantic regions. Score prediction differences at each stage highlight distinctions between MSE and machine vision perception. MSE-optimized features show significant score differences at lower-level stages (1st and 2nd), enhancing PSNR by improving textures. Conversely, machine vision-optimized features show significant score differences at all stages, indicating that both high- and low-level semantic features are crucial for analysis tasks. Consequently, images optimized for MSE without semantic supervision are visually smoother at low bitrate but may lose important semantic features, reducing accuracy. Paths optimized under LPIPS supervision retain more semantic information, ensuring the reconstructed image is semantically closer to the original one. By introducing task loss in Eq. (8), images decoded by ϕ_{cls} and ϕ_{seg} have more task-specific features for classification and segmentation respectively, leading to higher accuracy than ϕ_{perc}.

5.4 Diving into MPA

To further explore the configuration of the MPA, we perform a series of evaluations. Kodak [39], ImageNet-1K [14] and ADE20K [83] are used for evaluate BD-rate [6], top-1 accuracy and mIoU respectively. BD-rate is computed over the whole R-D curve. Top-1 accuracy and mIoU are computed

Table 1: Effects of path complexity

MLP Type	ϕ_{MSE} (BD-Rate ↓)	ϕ_{cls} (Acc. ↑)	ϕ_{seg} (mIoU ↑)
Bottleneck	19.51%	76.72%	37.41%
Inv. Bottleneck	16.04%	77.16%	37.76%

Table 2: Cross-validations on path choices

Task (Metric)	ϕ_{perc}	ϕ_{MSE}	ϕ_{cls}	ϕ_{seg}
MSE (BD-Rate ↓)	49.61%	**16.04%**	32.81%	34.19%
Cls. (Acc. ↑)	76.66%	60.59%	**76.77%**	73.57%
Seg. (mIoU ↑)	36.17%	28.34%	35.34%	**37.41%**

Table 3: Ablations on encoder

Components	BD-Rate ↓ against VTM
Full MPA	16.04%
w/o Predictors	16.25%
w/o ϕ_{hq}	17.05%
w/o ϕ_{lq}	17.18%

Table 4: Comparison of complexity

Models	#Param.	KFLOPs per pixel	Latency (ms)
MRIC [2]	69.14M	1118.17	11.89
TinyLIC [55]	28.46M	439.29	12.68
+ MLPs	+0.51M∼+2.04M	-56.68∼+0	-0.33∼+0
+ Predictors	+0.03M	+2.23	+0.09
+ Ⓢ & Ⓐ	+0	+0	+0.28
MPA	29.00M∼30.53M	384.84∼441.52	12.72∼13.05

at the lowest bitrate, i.e., 0.1521bpp on ImageNet-1K and 0.0948bpp on ADE20K. For the effects of complexity, Table 1 reveals that human vision-oriented tasks are more sensitive to the complexity of the paths, which suggests the use of inverted bottleneck MLPs, while machine vision-oriented paths can use bottleneck MLPs because the complexity has relatively less impact on the accuracy. As for the choice of paths, we conduct cross-validations as shown in Table 2, which demonstrates that the paths do learn the task-specific features to make their corresponding tasks perform optimally. To evaluate the effect of MPA in the encoder, we conduct ablations on each MPA component. We replace the predicted mask with a random mask and disable ϕ_{hq} and ϕ_{lq} during encoding. The results in Table 3 demonstrate that the predictors capture different features critical to high and low bitrate compression, and ϕ_{hq} and ϕ_{lq} are specialized to corresponding features respectively.

To evaluate the generalization of the learned features for the same task, we test ϕ_{cls} on Swin Transformer V2 [52]. Fig. 6 shows that ϕ_{cls} has a similar performance gain comparable to that on ConvNeXt [54] which is originally used for training. Note that ϕ_{cls} is only trained once without fine-tuning on SwinV2, reflecting the generalization of the learned features in MPA.

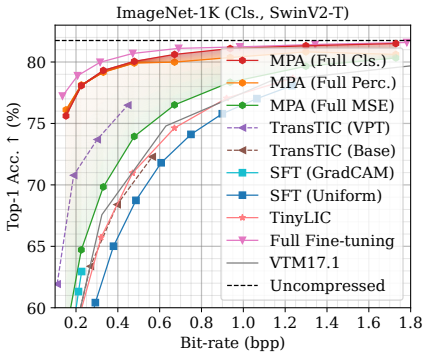


Figure 6: Validation on SwinV2-T [52]

The complexity of MPA is shown in Table 4. Here, the input size is 256×256×3. Latency is averaged over a batch of 16 images running 5000 times on a RTX3090 GPU through an encoder, a hyperprior entropy model, and a decoder. With the negligible computational overhead, MPA achieves comparable performance to the full fine-tuning model and has seamless transitions between tasks. The main limitations of MPA are its increased latency compared to the baseline and the need for separate training for each task. The former can be mitigated by developing a specialized operator to eliminate the frequent flattening, selection, and reassembly operations, while the latter can be addressed by leveraging multi-task learning techniques [72], which will be explored in future work.

6 Conclusion

In this paper, we propose Multi-Path Aggregation (MPA) to tackle the problem of coding for multi-task applications in an all-in-one manner. By aggregating task-specific paths, MPA can support a variety of tasks with seamless transitions between them for joint human-machine vision using a unified model and a unified representation. Merely fine-tuning as low as 1.89% additional parameters, MPA can achieve comparable performance to that of separable models optimized using dedicated criteria, showcasing its strong versatility and scalability. Future work will explore joint optimization of multiple paths to further unify training process and improve multi-task performance.

Acknowledgments and Disclosure of Funding

This work was supported in part by Natural Science Foundation of Jiangsu Province under Grant BK20241226 and BK20243038, and Jiangsu Provincial Key Research and Development Program under Grant BE2022155. The authors would like to express their sincere gratitude to the Interdisciplinary Research Center for Future Intelligent Chips (Chip-X) and Yachen Foundation for their invaluable support.

References

[1] Versatile video coding reference software version 17.1 (VTM-17.1). `https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-17.1`, July 2022.

[2] Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi-realism image compression with a conditional generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22324–22333, June 2023.

[3] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[4] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. In *Advances in Neural Information Processing Systems*, volume 35, pages 32897–32912. Curran Associates, Inc., 2022.

[5] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

[6] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *ITU SG16 Doc. VCEG-M33*, 2001.

[7] Lahiru D. Chamain, Fabien Racapé, Jean Bégaint, Akshay Pushparaja, and Simon Feltman. End-to-end optimized image compression for machines, a study. In *2021 Data Compression Conference (DCC)*, pages 163–172, 2021.

[8] Lahiru D. Chamain, Fabien Racapé, Jean Bégaint, Akshay Pushparaja, and Simon Feltman. End-to-end optimized image compression for multiple machine tasks, 2021.

[9] Tianlong Chen, Xuxi Chen, Xianzhi Du, Abdullah Rashwan, Fan Yang, Huizhong Chen, Zhangyang Wang, and Yeqing Li. Adamv-moe: Adaptive multi-task vision mixture-of-experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17346–17357, October 2023.

[10] Tong Chen and Zhan Ma. Variable bitrate image compression with quality scaling factors. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2163–2167, 2020.

[11] Yi-Hsin Chen, Ying-Chieh Weng, Chia-Hao Kao, Cheng Chien, Wei-Chen Chiu, and Wen-Hsiao Peng. Transtic: Transferring transformer-based image compression from human perception to machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23297–23307, October 2023.

[12] Hyomin Choi and Ivan V. Bajić. Scalable image coding for humans and machines. *IEEE Transactions on Image Processing*, 31:2739–2754, 2022.

[13] Ze Cui, Jing Wang, Shangyin Gao, Tiansheng Guo, Yihui Feng, and Bo Bai. Asymmetric gained deep image compression with continuous rate adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10532–10541, June 2021.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[15] Nikolaos Dimitriadis, Pascal Frossard, and François Fleuret. Pareto manifold learning: Tackling multiple tasks via ensembles of single-task models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8015–8052. PMLR, 23–29 Jul 2023.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[17] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 5547–5569. PMLR, 17–23 Jul 2022.

[18] Zhihao Duan, Ming Lu, Zhan Ma, and Fengqing Zhu. Lossy image compression with quantized hierarchical vaes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 198–207, January 2023.

[19] Zhihao Duan, Zhan Ma, and Fengqing Zhu. Unified architecture adaptation for compressed domain semantic inference. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(8):4108–4121, 2023.

[20] Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2005.

[21] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[22] Ruoyu Feng, Xin Jin, Zongyu Guo, Runsen Feng, Yixin Gao, Tianyu He, Zhizheng Zhang, Simeng Sun, and Zhibo Chen. Image coding for machines with omnipotent feature learning. In *Computer Vision – ECCV 2022*, pages 510–528, Cham, 2022. Springer Nature Switzerland.

[23] Changsheng Gao, Dong Liu, Li Li, and Feng Wu. Towards task-generic image compression: A study of semantics-oriented metrics. *IEEE Transactions on Multimedia*, 25:721–735, 2021.

[24] Noor Fathima Ghouse, Jens Petersen, Auke Wiggers, Tianlin Xu, and Guillaume Sautière. A residual diffusion model for high perceptual quality codec augmentation, 2023.

[25] Jacob Gildenblat and contributors. Pytorch library for cam methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

[26] Alon Harell, Anderson De Andrade, and Ivan V. Bajić. Rate-distortion in image coding for machines. In *2022 Picture Coding Symposium (PCS)*, pages 199–203, 2022.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[29] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[30] Yueyu Hu, Shuai Yang, Wenhan Yang, Ling-Yu Duan, and Jiaying Liu. Towards coding for human and machine vision: A scalable image coding approach. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020.

[31] Yuzheng Hu, Ruicheng Xian, Qilong Wu, Qiuling Fan, Lang Yin, and Han Zhao. Revisiting scalarization in multi-task learning: A theoretical perspective. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 48510–48533. Curran Associates, Inc., 2023.

[32] Shoma Iwai, Tomo Miyazaki, and Shinichiro Omachi. Controlling rate, distortion, and realism: Towards a single comprehensive neural image compression model. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2900–2909, January 2024.

[33] Shoma Iwai, Tomo Miyazaki, Yoshihiro Sugaya, and Shinichiro Omachi. Fidelity-controllable extreme image compression with generative adversarial networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8235–8242, 2021.

[34] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[35] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[36] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 709–727, Cham, 2022. Springer Nature Switzerland.

[37] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *the 3rd Int. Conf. on Learning Representations*, 2015.

[39] Eastman Kodak. Kodak lossless true color image suite (photocd pcd0992), 1993.

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[41] Nikolai Körber, Eduard Kromer, Andreas Siebert, Sascha Hauke, Daniel Mueller-Gritschneder, and Björn Schuller. Egic: Enhanced low-bit-rate generative image compression guided by semantic segmentation, 2023.

[42] Nam Le, Honglei Zhang, Francesco Cricri, Ramin Ghaznavi-Youvalari, and Esa Rahtu. Image coding for machines: an end-to-end learned approach. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1590–1594, 2021.

[43] Jooyoung Lee, Seyoon Jeong, and Munchurl Kim. Selective compression learning of latent representations for variable-rate image compression. In *Advances in Neural Information Processing Systems*, volume 35, pages 13146–13157. Curran Associates, Inc., 2022.

[44] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

[45] Huanyang Li and Xinfeng Zhang. Human-machine collaborative image compression method based on implicit neural representations. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pages 1–1, 2024.

[46] 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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[47] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[48] Jiaheng Liu, Guo Lu, Zhihao Hu, and Dong Xu. A unified end-to-end framework for efficient deep image compression. *arXiv preprint arXiv:2002.03370*, 2020.

[49] Jinming Liu, Heming Sun, and Jiro Katto. Improving multiple machine vision tasks in the compressed domain. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 331–337, 2022.

[50] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14388–14397, June 2023.

[51] Kang Liu, Dong Liu, Li Li, Ning Yan, and Houqiang Li. Semantics-to-signal scalable image compression with learned revertible representations. *International Journal of Computer Vision*, 129(9):2605–2621, 2021.

[52] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12009–12019, June 2022.

[53] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

[54] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, June 2022.

[55] Ming Lu, Fangdong Chen, Shiliang Pu, and Zhan Ma. High-efficiency lossy image coding through adaptive neighborhood information aggregation. *arXiv preprint arXiv:2204.11448*, 2022.

[56] Ming Lu, Peiyao Guo, Huiqing Shi, Chuntong Cao, and Zhan Ma. Transformer-based image compression. In *2022 Data Compression Conference (DCC)*, pages 469–469. IEEE, 2022.

[57] Pingchuan Ma, Tao Du, and Wojciech Matusik. Efficient continuous pareto exploration in multi-task learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6522–6531. PMLR, 13–18 Jul 2020.

[58] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6597–6607. PMLR, 13–18 Jul 2020.

[59] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. `https://github.com/pytorch/vision`, 2016.

[60] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12309–12318, June 2022.

[61] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11913–11924. Curran Associates, Inc., 2020.

[62] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.

[63] Michinari Momma, Chaosheng Dong, and Jia Liu. A multi-objective / multi-task learning framework induced by pareto stationarity. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15895–15907. PMLR, 17–23 Jul 2022.

[64] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035. Curran Associates, Inc., 2019.

[65] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[66] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[67] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[68] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[69] Myungseo Song, Jinyoung Choi, and Bohyung Han. Variable-rate deep image compression through spatially-adaptive feature transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2380–2389, October 2021.

[70] Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems*, volume 32, pages 4124–4133. Curran Associates, Inc., 2019.

[71] George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020), 2020.

[72] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3614–3633, 2022.

[73] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[74] Shurun Wang, Shiqi Wang, Wenhan Yang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. Towards analysis-friendly face representation with scalable feature and texture compression. *IEEE Transactions on Multimedia*, 24:3169–3181, 2022.

[75] Shurun Wang, Zhao Wang, Shiqi Wang, and Yan Ye. Deep image compression toward machine vision: A unified optimization framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6):2979–2989, 2023.

[76] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[77] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[78] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[79] Fuzhao Xue, Ziji Shi, Futao Wei, Yuxuan Lou, Yong Liu, and Yang You. Go wider instead of deeper. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8779–8787, Jun. 2022.

[80] Ning Yan, Changsheng Gao, Dong Liu, Houqiang Li, Li Li, and Feng Wu. Sssic: Semantics-to-signal scalable image coding with learned structural representations. *IEEE Transactions on Image Processing*, 30:8939–8954, 2021.

[81] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, June 2018.

[82] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, July 2017.

[83] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

[84] Yinhao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *International Conference on Learning Representations*, 2022.

Appendix

A Broader impacts

MPA provides a flexible and task-oriented optimization approach for a unified coding framework. By leveraging the different importance of features for different tasks and feature correlations across tasks, MPA can empower existing learned image compression models with the ability to reconstruct images with various orientations based on the needs of users or machines (such as fidelity and accuracy), significantly enhancing their versatility. This research not only proposes and sets a new baseline for multi-task coding but also offers a new avenue for the exploration of joint human-machine vision. By exploiting generalized features and few-step fine-tuning for a tiny proportion of parameters, MPA enables joint human-machine vision for multiple tasks in an all-in-one manner, without massive handcrafted techniques. This advancement represents a significant step toward creating a versatile coding framework that can adapt to a wide range of applications and requirements. This can have a positive societal impact by saving a massive amount of storage space, as only a unified model and a unified representation need to be stored rather than multiple coding models and representations. By reducing storage requirements, it can lower the cost of data storage and transmission, making it more accessible for a wide range of applications. Additionally, this reduction in storage can lead to more efficient use of resources and energy, contributing to environmental sustainability. Lower storage costs and improved resource efficiency can also facilitate broader adoption of advanced coding technologies in various industries, enhancing scalability and operational efficiency. Although there are limitations in scenarios requiring extremely high fidelity or accuracy, which may lead to potential negative societal impacts such as misjudgments on reconstructed images, these issues can be addressed by future work aimed at performance improvements or the development of solutions tailored to specific scenarios.

B Implementation details

B.1 Architecture

Our implementation is based on TinyLIC [55]. The multi-path mechanism is fulfilled by replacing the default Single-Path Residual Neighborhood Attention Block (SP-RNAB) with Multi-Path RNAB (MP-RNAB, i.e., MP-Block in Fig. 2) at the first three stages of the main encoder and decoder (see Fig. 7). To build a continuously variable-rate baseline, we introduce the scaling factor (SF) modulation $\boldsymbol{s}_q$ [10] and non-linear interpolation [13, 43] as Eq. (10). The SF layer and its inverse operation (ISF) are embedded for the latent representation and each Transformer Block. *Other settings are consistent with TinyLIC.*

$$\boldsymbol{s}_q = \begin{cases} \boldsymbol{s}_q, & \text{if } q \in \{1, 2, ..., Q_{\text{max}}\}, \\ \boldsymbol{s}_{\lfloor q \rfloor}^{1-(q-\lfloor q \rfloor)} \cdot \boldsymbol{s}_{\lceil q \rceil}^{q-\lfloor q \rfloor}, & \text{otherwise.} \end{cases} \tag{10}$$

In terms of path configuration, we use inverted bottleneck MLPs for human vision (i.e., ϕ_{perc} and ϕ_{MSE}) and bottleneck MLPs for machine vision (i.e., ϕ_{cls} and ϕ_{seg}), as shown in Fig 8. The bottleneck MLP is used to address low complexity demands, while the inverted bottleneck MLP is used to address high complexity demands. Specifically, the bottleneck MLP first down-samples input channels to C' followed by GELU activation, then up-samples it to C, while the inverted one first up-samples C to C'' with activation and then down-samples it to C. In our implementation, we set $C'' = 2C$ as TinyLIC [55] does, and set $C' = C/2$ to align the up/down-sampling ratio.

B.2 MPA in encoder

As we mentioned in Sec. 3.2, we also implement MPA in the encoder, i.e., ϕ_{hq} and ϕ_{lq}. Setting $\beta = 5$ in Eq. 2, the relationship between q and the aggregation ratio ρ_{enc} is visualized in Fig. 9, which fits an approximate linear relationship between bitrate and ρ_{enc} on both Kodak [39] and CLIC [71].

B.3 Reproducibility

All experiments are conducted on a server with two Intel Xeon Silver 4210 CPUs and a single NVIDIA RTX3090 GPU. Our implementation relies on the open-source repository of TinyLIC [55],

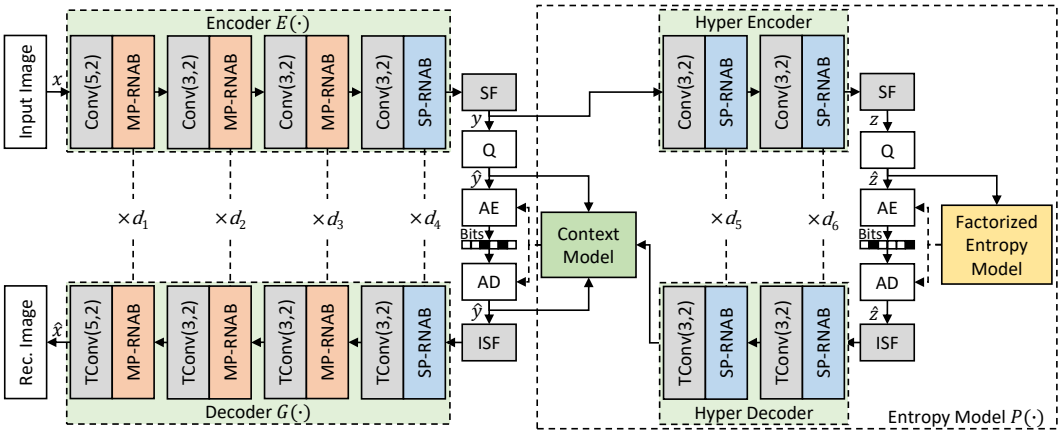
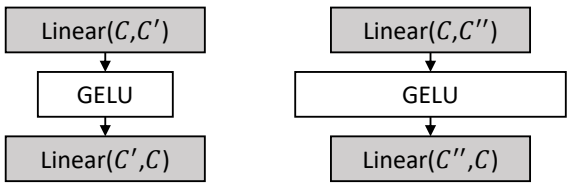


Figure 7: MPA in TinyLIC [55]. d_i is the number of RNABs used at i-th stage. Convolution Conv(k,s) and its transposed version TConv(k,s) apply the kernel at a size of $k \times k$ and a stride of s. Uniform Quantization is used in Q; AE and AD stand for respective Arithmetic Encoding and Decoding.



(a) Bottleneck MLP (b) Inverted Bottleneck MLP

Figure 8: Two types of MLP. Linear(C, C') denotes a fully-connected layer with C input channels and C' output channels. We set $C' = C/2$ and $C'' = 2C$ in our implementation.

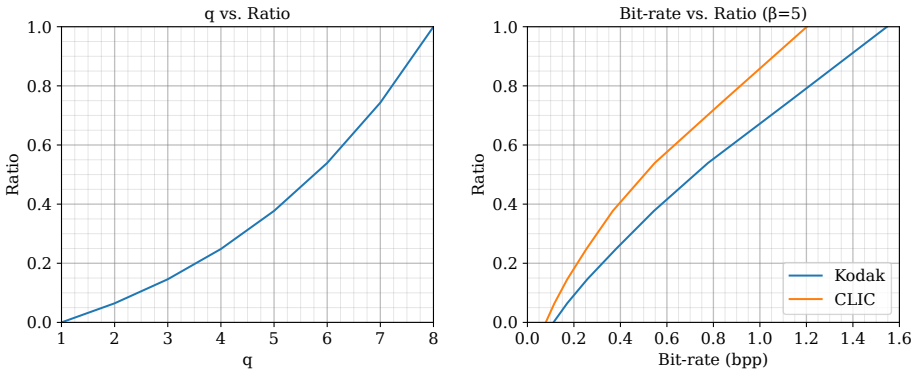


Figure 9: The relationship between q and the aggregation ratio ρ_{enc}.

which is based on PyTorch [64] and CompressAI [5]. The open-source baselines, i.e., SFT [69] and TransTIC [11], are re-tested using their public repository. For the comparison to baselines without public code and weights, we align our experimental setup with theirs. ConvNeXt Tiny [54] and Swin Transformer V2 Tiny [52] are created using timm [77] with officially published pre-trained weights. PSPNet50 [54] and its pre-trained weights are obtained from the authors' official repository. FullGrad [70] is visualized using an open-source PyTorch library [25].

C Comparison to other related work

Mixture-of-Experts (MoE) [34, 67] enables Transformer models [73, 16] to have non-uniform computing patterns, improving downstream task performance [44, 17, 21, 79], gaining multi-modality modelling capabilities [4, 76], or enabling multi-task learning [9]. They replace the Feed-Forward Network (FFN) with multiple experts and routing tokens via gating networks to the appropriate experts for inference. Unfortunately, the strengths of MoE are most predominantly represented in large language models due to the scaling law [37], which is not suitable for storage- and complexity-sensitive scenarios. In comparison, MPA is quite different from the common practice of MoE, as listed in Table 5.

Table 5: Comparison between MPA and MoE

Aspect	**MPA**	**MoE**
Architecture	Unified all-in-one model with multiple task-specific paths	Multiple expert models selected by a gating network
Optimization	Two-stage optimization with partial parameter fine-tuning	Joint optimization with expert selection via gating network
Routing	Leverages feature importance and correlations across tasks	Utilizes the expertise of different experts for specific inputs
Efficiency	Designed specifically for storage- and computation-sensitive multi-task scenarios	Specially used in large models to significantly increase parameter count for performance boost
Usage	User-defined selection of task paths	Expert allocation entirely decided by the gating network

Low-Rank Adaptation (LoRA) [29] is a technique that reduces the number of trainable parameters in large pre-trained models by factorizing weight matrices into low-rank forms, enabling efficient fine-tuning with minimal computational cost while maintaining performance across various tasks. Our experiments recognize the effectiveness of LoRA in Table 6. Although MPA involves more fine-tuning parameters, it also achieves better performance. Moreover, our work has distinct features. We utilize predictors to support coding for multi-task applications and smooth transitions between tasks within an all-in-one framework. The low-rank structure design of LoRA inspires the future work to consider improvements for the side path.

Table 6: R-D performance comparison between MPA and LoRA. The ranks r of LoRA are set to 64, 16, and 4, respectively, and the `lora_alpha` (NOT the α in MPA) is fixed to 1. We optimize MSE paths for low distortion. Other experimental settings are consistent with those in Sec. 4.

Dataset	BD-Rate against VTM ↓			
	MPA (7.27% ft.)	LoRA (3.04% ft.)	LoRA (0.83% ft.)	LoRA (0.28% ft.)
Kodak [39]	16.04%	16.26%	16.39%	16.83%
CLIC [71]	21.43%	21.80%	21.95%	22.59%

D Loss functions

Our loss functions follow the same forms and hyperparameters as those in [2], which have been thoroughly evaluated. For the task loss $\mathcal{L}_{\text{task}}$ we propose, we provide detailed ablation studies in Table 7 to demonstrate that our current combination achieves a competitive trade-off. Regarding the role of each term, LPIPS enriches the semantic information of the reconstructed images, improving generalization; MSE loss enforces pixel value consistency between the reconstructed and original images; and the cross-entropy loss directly optimizes classification accuracy.

Table 7: Ablations on loss terms in $\mathcal{L}_{\text{task}}$. We use the same experimental settings as in Sec. 4, and re-train the classification path for each case. Metrics are evaluated at 0.1521bpp on ImageNet-1K [14]. The top-3 results are underlined. $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{perc}}$ and $\mathcal{L}_{\text{ce}}$ denote MSE loss, perceptual loss and cross-entropy loss respectively.

No.	$\mathcal{L}_{\text{MSE}}$	$\mathcal{L}_{\text{perc}}$	$\mathcal{L}_{\text{ce}}$	PSNR ↑	LPIPS ↓	Top-1 Acc. ↑ on ConvNeXt-T [54]	Top-1 Acc. ↑ on SwinV2-T [52]
1	✓			<u>26.7dB</u>	0.330	62.22%	59.27%
2		✓		25.7dB	<u>0.229</u>	74.21%	72.25%
3			✓	24.2dB	0.305	<u>76.79%</u>	<u>75.16%</u>
4		✓	✓	25.7dB	<u>0.244</u>	<u>77.01%</u>	<u>75.96%</u>
5	✓		✓	<u>26.5dB</u>	0.275	76.17%	74.90%
6	✓	✓		<u>26.6dB</u>	<u>0.238</u>	74.28%	73.01%
7	✓	✓	✓	<u>26.5dB</u>	<u>0.244</u>	<u>76.77%</u>	<u>75.61%</u>

E Licenses for released assets

We list the licenses for the released assets we use in Table 8.

Table 8: Licenses for released assets

Asset	**License**
TinyLIC [55]	Apache-2.0 license
TransTIC [11]	Apache-2.0 license
SFT [69]	Not specified
ConvNeXt [54]	MIT license
PSPNet [82]	MIT license
Swin Transformer V2 [52]	MIT license
ResNet [27] (in TorchVision [59])	BSD-3-Clause license
Faster R-CNN [65] (in Detectron2 [78])	Apache-2.0 license
VTM17.1 [1]	BSD license
Flicker2W [48]	Custom (research-only)
DIV2K [3]	Custom (research-only)
CLIC [71]	Not specified
Kodak [39]	Not specified
ImageNet-1K [14]	Custom (research-only, non-commercial)
ADE20K [83]	Custom (research-only, non-commercial)
MS-COCO 2017 [46]	Custom

F More results

F.1 Quantitative results

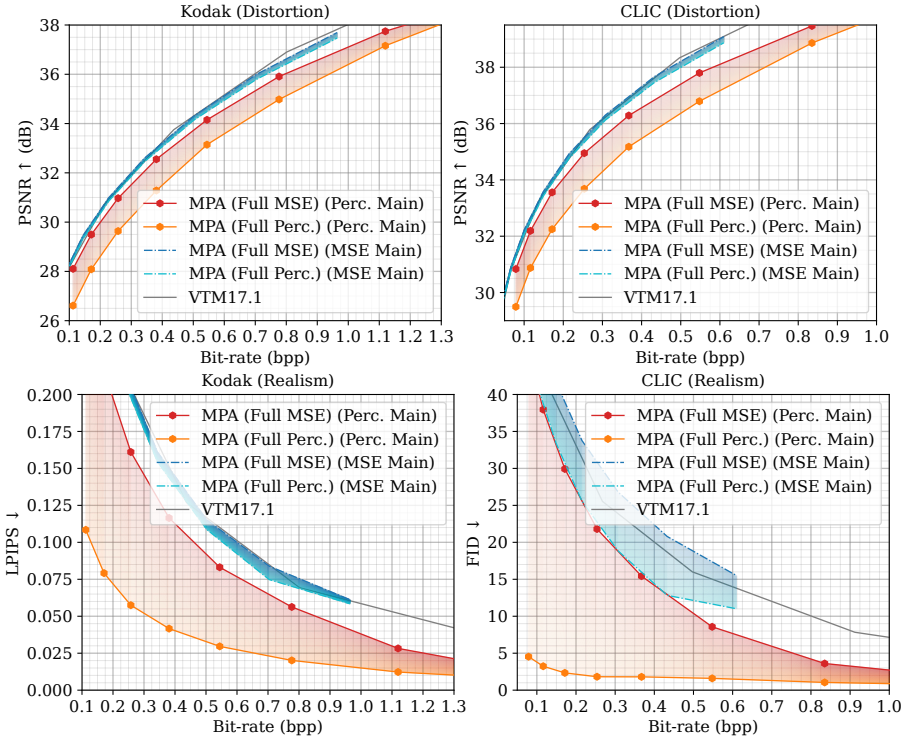


Figure 10: Multi-task performance with optimizing MSE path as the main path. The results reveal that perceptual optimization is more generalized than MSE optimization for multi-task applications. Although MSE optimization achieves lower distortion, its perceptual quality is more severely degraded compared to that of perceptual optimization, with a significantly narrower adjustable range.

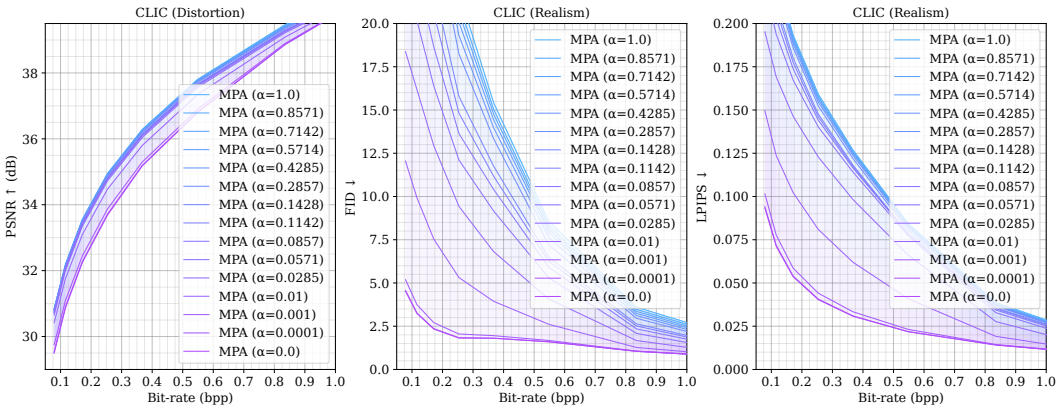


Figure 11: R-D and R-P performance of MPA with variable α on CLIC dataset [71].

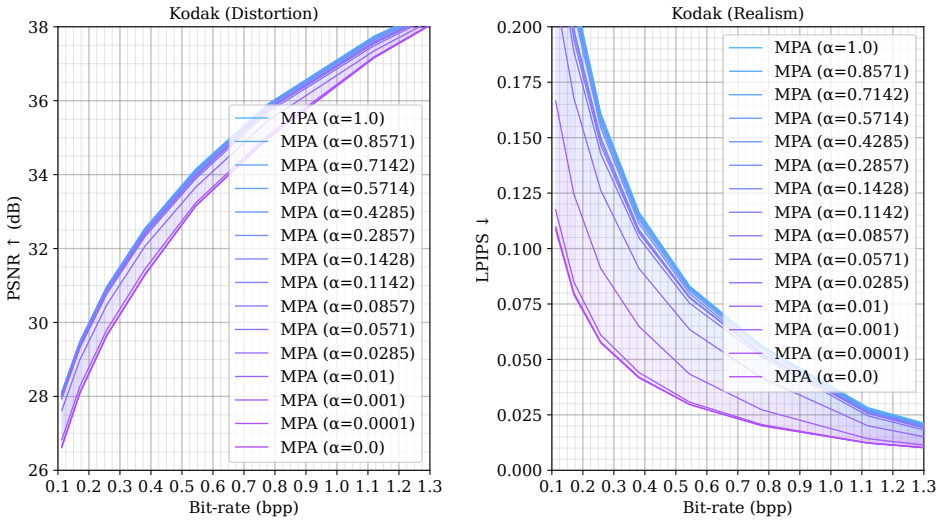


Figure 12: R-D and R-P performance of MPA with variable α on Kodak dataset [39].

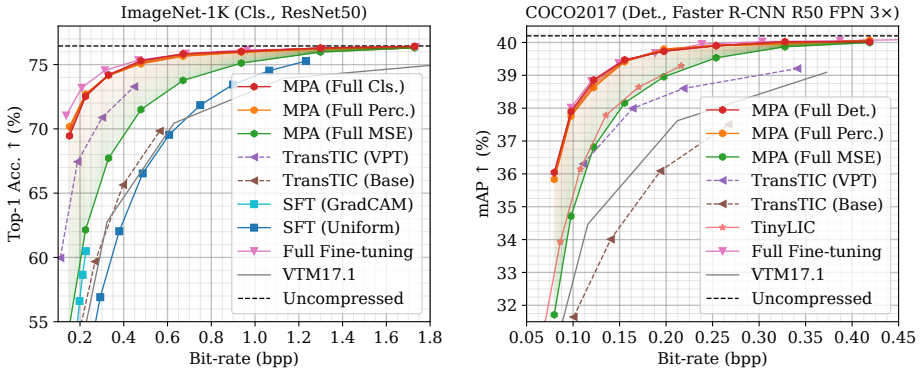


Figure 13: Classification and objection detection performance of MPA on ImageNet-1K [14] and MS-COCO 2017 [46]. ResNet50 [27] and Faster R-CNN R50 [65] are used for inference. Top-1 accuracy and mAP (mean Average Precision) are used as metrics for comparison.

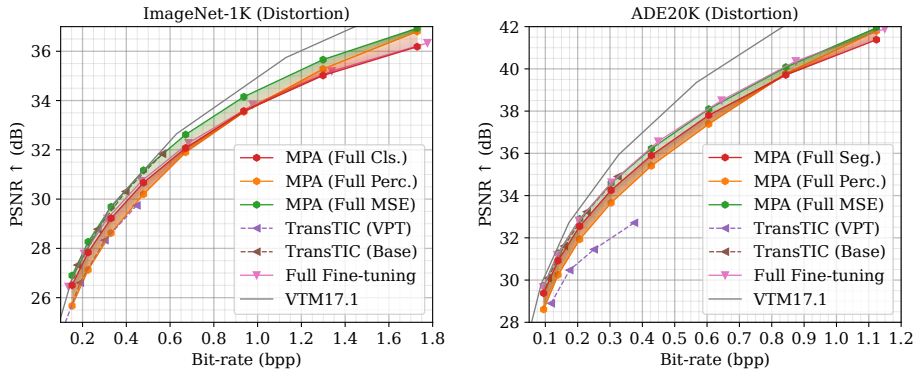


Figure 14: R-D performance of MPA on ImageNet-1K [14] and ADE20K [83].

F.2 Qualitative results

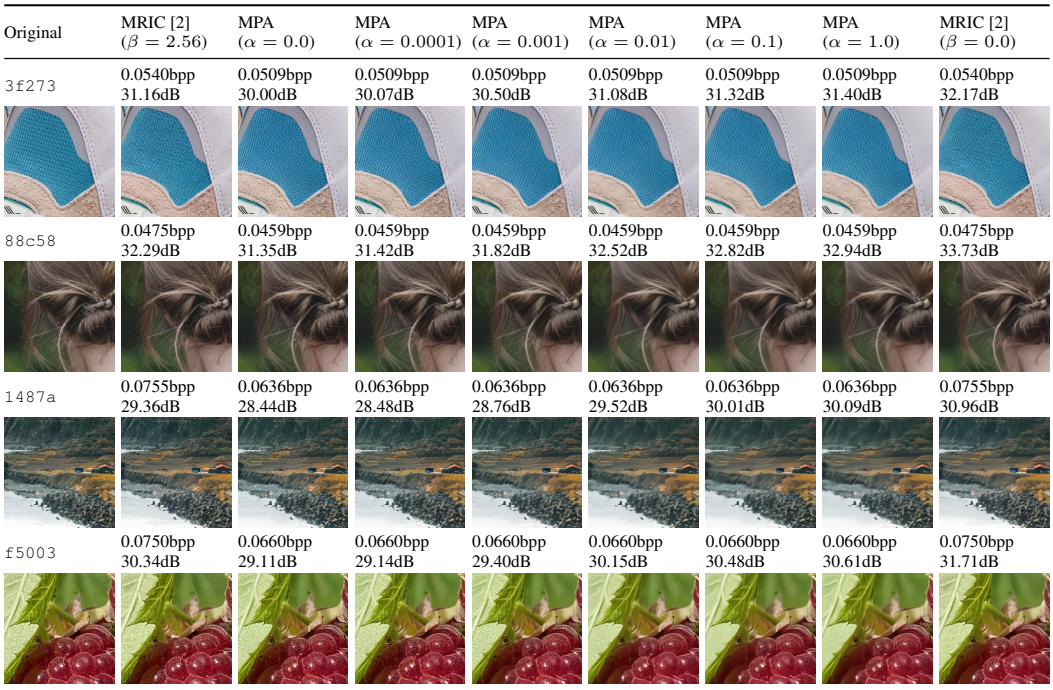


Figure 15: We visualize the reconstructed images of MPA with variable α compared to MRIC [2]. The image is from CLIC test set [71] and is cropped to 256×256×3. The selected paths are ϕ_{perc} and ϕ_{MSE}. Larger β for MRIC and smaller α for MPA stand for higher realism. Note that both the bitrate and the PSNR are calculated on uncropped images.

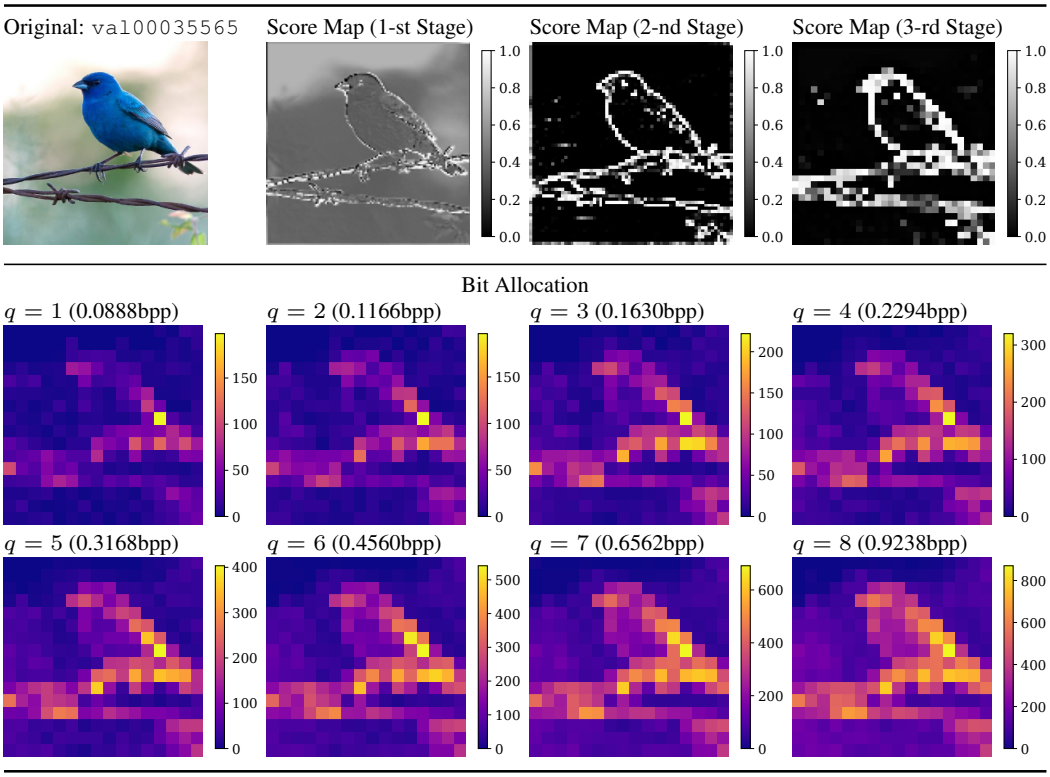


Figure 16: We visualize the bit allocation and score maps in the encoder. The image is from ImageNet validation set [14] and resized to 256×256×3. The 1st, 2nd, and 3rd stages are the first three stages of the encoder, consecutively.

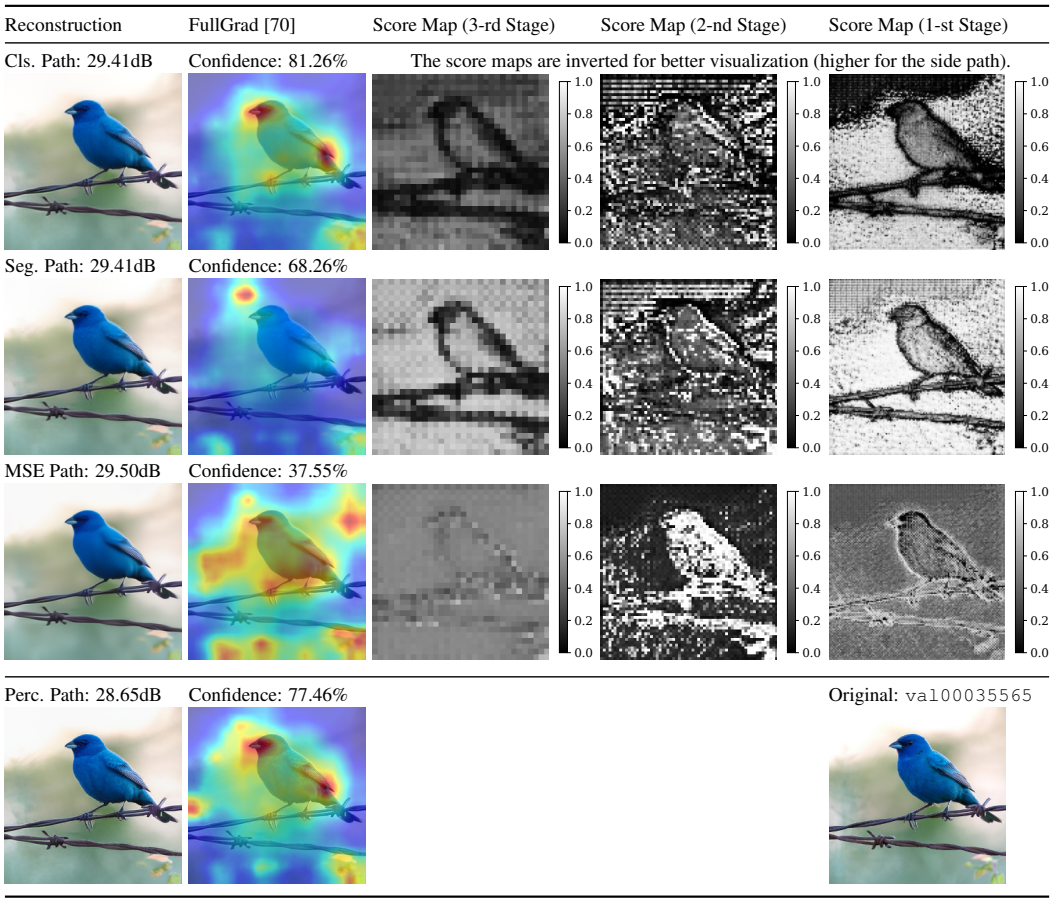


Figure 17: We visualize the reconstructed images, FullGrad [70] and score maps predicted by importance predictors in each path. The image is from ImageNet validation set [14] and resized to 256×256×3. The 3rd, 2nd, and 1st stages are the last three stages of the decoder, consecutively. The regions with warmer colors in FullGrad represent areas with larger gradients, indicating a stronger impact on the classification decision. q is set to 1 for a more distinct comparison. Note that the score maps are inverted for better visualization, i.e., larger scores indicate prioritized for entering the selected side path. The bitrate is 0.0888bpp.

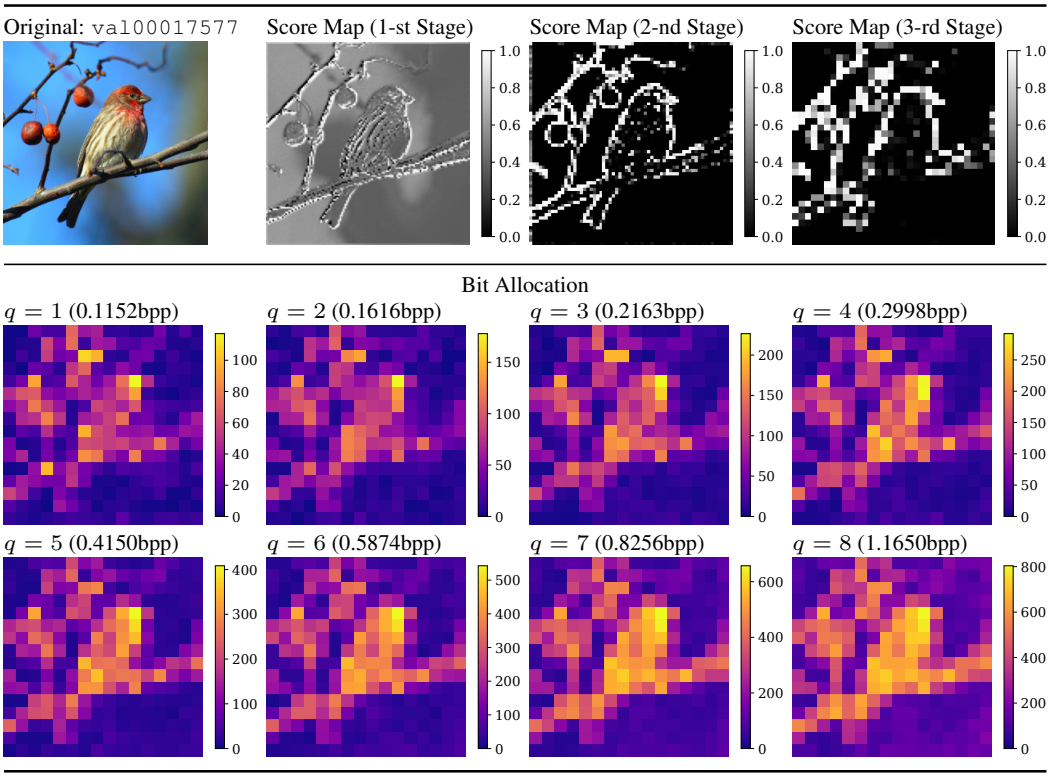


Figure 18: We visualize the bit allocation and score maps in the encoder. The image is from ImageNet validation set [14] and resized to 256×256×3. The 1st, 2nd, and 3rd stages are the first three stages of the encoder, consecutively.

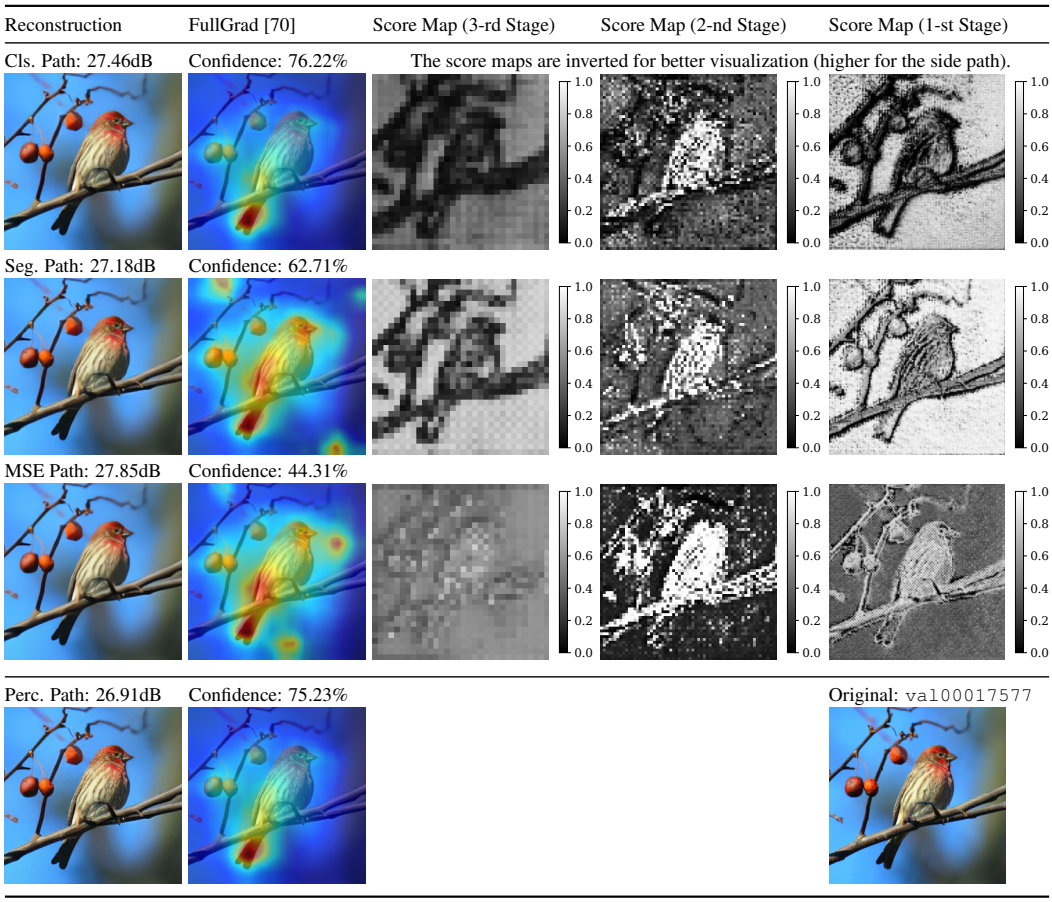


Figure 19: We visualize the reconstructed images, FullGrad [70] and score maps predicted by importance predictors in each path. The image is from ImageNet validation set [14] and resized to 256×256×3. The 3rd, 2nd, and 1st stages are the last three stages of the decoder, consecutively. The regions with warmer colors in FullGrad represent areas with larger gradients, indicating a stronger impact on the classification decision. q is set to 1 for a more distinct comparison. Note that the score maps are inverted for better visualization, i.e., larger scores indicate prioritized for entering the selected side path. The bitrate is 0.1152bpp.

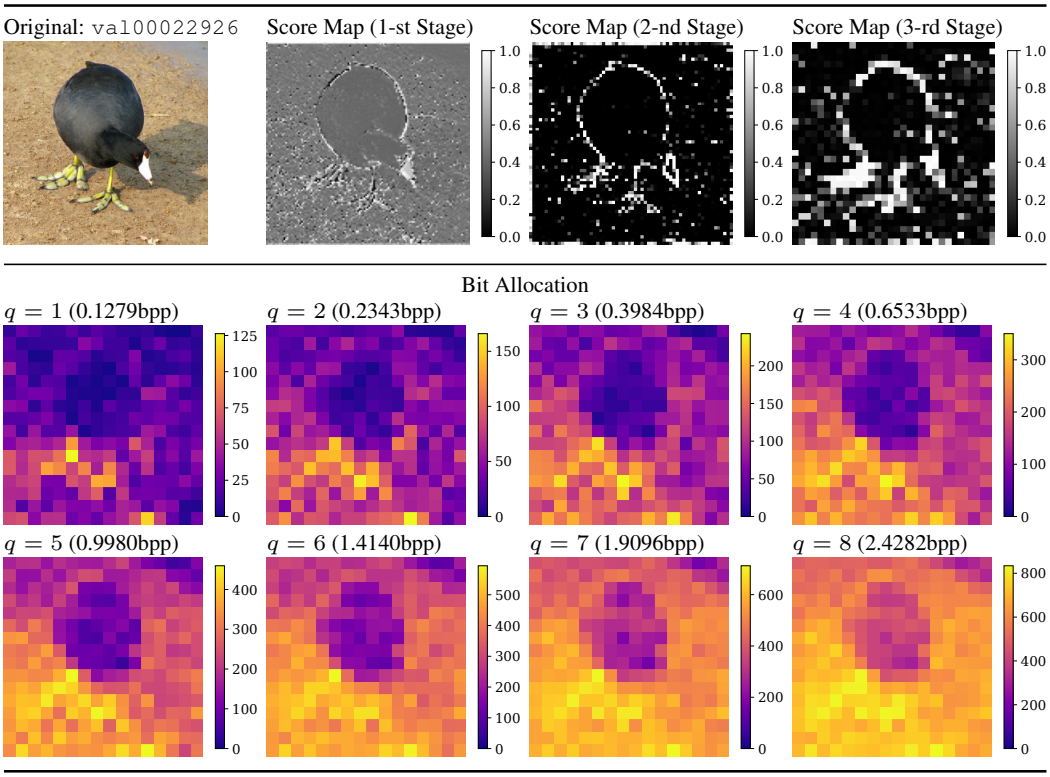


Figure 20: We visualize the bit allocation and score maps in the encoder. The image is from ImageNet validation set [14] and resized to 256×256×3. The 1st, 2nd, and 3rd stages are the first three stages of the encoder, consecutively.

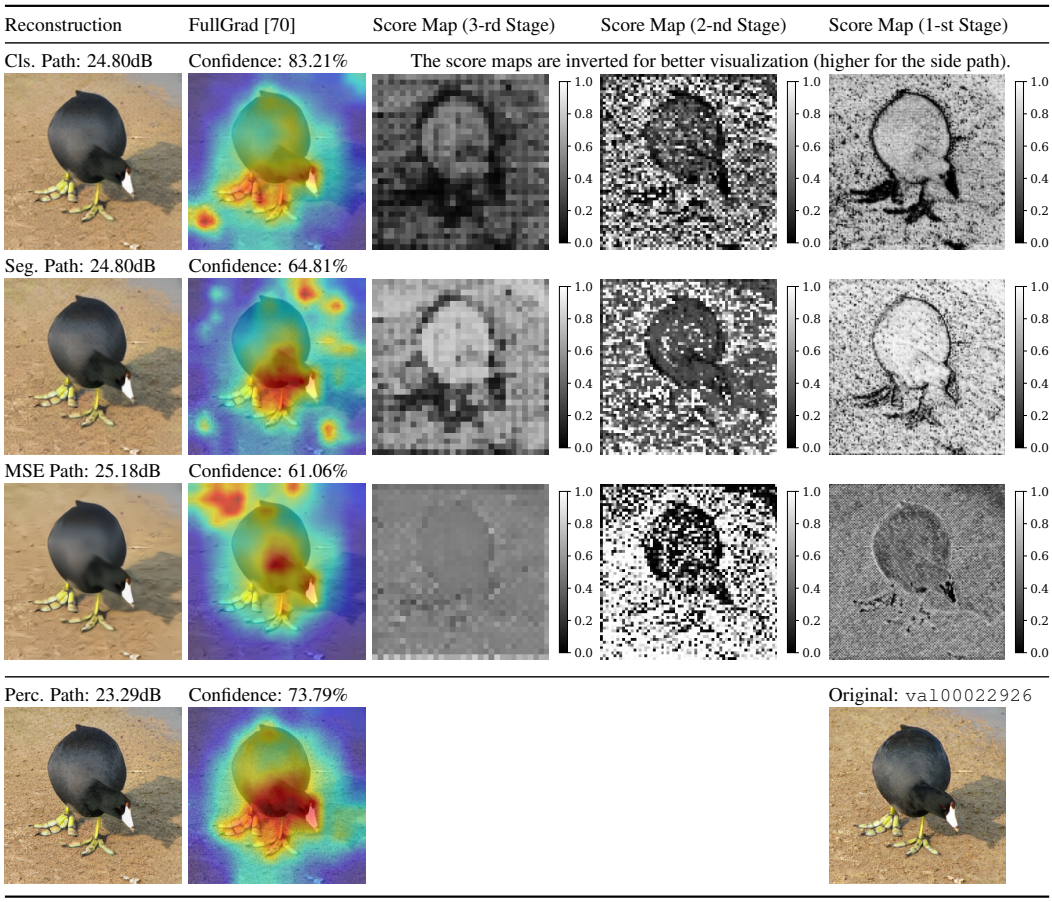


Figure 21: We visualize the reconstructed images, FullGrad [70] and score maps predicted by importance predictors in each path. The image is from ImageNet validation set [14] and resized to 256×256×3. The 3rd, 2nd, and 1st stages are the last three stages of the decoder, consecutively. The regions with warmer colors in FullGrad represent areas with larger gradients, indicating a stronger impact on the classification decision. q is set to 1 for a more distinct comparison. Note that the score maps are inverted for better visualization, i.e., larger scores indicate prioritized for entering the selected side path. The bitrate is 0.1279bpp.

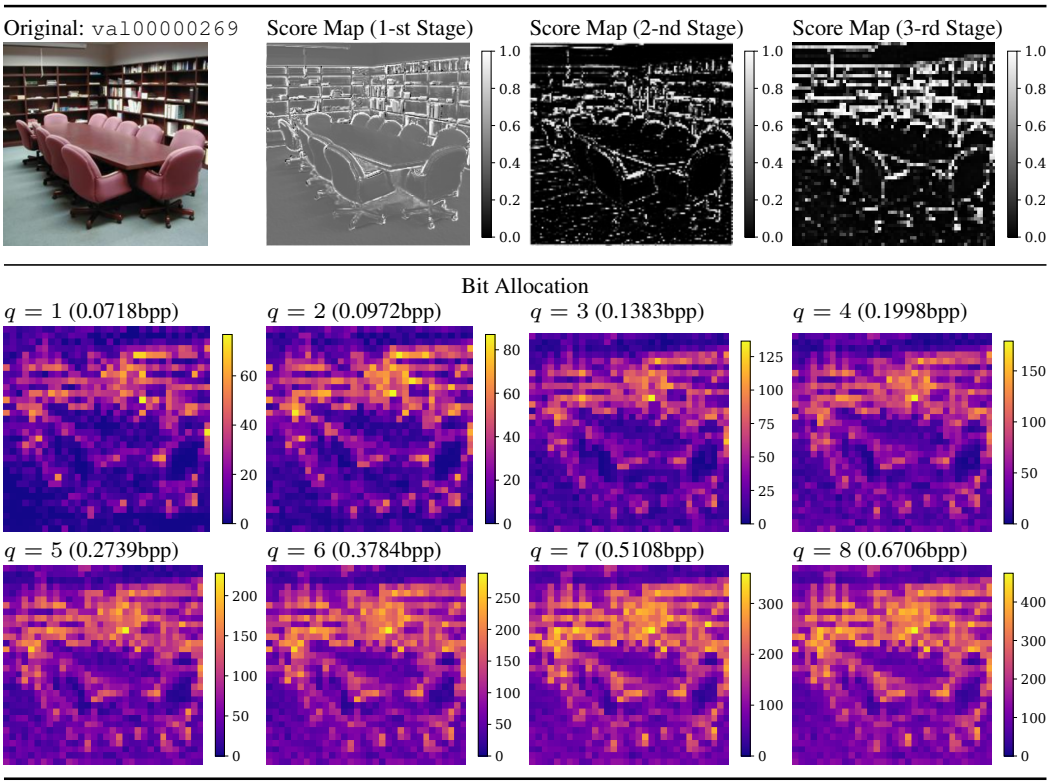


Figure 22: We visualize the bit allocation and score maps in the encoder. The image is from ADE20K validation set [83] and resized to 512×512×3. The 1st, 2nd, and 3rd stages are the first three stages of the encoder, consecutively.

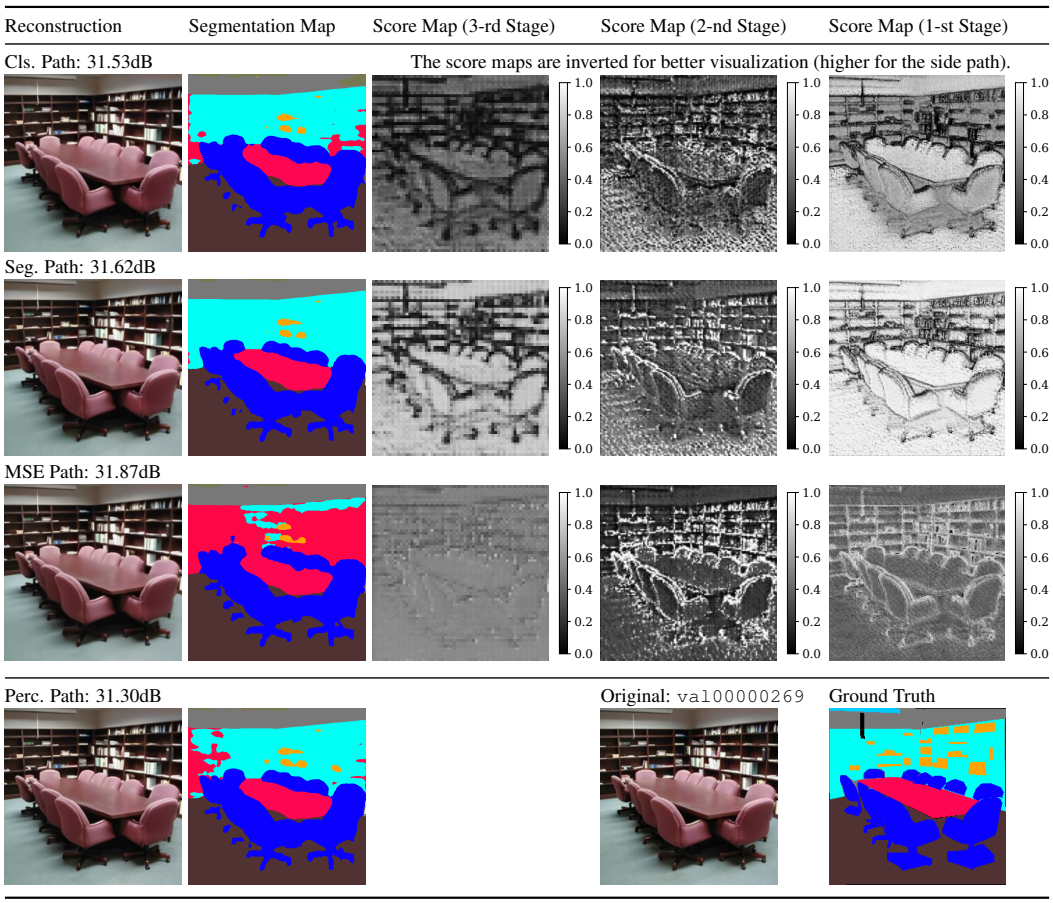


Figure 23: We visualize the reconstructed images, segmentation map and score maps predicted by importance predictors in each path. The image is from ADE20K validation set [83] and resized to 512×512×3. The 3rd, 2nd, and 1st stages are the last three stages of the decoder, consecutively. q is set to 1 for a more distinct comparison. Note that the score maps are inverted for better visualization, i.e., larger scores indicate prioritized for entering the selected side path. The bitrate is 0.0718bpp.

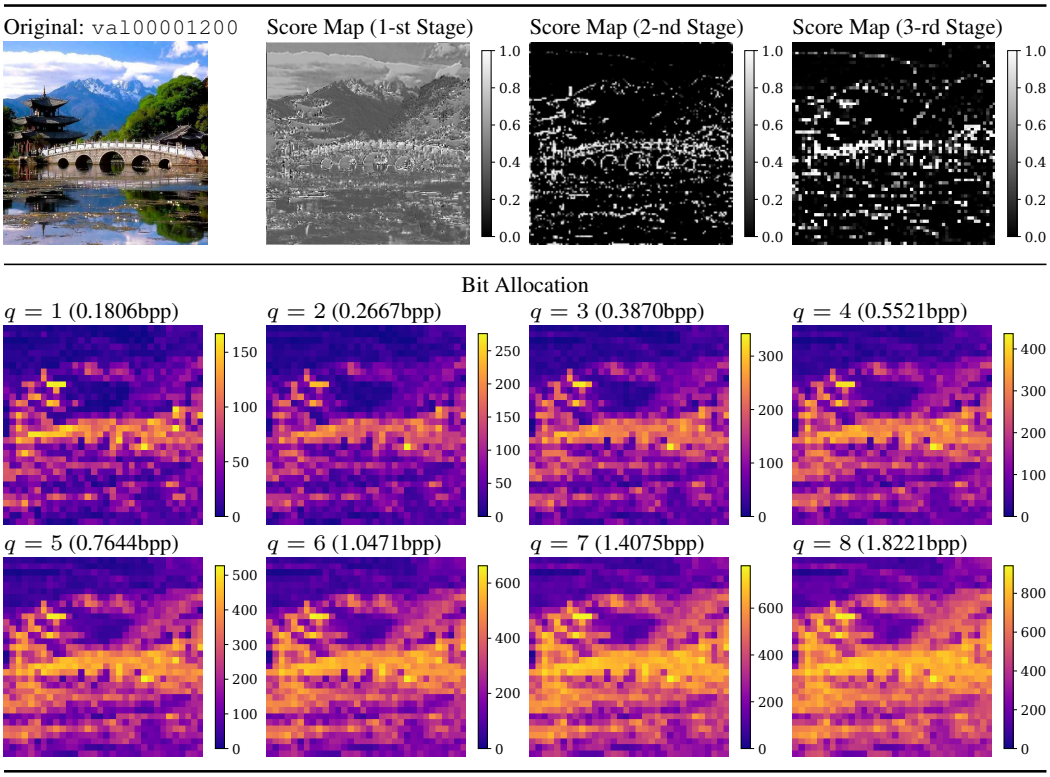


Figure 24: We visualize the bit allocation and score maps in the encoder. The image is from ADE20K validation set [83] and resized to 512×512×3. The 1st, 2nd, and 3rd stages are the first three stages of the encoder, consecutively.

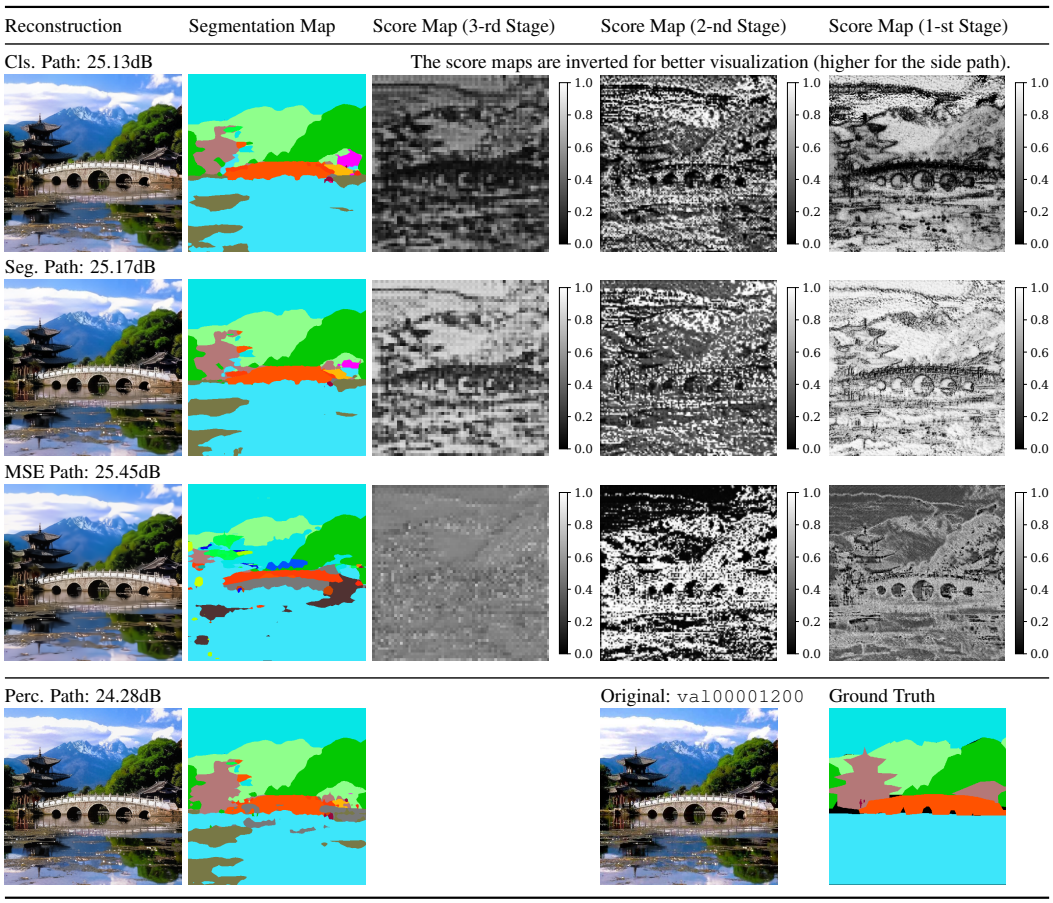


Figure 25: We visualize the reconstructed images, segmentation map and score maps predicted by importance predictors in each path. The image is from ADE20K validation set [83] and resized to 512×512×3. The 3rd, 2nd, and 1st stages are the last three stages of the decoder, consecutively. q is set to 1 for a more distinct comparison. Note that the score maps are inverted for better visualization, i.e., larger scores indicate prioritized for entering the selected side path. The bitrate is 0.1806bpp.

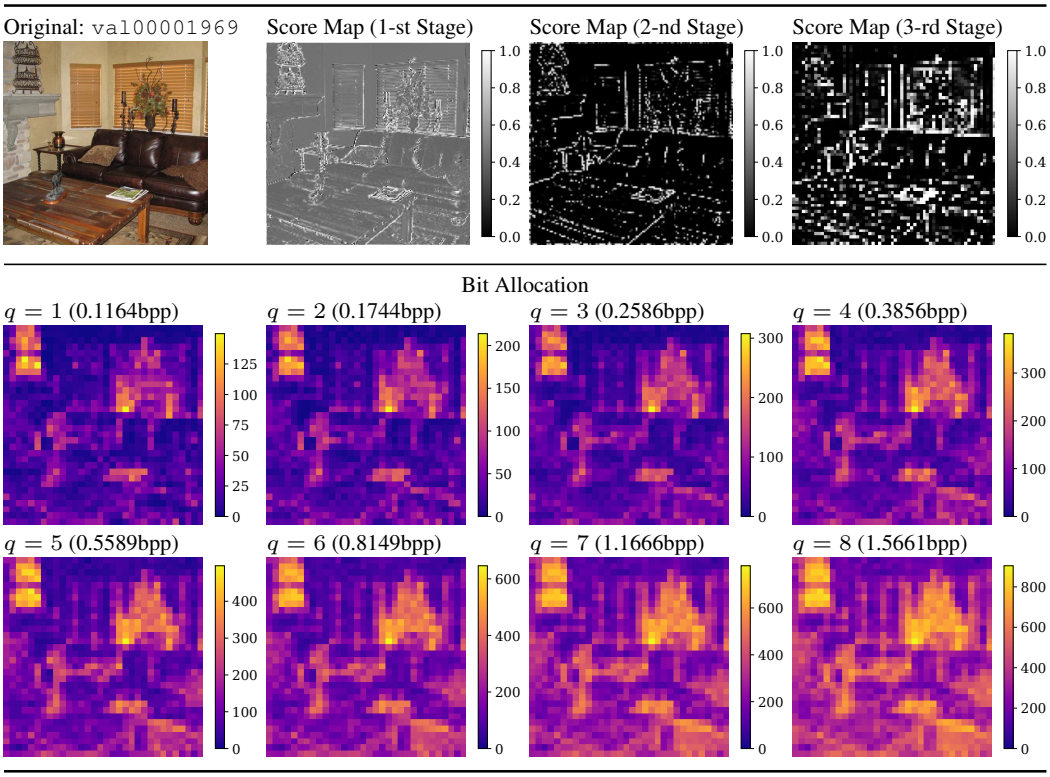


Figure 26: We visualize the bit allocation and score maps in the encoder. The image is from ADE20K validation set [83] and resized to 512×512×3. The 1st, 2nd, and 3rd stages are the first three stages of the encoder, consecutively.

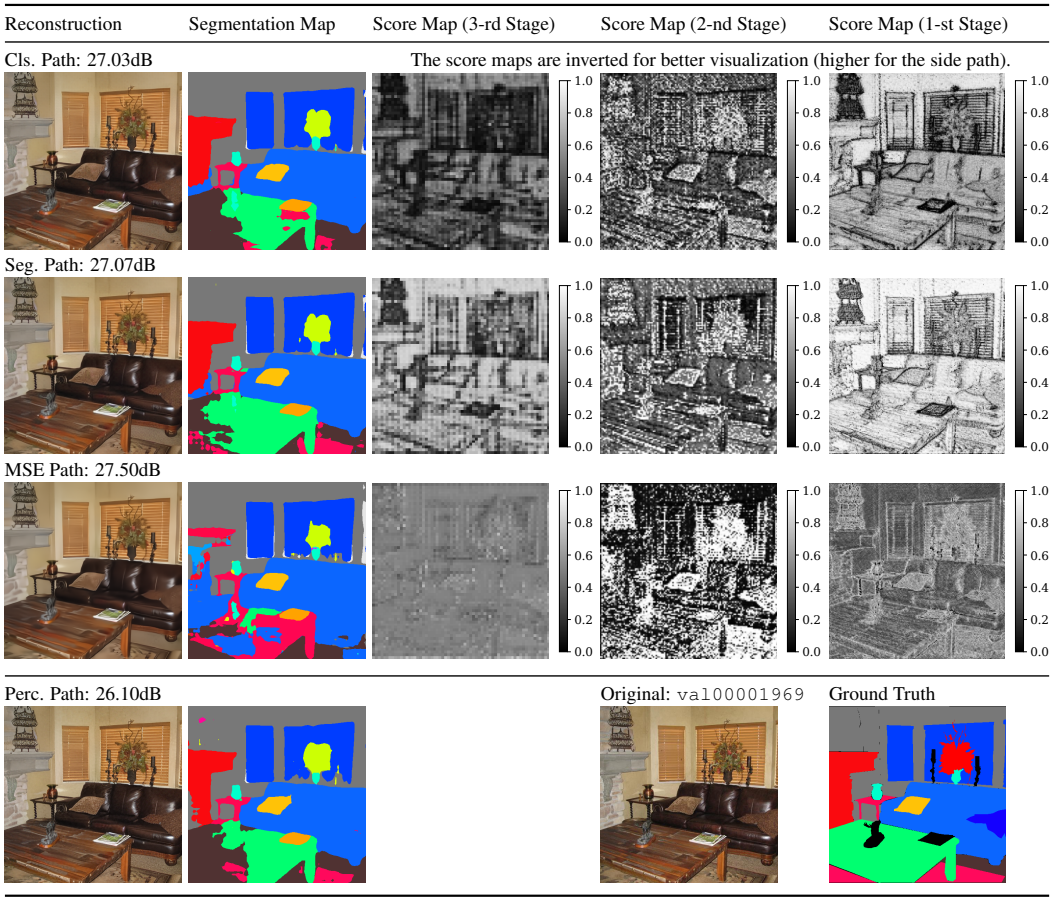


Figure 27: We visualize the reconstructed images, segmentation map and score maps predicted by importance predictors in each path. The image is from ADE20K validation set [83] and resized to 512×512×3. The 3rd, 2nd, and 1st stages are the last three stages of the decoder, consecutively. q is set to 1 for a more distinct comparison. Note that the score maps are inverted for better visualization, i.e., larger scores indicate prioritized for entering the selected side path. The bitrate is 0.1164bpp.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim the contributions and scope in the abstract and Sec. 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Sec. 5.4 and Appx. A.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experimental details are provided in Secs. 3, 4 and 5.1, and in Appx. B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the instructions in Sec. 4. Code will be available upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details in Sec. 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments do not involve statistical significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information on the computer resources in Appx. B.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impacts in Appx. A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The original papers are cited. The licenses are listed in Appx. E.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code is available at `https://github.com/NJUVISION/MPA`.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.