Supplementary Material for Image Compression for Machine and Human Vision With Spatial-Frequency Adaptation

Han Li[1], Shaohui Li[2(✉)], Shuangrui Ding[3], Wenrui Dai[1(✉)], Maida Cao[1], Chenglin Li[1], Junni Zou[1], and Hongkai Xiong[1]

[1] Shanghai Jiao Tong University
[2] Tsinghua Shenzhen International Graduate School, Tsinghua University
[3] The Chinese University of Hong Kong
{qingshi9974,daiwenrui,caomaida,lcl1985,zoujunni,xionghongkai}@sjtu.edu.cn , lishaohui@sz.tsinghua.edu.cn , ds023@ie.cuhk.edu.hk

A Preliminary of Learned Image Compression

Learned image compression model typically consists of two core modules: nonlinear transform and entropy model. The nonlinear transform including analysis transform (*i.e.,* encoder, g_a) and synthesis transform (*i.e.,* decoder, g_s). Given the raw image $\boldsymbol{x}$, the encoder $g_a(\cdot)$ maps it to the latent representation $\boldsymbol{y}$. Then, quantization operator $Q(\cdot)$ discretizes $\boldsymbol{y}$ to $\hat{\boldsymbol{y}}$. Finally, the reconstructed image $\hat{\boldsymbol{x}}$ is obtained by feeding the quantized latent $\hat{\boldsymbol{y}}$ to the synthesis transform $g_s(\cdot)$. This process can be formulated as follows:

$$\boldsymbol{y} = g_a(\boldsymbol{x}; \boldsymbol{\theta_a}), \tag{1}$$

$$\hat{\boldsymbol{y}} = Q(\boldsymbol{y}), \tag{2}$$

$$\hat{\boldsymbol{x}} = g_s(\hat{\boldsymbol{y}}; \boldsymbol{\theta_s}), \tag{3}$$

where $\boldsymbol{\theta_a}$ and $\boldsymbol{\theta_s}$ are the trainable parameters of encoder and decoder, respectively. To encode $\hat{\boldsymbol{y}}$ losslessly, the entropy model with side information $\hat{\boldsymbol{z}}$ is used to model each element of $\hat{\boldsymbol{y}}$ as a Gaussian distribution with the predicted parameters of mean $\boldsymbol{\mu}$ and scale $\boldsymbol{\sigma}$:

$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \tag{4}$$

Then, we can calculate the bitrates by:

$$R = -\log_2 p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) - \log_2 p_{\hat{\boldsymbol{z}}}(\hat{\boldsymbol{z}}). \tag{5}$$

If the decoded image $\hat{\boldsymbol{x}}$ is reconstructed for human vision, we usually measure its visual quality by calculating the mean square error over the raw image. However, this metric is not suitable when the image is reconstructed for machine vision. In this paper, we adopt the perceptual distortion loss to optimize the adpaters for better task performance.

B More Discussion about ICMH

We further discuss the differences and advantages of our approach over existing methods across various frameworks in detail.

Scalable coding framework Earlier work [7,21,22,30] achieves scalable coding for human and machine vision. Choi *et al.* [7] adopts a mutli-task pipeline that needs to jointly train the whole codec (*i.e.,* one encoder and multiple task-specific decoders) from scratch. Despite its superior performance on machine vision tasks, the multi-task pipeline dramatically degrades the R-D performance on human vision (about 1dB degradation in PSNR compared to optimizing only for human vision). In addition, training from scratch also brings huge training overhead and is restricted in adapting to newly coming machine vision tasks. In contrast, our Adapt-ICMH efficiently fine-tune the pre-trained human vision-oriented image codecs on machine vision without compromising the R-D performance of the original codec.

ICMH-Net [22], also based on the pre-trained image codec, selects partial quantized latent as the base bitstream for specific machine vision task. It does not sacrifice the R-D performance since the pre-trained image codec is frozen. However, the machine vision performance is unsatisfactory due to the use of a binary mask generated by Gumbel-Softmax to filter redundant latent, which is suboptimal and difficult to optimize stably. In contrast, our SFMA achieves latent adaptation through spatial-frequency modulation rather than simple spatial selection. Additionally, we have demonstrated the importance of updating shallow latent, while ICMH-Net only adapts the deepest latent.

Although our original framework wasn't specifically tailored for scalable coding, our SFMA can still integrate with scalable coding frameworks (*e.g.,* ICMH-Net) to significantly improve their rate-accuracy performance.

Single bitstream framework. [5,10,13,19] produce single bitstreams for each individual task and our framework also belongs to this category. Feng *et al.* [10] uses the group mask generated by the pre-analysis recognition models (*e.g.,* detection and segmentation models) to implement ROI coding for machine tasks. But it's not practical to deploy so many pre-analysis models at encoder side (usually user side). Liu *et al.* [19] performs channel selection and produces individual bitstreams for each specific machine task, but it also requires task-specific decoder. Chen *et al.* [5] transfers the transformer-based image codec from human vision to machine vision by visual prompt tuning, but it fails to be compatible with CNN-based image codecs, and brings significant additional computational complexity. In contrast, our proposed SFMA is lightweight and compatible with almost all the existing LIC models with the original image codec shared across human and machine vision.

generalized bitstream framework. There are also works [1,11,12] that uses a single bitstreams for multiple tasks. Feng *et al.* [11] learns a compressed om-

nipotent feature for multiple machine vision tasks. Bai *et al.* [1] jointly optimizes the quantized latent for human perception and classification task. However, it's still challenge to trade-off between multiple tasks and find a optimal solution.

C Details of Training Setting

Task-specific Perceptual Distortion Loss. In Eq.(1) of our main paper, we use the task-specific perceptual distortion loss $\mathcal{D}$ to optimize our SFMAs for machine vision tasks, allowing us to train the task-specific module without accessing the task-related label.

Specifically, we follow [5] to use pre-trained ResNet50[4] [15], Faster RCNN[5] [26], and Mask RCNN[6] [14] as the off-the-shelf recognition model for classification, object detection, and instance segmentation, respectively. Fig. 1 shows the network architectures of ResNet50-based Feature Pyramid Network (FPN), which is the backbone network of Faster RCNN [26] and Mask RCNN [14]. In particular, we take the features output by ResNet50 (*i.e.,* F_1, F_2, F_3, and F_4) to evaluate the perceptual loss for classification task:

$$\mathcal{D}(\boldsymbol{x}, \hat{\boldsymbol{x}}, \mathcal{G}) = \frac{1}{4}\sum_{j=1}^{4} \mathrm{MSE}\left(F_j(\boldsymbol{x}), F_j(\hat{\boldsymbol{x}})\right), \tag{6}$$

where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ denotes the raw and reconstructed images, respectively. For object detection and instance segmentation tasks, we take the features output by FPN (*i.e.,* P_2, P_3, P_4, P_5, and P_6) to calculate the perceptual loss:

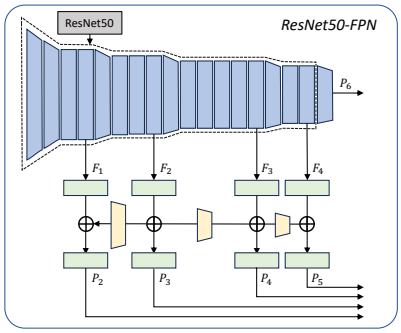


Fig. 1: Network architecture of ResNet50-FPN.

$$\mathcal{D}(\boldsymbol{x}, \hat{\boldsymbol{x}}, \mathcal{G}) = \frac{1}{5}\sum_{j=2}^{6} \mathrm{MSE}\left(P_j(\boldsymbol{x}), P_j(\hat{\boldsymbol{x}})\right). \tag{7}$$

Hyperparamters of Training Tab. 1 details the training hyperparamters of our all experiments for different machine tasks. We use NVIDIA GeForce RTX 4090 and Intel Xeon Platinum 8260 to conduct all our experiments.

[4] `https://download.pytorch.org/models/resnet50-0676ba61.pth`
[5] `https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/faster_rcnn_R_50_FPN_3x/137849458/model_final_280758.pkl`
[6] `https://dl.fbaipublicfiles.com/detectron2/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x/137849600/model_final_f10217.pkl`

Table 1: Training hyperparamters for experiments in the main paper.

	Classification	Detection	Segmentation
Optimizer	Adam	Adam	Adam
Batch size	16	8	8
Trade-off term λ	[1.8, 3.5, 6.7, 13]	[5, 10, 20, 50]	[5, 10, 20, 50]
Epochs	5	40	40
Learning rate schedule	MultiStepLR	-	-
Milestones	[2, 4]	-	-
Learning rate decay	0.5	-	-
Base learning rate	1e-4	1e-4	1e-4

D Details of Reproduction for Other Methods

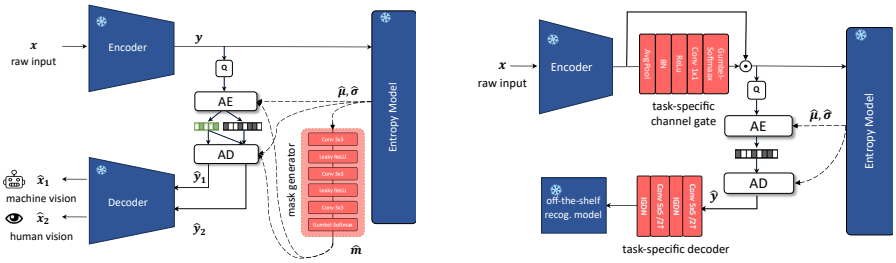


Fig. 2: Architecture of ICMH-Net [22] (left) and channel selection [19] (right).

We have compared our framework with SOTA tuning-based methods [5, 19, 22]. We use the result of TransTIC [5] published in the paper. However, [22] and [19] used different pre-trained recognition models from us[7], we cannot directly compare our framework with theirs. Since [22] and [19] are not open source, we reproduce their results by using the same training, evaluation settings, and dataset as ours. Fig. 2 shows the their sketch of architecture, we set the temperature parameter of hard version of Gumbel-Softmax to 1 for [19, 22].

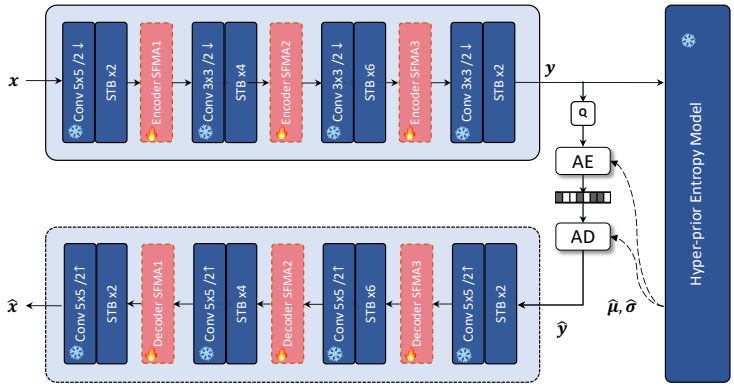
E Details of Architecture for Different Base Codecs

We illustrate the detailed network architecture for our Adapt-ICMH framework incorporating different image codec methods in Fig. 3. Note that these image codecs have different nonlinear transforms and entropy models, but our SFMA consistently shows the ability to achieve superior rate-accuracy performance for machine vision tasks, and do not affect the visual quality for human vision of the base codec. We also include the source of the pre-trained checkpoint of base codecs in the Tab. 2.

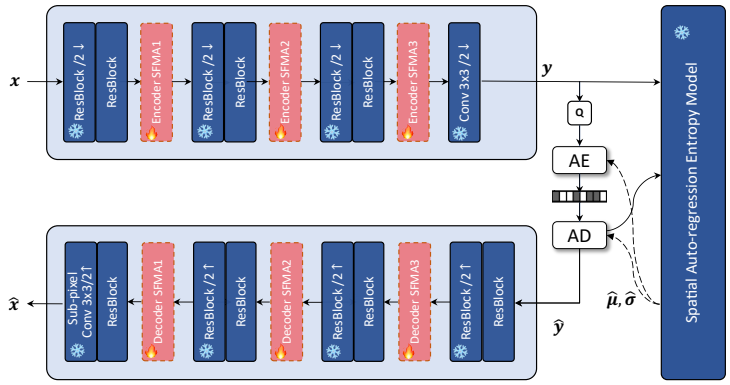
F Details of Scalable Coding Pipeline

We present the framework of scalable coding for machine and human vision in Fig. 4. Specifically, we only inject the SFMAs in the decoder of base codec and adopt the mask generator [22] to select the latent to be transmitted for

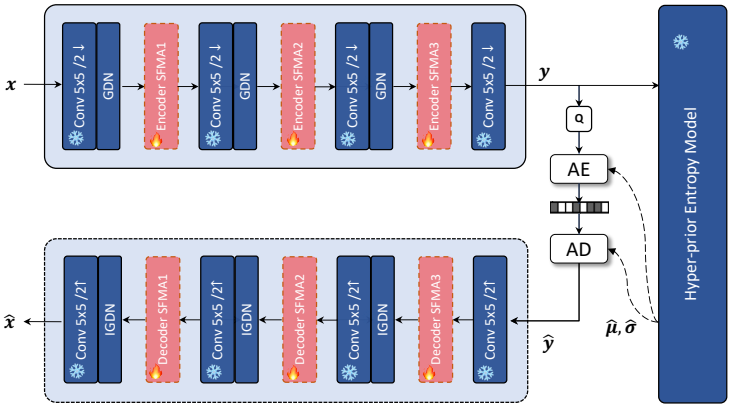
[7] [19] used the Deep Lab V3 for segmentation task. Although [22] claimed that they used ResNet50 for classification and Faster RCNN for detection, they didn't disclose which pre-trained checkpoints they utilized.



(a) *Lu2022-TIC* model [24] as base codec. STB denotes the Swin-Transformer Block [23]. Following [5], we adopt the simplified version of [24] for fair comparison.



(b) *Cheng2020-anchor* model [6] as base codec. ResBlock denotes the residual block, \2 ↓ denotes a stride on the first convolution, and \2 ↑ denotes a sub-pixel upsampling on the last convolution.



(c) *mbt2018-mean* model [25] as base codec. GDN denotes the Generalized Divisive Normalization layer [2].

Fig. 3: Details of network architecture for different base codecs.

Table 2: Source path of the implementations and pre-trained weights of base codecs

Model	Source Path
Lu2022-TIC	https://github.com/NYCU-MAPL/TransTIC/tree/master
Cheng2020-anchor	https://github.com/InterDigitalInc/CompressAI/tree/master
mbt2018-mean	https://github.com/InterDigitalInc/CompressAI/tree/master

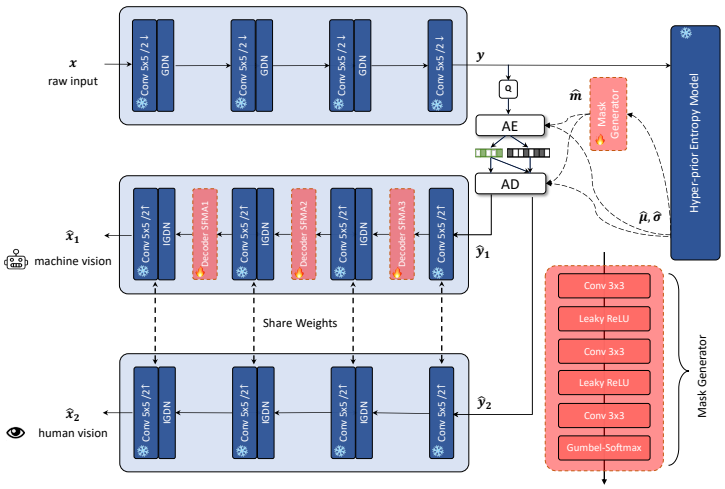


Fig. 4: Our scalable coding framework. The mask generator is proposed by [22] and we adopt pre-trained *mbt2018-mean* as base codec.

machine vision. Specifically, the binary spatial-channel mask $\hat{\boldsymbol{m}}$ is derived by inputting the entropy parameters $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\sigma}}$ into the mask generator. Thus, we obtain the masked quantized latent $\hat{\boldsymbol{y}}_1$ (*i.e.*, $\hat{\boldsymbol{y}}_1 = \hat{\boldsymbol{m}} \cdot \mathrm{Q}(\hat{\boldsymbol{y}})$) which is encoded as the base layer, while the remaining latent is encoded as the enhanced layer. On the decoder side, the base layer $\hat{\boldsymbol{y}}_1$ is decoded by the decoder with SFMAs to obtain the reconstructed image $\hat{\boldsymbol{x}}_1$ for machine vision, while the full latent $\hat{\boldsymbol{y}}_2$ is decoded by the decoder without SFMAs to obtain the reconstructed image $\hat{\boldsymbol{x}}_2$ for human vision.

We provide the additional rate-accuracy results on classification and instance segmentation tasks in Fig. 5. It demonstrates that our proposed SFMA can significantly benefit existing scalable coding framework, *i.e.,* ICMH-Net [22].

G Analysis on the Computational Complexity

We compare the computational complexity of our framework with others. Tab. 3 shows that our framework only introduces small computational complexity and the increase on the latency time can be ignored. Although full fine-tuning can achieve a satisfactory rate-accuracy performance without an increased computational complexity, it requires to store and deploy a copy of the entire codec.

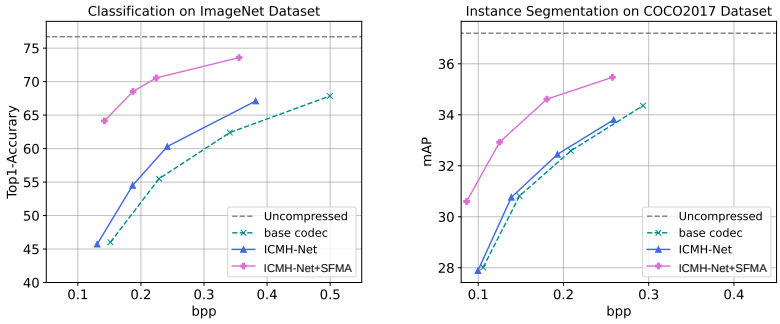


Fig. 5: Rate-accuracy results of classification (left) and instance segmentation (right).

Channel selection [19] can reduce the complexity at the decoder side, since it uses a lightweight task-specific decoder (shown in Fig. 2) for each machine vision task. However, it cannot take advantage of the powerful knowledge of the pre-trained synthesis transform, resulting in degraded rate-accuracy performance in the complex vision tasks (*e.g.,* detection and segmentation) compared to the base codec.

Table 3: Comparison on computational complexity evaluated on the ImageNet-val [8] dataset for classification task. We do not include the computation on the off-the-shelf recognition model. Ours-n indicates n middle dimensions. The BD-acc is presented for rate-accuracy performance comparison with the base codec of *Lu2022-TIC* [24] as the anchor.

Model	**KMACs/pixel**		**Latency (ms)**		**#Trainable Params(M)**	**BD-acc ↑**
	Enc.	**Dec.**	**Enc.**	**Dec.**		
base codec	142.5	188.5	120.1	35.8	-	-
full fine-tuning	142.5	188.5	120.1	35.8	7.51	**17.6%**
ICMH-Net [22]	159.1	205.1	126.7	43.3	3.98	3.3%
channel selection [19]	142.6	25.1	121.1	10.1	0.91	6.2%
TransTIC [5]	332.0	202.6	146.2	40.2	1.61	9.9%
Ours-32	149.7	195.7	121.7	37.6	0.14	16.4%
Ours-64	157.2	203.2	123.2	40.1	0.28	16.9%
Ours-128	173.4	219.4	124.1	40.9	0.62	**17.6%**

H Application on Larger Transformer-based Image Codecs

We further demonstrate the effectiveness of our framework on larger scale transformer-based image codecs, including *STF* [34], *TCM* [20], and *FAT* [16]. We conduct performance comparison with other ICMH frameworks [5, 19], it is noted that *ICMH-Net* [22] cannot support the channel-wise autoregression entropy model used in [16,20,34]. Specifically, we train each methods for one rate-accuracy point and report the classification results in Tab. 4. We observe that ours method still outperform other methods with seldom trainable parameters (less than 1% of the base codec).

Table 4: Classification comparison on ImageNet-*val* dataset using more large-scale transformer-based image codecs, including *STF* [34], *TCM* [20], and *FLIC* [16]. Acc. denotes the top-1 accuracy.

Method	**Classification**		**Trainable**
	bpp↓	Acc. ↑	Params ↓(M)
STF [34]			
full fine-tuning	**0.2559**	**75.98**	99.85(100%)
TransTIC [5]	0.4910	74.51	1.15(1.2%)
channel selection [19]	0.5217	73.21	2.49(2.5%)
Ours	<u>0.3418</u>	<u>75.18</u>	0.26(0.3%)
TCM [20]			
full fine-tuning	**0.2494**	<u>75.83</u>	76.56(100%)
TransTIC [5]	0.4252	75.60	1.30(1.7%)
channel selection [19]	0.4703	73.37	2.73(3.6%)
Ours	<u>0.3273</u>	**75.87**	0.53(0.7%)
FLIC [16]			
full fine-tuning	**0.2403**	**75.93**	70.97(100%)
TransTIC [5]	0.3775	75.65	1.22(1.7%)
channel selection [19]	0.4067	73.46	2.73(3.8%)
Ours	<u>0.3274</u>	<u>75.72</u>	0.36(0.5%)

I More Ablation Studies

I.1 Plugging SFMAs into Entropy Model

Our proposed SFMA is designed to fine-tune the nonlinear transform of the base codec for machine vision task. In this section, we also plug SFMAs into the entropy model to further explore its effectiveness. Specifically, SFMAs are plugged into the intermediate layer of hyper encoder h_a and hyper decoder h_s, which is similar to the process of SFMA for g_a and g_s. From Tab. 5 we observe that further plugging SFMAs into the entropy model cannot bring significant

performance improvement, but introduces more model complexity. Thus, we decide to only plug SFMAs into the nonlinear transform. This also demonstrates that it's the nonlinear transform rather than the entropy model that is the key difference between human and machine vision-oriented image compression.

Table 5: Ablations on plugging SFMAs into entropy model

g_a, g_s	h_a, h_s	**Classification**		**Detection**		**Segmentation**		Params
		BD-rate↓	BD-acc↑	BD-rate↓	BD-mAP↑	BD-rate↓	BD-mAP↑	(M)
✓		-82.00%	18.71	-56.17%	3.84	-52.65%	3.17	0.28
	✓	-1.56%	0.27	-1.32%	0.07	-1.11%	0.09	0.26
✓	✓	-81.27%	18.80	-56.94%	3.91	-53.21%	3.19	0.55

I.2 Compared with Naive Adapter

We also replace our SFMA with the naive adapter in [4] to perform feature adaptation. The naive adapter consists of two linear layers with a ReLU activation layer, which is limited in achieving spatial and frequency modulation like our SFMA. Tab. 6 shows that the naive adapter is inferior to our proposed SFA and FMA under similar trainable parameters, demonstrating the effectiveness of our SFMA.

Table 6: Ablations on using naive adapter. Naive-n denotes that the naive adapter with the middle dimension set to n.

Method	**Classification**		**Detection**		**Segmentation**		Params
	BD-rate↓	BD-acc↑	BD-rate↓	BD-mAP↑	BD-rate↓	BD-mAP↑	(M)
SFMA	-82.00%	18.71	-56.17%	3.84	-52.65%	3.17	0.28
SMA-only	-73.82%	16.30	-51.94%	3.61	-48.16%	2.92	0.16
FMA-only	-77.40%	17.29	-52.86%	3.32	-49.58%	2.90	0.12
Naive-64	-69.72%	15.61	-46.83%	2.82	-42.34%	2.35	0.11
Naive-96	-71.23%	15.92	-47.92%	2.94	-44.17%	2.47	0.16

J More Qualitative Results

We provide additional qualitative results for detection and segmentation tasks. These qualitative results further demonstrate the superiority of our framework, which can effectively reduce the latent redundancies for machine vision, thus achieving better task performance with lower bitrates.

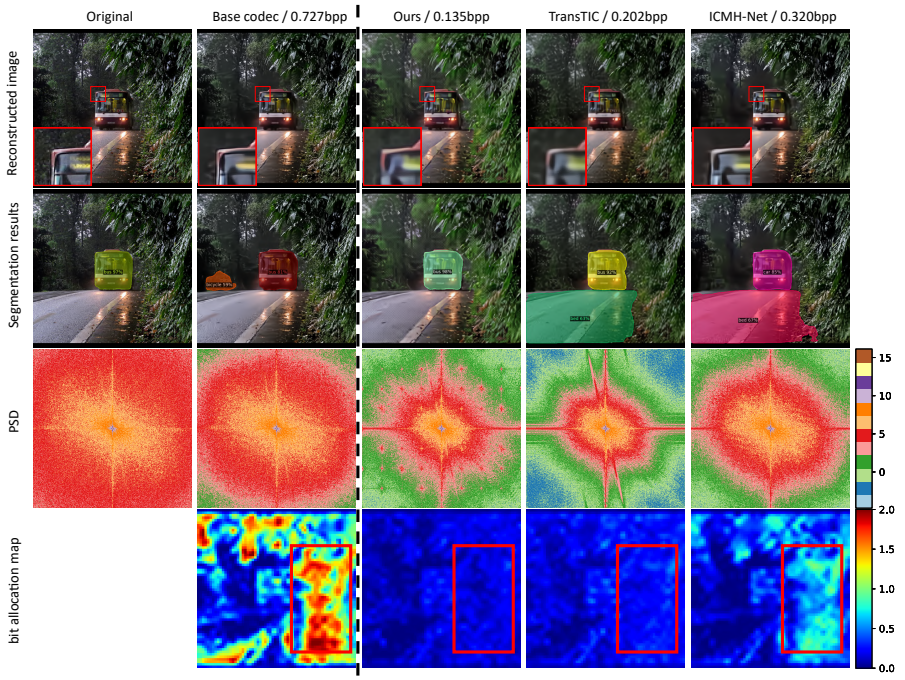


Fig. 6: Qualitative comparison of our Adapt-ICMH with other ICMH methods. **First row:** The original image and decoded image of each method. We show the decoded images for machine vision of three ICMH methods (left). **Second row:** The object detection results of each image. **Third row:** The log power spectral density maps of each image. **Bottom row:** The bit allocation maps for $\hat{\boldsymbol{y}}$ of each method.

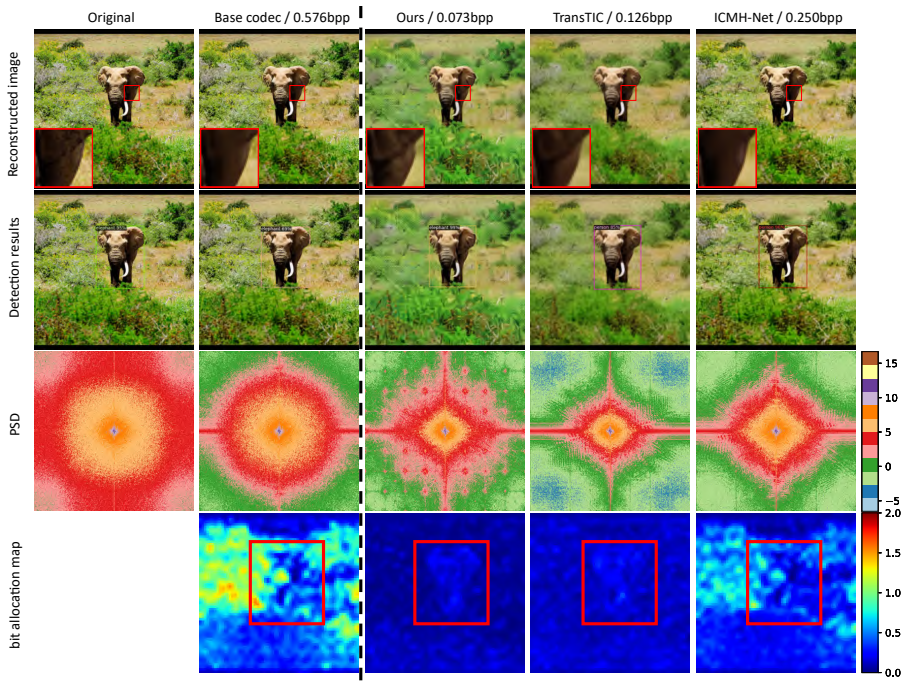


Fig. 7: Qualitative comparison of our Adapt-ICMH with other ICMH methods. **First row:** The original image and decoded image of each method. We show the decoded images for machine vision of three ICMH methods (left). **Second row:** The instance segmentation results of each image. **Third row:** The log power spectral density maps of each image. **Bottom row:** The bit allocation maps for $\hat{\boldsymbol{y}}$ of each method.

K Pytorch Implementation of SFMA

```
class SFMA(nn.Module):
    def __init__(self, in_dim=128, middle_dim=64,factor=1):
        super().__init__()
        self.s_down1 = nn.Conv2d(in_dim, middle_dim, 1, 1, 0)
        self.s_down2 = nn.Conv2d(in_dim, middle_dim, 1, 1, 0)
        self.s_dw = nn.Conv2d(middle_dim, middle_dim, 5, 1,
    2, groups=middle_dim)
        self.s_relu = nn.ReLU(inplace=True)
        self.s_up = nn.Conv2d(middle_dim, in_dim, 1, 1, 0)

        self.f_down = nn.Conv2d(in_dim, middle_dim, 1, 1, 0)
        self.f_relu1 = nn.ReLU(inplace=True)
        self.f_relu2 = nn.ReLU(inplace=True)
        self.f_up = nn.Conv2d(middle_dim, in_dim, 1, 1, 0)
        self.f_dw = nn.Conv2d(middle_dim, middle_dim, 3, 1,
    1, groups=middle_dim)
        self.f_inter = nn.Conv2d(middle_dim, middle_dim, 1,
    1, 0)
        self.sg = nn.Sigmoid()

    def forward(self, x):
        '''
        input:
        x: intermediate feature
        output:
        x_tilde: adapted feature
        '''
        _, _, H, W = x.shape

        y = torch.fft.rfft2(self.f_down(x), dim=(2, 3), norm=
    'backward')
        y_amp = torch.abs(y)
        y_phs = torch.angle(y)
        y_amp_modulation = self.f_inter(self.f_relu1(self.
    f_dw(y_amp)))
        y_amp = y_amp * self.sg(y_amp_modulation)
        y_real = y_amp * torch.cos(y_phs)
        y_img = y_amp * torch.sin(y_phs)
        y = torch.complex(y_real, y_img)
        y = torch.fft.irfft2(y, s=(H, W), norm='backward')

        f_modulate = self.f_up(self.f_relu2(y))
        s_modulate = self.s_up(self.s_relu(self.s_dw(self.
    s_down1(x)) * self.s_down2(x)))
        x_tilde = x + (s_modulate + f_modulate)*factor
        return x_tilde
```

Listing 1.1: Pytorch implementation of SFMA

L Limitation and Future Work

A potential limitation of our Adapt-ICMH is that it cannot directly achieve scalable coding for machine and human vision. However, our proposed SFMA can incorporate existing scalable coding ICMH frameworks [22] and boost their performance, as demonstrated in our paper. Further, we will extend our method into more machine vision tasks, such as pose estimation [3, 17, 18, 28, 33], person re-identification [27,29,32]. Additionally, while we only focus on image compression in this paper, video coding for machines (VCM) [9,31] is also a current topic of interest. In our future work, we aim to expand the scope of SFMA to encompass VCM to further demonstrate the superiority of our propose framework.

References

1. Bai, Y., Yang, X., Liu, X., Jiang, J., Wang, Y., Ji, X., Gao, W.: Towards end-to-end image compression and analysis with transformers. In: AAAI. vol. 36, pp. 104–112 (2022)
2. Ballé, J., Laparra, V., Simoncelli, E.P.: Density modeling of images using a generalized normalization transformation. In: ICLR (2016)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7291–7299 (2017)
4. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. In: NeurIPS. vol. 35, pp. 16664–16678 (2022)
5. Chen, Y.H., Weng, Y.C., Kao, C.H., Chien, C., Chiu, W.C., Peng, W.H.: Transtic: Transferring transformer-based image compression from human perception to machine perception. In: ICCV. pp. 23297–23307 (2023)
6. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: CVPR. pp. 7939–7948 (2020)
7. Choi, H., Bajić, I.V.: Scalable image coding for humans and machines. IEEE TIP **31**, 2739–2754 (2022)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
9. Duan, L., Liu, J., Yang, W., Huang, T., Gao, W.: Video coding for machines: A paradigm of collaborative compression and intelligent analytics. IEEE Transactions on Image Processing **29**, 8680–8695 (2020)
10. Feng, R., Gao, Y., Jin, X., Feng, R., Chen, Z.: Semantically structured image compression via irregular group-based decoupling. In: ICCV (2023)
11. Feng, R., Jin, X., Guo, Z., Feng, R., Gao, Y., He, T., Zhang, Z., Sun, S., Chen, Z.: Image coding for machines with omnipotent feature learning. In: ECCV. pp. 510–528. Springer (2022)
12. Feng, R., Liu, J., Jin, X., Pan, X., Sun, H., Chen, Z.: Prompt-icm: A unified framework towards image coding for machines with task-driven prompts. arXiv preprint arXiv:2305.02578 (2023)
13. Fischer, K., Brand, F., Kaup, A.: Boosting neural image compression for machines using latent space masking. IEEE TCSVT (2022)

14. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
16. Li, H., Li, S., Dai, W., Li, C., Zou, J., Xiong, H.: Frequency-aware transformer for learned image compression. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=HKGQDDTuvZ
17. Li, H., Shi, B., Dai, W., Chen, Y., Wang, B., Sun, Y., Guo, M., Li, C., Zou, J., Xiong, H.: Hierarchical graph networks for 3d human pose estimation. arXiv preprint arXiv:2111.11927 (2021)
18. Li, H., Shi, B., Dai, W., Zheng, H., Wang, B., Sun, Y., Guo, M., Li, C., Zou, J., Xiong, H.: Pose-oriented transformer with uncertainty-guided refinement for 2d-to-3d human pose estimation. In: AAAI. vol. 37, pp. 1296–1304 (2023)
19. Liu, J., Sun, H., Katto, J.: Improving multiple machine vision tasks in the compressed domain. In: ICPR. pp. 331–337. IEEE (2022)
20. Liu, J., Sun, H., Katto, J.: Learned image compression with mixed transformer-cnn architectures. In: CVPR. pp. 14388–14397 (2023)
21. Liu, K., Liu, D., Li, L., Yan, N., Li, H.: Semantics-to-signal scalable image compression with learned revertible representations. IJCV **129**(9), 2605–2621 (2021)
22. Liu, L., Hu, Z., Chen, Z., Xu, D.: Icmh-net: Neural image compression towards both machine vision and human vision. In: ACM MM. pp. 8047–8056 (2023)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
24. Lu, M., Guo, P., Shi, H., Cao, C., Ma, Z.: Transformer-based image compression. In: DCC. pp. 469–469 (2022)
25. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. In: NeurIPS. vol. 31 (2018)
26. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS. vol. 28 (2015)
27. Somers, V., De Vleeschouwer, C., Alahi, A.: Body part-based representation learning for occluded person re-identification. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 1613–1623 (2023)
28. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR. pp. 5693–5703 (2019)
29. Yan, S., Dong, N., Zhang, L., Tang, J.: Clip-driven fine-grained text-image person re-identification. IEEE Transactions on Image Processing (2023)
30. Yang, S., Hu, Y., Yang, W., Duan, L.Y., Liu, J.: Towards coding for human and machine vision: Scalable face image coding. IEEE TMM **23**, 2957–2971 (2021)
31. Yang, W., Huang, H., Hu, Y., Duan, L.Y., Liu, J.: Video coding for machines: Compact visual representation compression for intelligent collaborative analytics. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
32. Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., Hoi, S.C.: Deep learning for person re-identification: A survey and outlook. IEEE transactions on pattern analysis and machine intelligence **44**(6), 2872–2893 (2021)
33. Zheng, H., Li, H., Shi, B., Dai, W., Wang, B., Sun, Y., Guo, M., Xiong, H.: Actionprompt: Action-guided 3d human pose estimation with text and pose prompting. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). pp. 2657–2662. IEEE (2023)
34. Zou, R., Song, C., Zhang, Z.: The devil is in the details: Window-based attention for image compression. In: CVPR. pp. 17492–17501 (2022)