# Supplementary: Image Coding for Machines with Omnipotent Feature Learning

## 1 Comparison with SOTA ICM-related methods

First, we must emphasize that our method focuses on a new ICM paradigm of "one bitstream covers multiple different tasks", and we are also the first to report results on such wide range of intelligent tasks and widely accepted datasets. To our knowledge, there is currently no similar work has studied such general problem as we did. Recent ICM-related methods, *e.g.*, the traditional codec based RoI bit location scheme [9] and the learning based joint training codec [12], mostly only focus on specific AI tasks and report the corresponding results, which makes it hard to directly compare these methods with ours and guarantees fairness. Besides, most of these methods didn't release codes, which further makes the fair comparison become more difficult. Despite this, we still reproduced two SOTA ICM-related competitors [9,12] following their papers. The RoI based [9] is optimized and evaluated for every task. The end-to-end joint training based [12] is trained with object detetion on PASCAL VOC dataset and evaluated on all tasks. Table 1 shows the comparison, our method significantly outperforms them by a large margin (the lower the better).

**Table 1.** Comparison with two SOTA ICM methods: RoI based bit allocation (RoI) [9] and task-driven joint training (Joint) [12]. Three AI tasks of detection (Det.), instance segmentation (Ins.), and semantic segmentation (Sem.) are used for evaluation. Bjøntegaard Delta rate (BD-rate) saving w.r.t the AI task performances is taken as metric (more lower, more better). HEVC [17], VVC [3], and a SOTA learned based codec [4] (noted as cheng) are taken as anchors. HEVC is taken as the benchmark. Note that, the results with "*" are converted from the original paper. The best performance of each task is marked in bold.

| Datasets | HEVC+RoI [9] | VVC | VVC+RoI [9] | cheng | cheng+Joint [12] | Ours |
|---|---|---|---|---|---|---|
| Det. (VOC) | -17.6 | -9.0 | -32.9* | -0.9 | -14.4 | **-35.1** |
| Det. (COCO) | -17.2 | -14.4 | -32.4 | -10.7 | -3.0 | **-43.9** |
| Ins. (COCO) | -11.9 | -14.1 | -39.1* | -12.8 | -5.4 | **-42.8** |
| Sem. (City.) | 4.3 | -24.6 | -25.4 | -0.8 | 2.3 | **-72.0** |

## 2 Discussion about Image Coding for Machines (ICM)

In this section, we describe in detail how the approach we take to tackle the problem of image coding for machines (ICM) differs from those of related tasks.

### 2.1 Relationship to Image Coding for Human Perception

The initial purpose of lossy image compression [17,3,4] is to ensure the fidelity of the reconstructed image as much as possible. Such fidelity are often measured by objective metrics such as PSNR and MS-SSIM [7,20]. A reconstructed image with a small distortion is supposed to have a good viewing effect.

Except the traditional objective metrics, the human eye perception can be well indicated/reflected by the perceptual quality, which is related to the realism of the picture. For example, HiFiC [13] combines the learning based compression and GAN techniques [6] to get a lossy image compression algorithm with high visual perceptual quality, although the fidelity of compressed image is not very high. Moreover, Blau *et al.* [2] have demonstrated that there exists a trade-off of distortion and perception. Thus, balancing the trade-off of rate, distortion and perception is the goal of lossy compression for humans. In contrast, the case of image coding for machines can be regarded as balancing a trade-off of rate and intelligent tasks. However, since there exists lots of downstream tasks and even unknown ones, it is difficult to optimize them uniformly. Therefore, in this paper, we choose a generalized representation learning method, *i.e.*, Omni-ICM, to make the learned representation not biased to any task, and general enough for supporting different intelligent tasks.

### 2.2 Relationship to Information Bottleneck

Our solution for ICM that aims to learn the omnipotent feature can also be viewed as a particular instantiation of the more general information bottleneck framework [18,1]. Here we learn the omnipotent representation by maximize the mutual information between our representation and the target of instance discrimination, and meanwhile constrain the mutual information between our representation and the original data. This procedure can be formulated as:

$$\min_{\boldsymbol{\theta}} I(Z, Y; \boldsymbol{\theta}) \text{ s.t. } I(X, Z; \boldsymbol{\theta}) \leq I_c, \tag{1}$$

where $X$ indicates the original data, $Z$ indicates the latent representation, $Y$ indicates the optimization target and $\boldsymbol{\theta}$ indicates the functions parameterized by $\boldsymbol{\theta}$. And equivalently, with the introduction of a Lagrange multiplier $\beta$ to control the trade-off, it can be formulated as maximize the objective function:

$$R_{IB}(\boldsymbol{\theta}) = I(Z, Y; \boldsymbol{\theta}) - \beta I(Z, X; \boldsymbol{\theta}). \tag{2}$$

But differently, our work pays more attentions on how to achieve a good trade-off between compression efficiency and AI tasks generalization, which is not only a naive application or extension of information bottleneck.

### 2.3 Relationship to Self-supervised Learning

Methods in self-supervised learning (SSL) [10,21,15,14] are proposed to learn general representations for downstream tasks by solving various pretext tasks

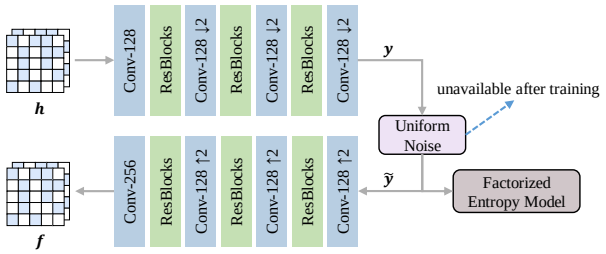


**Fig. 1.** The architecture of information filtering (IF) module. Each "Resblocks" in the figure is stacked by three ResBlocks. And a ResBlock consists of two conlutional layers (with $3 \times 3$ kernel size, 128 input channels and 128 output channels) which involved by a single shortcut.

on large-scale unlabeled datasets. There are mainly two differences between SSL and our method here. One is that the self-supervised learning targets at good initialization weights through pre-training. In subsequent task migration, the entire network is often fine-tuned according to downstream tasks. Our method targets at a general representation learning. Once the training is over, the original image is no longer visible to the machines, and replaced by representation extracted from the original image. Thus, the network weights of extracting this representation (backbone head as described in the Section 3.2 of main text) are not allowed to be updated. The second point is that SSL has no explicit constrains about entropy but we did, for the reason that we focus on both of the representation ability and the information quantity. In a word, we need to balance the trade-off between generalization and the amount of information of the representation.

### 2.4 Relationship to Meta-learning

Meta-learning [19,8,16,5], also called as learning-to-learn, provides an alternative paradigm where a machine learning model gains experience over multiple learning episodes - often covering a distribution of related tasks. The experience of mentioned procedure would help improve the future learning performance. Similar to SSL mentioned in Section 2.3, the pre-trained model are often fine-tuned for downstream tasks with all parameters updated. In addition, meta-learning also does not explicitly need a representation with low entropy thus easy to compressing.

## 3 Architecture Details

**Information Filtering (IF) Module.** The detailed architecture of the information filtering (IF) module illustrate in Figure 1. Extra residual blocks are used to increase receptive filed and improve non-linear transformation capability [4]. As for the size of IF module, compared with baseline (ResNet-50) with 25.56M

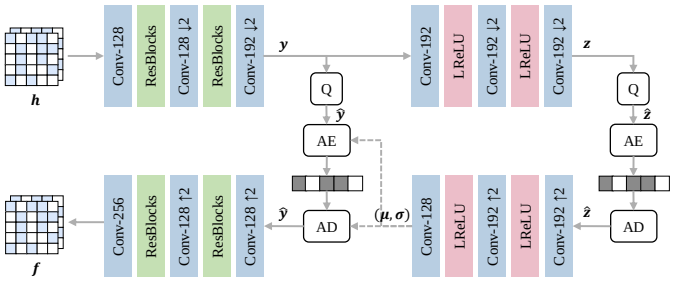


**Fig. 2.** The architecture of the learning-based feature compressor. Each "Resblocks" in the figure is stacked by three ResBlocks. And a ResBlock consists of two conlutional layers (with $3 \times 3$ kernel size, 128 input channels and 128 output channels) which involved by a single shortcut. LReLU indicates LeakyReLU.

parameters, our method just adds an additional information filter (IF) module, with only 8.24M parameters increased.

**Feature Compression Codec.** The detailed architecture of the learning-based feature compression codec is illustrated in Figure 2. Note that, here the several residual blocks are used to increase receptive filed and improve the entire rate-distortion performance.

## 4 More Experimental Results

More experimental results are illustrated in Fig. 3, 4, 5, 6. We additionally compare our method with the competitors of supervised fine-tuning. For this case, we train task models with the ImageNet pre-trained weights with supervised training as initialization and evaluate on them. As shown in these figures, the performances of baselines that fine-tuning on contrastive learning pre-trained models are better than those of fine-tuning on supervised learning pre-trained models. And the baselines of our Omni-ICM have a drop compared with fully contrastive pre-training. The degrees of decline vary according to the datasets and tasks. As for the case of coding for intelligent tasks (the curve part of the paradigms), results on task models fine-tuning on contrastive pre-training are better than those fine-tuning on supervised pre-training, and our methods performs better than both of them.

## 5 Details about Feature Reconstruction

**Decoder Architecture.** The architecture of the decoders (mentioned in Section 4.6 of the text) for reconstruction from features are stacked by convolutional layers and ResBlocks, which is illustrated in Fig. 7. Residual blocks are also used to increase receptive filed and improve non-linear transformation capability. These two decoders share the same architecture and training schedule. We train them

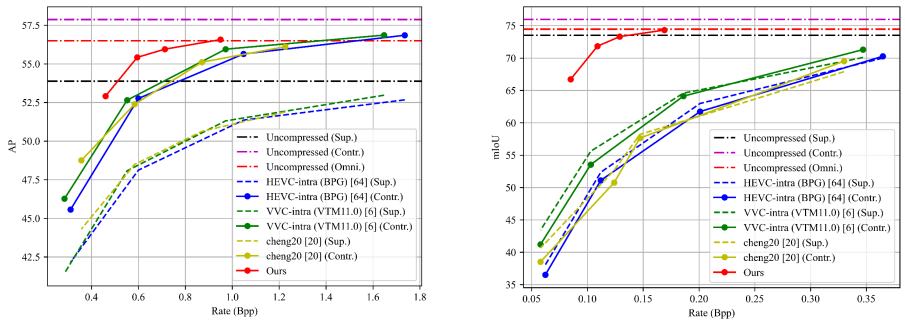


**Fig. 3. Object detection on PASCAL VOC (left) and semantic segmentation on Cityscapes (right).** Dotted lines indicate the results of uncompressed data as input. Dashed lines indicate the results of fine-tuning with ImageNet supervised pre-training weights.

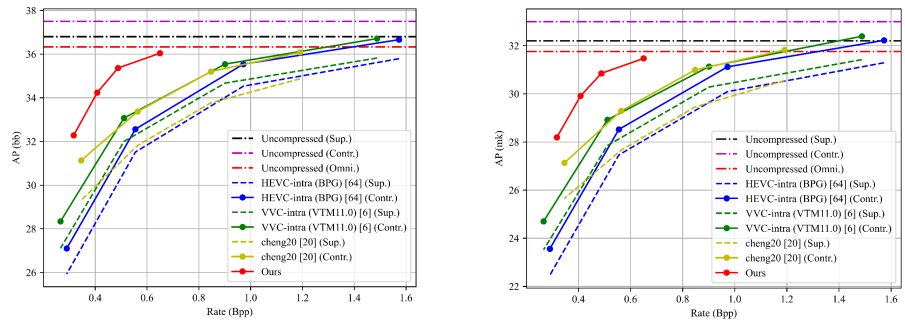


**Fig. 4. Object detection and instance segmentation on MS coco.** Dotted lines indicate the results of uncompressed data as input. Dashed lines indicate the results of fine-tuning with ImageNet supervised pre-training weights.

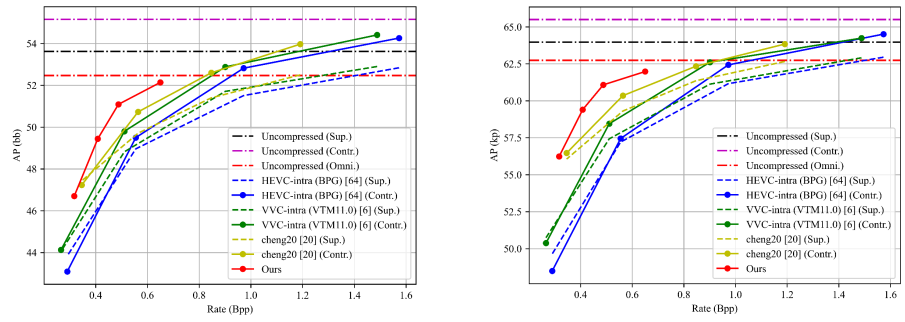


**Fig. 5. Pose estimation on MS COCO.** Dotted lines indicate the results of uncompressed data as input. Dashed lines indicate the results of fine-tuning with ImageNet supervised pre-training weights.

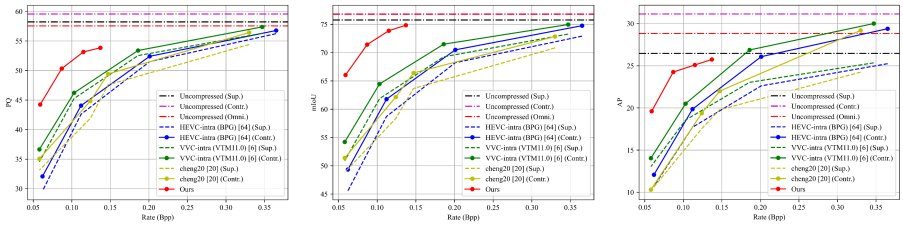


**Fig. 6. Panoptic segmentation on Cityscapes.** Dotted lines indicate the results of uncompressed data as input. Dashed lines indicate the results of fine-tuning with ImageNet supervised pre-training weights.

for $200,000$ iterations with batch size of 16. Adam optimizer[11] is employed and the learning rate is set as $5 \times 10^{-5}$. Data augmentation is $256 \times 256$ random cropping.

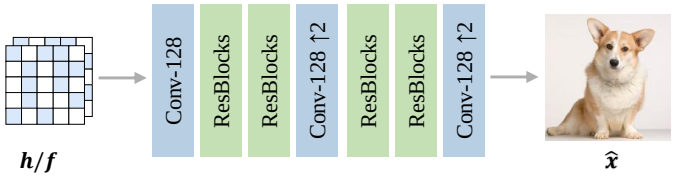


**Fig. 7.** The architecture of decoder for reconstruction from features. Each "Resblocks" in the figure is stacked by three ResBlocks. And a ResBlock consists of two conlutional layers (with $3 \times 3$ kernel size, 128 input channels and 128 output channels) which involved by a single shortcut.

**More reconstruction results.** Fig. 8, 9, 10 show more results of reconstruction of features before and after IF module on several Kodak images. We can see that the images reconstructed from $h$ contain slight color different, and textures are relatively complete. But, the images that reconstructed from $f$ suffer from more obvious color jitter and texture distortion. This indicate that our information filtering (IF) module indeed filter out these color and texture information that have a slight influence on intelligent analytics.


**Fig. 8.** Reconstruction of features before and after IF module on Kodak 4 image. The numbers on the top of the crop images indicate PSNR (dB) / MS-SSIM of an entire image.

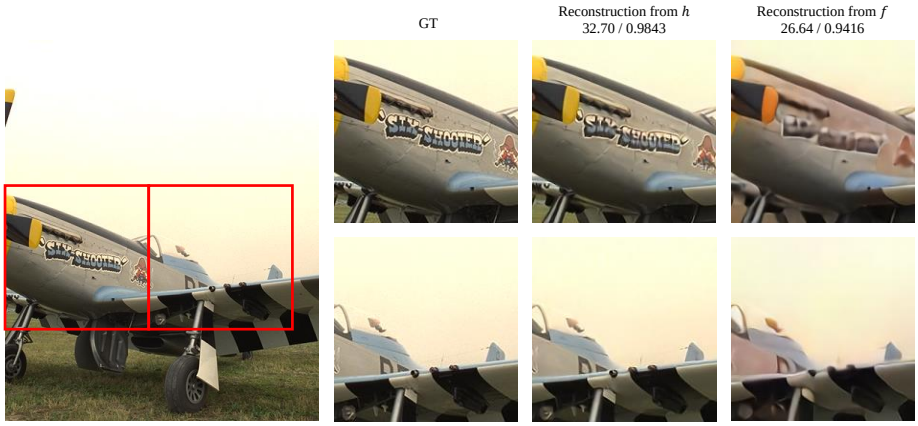


**Fig. 9.** Reconstruction of features before and after IF module on Kodak 20 image. The numbers on the top of the crop images indicate PSNR (dB) / MS-SSIM of an entire image.


**Fig. 10.** Reconstruction of features before and after IF module on Kodak 24 image. The numbers on the top of the crop images indicate PSNR (dB) / MS-SSIM of an entire image.

## References

1. Alemi, A.A., Fischer, I., Dillon, J.V., Murphy, K.: Deep variational information bottleneck. arXiv preprint arXiv:1612.00410 (2016)
2. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: CVPR. pp. 6228–6237 (2018)
3. Bross, B., Wang, Y.K., Ye, Y., Liu, S., Chen, J., Sullivan, G.J., Ohm, J.R.: Overview of the versatile video coding (vvc) standard and its applications. TCSVT (2021)
4. 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)
5. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS **27** (2014)
7. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th international conference on pattern recognition. pp. 2366–2369. IEEE (2010)
8. Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: A survey. arXiv preprint arXiv:2004.05439 (2020)
9. Huang, Z., Jia, C., Wang, S., Ma, S.: Visual analysis motivated rate-distortion model for image coding. In: 2021 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2021)
10. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. TPAMI (2020)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

12. Le, N., Zhang, H., Cricri, F., Ghaznavi-Youvalari, R., Rahtu, E.: Image coding for machines: An end-to-end learned approach. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1590–1594. IEEE (2021)
13. Mentzer, F., Toderici, G.D., Tschannen, M., Agustsson, E.: High-fidelity generative image compression. NeurIPS **33**, 11913–11924 (2020)
14. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV. pp. 69–84. Springer (2016)
15. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR. pp. 2536–2544 (2016)
16. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
17. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. TCSVT **22**(12), 1649–1668 (2012)
18. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv preprint physics/0004057 (2000)
19. Vanschoren, J.: Meta-learning: A survey. arXiv preprint arXiv:1810.03548 (2018)
20. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
21. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV. pp. 649–666. Springer (2016)