Supplementary Information

This document provides more details about our proposed method and comparison.

1.1. Related Work

1.1.1. Image Coding for Machines

Image coding for machines (ICM) focuses on efficiently compressing and transmitting images to support downstream intelligent tasks. Scalable coding approaches [1, 2, 3] utilize multiple bit-streams and decoders to accommodate both machine and human vision. Hybrid methods, commonly referred to as image coding for machine and human (ICMH) [4, 5, 6], adapt pre-trained codecs originally designed for human vision to facilitate machine vision tasks, achieving a balance between the two requirements.

Feature compression techniques [7] target the compression of intermediate features extracted by neural networks, optimizing them for specific tasks. Recent advancements include omnipotent feature learning [8], as well as pre- and post-processing methods [9, 10, 11] to enhance generalization and efficiency. Additionally, feature optimization strategies [12, 13, 14] aim to further improve coding efficiency by tailoring features for specific downstream tasks.

1.1.2. Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) has been extensively studied in natural language processing (NLP) [15, 16, 17], allowing large-scale pre-trained models to be fine-tuned for downstream tasks with minimal modifications to their parameters. With the rise of vision transformers, PEFT techniques have been successfully adapted for computer vision applications. Key approaches include prompt tuning [18] and LoRA (Low-Rank Adaptation) [19, 20], which enable efficient task-specific adaptation while minimizing computational and storage costs.

In the domain of learned image compression, PEFT has shown great potential. For example, Feng *et al.* [21] introduced prompt-based methods for machine-oriented image coding, while Chen *et al.* [4] applied prompt tuning for ICMH. Other approaches, such as [22], focus on transmitting auxiliary adapter parameters to enable content-adaptive optimization, further enhancing task-specific performance.

Building on these advancements, we propose a new method that leverages the strengths of PEFT in ICM. Our approach utilizes pre-trained vision backbones with a low-rank adaptation mechanism to address the limitations of existing ICM methods, which often rely on task-specific or general-purpose models trained from scratch for each task. By significantly reducing training overhead and energy consumption, our method enables robust performance across diverse tasks without requiring full fine-tuning.

1.2. Detailed Training Settings

We list detailed training information in Table 1, including data augmentation, hyperparameters, and training devices.

NYUD-v2 [23] contains various indoor scenes such as offices and living rooms with 795 training and 654 testing images. It provides different dense labels, including semantic segmentation, monocular depth estimation, surface normal estimation, and object boundary detection. **PASCAL-Context** is formed from the PASCAL dataset [24]. It has 4,998 images in the training split and 5,105 in the testing split, covering both indoor and outdoor scenes. This dataset provides pixel-wise labels for semantic segmentation, human parsing, and object boundary detection. Additionally, [25] generates surface normal and saliency labels for this dataset.

Table 1. Training Hyperparameters.

Training set	NYUD-v2	PASCAL-Context
# images	795 training	4,998 training
Image size	Around 448×576	Around 512×512
Data augment.	Crop, h-flip, Scale	Crop, h-flip, Scale
Optimizer	Adam	Adam
Learning rate	2×10^{-5}	2×10^{-5}
LR schedule	Polynomial	Polynomial
Batch size	2	2
# iterations	160k	160k

1.3. Detailed Neural Network Architectures

The EVA-02 pre-trained vision backbone used in our framework is a four-stage plain ViT with modified blocks. The detailed architecture of its Transform Vision (TrV) block is depicted in Fig. 1. The block incorporates a 2D rotary position embedding (RoPE) [26] for injecting positional information, a gated linear unit (GLU) [27, 28] with a sigmoid linear unit (SiLU) [29] as the feedforward network, and sub-layer normalization (sub-LN) [30] as the normalization layer. The pre-training strategies for EVA-02 are detailed in [31]. We utilize the pre-trained weights directly available via timm[1].

1.4. Quantitative Results

We present the BD-Rate, BD-Accuracy, and the number of trainable parameters for various methods in Tab. 3 and Tab. 4. Note that when calculating BD-Rate and BD-Accuracy, only the overlapping regions are considered. Our proposed method exhibits substantial improvements over both the anchor method and the Pre-processor method in terms of BD-Rate and BD-Accuracy across all tasks, while maintaining comparable performance. For instance, the proposed DoRA FT achieves an impressive -80.03% BD-Rate and 17.91% BD-mIoU compared to the VVC + Full FT Baseline

[1] https://huggingface.co/timm/eva02_large_patch14_448.mim_m38m_ft_in22k

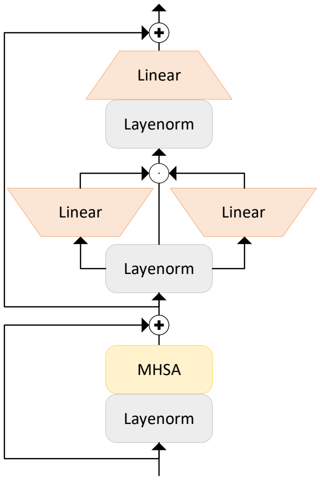


Fig. 1. Transform Vision (TrV) block

anchor. In contrast, the Pre-processor + DoRA FT Baseline achieves only -39.60% BD-Rate and 6.27% BD-mIoU on the NYUD-v2 semantic segmentation task. Furthermore, the total trainable parameters for these methods are 84M and 87M, respectively, underscoring the superior parameter efficiency of the proposed framework.

1.5. Complexity

Table 4 provides a comprehensive comparison of the complexity of various methods, including training time, trainable parameters, GPU memory requirements, and storage overhead for handling both single and multiple tasks. Among these methods, DoRA FT, the approach proposed in this paper, demonstrates distinct advantages in terms of training overhead and storage burden.

Compared to the Full FT, DoRA FT achieves a significant reduction in trainable parameters, from 387M to 84M—a decrease of nearly 78%. Additionally, DoRA FT lowers GPU memory usage from 22.55GB to 19.46GB, which further underscores its efficiency. In terms of storage requirements, DoRA FT exhibits substantial improvements over Full FT. For single tasks, the storage requirement is the same (1.44GB). However, for additional tasks, DoRA FT requires just 0.31GB of storage per task, as only the DoRA layers and the decoder need to be stored. This represents a reduction of over 75% compared to the 1.44GB required by Full FT. This feature makes DoRA FT appealing for multi-task scenarios where storage overhead can escalate rapidly. Additionally, for each task, DoRA FT completes training 3 hours faster than Full FT. When scaled to handle multiple tasks, this time savings become increasingly significant, enabling faster deployment and iteration, saving training resources, and reducing training energy consumption.

Compared to other methods, such as the VVC + Full FT Baseline and Pre-processor + Full FT Baseline, DoRA FT consistently demonstrates superior performance. While the VVC + Full FT Baseline incurs high GPU memory usage (21.63GB) and substantial trainable parameters (330M), DoRA FT reduces these costs. Similarly, the Pre-processor + Full FT Baseline involves significant training time and storage overhead, making it less energy efficient than DoRA FT for multi-tasks.

Overall, DoRA FT's combination of reduced trainable parameters, lower GPU memory usage, minimized storage requirements, and faster training time establishes it as a highly efficient and scalable solution. These advantages position DoRA FT as a practical choice for resource-constrained environments and applications surpassing both baseline and other ICM methods in terms of overall efficiency.

1.6. Task Interaction Results

In the main experiment, separate decoders are used for different tasks. However, recent work [32] has demonstrated that modeling task relationships can significantly enhance performance. To explore this, we propose a straightforward approach to integrate task interactions. Specifically, task features are concatenated along the spatial dimension in the decoder to form a multitask feature representation, $\mathbf{F}_s \in \mathbb{R}^{T\times H_s\times W_s\times C_s}$, where T, H_s, W_s, and C_s represent the number of tasks, spatial dimensions, and channels at stage s, respectively. Using this representation, the query, key, and value projections are defined as follows: $\mathbf{Q}_s = \mathbf{W}_s^q(\mathrm{Conv}(\mathbf{F}_s)) \in \mathbb{R}^{\frac{TH_sW_s}{4}\times C_s}$, $\mathbf{K}_s = \mathbf{W}_s^k(\mathrm{Conv}(\mathbf{F}_s, k_s))$, $\mathbf{V}_s = \mathbf{W}_s^v(\mathrm{Conv}(\mathbf{F}_s, k_s))$, where $\mathbf{K}_s, \mathbf{V}_s \in \mathbb{R}^{\frac{TH_sW_s}{k_s^2}\times C_s}$. The self-attention score matrix is then computed as: $\mathbf{A}_s = \mathbf{Q}_s\mathbf{K}_s^T \in \mathbb{R}^{\frac{TH_sW_s}{4}\times\frac{TH_sW_s}{k_s^2}}$. In this experiment, the encoder from the main experiment is reused, and only the decoder is retrained.

All other configurations remain consistent with the main method. Additionally, our low-complexity decoder facilitates efficient self-attention across both the spatial and task dimensions, maintaining acceptable computational complexity. The results, shown in Fig. 2, clearly demonstrate that incorporating task interactions yields significant performance improvements. These findings highlight the potential of task interaction modeling and underscore its importance as a promising area for further research.

2. REFERENCES

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

[2] Ning Yan, Changsheng Gao, Dong Liu, Houqiang Li, Li Li, and Feng Wu, "Sssic: semantics-to-signal scalable image coding with learned structural representa-

Table 2. Comparison of complexity of various methods

Method	**Training time**	**Trainable parameters**	**GPU memory**	**One task storage space**	**Additional task △ storage space**
VVC + Full FT Baseline	35h	330M	21.63GB	1.23GB	1.23GB
VVC + DoRA FT Baseline	32h	27M	18.76GB	1.23GB	0.10GB
Pre-processor + Full FT Baseline	44h + 35h	60M + 330M	14.52GB + 21.63GB	0.22GB + 1.23GB	0.22GB + 1.23GB
Pre-processor + DoRA FT Baseline	44h + 32h	60M + 27M	14.52GB + 18.76GB	0.22GB + 1.23GB	0.22GB + 0.10GB
Full FT	38h	387M	22.55GB	1.44GB	1.44GB
DoRA FT	35h	84M	19.46GB	1.44GB	0.31GB

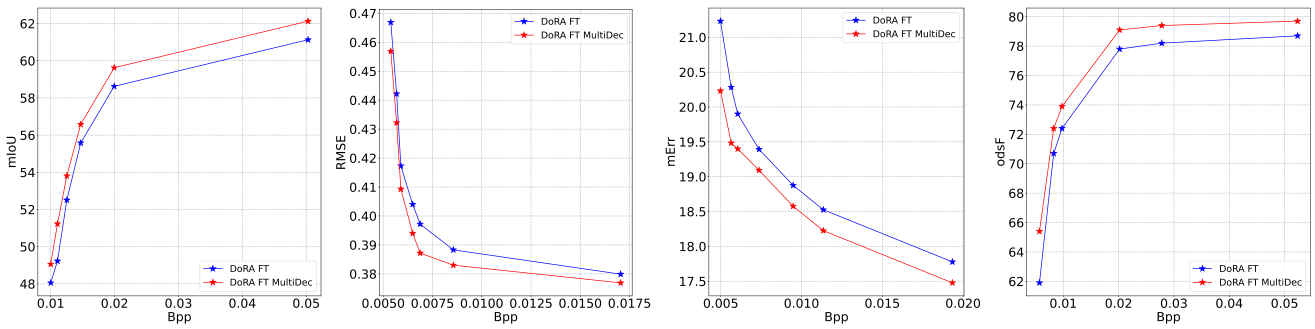
Training Conditions: 1 × Nvidia A40 GPU, AMD EPYC 7662 CPU, 1024GB RAM. Bold represents better performance.

Table 3. Comparison of rate-accuracy performance and the number of trainable parameters under different machine vision tasks and different methods on the NYUD-v2 dataset. VVC + Full FT Baseline is used as the anchor to calculate BD-Rate and BD-Accuracy. Arrows indicate whether lower is better (↓) or higher is better (↑).

Method	**Semseg**		**Depth**		**Normal**		**Boundary**		**Total Trainable**	**Backbone Trainable**
	BD-Rate ↓	**BD-mIoU ↑**	**BD-Rate ↓**	**BD-RMSE ↓**	**BD-Rate ↓**	**BD-mErr ↓**	**BD-Rate ↓**	**BD-odsF ↑**	**Params ↓ (M)**	**Params ↓ (M)**
VVC + Full FT Baseline	0%	0%	0%	0	0%	0	0%	0%	330M	305M
VVC + DoRA FT Baseline	-27.16%	4.20%	-37.66%	-0.06	7.92%	0.34	33.64%	-3.31%	27M	1M
Pre-processor + Full FT Baseline	-18.90%	2.65%	-19.61%	-0.03	-23.85%	-1.20	-17.70%	1.90%	390M	305M
Pre-processor + DoRA FT Baseline	-39.60%	6.27%	-50.87%	-0.09	-17.57%	-0.84	10.78%	-1.13%	87M	1M
Full FT	-76.68%	19.79%	-87.66%	-0.92	-88.14%	-8.68	-77.74%	13.46%	387M	305M
DoRA FT	-80.03%	17.91%	-97.08%	-0.71	-89.39%	-8.97	-67.27%	8.84%	84M	1M

Table 4. Comparison of rate-accuracy performance and the number of trainable parameters under different machine vision tasks and different methods on the PASCAL- Context dataset. VVC + Full FT Baseline is used as the anchor to calculate BD-Rate and BD-Accuracy. Arrows indicate whether lower is better (↓) or higher is better (↑).

Method	**Semseg**		**Parsing**		**Saliency**		**Normal**		**Boundary**	
	BD-Rate ↓	**BD-mIoU ↑**	**BD-Rate ↓**	**BD-mIoU ↑**	**BD-Rate ↓**	**BD-maxF ↑**	**BD-Rate ↓**	**BD-mErr ↓**	**BD-Rate ↓**	**BD-odsF ↑**
VVC + Full FT Baseline	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
VVC + DoRA FT Baseline	-18.61%	3.89%	-15.42%	3.13%	13.71%	-0.98%	-19.04%	-0.67%	-1.83%	0.25%
Pre-processor + Full FT Baseline	-20.05%	3.66%	-21.45%	4.25%	-32.47%	2.81%	-37.47%	-1.54%	-22.04%	3.79%
Pre-processor + DoRA FT Baseline	-33.85%	6.89%	-34.78%	6.80%	-22.03%	1.85%	-52.00%	-2.17%	-21.27%	3.46%
Full FT	-85.57%	29.75%	-91.16%	26.94%	-92.30%	66.44%	-92.96%	-8.55%	-86.20%	23.13%
DoRA FT	-85.59%	26.04%	-92.19%	23.64%	-92.07%	32.58%	-93.24%	-8.82%	-82.46%	24.01%



(a) Semseg

(b) Depth

(c) Normal

(d) Boundary

Fig. 2. **Comparison of separate coder and multitask decoder on NYUD-v2 dataset.**

tions," *IEEE Transactions on Image Processing*, vol. 30, pp. 8939–8954, 2021.

[3] Shuai Yang, Yueyu Hu, Wenhan Yang, Ling-Yu Duan, and Jiaying Liu, "Towards coding for human and machine vision: Scalable face image coding," *IEEE Transactions on Multimedia*, vol. 23, pp. 2957–2971, 2021.

[4] 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," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23297–23307, 2023.

[5] Lei Liu, Zhihao Hu, Zhenghao Chen, and Dong Xu, "Icmh-net: Neural image compression towards both machine vision and human vision," *Proceedings of the ACM International Conference on Multimedia*, pp. 8047–8056, 2023.

[6] Han Li, Shaohui Li, Shuangrui Ding, Wenrui Dai, Maida Cao, Chenglin Li, Junni Zou, and Hongkai Xiong, "Image compression for machine and human vision with spatial-frequency adaptation," *Proceedings of the European Conference on Computer Vision*, pp. 382–399, 2025.

[7] Zhuo Chen, Kui Fan, Shiqi Wang, Lingyu Duan, Weisi Lin, and Alex Chichung Kot, "Toward intelligent sensing: Intermediate deep feature compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 2230–2243, 2019.

[8] 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," *Proceedings of the European Conference on Computer Vision*, pp. 510–528, 2022.

[9] Guo Lu, Xingtong Ge, Tianxiong Zhong, Qiang Hu, and Jing Geng, "Preprocessing enhanced image compression for machine vision," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[10] Mingyi Yang, Fei Yang, Luka Murn, Marc Gorriz Blanch, Juil Sock, Shuai Wan, Fuzheng Yang, and Luis Herranz, "Task-switchable pre-processor for image compression for multiple machine vision tasks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[11] Binzhe Li, Shurun Wang, Shiqi Wang, and Yan Ye, "High efficiency image compression for large visual-language models," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[12] Wenhan Yang, Haofeng Huang, Yueyu Hu, Ling-Yu Duan, and Jiaying Liu, "Video coding for machines: Compact visual representation compression for intelligent collaborative analytics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[13] Xihua Sheng, Li Li, Dong Liu, and Houqiang Li, "Vnvc: A versatile neural video coding framework for efficient human-machine vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[14] Yuan Tian, Guo Lu, Yichao Yan, Guangtao Zhai, Li Chen, and Zhiyong Gao, "A coding framework and benchmark towards low-bitrate video understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[15] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al., "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

[16] 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," *Proceedings of the International Conference on Learning Representations*, 2022.

[17] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen, "Dora: Weight-decomposed low-rank adaptation," 2024.

[18] Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan, "Promptir: Prompting for all-in-one image restoration," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 36, 2024.

[19] Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang, "Parameter-efficient model adaptation for vision transformers," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, pp. 817–825, 2023.

[20] Ahmed Agiza, Marina Neseem, and Sherief Reda, "Mtlora: Low-rank adaptation approach for efficient multi-task learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16196–16205, 2024.

[21] Ruoyu Feng, Jinming Liu, Xin Jin, Xiaohan Pan, Heming Sun, and Zhibo Chen, "Prompt-icm: A unified framework towards image coding for machines with task-driven prompts," 2023.

[22] Yue Lv, Jinxi Xiang, Jun Zhang, Wenming Yang, Xiao Han, and Wei Yang, "Dynamic low-rank instance adaptation for universal neural image compression," *Proceedings of the ACM International Conference on Multimedia*, pp. 632–642, 2023.

[23] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, "Indoor segmentation and support inference from rgbd images," *Proceedings of the European Conference on Computer Vision*, pp. 746–760, 2012.

[24] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *Interna-*

tional Journal of Computer Vision, vol. 88, pp. 303–338, 2010.

[25] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos, "Attentive single-tasking of multiple tasks," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1851–1860, 2019.

[26] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, pp. 127063, 2024.

[27] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," *Proceedings of the International Conference on Machine Learning*, pp. 933–941, 2017.

[28] Noam Shazeer, "Glu variants improve transformer," 2020.

[29] Dan Hendrycks and Kevin Gimpel, "Gaussian error linear units (gelus)," 2016.

[30] Hongyu Wang, Shuming Ma, Shaohan Huang, Li Dong, Wenhui Wang, Zhiliang Peng, Yu Wu, Payal Bajaj, Saksham Singhal, Alon Benhaim, et al., "Foundation transformers," 2022.

[31] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao, "Eva-02: A visual representation for neon genesis," *Image and Vision Computing*, vol. 149, pp. 105171, 2024.

[32] Anonymous, "Which tasks should be compressed together? a causal discovery approach for efficient multitask representation compression," 2024.