Test-Time Fine-Tuning of Image Compression Models for Multi-Task Adaptability

Supplementary Material

This material consists of the following items. These additional discussion and pieces of information will aid in understanding the proposed method.

- Discussion;
- Comparison of Computational Efficiency;
- Another Example of Applying SVD-LoRA;
- More Ablation Studies;
- More Tasks and Vision models;
- More Qualitative Results.

A1. Discussion

As mentioned in the Sec. 1, future compressors are expected to prioritize machine vision tasks as their main applications rather than human perception, ultimately addressing multiple open-set machine vision tasks. In other words, while ensuring superior performance in closed-set tasks as a fundamental capability, there will be a growing demand for effectively handling open-set tasks. Multi-task and single-task bitstream LICs are actively researched to address these needs. However, many previous methods struggle to fully accommodating open-set tasks. A pre-trained, single-model-based network still faces challenges in adapting to unseen dataset distributions and tasks. The proposed method provides a effective approach for handling open-set tasks. By applying SVD-LoRA to both the encoder and decoder of the backbone LIC model, the method enables fully instance-specific Test-Time Fine-Tuning (TTFT) for adaptation to targeted open-set tasks. While the proposed TTFT method introduces additional time overhead during the encoding stage, this issue can be controlled by adjusting TTFT iterations, as described in Sec. 5.4. Moreover, certain compression systems can inherently address this issue. For example, compression systems in satellite and medical imagery do not require a low-latency encoding process. In these systems, performance is more critical than time overhead.

Our proposed method can provide limitless potential for performance improvement in line with the advancements in learned image compression. As mentioned in Sec. 4.1 and Sec. A3, the proposed method can be applied effectively regardless of whether the architecture is transformer-based or CNN-based. In other words, as the backbone network advances, the proposed approach will also evolve in parallel with negligible increases in model size and inference cost. As shown in Sec. A2, the proposed method maintains cost efficiency at a level comparable to the backbone network while simultaneously surpassing SOTA performance, as demonstrated in Sec. 5.

Table A1. Comparison of kMACs/pixel and model size. Compared to previous SOTA method, our approach achieves superior performance as mentioned in Sec. 5 while requiring fewer computations and a smaller model size. Additionally, our approach preserves computational efficiency and model size comparable to the backbone LIC model.

	kMACs/pixel		Params (M)	
	Encoder	Decoder	Encoder	Decoder
TIC	142.31	142.53	3.64	3.86
TransTIC	332.03	202.6	5.24	3.89
Ours (rank 2)	142.31	142.53	3.69	3.92
Ours (rank 4)	**142.31**	**142.53**	**3.72**	**3.94**
Ours (rank 8)	142.31	142.53	3.76	3.99

A2. Comparison of Computational Efficiency

Table A1 demonstrates the cost efficiency of our approach. In the Sec. 5, the proposed method exhibited superior performance over the SOTA performance of TransTIC in both closed-set and open-set tasks. Moreover, our approach maintains computational efficiency, as measured in kMACs/pixel, and model size at levels comparable to the backbone LIC model, whereas TransTIC requires significantly more computational resources and larger parameter sizes. In Table A1, bold text indicates the setting conditions from Sec. 5. Although adjusting the rank in our method does lead to changes in model size, the extent of variation is minimal compared to that of TransTIC. The performance impact of rank adjustments is discussed in Sec. A4.2.

A3. Another Example of Applying SVD-LoRA

As we mentioned in Sec. 4.1, the proposed method is compatible with various backbone LIC architectures, including not only transformer-based models but also CNN models. Fig. A1 shows an example of applying SVD-LoRA to the CNN-based codec [14]. To verify the performance, we applied the closed set scenario for object detection, as previously described in Sec. 5. Consequently, Fig. A2 demonstrates that the application of our method is also effective for CNN-based codecs.

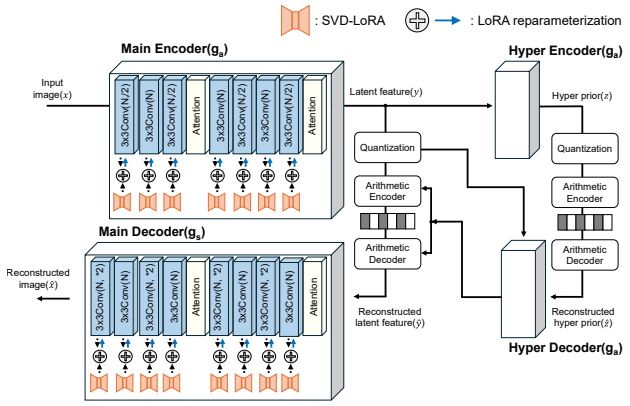

Figure A1. An example of applying SVD-LoRA to the CNN-based codec [14]. The SVD-LoRA is only applied to the CNN layers of the main encoder and decoder, which are known to have the most significant impact on task adaptation.

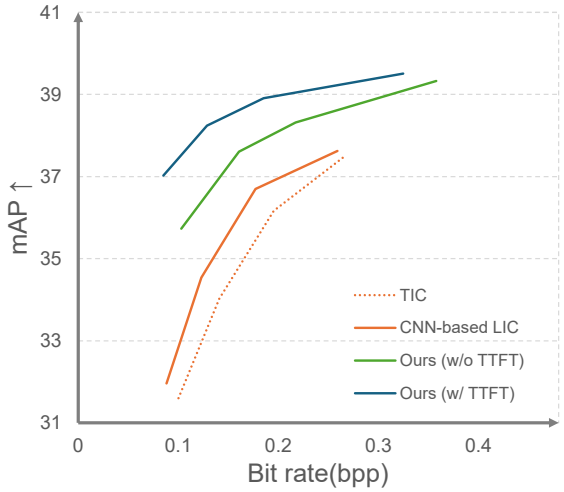

Figure A2. Rate-accuracy performance when our approach is applied to a CNN-based codec. Experiments demonstrate that the application of our method is also effective for CNN-based codecs.

A4. More Ablation Studies

A4.1. LoRA vs. SVD-LoRA

Fig. A3 shows the rate-accuracy performance comparison between LoRA and SVD-LoRA. For this performance evaluation, we employed a closed set scenario without TTFT for object detection, as described in Sec. 5. In the case of LoRA, it is applied exclusively to the encoder, since applying LoRA to the decoder would prevent support for TTFT, as mentioned in Sec. 5.4. In terms of performance, SVD-LoRA slightly outperforms LoRA, indicating that SVD-LoRA is more effective in learning the intrinsic dimension of weights at the same rank level.

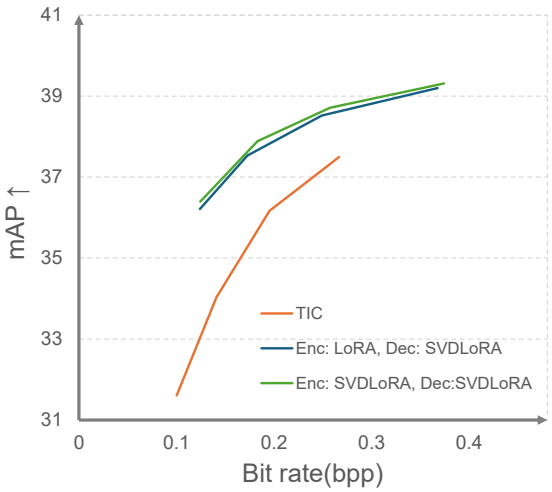

Figure A3. Ablation on LoRA and SVD-LoRA. SVD-LoRA slightly outperforms LoRA, demonstrating its superior ability to capture the intrinsic dimension of weights at the same rank.

Table A2. TTFT time comparison with varying SVD-LoRA ranks. The time overhead increases with rank, but the difference is negligible since the SVD-LoRA weights constitute only a small portion of the entire model.

	TTFT Time (sec)
Rank 2	13.38
Rank 4	**14.13**
Rank 8	14.26

A4.2. Varying the Rank of SVD-LoRA

Fig. A4 and Table A2 respectively show the performance and TTFT time comparison with different SVD-LoRA ranks. For the performance evaluation, we employed a closed set scenario for object detection, as described in Sec. 5. Additionally, the TTFT time is measured under the same conditions as Sec. 5.4. The proposed method in Sec. 5 used a rank of 4, and in this ablation study, we compared ranks 2 and 8. As illustrated in Fig. A4, the performance improvement from rank 2 to rank 4 is noticeable, whereas the improvement from rank 4 to rank 8 is minimal. Whether the LoRA rank is sufficient depends on the intrinsic dimensional complexity of the weights required during the task transfer process. Experimentally, we verified that rank 4 is appropriate for the dataset and task complexity we employed. As shown in Table A2, the difference in TTFT time with varying ranks is negligible. As indicated in Table A1, the number of learnable parameters due to varying ranks is minimal compared to the overall model size, leading to an insignificant impact on TTFT time.

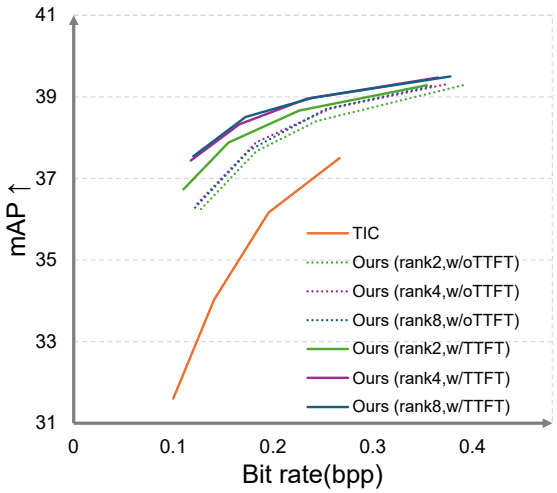

Figure A4. Ablation on the effect of varying SVD-LoRA rank. The performance improves noticeably from rank 2 to 4 but shows minimal gain from rank 4 to 8.

A4.3. Full fine-tuning on encoder side

Table A3 shows an ablation study on encoder full fine-tuning for open-set object detection, with applying 40 TTFT iterations. While Variant (encoder full fine-tuning) resulted in a slight performance improvement, it requires 2.82M learnable parameters for each instance-specific TTFT, which leads to an increase in TTFT time overhead.

Table A3. Ablation study on full fine-tuning of encoder. Variant (encoder full fine-tuning) slightly improves performance but requires a large number of parameters for each instance-specific TTFT.

Method		BD-Rate(%)	BD-mAP ↑	#Params
Ours	Enc: SVD-LoRA(rank 4) Dec: SVD-LoRA(rank 4)	-48.94	2.69	0.05 M
Variant	Enc: Full fine tuning Dec: SVD-LoRA(rank 4)	-50.86	3.01	**2.82 M**

A5. More Tasks and Vision models

We conducted additional experiments on other tasks, including pose estimation [34] and depth estimation [63], using the datasets [2] and [22], respectively. Further experiments were also performed with other models [8, 35] for object detection and instance segmentation tasks. The results in Table A4 demonstrates that our method is effective in different settings for open-set adaptation.

A6. More qualitative results

Fig. A5, Fig. A6, and Fig. A7 present qualitative results across different machine vision tasks. The results are compared with the backbone TIC and the SOTA competing method, TransTIC, all evaluated at the quality level 1. The analysis indicates that due to the compression loss in the backbone model, texture quality is noticeably degraded, resulting in lower performance scores on vision tasks compared to the original image without compression. TransTIC struggles to recover from this compression loss, leading to suboptimal performance. In contrast, the proposed method, which utilizes TTFT, demonstrates a notable ability to restore some of the texture, achieving higher performance scores.

Table A4. Performance evaluation with more diverse datasets and tasks. Experiments show that our method is effective in different settings for open-set adaptation.

Method	Pose estimation [34]		Depth estimation [63]		Object detection w/ [8]		Instance segmentation w/ [35]	
	BD-Rate(%)	BD-PCKh@0.5 ↑	BD-Rate(%)	BD-Error ↓	BD-Rate(%)	BD-mAP ↑	BD-Rate(%)	BD-mAP ↑
TransTIC(Open)	-32.73	1.10	-33.81	-1.13	-44.94	3.05	-44.22	2.42
Ours(w/o TTFT, Open)	-26.50	0.74	-42.61	-1.14	-32.48	2.09	-30.35	1.96
Ours(w/ TTFT, Open)	**-35.30**	**1.25**	**-66.65**	**-1.80**	**-47.21**	**3.21**	**-48.06**	**2.89**

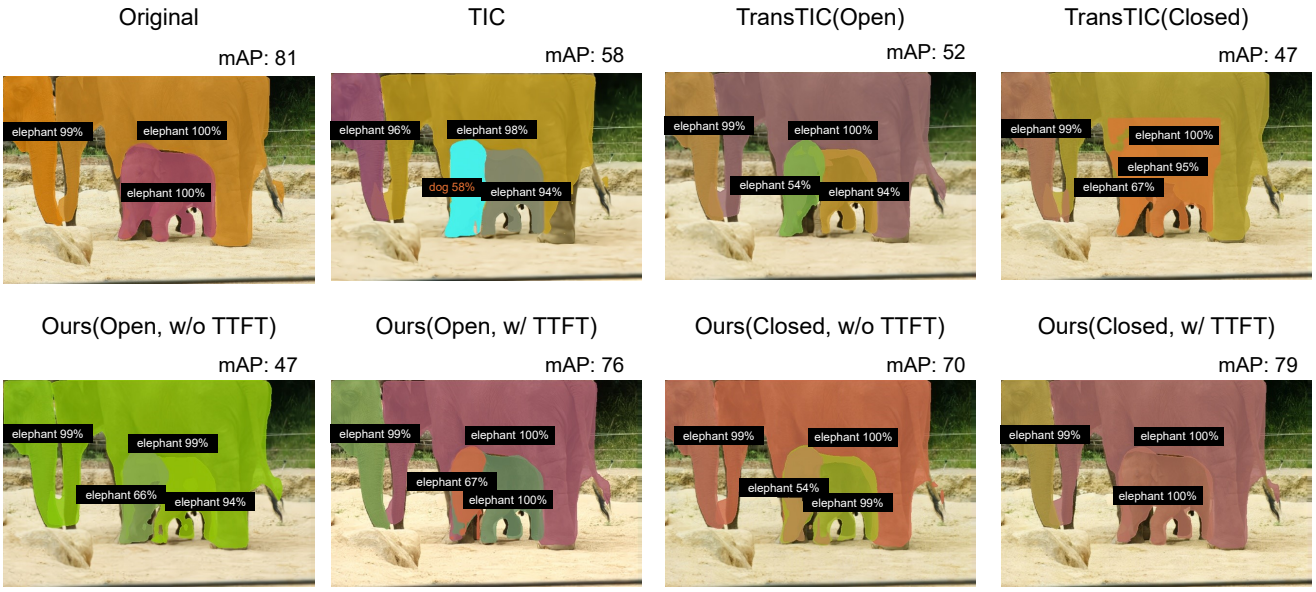

Figure A5. Visualization of instance segmentation results. Ours (w/TTFT) effectively handles both closed-set and open-set tasks, while the SOTA competing method struggles.

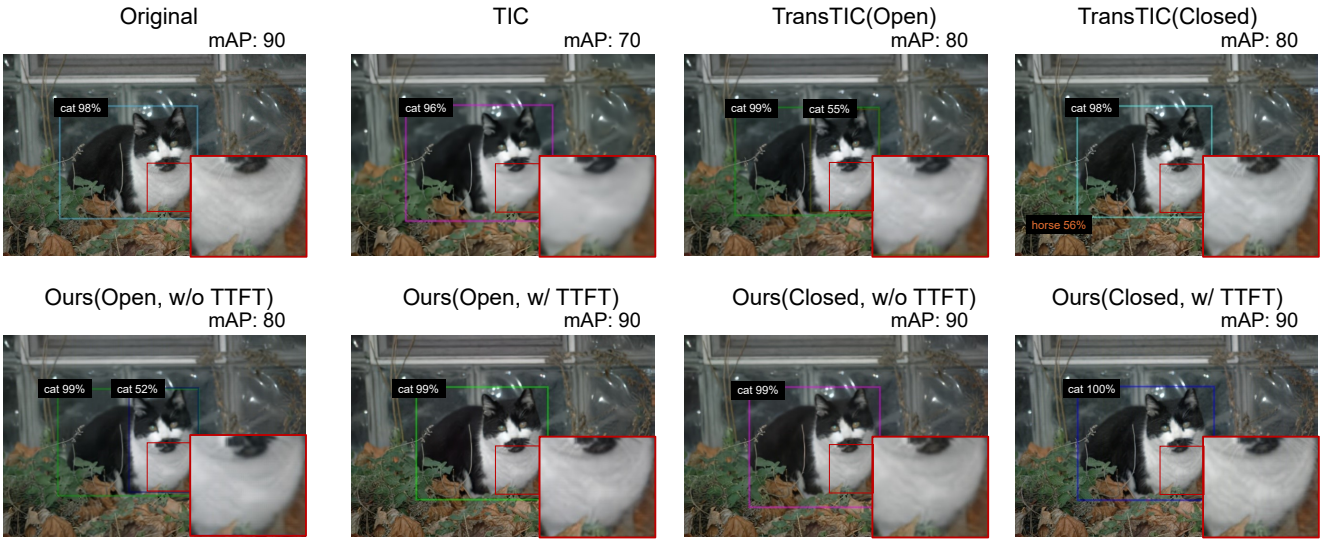

Figure A6. Visualization of object detection results. Ours(w/TTFT) effectively restores texture, achieving performance scores comparable to the original image in both closed-set and open-set tasks.

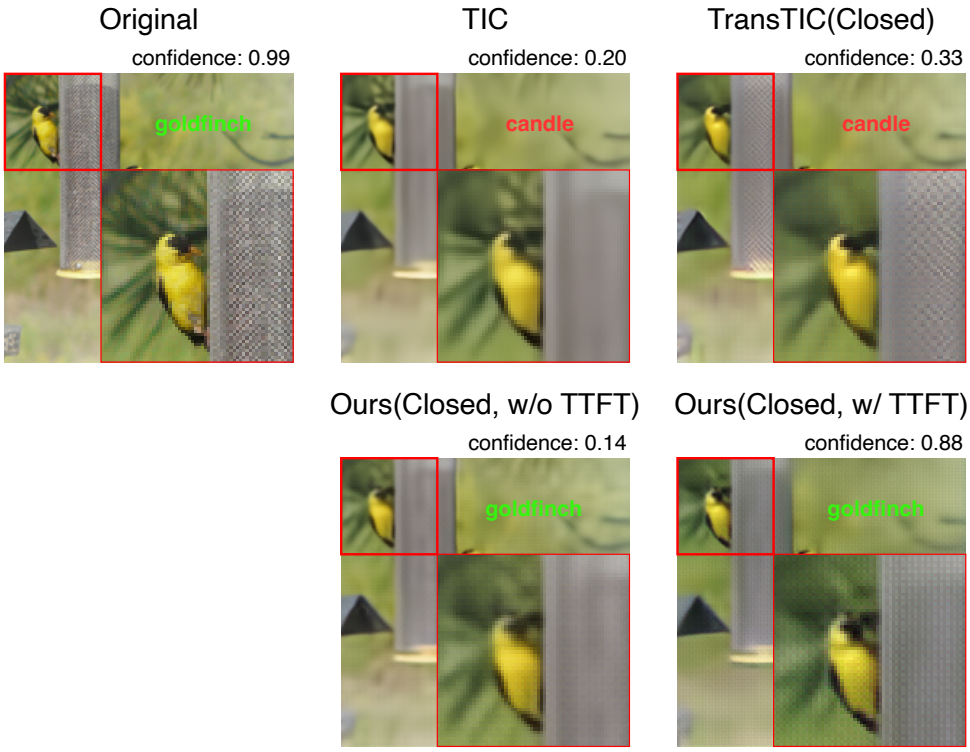

Figure A7. Visualization of classification results. Ours (w/TTFT) classifies the correct label with a significantly higher confidence score, while the SOTA competing method misclassifies.