What and Where to Adapt: Structure–Semantics Co-Tuning for Machine Vision Compression via Synergistic Adapters

Supplementary Material

This supplement provides additional discussion on:

A. PEFT in ICMH

We introduce a paradigm shift in Learned Image Compression (LIC) by incorporating Parameter-Efficient Fine-Tuning (PEFT) to resolve the scalability bottleneck in multi-task deployment. While traditional full fine-tuning yields task-specific gains, it necessitates maintaining separate, heavyweight checkpoints for every downstream application, rendering deployment prohibitively expensive in terms of storage and version management. In contrast, our plug-and-play adapter design offers a superior performance-flexibility trade-off. This architecture disentangles optimization goals: the frozen base codec preserves optimal perceptual quality for human viewing, while the insertion of lightweight adapters maximizes machine-task accuracy. Consequently, this enables rapid, low-cost adaptation to diverse analytics scenarios, where maintaining a library of tiny adapters is orders of magnitude more efficient than storing N independent, fully fine-tuned models.

Table 1. Performance comparison on different tasks of ICMH.

Tasks	Models	Bit Rate	mAP (↑)	PSNR (↑)	PSNR w/o Adapters (↑)
object detection	base codec	0.056	27.123	30.286	30.286
	S^2-CoT	0.0814	36.767	27.044	30.286
instance segmentation	base codec	0.056	24.81	30.29	30.29
	S^2-CoT	0.103	34.64	28.18	30.29

To clarify the benefits of our task-oriented modules and the associated trade-offs, we provide a detailed quantitative analysis in Tab. 1. Our S^2-CoT framework employs Parameter-Efficient Fine-Tuning adapters to decouple machine vision optimization from human perceptual requirements. As shown in Tab. 1, activating the adapters significantly boosts downstream performance, exemplified by substantial gains in object detection metrics, with only a marginal increase in bitrate. While optimizing specifically for machine accuracy involves a trade-off in pixel-wise fidelity as reflected in the decrease of PSNR, this aligns with the consensus that semantic feature preservation often diverges from pixel-perfect reconstruction. Crucially, the primary benefit of introducing these extra modules lies in scalability and flexibility. As evidenced by the "PSNR w/o Adapters" column, our framework maintains the capability to revert to the optimal human-perceptual quality of the base codec simply by deactivating the adapters. This design allows a single system to support diverse transmission demands: it provides high-efficiency feature coding for machine analytics when adapters are active and high-fidelity reconstruction for human viewing when bypassed, achieving the best of both worlds without altering the codec.

B. More Related Work

This section supplements the main paper's related work by providing a broader historical and technical context. We

first review traditional codecs to establish a historical baseline, followed by a summary of the general evolution of LIC. This provides the foundational context for the more specialized methods discussed in the main paper.

B.1. Traditional Image Compression

Traditional image compression methods, such as JPEG [23] and JPEG2000 [3, 7], rely on discrete cosine transform and wavelet decomposition to exploit perceptual redundancy. The objective of optimization is to reduce image distortion. These methods have been successful in removing spatial redundancy from images and are able to satisfy the image quality requirements of human vision to a certain extent. However, given that these systems were not originally designed for machine vision tasks, applying them to such contexts results in a serious degradation of downstream performance due to the failure to preserve semantic integrity.

B.2. Learned Image Compression

Learned image compression [1, 2, 6, 8, 12, 14, 17, 20, 25, 26] has revolutionized the field by employing advanced end-to-end optimized neural networks to surpass the rate-distortion performance of traditional codecs [3, 7, 23]. Foundational works by Ballé *et al.* [1, 2] established this paradigm, demonstrating the power of learning nonlinear transforms from data.

Subsequent research has advanced LIC by exploring increasingly powerful architectures for the nonlinear transform backbone. Architectures have evolved from early CNN-based designs [6] to more powerful Transformer-based models [16, 26] that better capture both local and global dependencies in the image. Despite these significant architectural advancements, the majority of these foundational LIC methods are optimized for human-centric metrics (*e.g.*, PSNR, MS-SSIM). Consequently, they exhibit limited adaptability for diverse machine vision tasks, which prioritize the preservation of task-critical semantic, rather than pixel-level, fidelity.

C. Theoretical Analysis and Empirical Analysis of the Structure-Semantics Synergy

This section complements Sec. 3.1 of the main paper by providing a comprehensive and rigorous theoretical derivation of the coding redundancy penalty (ΔR) and presenting a detailed analysis of the synergy between the Structural Fidelity Adapter (SFA) and the Semantic Context Adapter (SCA). We demonstrate how the proposed S^2-CoT framework effectively aligns the structural representation with statistical modeling to minimize redundancy.

C.1. Derivation of Coding Redundancy Penalty

In a standard LIC framework, the non-linear encoder-decoder (also called transform) (g_a, g_s) and the entropy model (h_a, h_s) are jointly optimized. The encoder g_a maps an input x to a compact latent representation y, while the entropy model estimates the probability distribution $p_{\hat{y}}$ of its quantized version $\hat{y}$ to minimize the bitrate.

In hyperprior-based models [2], the latent y is typically modeled as a Gaussian distribution conditioned on the hyper-latent $\hat{z}$:

$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) \sim \mathcal{N}(\mu, \sigma^2) \tag{1}$$

where the Gaussian parameters (μ, σ) are predicted by the hyperprior decoder h_s:

$$(\mu, \sigma) = h_s(\hat{z}; \theta_{h_s}), \quad \text{with} \quad \hat{z} = Q(h_a(y; \theta_{h_a})) \tag{2}$$

The theoretical bitrate $R(\hat{y})$ is given by the cross-entropy between the marginal distribution of the latents and the estimated probability model:

$$R(\hat{y}) = \mathbb{E}_{\hat{y}\sim p_{\text{data}}}[-\log_2(p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}))] \tag{3}$$

C.2. Consequence of Isolated Adaptation

When we perform Parameter-Efficient Fine-Tuning (PEFT) using only the structural adapter in the encoder-decoder (g_a, g_s) while keeping the entropy model frozen [15], the latent representation shifts from y to a task-adapted feature space y'. This shift alters the statistical properties of the distribution, denoted as p'_{data}.

However, since the entropy model parameters $(\theta_{h_a}, \theta_{h_s})$ remain frozen, the predicted Gaussian parameters (μ', σ') are derived from the pre-trained (human-oriented) priors:

$$(\mu', \sigma') = h_s(\hat{z}'; \theta_{h_s}), \quad \text{where} \quad \hat{z}' = Q(h_a(y'; \theta_{h_a})) \tag{4}$$

These parameters (μ', σ') are suboptimal for the new distribution y', creating a deviation from the ideal parameters $(\mu_{\text{ideal}}, \sigma_{\text{ideal}})$ that would model the task-adapted latents. This statistical misalignment leads to a non-negligible coding redundancy penalty ΔR, defined as the Kullback-Leibler (KL) divergence between the ideal distribution and the frozen model's estimation.

Substituting this into the bitrate equation (Eq. (2) in the main paper), we derive:

$$\begin{aligned} R(\hat{y}') &= \mathbb{E}[-\log_2(p_{\text{frozen}}(\hat{y}'|\hat{z}'))] \\ &= \underbrace{\mathbb{E}[-\log_2(p_{\text{ideal}}(\hat{y}'|\hat{z}'))]}_{\text{Ideal Bitrate}} + \underbrace{D_{\text{KL}}(p_{\text{ideal}}||p_{\text{frozen}})}_{\text{Penalty } \Delta R} \end{aligned} \tag{5}$$

Here, $\Delta R > 0$ represents the extra bits consumed due to the failure of the frozen entropy model to capture the structural changes introduced by the transform-only adapter.

C.3. Synergy via Semantic Context Adapter

The proposed context adapter is explicitly designed to eliminate this penalty. By inserting semantic context adapter

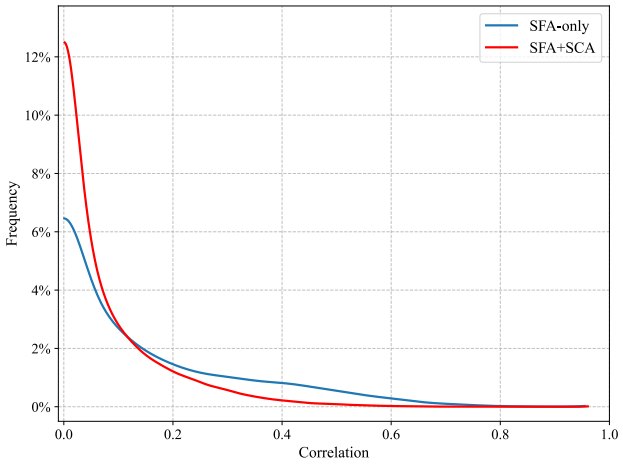

Figure 1. Pairwise channel similarities of the latent representation y for different fine-tuning strategies. Incorporating SCA markedly reduces the inter-channel correlation.

into the entropy model, we introduce a learnable correction term that adapts the frozen priors to the new latent statistics:

$$(\mu_{\text{tuned}}, \sigma_{\text{tuned}}) = \text{SCA}(h_s(\hat{z}'; \theta_{h_s})) \tag{6}$$

Through the joint optimization of S^2-CoT, the SCA effectively minimizes the KL divergence term, driving $\Delta R \to 0$. This ensures that the structural gains provided by SFA are preserved without incurring a bitrate penalty, realizing the theoretical basis of our structure-semantics synergy.

C.4. Correlation & Redundancy Analysis

To empirically validate the theoretical derivation above, we analyze the statistical properties of the latent representations produced by different adaptation strategies. We compare the baseline using SFA-only (isolated structural tuning) against our proposed SFA+SCA (S^2-CoT) framework.

C.4.1. Inter-channel correlation

We first calculate the pairwise channel similarity of the latent y to estimate inter-channel redundancy. High correlation implies that multiple channels encode similar information, wasting bitrate.

As shown in Fig. 1, the SFA-only baseline exhibits a heavy tail and a low initial value in the similarity distribution, indicating significant unresolved redundancy. In contrast, the SFA+SCA model shows a steeper decay, suggesting that the SCA effectively decorrelates the features. This confirms that the SCA refines the channel context to learn a more compact, orthogonal representation for machine tasks.

C.4.2. Spatial correlation

We extend the analysis to the spatial dimension by measuring the correlation between adjacent spatial positions in the normalized latents $(y-\mu)/\sigma$. Ideally, an optimized entropy model should whiten the latents, resulting in near-zero spatial correlation.

As visualized in Fig. 2, the SFA-only strategy results in a high average spatial correlation ($\rho = 0.2914$), reflecting the frozen entropy model's inability to capture the new spatial structures. The synergistic SFA+SCA approach substantially reduces this to $\rho = 0.2489$. This reduction serves as direct evidence that the S^2-CoT framework successfully aligns the statistical model with the structural features.

C.4.3. Analysis of quantization deviation

To further evaluate the precision of the probability estimation, we analyze the "scaling deviation" metric ϵ, which acts as a proxy for quantization loss [24]. It measures the discrepancy between the latent y and the estimated mean μ:

$$\epsilon = \frac{|Q(y-\mu) - (y-\mu)|}{\sum |y|}. \tag{7}$$

A smaller ϵ indicates that the entropy model's predicted mean μ is closer to the actual latent center, minimizing quantization error.

Fig. 3 presents the scaled deviation maps. The SFA-only approach yields high deviation errors (warmer regions), confirming the mismatch where the frozen priors fail to center the task-adapted latents. Conversely, the addition of SCA in our framework significantly suppresses these errors (cooler regions), particularly in the background. This validates that SCA effectively corrects the statistical shift $\Delta\mu$, ensuring precise quantization and efficient coding.

C.4.4. Visualizing entropy distribution

Meanwhile, we provide a qualitative view of the latent representation energy. Fig. 4 displays the five channels with the highest entropy. The latents from the SFA+SCA strategy exhibit significantly less perceptible structure (more "noise-like") compared to SFA-only. In compression theory, a more unstructured, whitened latent representation indicates better decorrelation and higher coding efficiency. This visually confirms that our S^2-CoT effectively removes semantic redundancy while preserving the task-critical information in the decoder features.

C.4.5. Quantitation of redundancy elimination

As visualized in the rate-accuracy curve for object detection on COCO2017 (see Fig. 5), the deployment of our full S^2-CoT framework (SFA+SCA) establishes a superior Pareto frontier compared to the isolated SFA-only baseline. The distinct leftward shift, highlighted by the blue arrows, signifies a substantial reduction in bitrate (bpp) at equivalent mean Average Precision (mAP) levels. This empirical evidence validates that the SCA effectively rectifies the statistical misalignment induced by structural fine-tuning, thereby

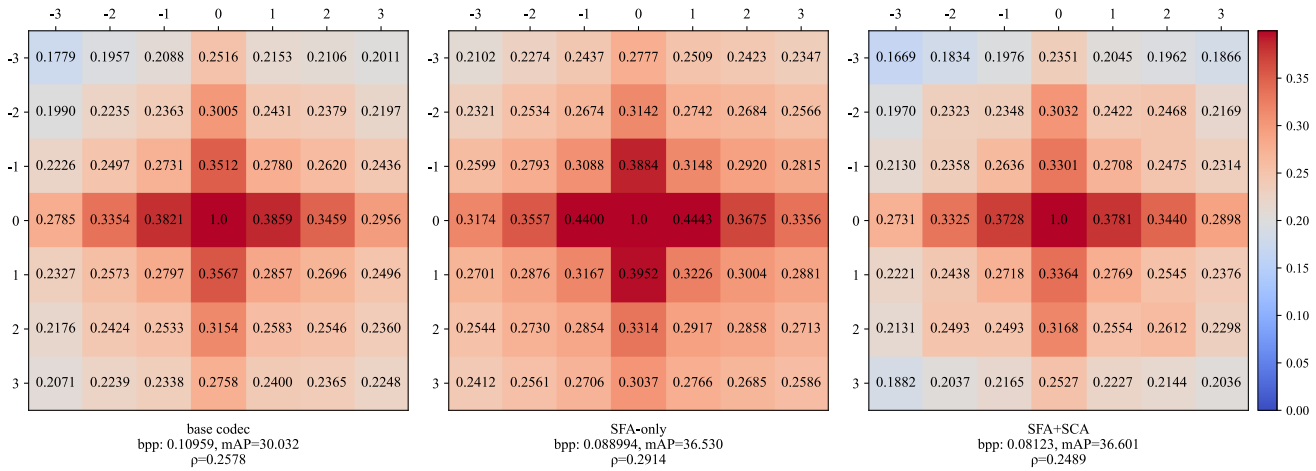

Figure 2. Spatial correlation of $(y - \mu)/\sigma$ for models trained with λ=0.5. SFA+SCA (right) reduces average spatial correlation compared to SFA-only (middle) and the base codec (left), which benefits subsequent machine vision tasks.

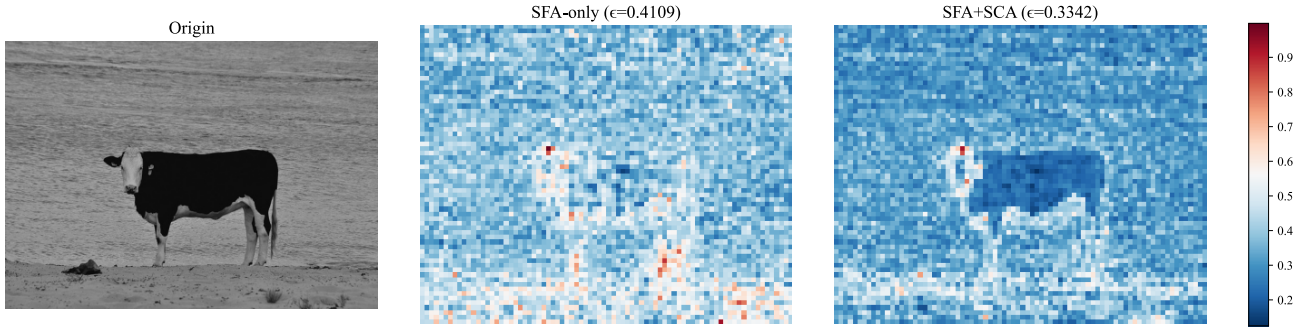

Figure 3. Scaled deviation map of two strategies: SFA-only and SFA+SCA.

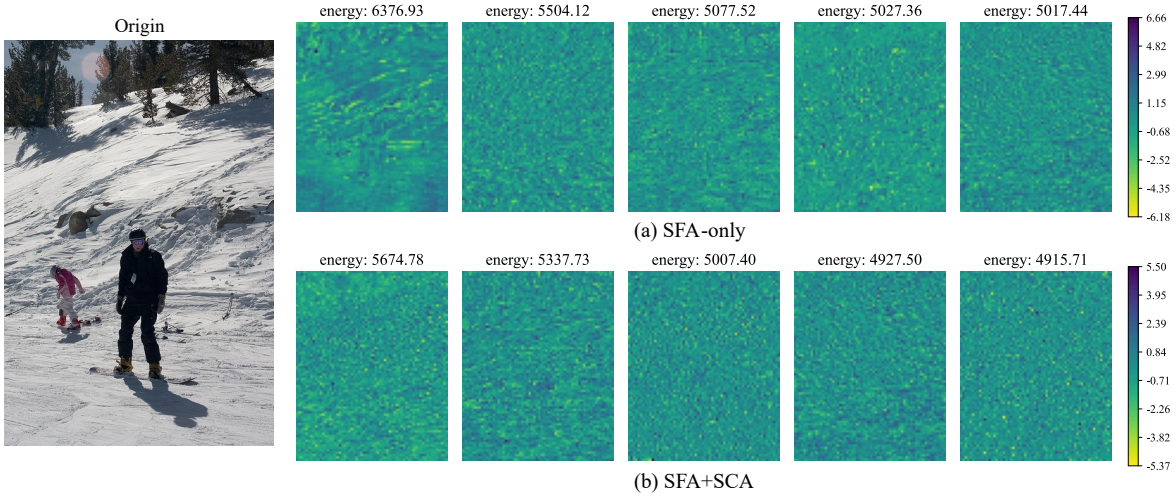

Figure 4. Each row corresponds to a different strategy and shows the energy of five channels with the highest entropy.

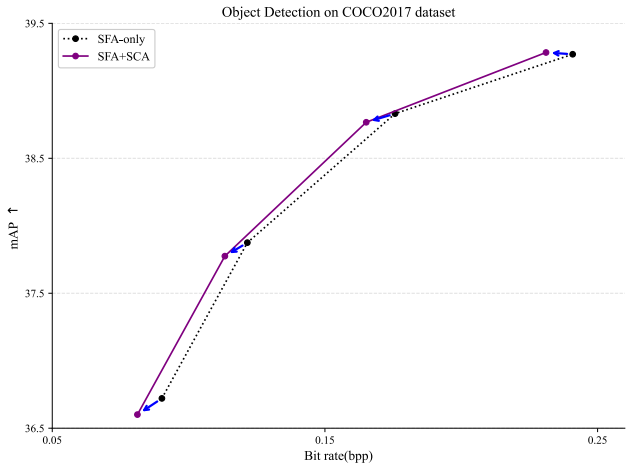

Figure 5. Object detection performance on the COCO2017 dataset using *Lu2022-TIC* as the base codec. Incorporating SCA reduces the bitrate without compromising detection accuracy.

eliminating coding redundancy and yielding a more compact, task-optimal latent representation.

D. Task-specific Perceptual Distortion Loss

$$\mathcal{L} = \mathcal{R}(\hat{\mathbf{y}}) + \lambda \cdot \mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}; \mathcal{G}) \tag{8}$$

In Eq. (8) of the main paper, we employ a task-specific perceptual distortion loss, $\mathcal{D}$, to optimize our adapters for downstream machine vision tasks. A key advantage of this approach is that it enables end-to-end training of the task-specific modules without requiring access to ground-truth task labels (*e.g.*, bounding boxes or segmentation masks).

Our implementation strictly follows the established setup in [5, 15]. Specifically, we utilize a frozen, pre-trained recognition model, $\mathcal{G}$, as a feature extractor to measure the distance between the original image, $\mathbf{x}$, and the reconstructed image, $\hat{\mathbf{x}}$, in the feature space. The choice of $\mathcal{G}$ depends on the downstream task:

- Classification: ResNet50 [10]
- Object Detection: Faster R-CNN [21] (with a ResNet50-FPN backbone)
- Instance Segmentation: Mask R-CNN [11] (with a ResNet50-FPN backbone)

D.1. Loss for Classification Task

For the classification task, the perceptual distortion is evaluated by computing the Mean Squared Error (MSE) between feature maps extracted from different stages of the ResNet50 backbone, as shown in Fig. 6. We extract the outputs from its four main stages (denoted as F_1,F_2,F_3,F_4). The loss is defined as Eq. (9):

$$\mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}; \mathcal{G}) = \frac{1}{4} \sum_{j=1}^{4} \text{MSE}\left(F_j(\mathrm{x}), F_j(\hat{\mathrm{x}})\right) \tag{9}$$

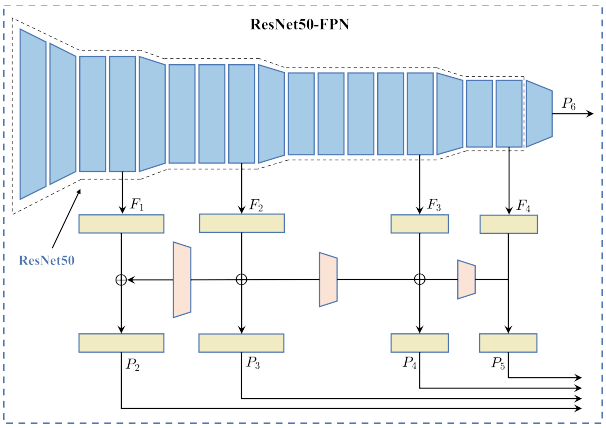

Figure 6. ResNet50-based FPN architecture, indicating the features selected for perceptual loss evaluation.

where $F_j(\cdot)$ denotes the feature map from the j-th stage of the model $\mathcal{G}$, ResNet50.

D.2. Loss for Detection and Segmentation Tasks

For dense prediction tasks, object detection, and instance segmentation, the perceptual distortion is evaluated by computing the MSE between feature maps from the Feature Pyramid Network (FPN) of the respective downstream model. We extract the five output levels of the FPN (denoted as P_2,P_3,P_4,P_5, P_6). The loss is defined as Eq. (10):

$$\mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}; \mathcal{G}) = \frac{1}{5} \sum_{j=2}^{6} \text{MSE}\left(P_j(\mathbf{x}), P_j(\hat{\mathbf{x}})\right) \tag{10}$$

where $P_j(\cdot)$ denotes the feature map from the j-th pyramid level of the FPN. This method effectively measures the fidelity of the multi-scale features that are critical for these downstream tasks.

E. Experimental Settings of Training

E.1. Experimental Equipment

All experiments were conducted on a single NVIDIA L40 (48GB) or GeForce RTX 4090 (24GB) GPU. Our implementation is based on PyTorch v2.5, and all models were trained using CUDA 12.4.

E.2. Hyperparameter Settings

Tab. 2 summarizes the key hyperparameters used for training across all machine vision tasks, namely classification, object detection, and instance segmentation.

F. More Main Experimental Results

As referenced in the main paper, we provide a full quantitative comparison for object detection and instance segmentation tasks in Tab. 3. The experiments are conducted

Table 2. Training hyperparameter settings for the main trials.

	Classification	Detection	Segmentation
Optimizer	Adam	Adam	Adam
Batch size	16	8	8
Epochs	8	40	40
Base learning rate	1e-4	1e-4	1e-4
Lu2022-TIC			
Trade-off term λ	[2.5, 3.5, 5, 6.7, 13]	[0.5, 0.9, 1.8, 3.2]	[0.35, 0.5, 0.9, 1.8, 3.2]
Cheng2020-anchor			
Trade-off term λ	-	[0.5, 0.9, 1.8, 3.2, 5.0]	[0.9, 1.8, 3.2, 5.0]
DCAE & ELIC			
Trade-off term λ	-	[0.5, 0.9, 1.8, 3.2]	[0.5, 0.9, 1.8, 3.2]

on the Cheng2020-anchor framework, with the evaluation metrics including BD-Rate savings and BD-mAP gains [4]. Notably, we have included an additional entry, Ours ($C' = 4, r = 32$), to demonstrate the performance of a more compact version of the S^2-CoT. These detailed results provide a comprehensive view of our method's performance and its competitive standing against other methods.

G. Detailed Ablation Studies

G.1. Detailed Ablation on Adapt-ICMH and SCA

To further investigate whether the performance gains of S^2-CoT stem from parameter capacity or architectural design, we provide an extended comparison with scaled variants of Adapt-ICMH [15]. As shown in Tab. 4, our method (d) outperforms all Adapt-ICMH variants across both tasks. Notably, even when Adapt-ICMH is scaled to a higher parameter count (*e.g.*, 0.64M in method (e)), it still falls short of S^2-CoT (0.42M). Conversely, while the lightweight Adapt-ICMH-64 (g) reduces overhead, its performance gap widens significantly. These results confirm that the efficacy of S^2-CoT is not a byproduct of parameter scaling. Instead, the gains are fundamentally driven by our SFA design, particularly the dynamic soft fusion mechanism and the role-matched synergy between SFA and SCA, which together confirm that structural optimization is more critical for performance than merely expanding the parameter budget.

G.2. Detailed Ablation Study on SCA Placement

To determine the optimal placement of the SCA within the hyperprior entropy model, we conducted a series of ablation studies trained for 18 epochs. We designed multiple experimental schemes, where each scheme involves inserting SCA at different stages of the hyper-encoder h_a and hyper-decoder h_s. The specific insertion placements for each scheme correspond to the positions labeled hp1–hp5 and sp1–sp5 in the architecture shown in Fig. 7. To quantify the performance of each placement, we measured the gain in mAP relative to the Adapt-ICMH [15]. Specifically, for each experimental run, we computed the predicted mAP of Adapt-ICMH at the same bitrate using a fitted polynomial Rate-mAP curve. The final gain, ΔmAP, is then calculated as the difference between our method's achieved mAP and this predicted mAP value. A higher positive value indicates a more significant performance improvement attributable to the SCA's placement.

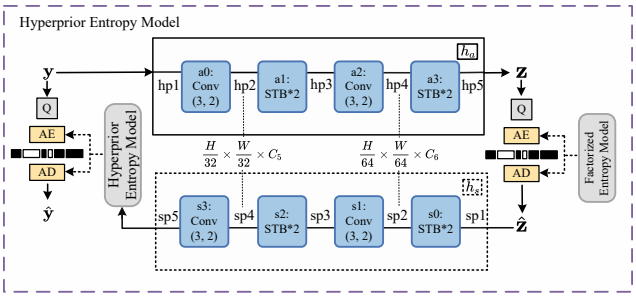

Figure 7. Architecture of the hyperprior entropy model with location markers. The figure illustrates the composition of the hyper-encoder h_a and hyper-decoder h_s.

The experimental results are summarized in Tab. 5. Although placing the adapter in the reverse bottleneck (hp1, sp5) or asymmetric layers (hp3, hp5, sp3, sp5) yields promising performance, these configurations typically come with increased parameter overhead. In contrast, inserting adapters into symmetric middle layers achieves a better trade-off between efficiency and performance, demonstrating superior overall effectiveness. This suggests that for the LIC architecture, the features at the intermediate middle spatial resolution retain a more optimal balance of semantic information and channel detail for the adapter to effectively operate on.

The most compressed high layer features at the classic bottleneck, while semantically rich, may have discarded spatial and channel cues that are crucial for the adapter's refinement process in the model. The superior performance at hp3 and sp4, coupled with its minimal parameter footprint, validates the selection of the symmetric middle layer insertion strategy for our final architecture.

G.3. Detailed Hyperparameters Analysis

This subsection supplements the main paper by providing a comprehensive analysis of the key hyperparameters governing our adapters' capacity: the Structural Fidelity Adapter's (SFA) middle dimension, C', and the Semantic Context Adapter's (SCA) reduction ratio, r. C' controls the SFA's representational power for structural features, while r ad-

Table 3. Extended quantitative results for object detection and instance segmentation tasks based on the *Cheng2020-anchor* base codec. This table provides the full results that are referenced in the main paper. Best results are in **bold**, second-best are underlined.

Base	Method	Venue	Object Detection		Instance Segmentation		Trainable Params↓ (M)
			BD-Rate↓	BD-mAP↑	BD-Rate↓	BD-mAP↑	
Cheng2020 -anchor	full fine-tuning	–	-59.015%	4.699	-74.052%	3.869	26.60 (100.00%)
	Channel Selection	ICPR'22	-11.520%	0.723	-5.156%	0.230	1.34 (5.04%)
	ICMH-Net	ACM MM'23	-8.365%	0.503	-11.980%	0.737	4.43 (16.65%)
	Adapt-ICMH	ECCV'24	-49.245%	3.121	-60.657%	3.389	0.41 (1.54%)
	SVD-LoRA	CVPR'25	-36.354%	2.331	-38.087%	1.899	**0.14** (0.53%)
	Ours (C'=4, r=32)	–	-56.337%	3.784	-52.609%	3.305	0.40 (1.50%)
	Ours	–	**-61.578%**	**4.153**	**-63.607%**	**3.826**	0.74 (2.78%)

Table 4. Ablation study of Adapt-ICMH [15] and SCA synergy (extension of Tab. 5 in the main paper).

Method	SFA	SCA		Object Detection		Instance Segmentation		Trainable Params↓ (M)
		h_a	h_s	BD-Rate↓	BD-mAP↑	BD-Rate↓	BD-mAP↑	
(d) Ours	✓	✓	✓	**-60.824%**	**4.014**	**-61.784%**	**3.480**	0.42 (5.59%)
(e)	Adapt-ICMH-128	✓	✓	-58.399%	3.709	-60.936%	3.413	0.64(8.52%)
(f)	Adapt-ICMH-96	✓	✓	-59.811%	3.665	-59.309%	3.341	0.48(6.39%)
(g)	Adapt-ICMH-64	✓	✓	-57.894%	3.894	-57.851%	3.356	**0.30** (3.99%)

Table 5. Ablation study on the placement of SCA within the entropy model. Gain (ΔmAP) is measured as the difference in mAP between our method's result and the value predicted by the Adapt-ICMH Rate-mAP curve at the same bitrate.

Placement in h_a, h_s	Core Idea	ΔmAP ↑	Params↓
hp1, sp5	reverse bottleneck	+0.459	0.497
hp5, sp1	classic bottleneck	-0.496	0.421
hp3, sp4 or hp3, sp3	symmetrical middle layer	+0.479	0.421
hp2, hp4, sp2, sp4	symmetrical middle high layer	+0.242	0.438
hp3, hp5, sp1, sp3	symmetrical middle high layer	+0.343	0.438
hp3, hp5, sp3, sp5	asymmetric layer	+0.467	0.504

justs the SCA's bottleneck for context modeling.

The complete results of the controlled ablation study are presented in Tab. 6. The data reveals a clear trade-off between performance and parameter complexity across various settings. The configuration selected for our main experiments (C'=64, r=8) was determined from this analysis to provide the optimal balance between task accuracy and efficiency, noting their coupled influence.

H. Generalization to YOLO Detectors

At a compression ratio of 296:1, S^2-CoT maintains robust performance (*e.g.*, 47.760 mAP on YOLOv11m), narrowing the gap to the unconstrained original image upper bound presented in Tab. 7. Meanwhile, we also evaluate S^2-CoT across multiple downstream detectors, including Faster-RCNN, YOLOv11m [13], and YOLO26x [22]. The

Table 6. Comprehensive ablation study on the SFA middle dimension C' and the SCA reduction ratio r.

C'	r	Detection		Segmentation		Params ↓(M)
		BD-Rate↓	BD-mAP↑	BD-Rate↓	BD-mAP↑	
32	32	-60.276%	3.852	-60.255%	3.386	0.28 (3.73%)
32	16	-57.622%	3.875	-61.991%	3.462	0.29 (3.86%)
64	8	-60.824%	4.014	-61.784%	3.480	0.42 (5.59%)
96	4	-61.329%	4.063	-62.429%	3.529	0.57 (7.59%)
128	4	-62.763%	4.165	-63.657%	3.625	0.72 (9.59%)

Table 7. Performance comparison. Top: compression ratio and mAP on original images. Bottom: performance on three detectors.

	Original Image	LIC		Traditional
		Base Codec	**Ours**	JPEG(Q10)
Compression Ratio	None	219:1	**296:1**	67:1
mAP↑	51.302	45.231	**47.760**	15.481

Methods	Base Codec		**Ours**	
	bpp↓	mAP↑	bpp↓	mAP↑
Faster-RCNN	0.1096	31.606	**0.0812**	**36.601**
YOLOv11m	0.1096	45.231	**0.0812**	**47.760**
YOLO26x	0.1096	49.972	**0.0812**	**52.410**

Table 8. More Detailed Comparison on computational complexity.

Model	KMACs/pixel		CPU Average Latency (ms)		GPU Average Latency (ms)		Trainable Params ↓ (M)	BD-Acc ↑
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.		
base codec	132.64	176.42	94.87	94.75	33.22	31.18	-	-
full fine-tuning	132.64	176.42	94.87	94.75	33.22	31.18	7.51	17.7
Adapt-ICMH[15]	147.39	191.16	111.01	110.14	29.49	25.34	0.29	16.9
Ours	153.57	197.34	109.49	101.03	29.60	28.45	0.42	17.4

results demonstrate consistent mAP gains while reducing the bpp from 0.1096 to 0.0812, confirming that our method provides a detector-agnostic and highly efficient solution for machine vision compression.

I. Computational Complexity and Efficiency

Complementing the efficiency analysis in the main paper, we provide a comprehensive breakdown of computational complexity and inference latency in Tab. 8. Acknowledging the hardware-dependent nature of runtime performance, we evaluate the average encoding and decoding latency on both an INTEL(R) XEON(R) SILVER 4510 CPU and an NVIDIA L40 GPU. The results verify that our proposed framework maintains competitive inference speeds with negligible computational overhead (in KMACs/pixel) compared to full fine-tuning, validating its practical efficiency alongside superior parameter reduction.

J. Integration Details of Adapters

To demonstrate the architectural universality and seamless plug-and-play capability of our proposed framework, we detail the integration of SFA and SCA into four distinct, state-of-the-art learned image compression baselines. These codecs represent a diverse spectrum of architectural paradigms, encompassing both Transformer-based models (*Lu2022-TIC* [19] and *DCAE* [18]) and CNN-based models (*Cheng2020-anchor* [6] and *ELIC* [9]). Crucially, they feature highly heterogeneous entropy modeling mechanisms, ranging from standard hyperpriors and spatial autoregression to sophisticated dictionary-based cross-attention and channel-conditional backward-adaptive models, each tailored to distinct compression scenarios.

As illustrated in Fig. 8, Fig. 9, Fig. 10, and Fig. 11, our dual-adapter framework maintains a consistent integration strategy despite the structural variations of the base codecs. The SFA modules are strategically interleaved within the multiscale stages of the encoder-decoder backbones (g_a, g_s) to preserve high-fidelity structural information in the feature domain. Concurrently, the SCA modules are embedded within the symmetrical intermediate layers of the entropy models (h_a, h_s). This placement allows the SCA to refine the semantic context effectively, regardless of whether the underlying probability estimation relies on local spatial neighbors, channel contexts, or global dictionary priors. This uniform applicability across diverse backbones and statistical models strongly validates the robustness of our structure-semantics synergy.

K. More Results on the Diverse Base Codec

To further substantiate the scalability and architectural universality of our S^2-CoT framework, we extend our evaluation to two advanced, high-performance base codecs: *DCAE* [18], a large-scale Transformer-based model utilizing dictionary-based cross-attention, and *ELIC* [9], a sophisticated CNN-based model featuring unevenly grouped space-channel contextual coding. As presented in Tab. 9, our method demonstrates exceptional robustness when applied to these complex architectures. Specifically, on the *DCAE* codec, S^2-CoT outperforms existing PEFT methods, such as TransTIC and Adapt-ICMH, across both object detection and instance segmentation tasks. Notably, as visualized in Fig. 12 (a), our method establishes a rate-accuracy frontier that closely parallels the theoretical upper bound of full fine-tuning, all while adding only a negligible fraction of trainable parameters relative to the large-scale backbone.

Similarly, the experiments on the *ELIC* codec validate the superior efficacy of our structure-semantics synergy. As shown in the bottom part of Tab. 9 and Fig. 12 (b), S^2-CoT consistently dominates the performance landscape, delivering superior coding efficiency and detection accuracy compared to recent state-of-the-art adaptations like SVD-LoRA and Adapt-ICMH. These results collectively confirm that our synergistic adaptation strategy is model-agnostic, effectively mitigating the domain gap for machine vision tasks even when integrated into highly optimized, state-of-the-art compression baselines.

L. Quantitative Results on Classification Task

This section provides detailed quantitative data on classification task for the generalization experiment on the *Lu2022-TIC* codec, supplementing the rate-accuracy curves in the main paper.

The BD-metrics in Tab. 10 quantify the performance of different Parameter-Efficient Fine-Tuning (PEFT) ways.

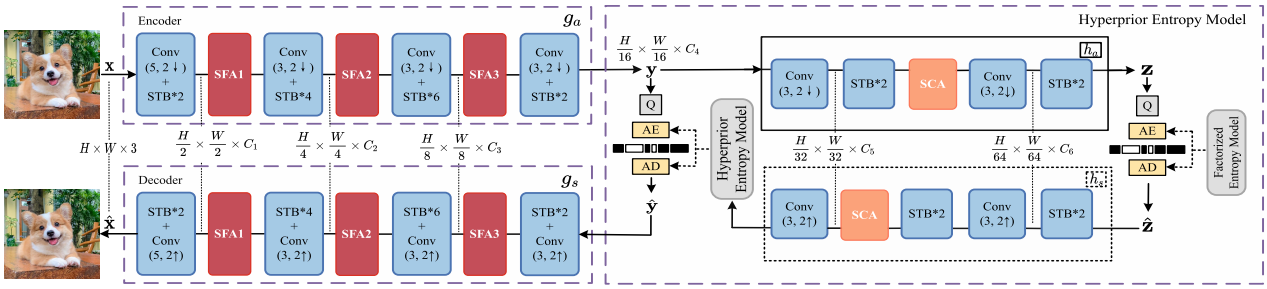

Figure 8. Our SFA and SCA are integrated into the *Lu2022-TIC* codec [19]. STB denotes the Swin-Transformer Block. Conv(n, 2↑) denotes a transposed convolution (kernel size=n) with a stride of 2 for upsampling, and Conv(n, 2↓) denotes a convolution (kernel size=n) with a stride of 2 for downsampling.

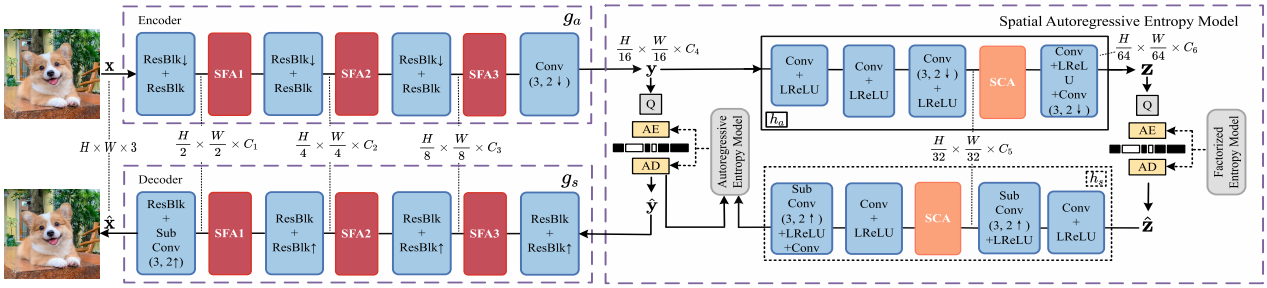

Figure 9. Our SFA and SCA are integrated into the *Cheng2020-anchor* codec [6]. ResBlk denotes a residual block, where ResBlk↓ indicates a downsampling residual block with a stride of 2. Conv represents a standard 3×3 convolution. LReLU denotes Leaky ReLU. Sub Conv(3, 2↑) denotes a sub-pixel convolution (kernel size=3) with a stride of 2 for upsampling.

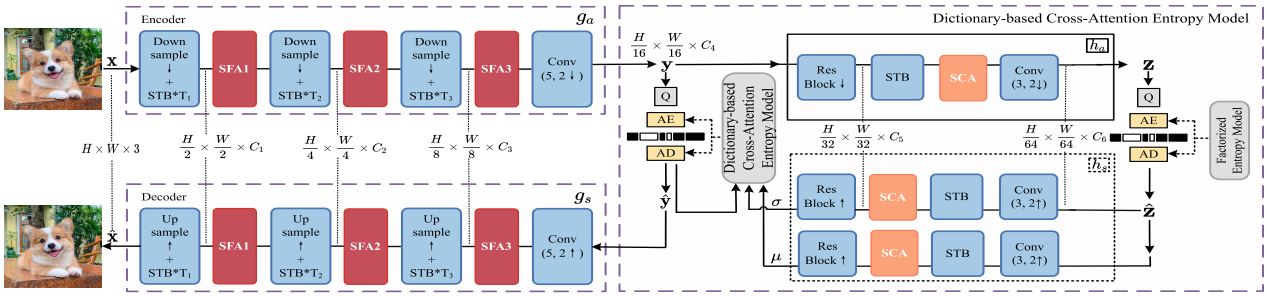

Figure 10. Our SFA and SCA are integrated into the *DCAE* codec [18]. Downsample denotes the ResidualBottleneckBlockWithStride blocks, where STB indicates the SwinBlockWithConvMulti blocks. Conv(5, 2↓) represents a 5×5 convolution with a stride of 2. Upsample denotes the ResidualBottleneckBlockWithUpsample blocks. Conv(5, 2↑) represents a 5×5 convolution with a stride of 2 for upsampling.

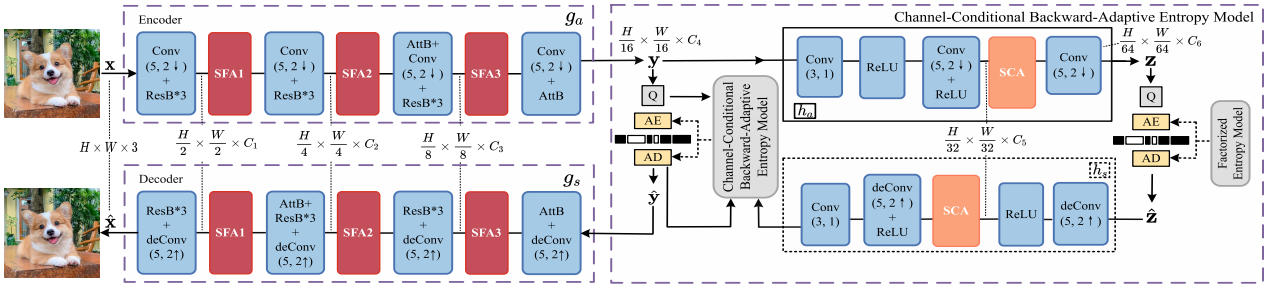

Figure 11. Our SFA and SCA are integrated into the *ELIC* codec [9]. Conv(5, 2↓) represents a 5×5 convolution with a stride of 2. ResB denotes residual blocks, and AttB denotes attention blocks.. Conv(3, 1) represents a standard 3×3 convolution with a padding of 1. The deConv(5, 2↑) denotes a 5×5 transposed convolution with a stride of 2, serving as a learnable spatial upsampling operation.

Table 9. Quantitative comparison for object detection and instance segmentation tasks based on the *DCAE* and *ELIC* codec. Our method is competitive with full fine-tuning against others. Our method demonstrates superior BD-metric performance and parameter efficiency.

Base	Method	Venue	Object Detection		Instance Segmentation		Trainable Params↓ (M)
			BD-Rate↓	BD-mAP↑	BD-Rate↓	BD-mAP↑	
DCAE	full fine-tuning	–	-84.677%	4.002	-81.897%	3.049	119.40 (100.00%)
	Channel Selection	ICPR'22	-24.498%	0.977	-18.661%	0.747	2.68 (2.24%)
	TransTIC	ICCV'23	-49.269%	2.339	-50.120%	2.271	1.13 (0.95%)
	Adapt-ICMH	ECCV'24	-64.608%	2.863	-63.240%	2.404	0.36 (0.30%)
	SVD-LoRA	CVPR'25	-58.166%	2.300	-54.371%	2.229	**0.17** (0.14%)
	Ours	–	**-68.882%**	**3.361**	**-68.363%**	**2.743**	0.42 (0.35%)
ELIC	full fine-tuning	–	-70.409%	5.745	-70.153%	4.158	33.79 (100.00%)
	Channel Selection	ICPR'22	-23.235%	1.518	-11.041%	0.591	2.68 (7.93%)
	Adapt-ICMH	ECCV'24	-54.844%	3.379	-51.173%	2.861	**0.41** (1.21%)
	SVD-LoRA	CVPR'25	-41.190%	2.149	-42.721%	1.948	**0.41** (1.21%)
	Ours	–	**-60.046%**	**4.051**	**-59.214%**	**3.174**	0.74 (2.19%)

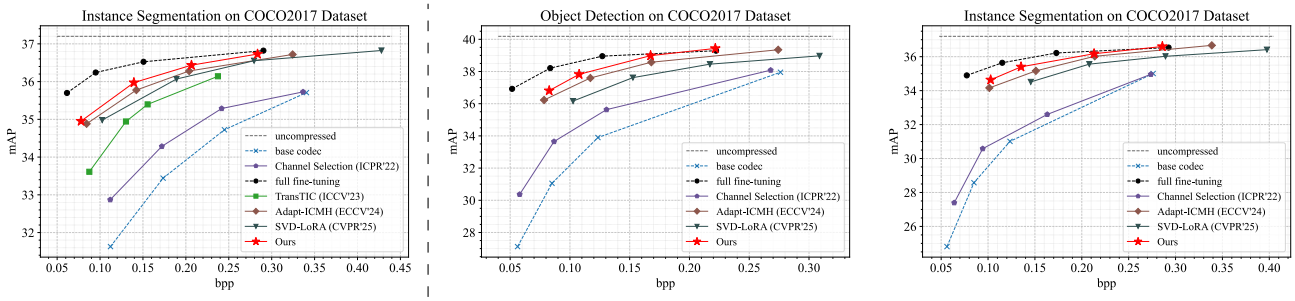

Figure 12. Comparison of rate-accuracy performance across various tasks and base codecs. (a) The instance segmentation results on the *DCAE* base codec. (b) Object detection (left) and instance segmentation (right) performance on the *ELIC* base codec.

The data confirms that our method significantly outperforms other PEFT approaches in both BD-Rate savings and BD-Acc gains. This validates the broad applicability of our method, showing that the synergistic strategy can be successfully extended to varying tasks beyond object detection, such as classification, without compromising efficiency.

Table 10. Quantitative comparison of PEFT methods on the *Lu2022-TIC* codec for the classification task. Our method demonstrates superior BD-metric performance and parameter efficiency.

Method	Classification		Params
	BD-Rate↓	BD-Acc↑	↓(M)
full fine-tuning	-	17.688	7.51 (100%)
Channel Selection	-37.178%	6.278	0.92 (12.25%)
TransTIC	-58.529%	9.956	1.62 (21.57%)
ICMH-Net	-18.759%	3.360	3.98 (53.00%)
Adapt-ICMH	-88.573%	16.901	0.29 (3.86%)
SVD-LoRA	-50.162%	7.920	0.09 (1.20%)
Ours	**-93.453%**	**17.351**	0.42 (5.59%)

M. Sensitivity & Necessity of Co-Tuning

We investigate deploying SCA in isolation, and Tab. 11 shows its solitary application causes counter-intuitive performance deterioration, highlighting the entropy model's extreme hypersensitivity to uncoordinated adjustments. By synchronizing structural and statistical adaptation, our S^2-CoT framework converts this degradation into synergistic gains, achieving SOTA across diverse base codecs.

Table 11. Isolated SCA reveals the entropy model's inherent hypersensitivity, where even minimal parameter addition leads to performance collapse, while our S^2-CoT reverses this trend.

Method	Object Detection		Trainable
Lu2022-TIC	BD-Rate ↓	BD-mAP ↑	Params ↓ (M)
+SFA	-58.077%	3.842	0.40 (5.33%)
+SCA	+8.160%	-0.511	0.02 (0.27%)
+SFA +SCA	-60.824%	4.014	0.42 (5.59%)

N. Framework Paradigm and Modularity

The proposed S^2-CoT framework introduces two types of lightweight adapters to efficiently regulate key components of the transform module and the entropy model. Specifically, the SFA harmonizes spatial and frequency representations for flexible feature modulation, while the SCA focuses on calibrating channel-wise statistics in the entropy model to facilitate precise probability estimation. From an architectural perspective, S^2-CoT establishes a structured, efficient fine-tuning scheme. Benefiting from a core design that decouples the regulation mechanism from the backbone network, the proposed adapters essentially function as interchangeable plug-ins. This implies that the framework is not dependent on specific component implementations: the modules used to regulate these two key structures can be substituted by analogous architectures, rather than being confined to the designs presented herein. Simultaneously, this modularity enables S^2-CoT to be seamlessly integrated into diverse LIC architectures.

O. PyTorch Implementation of SFA and SCA

To provide a deeper understanding of our S^2-CoT framework, we detail the PyTorch implementation of its two core components: SFA and SCA. These adapters, designed to achieve a crucial structure–semantics synergy, are implemented as lightweight and plug-and-play network blocks. The full, runnable source code for the S^2-CoT can be found in the code folder of the supplementary material.

```

class SFA(nn.Module):
"""
Spatial-Frequency Structural Fidelity Adapter: Feature Enhancement via Three Stages:
1. Channel Excitation and Bottleneck Projection.
2. Spatial-Frequency Dual-Branch Modulation.
3. Soft Fusion.
"""

def __init__(self, in_dim: int = 128, middle_dim: int = 64, r: int = 16, se_factor: float = 1.0, adapt_factor: float = 1.0):
super().__init__()
self.adapt_factor = adapt_factor

# Channel
self.c_squeeze = nn.AdaptiveAvgPool2d(1)
self.c_excite = nn.Sequential(
nn.Conv2d(in_dim, in_dim // r, 1, bias=False),
nn.ReLU(),
nn.Conv2d(in_dim // r, in_dim, 1, bias=False),
nn.Sigmoid()
)
self.down1 = nn.Conv2d(in_dim, middle_dim, kernel_size=1)
self.se_alpha = nn.Parameter(torch.tensor(se_factor))

# Spatial branch
self.s_gate = nn.Conv2d(in_dim, middle_dim, kernel_size=1)
self.s_dw3 = nn.Conv2d(middle_dim, middle_dim, kernel_size=5, padding=2, groups=middle_dim)
self.s_up_proj = nn.Conv2d(middle_dim, in_dim, kernel_size=1)

# Frequency branch
self.f_dw_conv = nn.Conv2d(middle_dim, middle_dim, kernel_size=3, padding=1, groups=middle_dim)
self.amplitude_mlp = nn.Conv2d(middle_dim, middle_dim, kernel_size=1)
self.f_up_proj = nn.Conv2d(middle_dim, in_dim, kernel_size=1)
self.amplitude_gelu = nn.GELU()
self.ifft_relu = nn.ReLU()
self.sigmoid = nn.Sigmoid()

# Shared
self.sf_dw3 = nn.Conv2d(in_dim, in_dim, kernel_size=3, padding=1, groups=in_dim, bias=False)
self.sf_relu = nn.ReLU()

# Fusion
self.sf_mlp = nn.Sequential(
nn.Conv2d(in_dim * 2, in_dim // 2, kernel_size=1, bias=False),
nn.ReLU(),
nn.Conv2d(in_dim // 2, in_dim, kernel_size=1, bias=False)
)

def forward(self, x: torch.Tensor) -> torch.Tensor:
_, _, H, W = x.shape

# Channel
c_weights = self.c_squeeze(x)
c_weights = self.c_excite(c_weights)
c = c_weights * x * self.se_factor
down = self.down1(x + c)

# Spatial branch
```

```
        gated = self.s_dw3(down) *
            self.s_gate(x)
        s = self.s_up_proj(torch.relu(gated))

        # Frequency branch
        fft = torch.fft.rfft2(down, dim=(2,
            3), norm='backward')
        amplitude, phase = torch.abs(fft),
            torch.angle(fft)
        amplitude = self.amplitude_mlp
        (self.amplitude_gelu(
        self.f_dw_conv(amplitude)))
        modulated = amplitude *
            self.sigmoid(amplitude)
        complex = torch.complex(modulated *
            torch.cos(phase), modulated *
            torch.sin(phase))
        ifft = torch.fft.irfft2(complex, s=(H,
            W), norm='backward')
        f =
            self.f_up_proj(self.ifft_relu(ifft))

        # Shared
        s = self.sf_relu(self.sf_dw3(s))
        f = self.sf_relu(self.sf_dw3(f))

        # Fusion
        fused = self.sf_mlp(torch.cat([s, f],
            dim=1))

        return x + c + fused *
            self.adapt_factor
```

```

    class SCA(nn.Module):
    """
    Semantic Context Adapter (SCA).
    """

    def __init__(self, channels: int,
        reduction_ratio: int = 8,
        adapt_factor: float = 1.0):
    super().__init__()
    self.adapt_factor = adapt_factor
    hidden_dim = max(4, channels //
        reduction_ratio)
    self.refine = nn.Sequential(
    nn.Conv2d(channels, hidden_dim, 1,
        bias=False),
    nn.ReLU(),
    nn.Conv2d(hidden_dim, channels, 1,
        bias=False)
    )
    self.se_inter = nn.Sequential(
    nn.AdaptiveAvgPool2d(1),
    nn.Conv2d(channels, hidden_dim, 1),
    nn.ReLU(),
    nn.Conv2d(hidden_dim, channels, 1),
    nn.Sigmoid()
    )

    def forward(self, x: torch.Tensor) ->
        torch.Tensor:
    refine = x + self.refine(x)
    se = self.se_inter(refine)
    return se * refine * self.adapt_factor
```

P. Future Work

The proposed S^2-CoT achieves state-of-the-art performance across multiple datasets and backbones, pushing the boundaries of task-oriented image compression with acceptable overhead. Given the constraints of computational resources and parameter size on practical deployment, our future work will focus on exploring lighter compression frameworks. We aim to optimize the trade-off between performance and efficiency by investigating diverse adapter structures and utilizing partial codec components. Furthermore, we plan to validate our method on a broader spectrum of downstream tasks and extend the current single-task adaptation paradigm to multi-task learning scenarios, thereby reducing training costs and enhancing generalization capabilities.

Q. More Qualitative Results

To complement the visual comparisons presented in the main paper, we provide additional qualitative results for the S^2-CoT framework in this section. As shown in the Fig. 13 - Fig. 18, our method consistently achieves superior visual fidelity, demonstrating enhanced structural integrity and fewer spurious high-frequencies compared to competing methods, even at low bitrates. These results further substantiate the claim that S^2-CoT effectively preserves the critical semantic information necessary for both human perception and downstream machine vision tasks.

R. Generalization of Synergy to Advanced Entropy Models Across Diverse Base Codecs

While Sec. C establishes the theoretical foundation using a standard Gaussian hyperprior model, state-of-the-art learned image compression frameworks employ significantly more sophisticated entropy modeling mechanisms to capture complex dependencies. In this section, we mathematically demonstrate that the proposed S^2-CoT paradigm is not limited to basic architectures but establishes a unified synergistic mechanism across diverse entropy models.

By mathematically decomposing the probability estimation processes of *Cheng2020-anchor* (Autoregressive GMM), *DCAE* (Dictionary-based Prior) and *ELIC* (Spatial-Channel Context), we show that the hyperprior feature ψ serves as the fundamental anchor for probability parameterization in all cases. Consequently, our SCA, by refining

this shared context, effectively propagates synergistic corrections through these complex probability chains.

R.1. Synergy in Autoregressive GMMs

Sophisticated codecs like *Cheng2020-anchor* [6] enhance coding efficiency by combining channel-wise autoregression with Gaussian Mixture Models (GMM). The conditional probability for a latent $\hat{y}_i$ is formulated as:

$$p_{\hat{y}_i|\hat{\mathbf{z}},\hat{\mathbf{y}}_{<i}}(\hat{y}_i|\cdot) = \sum_{k=1}^{K} w_i^{(k)} \mathcal{N}\left(\mu_i^{(k)}, {\sigma_i^{(k)}}^2\right) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right), \tag{11}$$

where the mixture parameters are jointly predicted by an entropy parameter network g_{ep}:

$$\left(w_i^{(k)}, \mu_i^{(k)}, \sigma_i^{(k)}\right)_{k=1}^{K} = g_{ep}(\boldsymbol{\psi}, \boldsymbol{\phi}_i; \boldsymbol{\theta}_{ep}). \tag{12}$$

Here, $\boldsymbol{\phi}_i$ represents the autoregressive context from previous latents $\hat{y}_{<i}$, and $\boldsymbol{\psi} = h_s(\hat{\mathbf{z}}; \boldsymbol{\theta}_{h_s})$ is the hyperprior feature derived from the hyper-decoder.

Synergy Mechanism: Even with the introduction of temporal context $\boldsymbol{\phi}_i$, the GMM parameters remain functionally dependent on the hyperprior feature $\boldsymbol{\psi}$. By inserting the SCA into the hyperprior pathway, our method dynamically calibrates $\boldsymbol{\psi}$ to $\boldsymbol{\psi}' = \text{SCA}(\boldsymbol{\psi})$. This calibrated feature acts as a corrected condition for g_{ep}, guiding the GMM to generate an optimal mixture distribution that aligns with the SFA-adapted latent space, thereby minimizing the coding cost $\mathbb{E}[-\log_2 p]$.

R.2. Synergy in Dictionary Entropy Models

The state-of-the-art *DCAE* codec [18] introduces a dictionary-based cross-attention mechanism to capture long-range dependencies. Its bitrate calculation involves a dictionary-based prior $\boldsymbol{\delta}$:

$$R(\hat{\mathbf{y}}) = \sum_{k} \mathbb{E}\left[-\log_2 p\left(\hat{y}^k \mid \boldsymbol{\psi}, \boldsymbol{\phi}^k, \boldsymbol{\delta}^k; \boldsymbol{\theta}_{ep}\right)\right], \tag{13}$$

where the dictionary prior is computed via cross-attention: $\boldsymbol{\delta}^k = \text{Attn}([\boldsymbol{\psi}, \boldsymbol{\phi}^k], \mathbf{D})$.

Synergy Mechanism: In this architecture, the hyperprior $\boldsymbol{\psi}$ serves a dual role: it is a direct condition for the entropy parameters and a query/key component for querying the learned dictionary $\mathbf{D}$. Therefore, the distributional shift analyzed in Sec. C would be exacerbated here, as an uncalibrated $\boldsymbol{\psi}$ would retrieve erroneous priors from the dictionary. Our S^2-CoT framework resolves this by tuning $\boldsymbol{\psi}$ via SCA, ensuring that the dictionary lookup retrieves task-relevant priors. This validates that SCA is not merely a local fix but a fundamental component that restores the operational integrity of advanced attention-based models.

R.3. Context Synergy in Spatial-Channel

Recent models like *ELIC* codec [9] further decouple the latent space into anchor (y_{ac}) and non-anchor (y_{na}) groups to exploit uneven spatial-channel correlations. The total bitrate is decomposed as:

$$R(\hat{\mathbf{y}}) = \sum \left(R_{\hat{y}_{ac}} + R_{\hat{y}_{na}}\right), \tag{14}$$

$$R_{\hat{y}_{ac}} = \mathbb{E}\left[-\log_2 p\left(\hat{y}_{ac} \mid \boldsymbol{\psi}, \boldsymbol{\theta}_{ch}\right)\right], \tag{15}$$

$$R_{\hat{y}_{na}} = \mathbb{E}\left[-\log_2 p\left(\hat{y}_{na} \mid \boldsymbol{\psi}, \boldsymbol{\theta}_{ch}, \boldsymbol{\theta}_{sp}\right)\right]. \tag{16}$$

Here, $\boldsymbol{\theta}_{ch}$ and $\boldsymbol{\theta}_{sp}$ denote the parameters for channel and spatial context transforms, respectively.

Synergy Mechanism: Crucially, both the anchor and non-anchor probability estimates are conditioned on the shared hyperprior context $\boldsymbol{\psi}$. The SCA essentially functions as a global context modulator. By refining $\boldsymbol{\psi}$, the SCA simultaneously optimizes the base probability for anchor latents and the conditional priors for non-anchor latents. This ensures that the complex spatial-channel grouping logic in *ELIC* operates on statistically aligned features, preserving the synergy between the structural transform and the contextual entropy model.

R.4. Unified Perspective

Across all the aforementioned formulations, spanning from simple Gaussian models to complex dictionary-augmented GMMs, a pivotal common structural invariant emerges: the hyperprior pathway provides the core foundational latent-side information $\boldsymbol{\psi}$ that parameterizes the underlying conditional distributions.

Our S^2-CoT framework leverages this invariant. Instead of redesigning adapters for every specific entropy sub-module (*e.g.*, autoregressive heads or attention blocks), we strategically place the SCA to refine the root information $\boldsymbol{\psi}$. This establishes a model-agnostic synergy, where a lightweight modulation at the semantic root effectively propagates statistical alignment throughout the entire entropy modeling hierarchy.

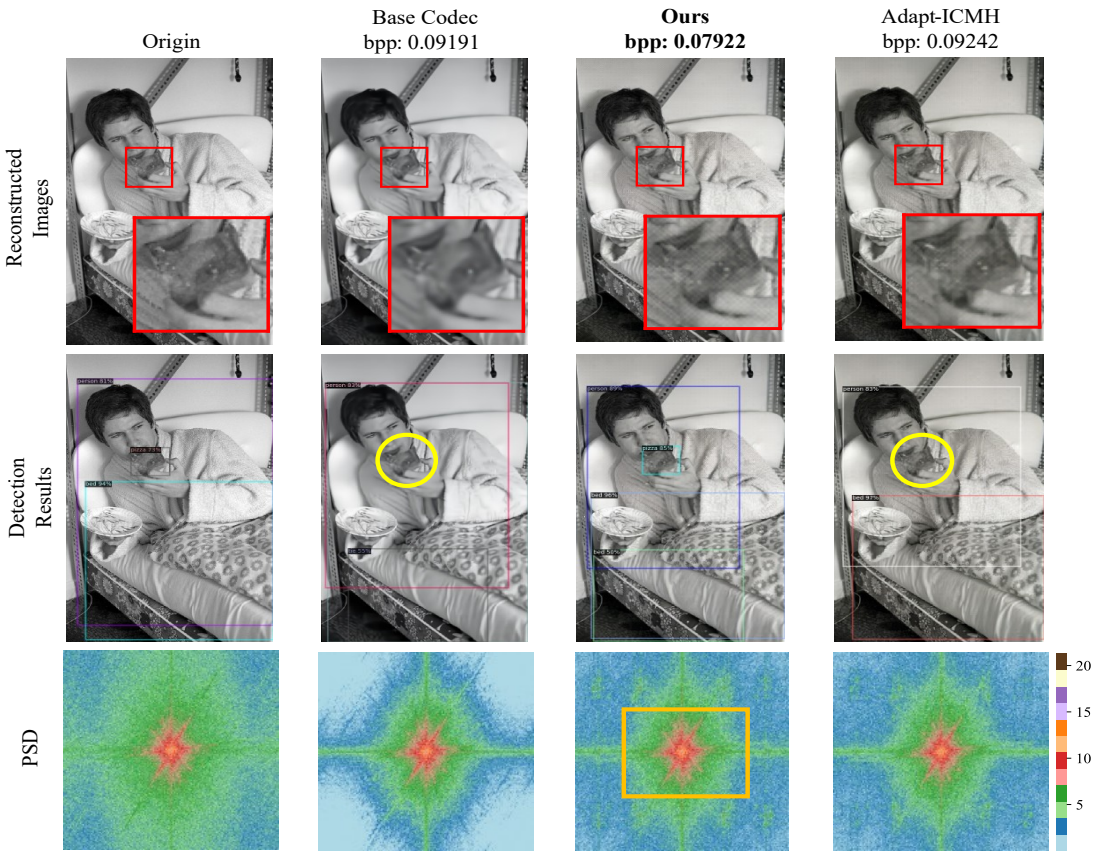

Figure 13. More detection qualitative results.

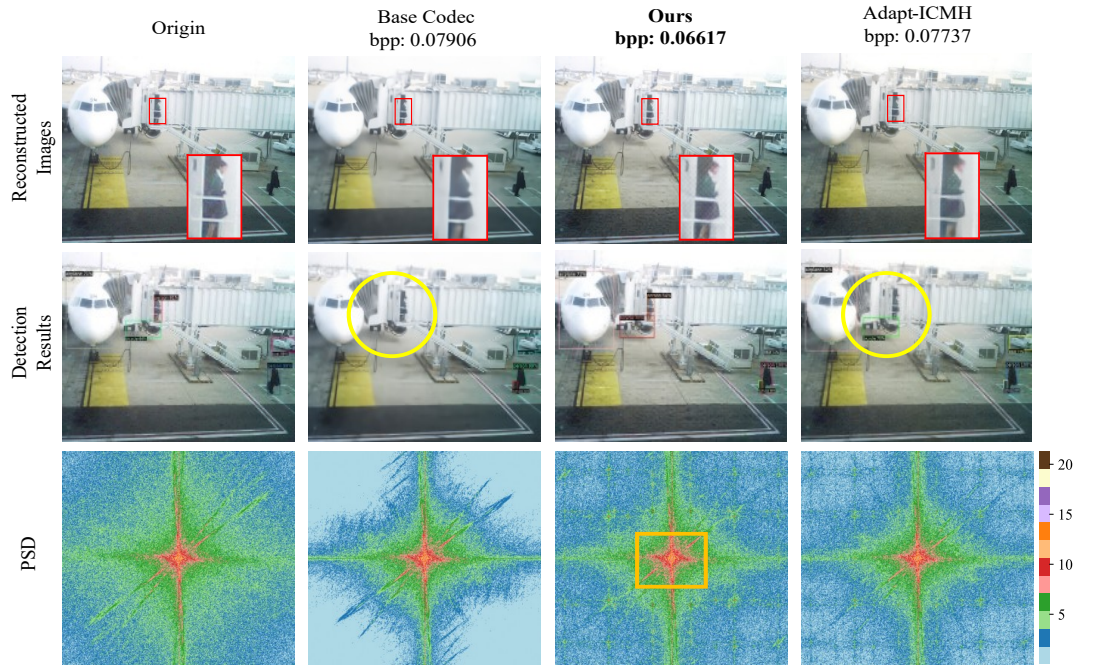

Figure 14. More detection qualitative results.

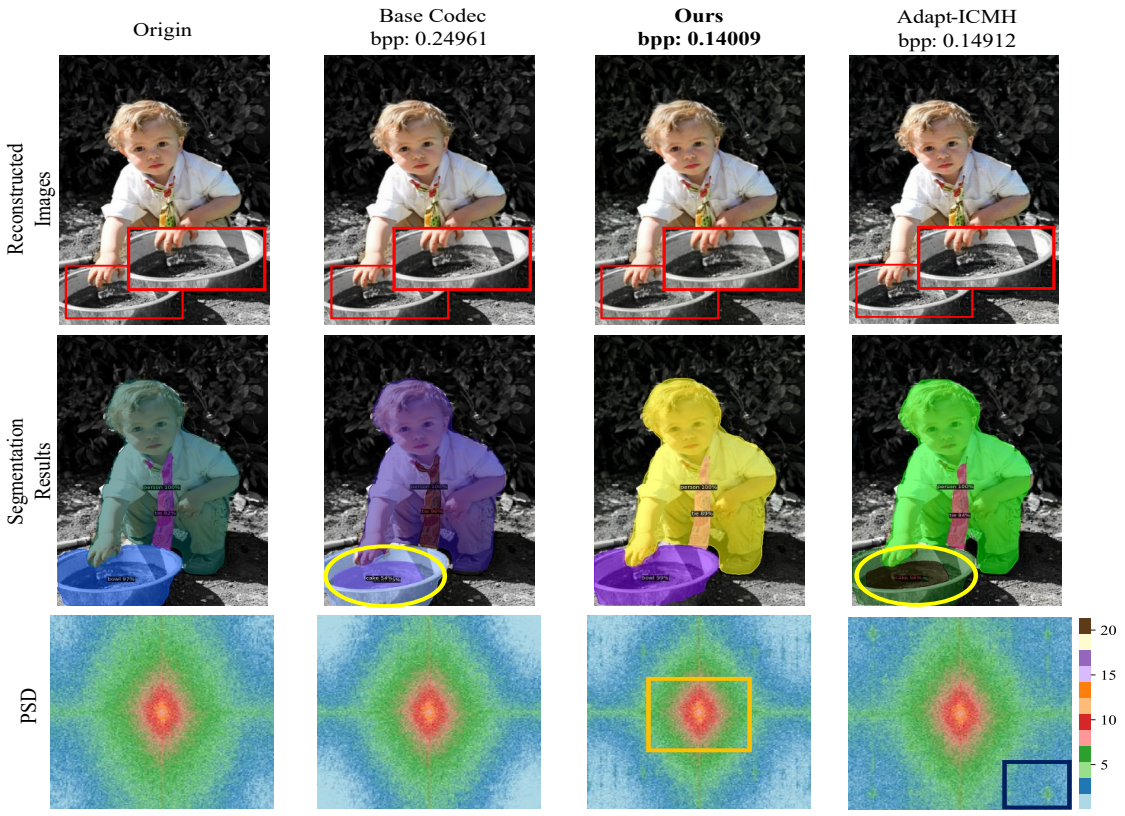

Figure 15. More segmentation qualitative results.

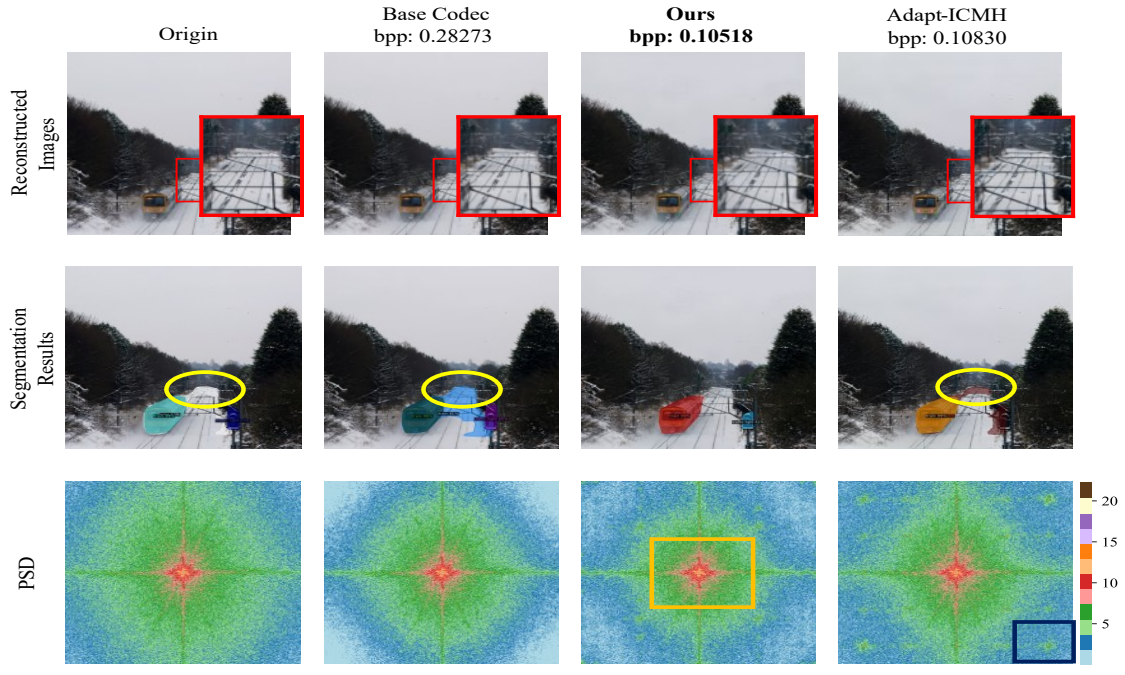

Figure 16. More segmentation qualitative results.

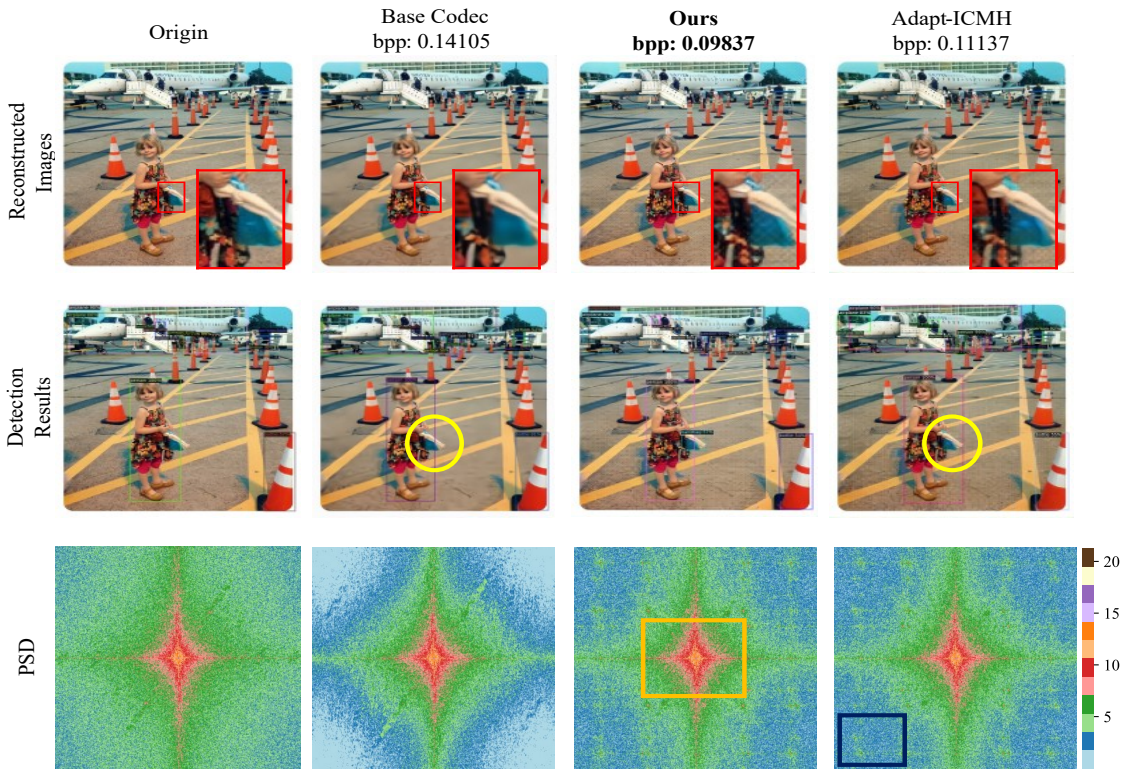

Figure 17. More detection qualitative results.

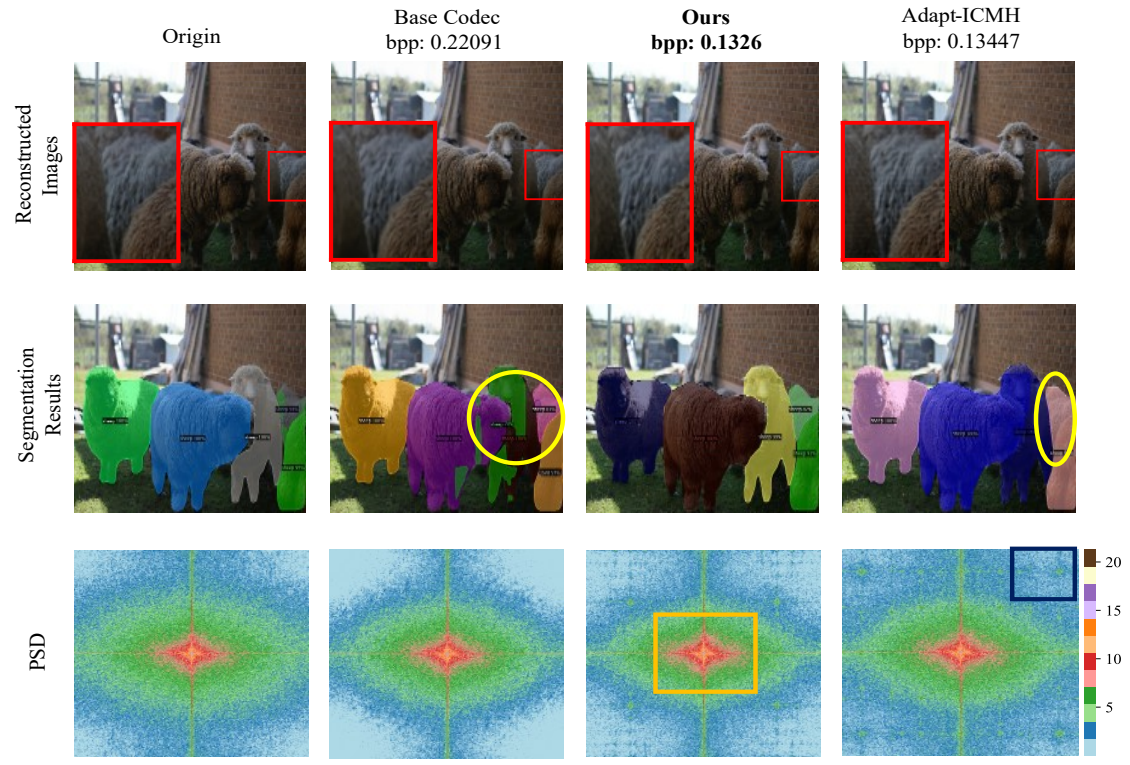

Figure 18. More segmentation qualitative results.

References

[1] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017. 2

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018. 2

[3] Joan Bartrina-Rapesta, Joan Serra-Sagrista, Francesc Auli-Llinas, and Juan Munoz Gomez. Jpeg2000 roi coding method with perfect fine-grain accuracy and lossless recovery. In *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pages 558–562, 2009. 2

[4] Gisle Bjøntegaard. Calculation of average psnr differences between rd-curves. 2001. 6

[5] 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, 2023. 5

[6] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7939–7948, 2020. 2, 8, 9, 13

[7] C. A. Christopoulos, T. Ebrahimi, and A. N. Skodras. Jpeg2000: the new still picture compression standard. In *Proceedings of the 2000 ACM Workshops on Multimedia*, page 45–49, 2000. 2

[8] Minghao Han, Shiyin Jiang, Shengxi Li, Xin Deng, Mai Xu, Ce Zhu, and Shuhang Gu. Causal context adjustment loss for learned image compression. In *Advances in Neural Information Processing Systems*, pages 133231–133253, 2024. 2

[9] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5718–5727, 2022. 8, 9, 13

[10] 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, 2016. 5

[11] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 5

[12] Xiu Ji, Xiao Yang, Zheyu Yue, Hongliu Yang, and Boyang Zheng. Deep learning image compression method based on efficient channel-time attention module. *Scientific Reports*, 15(1):15678, 2025. 2

[13] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024. 7

[14] Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Frequency-aware transformer for learned image compression. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[15] 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. In *European Conference on Computer Vision*, pages 382–399. Springer, 2024. 2, 5, 6, 7, 8

[16] Jinming Liu, Xin Jin, Ruoyu Feng, Zhibo Chen, and Wenjun Zeng. Composable image coding for machine via task-oriented internal adaptor and external prior. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5, 2023. 2

[17] 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, 2023. 2

[18] Jingbo Lu, Leheng Zhang, Xingyu Zhou, Mu Li, Wen Li, and Shuhang Gu. Learned image compression with dictionary-based entropy model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12850–12859, 2025. 8, 9, 13

[19] Ming Lu, Peiyao Guo, Huiqing Shi, Chuntong Cao, and Zhan Ma. Transformer-based image compression. *arXiv preprint arXiv:2111.06707*, 2021. 8, 9

[20] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10794–10803, 2018. 2

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 5

[22] Ranjan Sapkota, Rahul Harsha Cheppally, Ajay Sharda, and Manoj Karkee. Yolo26: key architectural enhancements and performance benchmarking for real-time object detection. *arXiv preprint arXiv:2509.25164*, 2025. 7

[23] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991. 2

[24] Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 162–170, 2021. 3

[25] Fanhu Zeng, Hao Tang, Yihua Shao, Siyu Chen, Ling Shao, and Yan Wang. Mambaic: State space models for high-performance learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18041–18050, 2025. 2

[26] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17492–17501, 2022. 2