SCALABLE FEATURE CODING FOR DISTRIBUTED LARGE VISION-LANGUAGE MODEL INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Feature coding enables distributed inference of large vision-language models (LVLMs) by compressing visual representations at the edge and transmitting them to the cloud for downstream processing. However, using a single coded representation for tasks with different information requirements can waste bandwidth and incur unnecessary computation in the cloud-side large language model (LLM). To address these problems, we propose the first **Scalable Feature Coding (SFC)** framework for LVLMs. SFC organizes visual representations into a hierarchical bitstream, comprising a base layer that encodes coarse-grained global semantics and two enhancement layers that provide additional information for object-level and pixel-level semantics, respectively. This hierarchy enables coarse-grained tasks to operate on fewer visual tokens while supplying finer-grained information on demand. To further improve coding efficiency, we analyze correlations across and within information layers and introduce two complementary context models. Cross-layer context modeling (CL-CTX) conditions the entropy model of each enhancement layer on previously decoded layers, exploiting inter-layer dependencies for more accurate probability estimation. Intra-layer context modeling (IL-CTX) further reduces coding redundancy by capturing dependencies within a layer. Experiments with InternVL on image captioning, object detection, and instance segmentation demonstrate that SFC achieves 18.37% – 51.30% BD-BR and 7.5% – 55.6% end-to-end LVLM inference latency over baseline methods.

1 INTRODUCTION

LVLMs have achieved remarkable performance in open-ended visual understanding (Chen et al., 2024b; Tang et al., 2025). However, their computational and memory requirements constrain deployment on resource-limited devices. Distributed inference addresses this constraint by placing visual processing at the edge and language generation in the cloud (Li et al., 2025; Yuan et al., 2026). This partition reduces the model workload on the device but requires intermediate visual representations to cross the communication link. These representations are usually high-dimensional features, making transmission a substantial source of LVLM latency (Yuan et al., 2026; Hsu et al., 2026). *Efficient compression of the intermediate representation is therefore critical to practical distributed inference.*

Feature coding addresses this problem by compressing intermediate representations into compact bitstreams Gao et al. (2025b); Chen et al. (2026); Yuan et al. (2026). Although these advances improve the coding efficiency of individual features, they do not provide a hierarchical bitstream that can be adapted to the requirements of different tasks. Consequently, even for a coarse-grained task, they encode and transmit the full feature representation, incurring unnecessary transmission costs and LLM inference computation.

This observation raises a natural question: *Can visual features be coded according to the information requirements of different tasks?* To address this question, we propose Scalable Feature Coding (SFC), which organizes visual features into a hierarchical bitstream for task-dependent transmission. For coarse-grained tasks, SFC transmits only a subset of the features in a base layer, reducing transmission and inference costs. For fine-grained tasks, enhancement layers provide the additional information needed.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

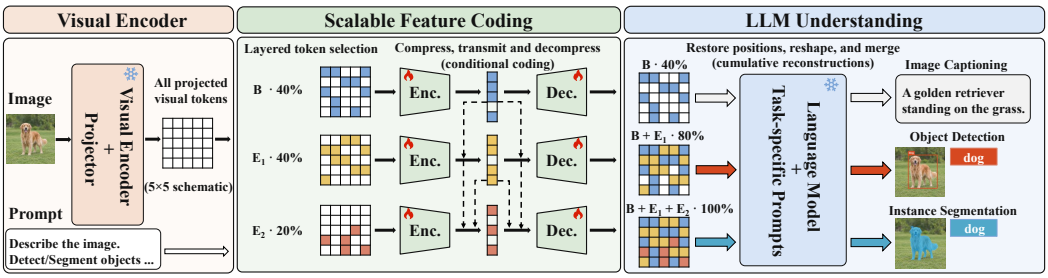


Figure 1: SFC-based distributed LVLM inference framework. Edge-extracted visual tokens are encoded into one base layer and two enhancement layers for transmission. The cloud decodes corresponding layers for a specific task and feed them to LLM for understanding.

As illustrated in Figure 1, SFC comprises three stages: edge-side visual encoding, scalable feature coding, and cloud-side LLM inference. At the edge, the visual encoder extracts visual features and partitions them into a base layer and two enhancement layers. The base layer is coded independently, while each enhancement layer is coded using previously decoded layers as context. To improve compression efficiency, the cross-layer context model (CL-CTX) and intra-layer context model (IL-CTX) are developed to capture dependencies between layers and dependencies within each layer. In the cloud, only the layers required for a given task are decoded before the reconstructed tokens are passed to the LLM for inference. Our main contributions are summarized as follows:

- We propose SFC, the first scalable feature coding framework for large models that simultaneously reduces transmission overhead and inference computation.
- We characterize vision tasks into three levels of granularity and develop an information layering method that partitions visual features into a base layer and two enhancement layers.
- We design CL-CTX and IL-CTX to improve compression efficiency by exploiting cross-layer and intra-layer dependencies, respectively.
- Extensive experiments show that SFC achieves superior rate-accuracy performance, computational efficiency, and end-to-end LVLM latency compared with existing methods.

2 Related Work

2.1 Feature Coding

CNN-oriented feature coding. Representative CNN-base feature coding studies improve compression efficiency through learned transforms, task-aware bit allocation, and the modeling of spatial, temporal, inter-scale, or cross-feature dependencies (Chen et al., 2020; Yamazaki et al., 2022; Alvar & Bajić, 2021; Suzuki et al., 2022; Liu et al., 2023b; Kim et al., 2024). Scalable methods organize coded features into base and enhancement layers, allowing each task to decode only the required portion of a shared bitstream (Wang et al., 2019; Choi & Bajić, 2021; Yan et al., 2021; Tu et al., 2021; Choi & Bajić, 2022; Liu et al., 2023a; Chen et al., 2024a). Their information layers are derived from CNN-specific structures, such as intermediate feature hierarchies, predefined channel subsets, and semantic-to-signal refinement paths. However, these CNN-specific layering strategies do not provide a mechanism for organizing visual tokens into cumulative information layers for LVLMs.

Large model feature coding. Recent studies have extended feature coding from CNNs to large models. Among them, LaMoFC (Gao et al., 2025b) establishes the first systematic benchmark with representative feature datasets, standardized test conditions, and coding baselines. Subsequent methods address complementary aspects: DT-UFC (Gao et al., 2025a) improves universality by aligning heterogeneous feature distributions, whereas VTC (Feng et al., 2026) models spatial redundancy in global and patch tokens. TOFC (Yuan et al., 2026) combines token merging with learned coding for multimodal inference, while VQFC (Chen et al., 2026) uses entropy-constrained vector quantization to reduce codec complexity. *Despite these advances, they encode and transmit the full visual representation without accounting for the semantic granularity required by downstream tasks.* This incurs unnecessary bitrate overhead and redundant computation during cloud-side task inference.

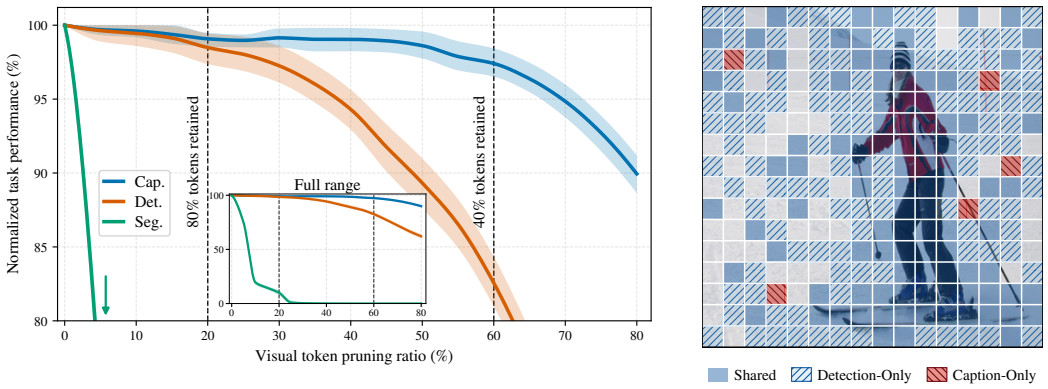


Figure 2: Analysis of task sensitivity to token reduction and token overlap across tasks. **Left:** Normalized performance shows that sensitivity to token reduction increases from captioning to detection and instance segmentation, indicating different information requirements. **Right:** Captioning and detection select many of the same tokens, suggesting that shared information can form a base layer, while task-specific information can be supplied through enhancement layers.

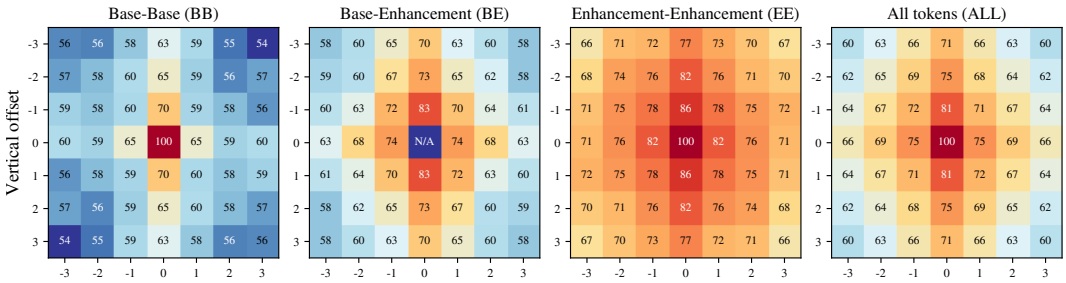


Figure 3: Spatial correlations within and across information layers. The heatmaps report Pearson correlations multiplied by 100 under a shared scale. Strong correlations within the enhancement layer and across layers indicate substantial redundancy that can be exploited for compression, motivating IL-CTX and CL-CTX, respectively.

2.2 Distributed LVLM Inference

The computation and memory demands of LVLMs motivate splitting inference between resource-constrained devices and more capable servers. Existing methods explore edge–cloud functional decomposition, heterogeneous-device model partitioning, and pipelined execution (Li et al., 2025; Yoon et al., 2025; Zhang et al., 2025; Ye et al., 2025). Related studies address communication efficiency by optimizing transmitted representations, adapting semantic transmission, or enabling collaborative decoding (Yuan et al., 2026; Hsu et al., 2026; Jia et al., 2026). However, most distributed inference frameworks focus on architecture design, model partitioning, and pipeline execution without explicitly addressing the compression of transmitted intermediate features. *Directly transmitting these high-dimensional features can substantially increase communication delay and end-to-end latency, making feature coding essential for efficient distributed LVLM inference.*

3 Motivation and Analysis

3.1 Task Scalability Analysis

Conventional image-transmission and feature-transmission pipelines deliver a complete coded image or feature regardless of downstream tasks. However, research in token compression show that removing some tokens does not degrade task performance (Yang et al., 2025; Tao et al., 2025; Cai et al., 2025; He et al., 2025). This motivates transmitting only the information required by each task.

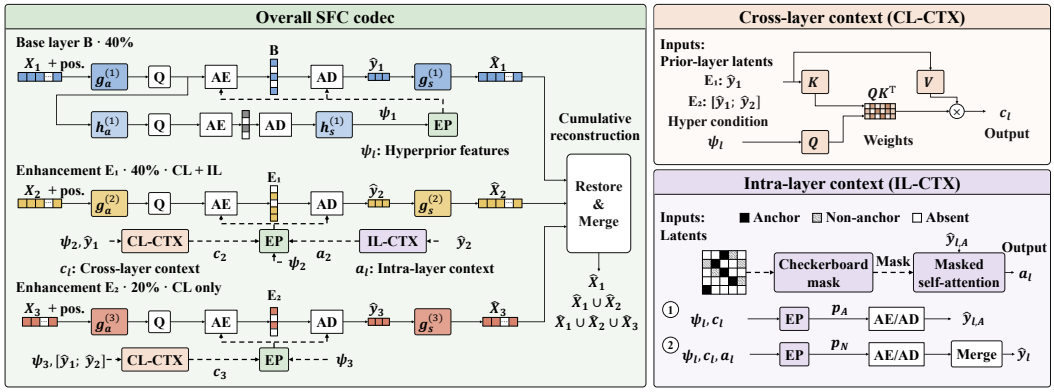


Figure 4: Architecture of the proposed SFC codec and its context models. **Left:** Overall SFC codec. The three-layer codec produces a base layer B and two enhancement layers E_1 and E_2 (Section 4.1). The base layer uses a hyperprior; E_1 additionally uses CL-CTX and IL-CTX, while E_2 uses CL-CTX only. **Upper right:** CL-CTX attends to previously decoded latents to derive cross-layer context (Section 4.2). **Lower right:** IL-CTX uses decoded checkerboard anchors to condition entropy estimation for non-anchor latents in a two-pass scheme (Section 4.3).

To investigate this possibility, we perform verification with InternVL2.5 model (Tang et al., 2025) on image captioning, object detection, and instance segmentation tasks. We progressively reduce the token budget using several pruning methods (Chen et al., 2025; Alvar et al., 2025; Dong et al., 2026) and measure the task performance. Figure 2(left) reports mean normalized performance, with shaded bands showing the range across methods. Captioning remains close to its original performance even when only 40% of the tokens are retained. Detection degrades more noticeably as tokens are removed, whereas instance segmentation is sensitive even to modest token reduction.

Figure 2(right) visualizes the spatial positions of tokens selected with visual–language guidance (Dong et al., 2026), retaining 40% for captioning and 80% for detection. The detection tokens include most of the captioning tokens. Their substantial overlap suggests that tasks can share a base representation and obtain additional information through enhancement layers. These observations motivate our scalable coding framework and provide the preliminary evidence.

3.2 Redundancy Analysis

Having motivated scalable coding, we next examine the statistical dependencies that remain after information layering. We use diversity-guided ordering to assign 40% of the tokens to the base layer and the remaining 60% to the enhancement layer. Figure 3 reports the mean Pearson correlation between token pairs at different displacements.

At nonzero displacements, correlations are lowest within the base layer and highest within the enhancement layer. Correlations across layers are also substantial, approaching those of the complete representation. These observations guide our codec design: CL-CTX exploits cross-layer dependencies by conditioning the entropy model for enhancement tokens on previously decoded layers, while IL-CTX exploits dependencies within a layer by using decoded tokens to model the remaining tokens.

4 The Proposed Method

4.1 Framework of SFC

Information layering. Let $X = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N$ denote the projected visual features, where N is the number of tokens. Following the analysis in Section 3.1, we partition this representation into three information layers to support global, object-level, and pixel-level perception. To evaluate the effect of different partitions, we consider two configurations: SFC-2L with one enhancement layer and SFC-3L with two enhancement layers. SFC-2L allocates approximately 40% of the features to

the base layer and the remaining 60% to the enhancement layer. SFC-3L allocates approximately 40%, 40%, and 20% to the base and two enhancement layers, respectively.

We implement information layering by varying the number of tokens assigned to each layer. Since the base layer must support global understanding with a limited token budget, we use diversity-guided ordering to select tokens that cover a broad range of visual content. Specifically, a greedy farthest-first traversal starts with the token having the largest nearest-neighbor distance and repeatedly selects the token farthest from its nearest selected token, using Euclidean distances between unit-normalized features. The resulting order assigns the earliest tokens to the base layer and subsequent tokens to the enhancement layers according to the specified proportions. Appendix A.1 provides the complete algorithm.

Scalable coding architecture. Following the information layering, we design a scalable three-layer framework that supports global, object-level and pixel-level tasks. As shown in Figure 4, SFC produces a hierarchical bitstream: a base layer B and two enhancement layers E_1 and E_2. All codec branches adopt the same hyperprior architecture (Ballé et al., 2018), consisting of a main branch with analysis and synthesis transforms $g_a^{(l)}$ and $g_s^{(l)}$, and a hyperprior branch with transforms $h_a^{(l)}$ and $h_s^{(l)}$. For layer l, these transforms are defined as

$$y_l = g_a^{(l)}(X_l; I_l), \quad z_l = h_a^{(l)}(y_l), \quad \psi_l = h_s^{(l)}(\hat{z}_l), \quad \hat{X}_l = g_s^{(l)}(\hat{y}_l; I_l), \tag{1}$$

where X_l and I_l denote the input tokens and their position indices, y_l and z_l are the main latent and hyperlatent, and ψ_l is the hyperprior information. A hat denotes a quantized latent or reconstructed feature. For clarity, Figure 4 shows the hyperprior branch only for the base layer.

The base layer is coded independently, with its main-latent entropy model conditioned only on ψ_1. Enhancement layers additionally exploit cross-layer and intra-layer context to improve coding efficiency. For layer $l > 1$, CL-CTX combines ψ_l and $\{\hat{y}_i\}_{i=1}^{l-1}$ to produce cross-layer context c_l. IL-CTX derives intra-layer context a_l from $\hat{y}_l$. The contexts c_l and a_l, together with ψ_l, condition the entropy model to achieve accurate probability estimation. In the three-layer configuration, E_1 uses both context models, whereas E_2 uses CL-CTX only, as detailed in Sections 4.2 and 4.3.

Codec architectures. All layers use the same Transformer-based transform architecture with separate parameters. The analysis transform $g_a^{(l)}$ first projects each visual token from 4096 to 512 dimensions to reduce the encoder's computational complexity, then adds a learned positional embedding corresponding to its original spatial position. The resulting sequence passes through two pre-norm Transformer blocks, followed by layer normalization and a 512-dimensional linear projection to produce the main latent. Each Transformer block contains 8-head self-attention and a feed-forward network with dimensions 512–1024–512 and GELU activation; both components use residual connections with learnable LayerScale. The synthesis transform $g_s^{(l)}$ mirrors the analysis architecture and reconstructs 4096-dimensional visual tokens from the decoded latents.

4.2 Cross-Layer Context Modeling

CL-CTX exploits the cross-layer correlations identified in Section 3.2 to improve entropy modeling for enhancement layers.

For enhancement layer $l > 1$, CL-CTX uses the current hyperprior information ψ_l as the query source and the cumulative reconstructed latents $\{\hat{y}_i\}_{i=1}^{l-1}$ as memory.

$$c_l = \mathrm{CrossAttn}(\psi_l, \{\hat{y}_i\}_{i=1}^{l-1}, \{\hat{y}_i\}_{i=1}^{l-1}). \tag{2}$$

As illustrated in the upper-right panel of Figure 4, CL-CTX independently projects ψ_l and $\{\hat{y}_i\}_{i=1}^{l-1}$ to 512 dimensions and adds learned position embeddings indexed by their original patch-grid locations. An 8-head cross-attention module uses the projected hyperprior information as queries and the preceding-layer latents as keys and values, followed by a linear projection that produces c_l.

For E_1, the memory contains the reconstructed base-layer latent $\hat{y}_1$. For E_2, it contains the cumulative reconstructed latents $\{\hat{y}_i\}_{i=1}^{2}$ from the base and first enhancement layers. Cross-attention captures dependencies between layers with different sequence lengths without requiring explicit

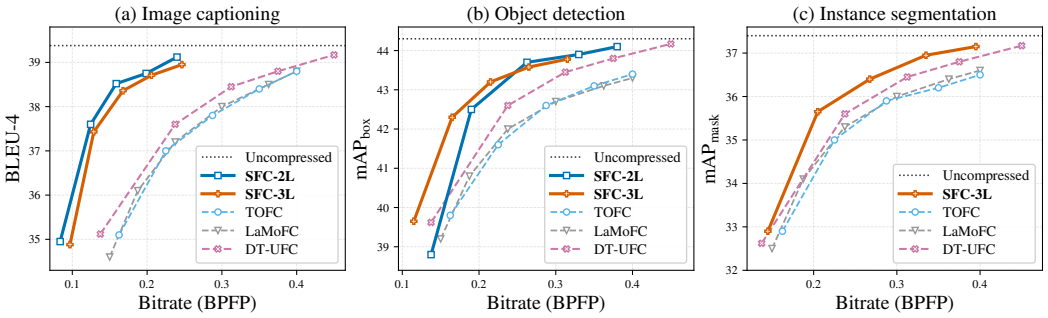


Figure 5: R-A performance comparisons between SFC methods and feature coding baselines.

Table 1: BD-BR (%) comparisons between SFC methods with baselines. LaMoFC is used as the anchor for BD-BR computation. Negative values indicate bitrate savings under the same accuracy.

	Seen tasks during training			Unseen tasks during training	
Method	Captioning	Detection	Instance Seg.	Semantic Seg.	Visual Grounding
TOFC	+0.26	+1.32	+1.78	−0.88	+5.93
DT-UFC	−11.58	−15.81	−10.05	−10.19	−8.02
SFC-2L	**−51.30**	−23.02	/	**−15.00**	**−16.31**
SFC-3L	−46.14	**−34.65**	**−18.37**	−7.06	−11.42

alignment. The resulting context c_l is combined with ψ_l to predict the probability distribution of the current main latent. Since both $\{\hat{y}_i\}_{i=1}^{l-1}$ and ψ_l are available at the encoder and decoder prior to decoding $\hat{y}_l$, identical context can be constructed at both ends.

4.3 Intra-Layer Context Modeling

IL-CTX uses two-pass checkerboard context (He et al., 2021) to exploit the intra-layer correlations identified in Section 3.2, avoiding fully sequential autoregressive decoding (Minnen et al., 2018).

Let (r_i, c_i) denote the row and column of token i on the original 16×16 patch grid. Positions satisfying $(r_i + c_i) \bmod 2 = 0$ form the anchor set $\mathcal{A}_l$, and the remaining positions form the non-anchor set $\mathcal{N}_l$. The probability parameters of the anchors are first predicted in parallel using the hyperprior and cross-layer context. After the anchors are decoded, IL-CTX projects the concatenated hyperprior and cross-layer context to a 512-dimensional query and the decoded anchor latents to 512-dimensional keys and values. An 8-head masked attention module attends only to anchor positions, followed by a linear projection that produces the intra-layer context a_l. This context is combined with ψ_l and c_l to predict the Gaussian distribution parameters of the non-anchor latents. The resulting two-pass factorization is

$$p(\hat{y}_l \mid \psi_l, c_l) = \prod_{i \in \mathcal{A}_l} p(\hat{y}_{l,i} \mid \psi_l, c_l) \prod_{j \in \mathcal{N}_l} p(\hat{y}_{l,j} \mid \psi_l, c_l, a_l). \tag{3}$$

This design requires only two decoding passes and allows the probability parameters within each pass to be predicted in parallel. IL-CTX is applied only to E_1 in both evaluated configurations, whereas E_2 uses CL-CTX alone because it contains only 20% of the tokens and is sparsely distributed across the patch grid, providing limited intra-layer context. Appendix A.2 summarizes the context used by each layer.

4.4 Training Strategy

Layer-wise training. We train the layer-specific codecs sequentially to provide a stable initialization for joint training. The base-layer codec is trained first. For each enhancement layer, only its codec and associated context modules are optimized, while the preceding codecs that supply context remain frozen. Training of each layer ends when its validation loss stabilizes and no longer decreases, after which training proceeds to the next layer.

Table 2: Computation and latency comparisons between SFC-3L and the original baseline using all visual tokens. Instance segmentation is omitted because it uses all visual tokens.

	Image Captioning			Object Detection		
Measure	Original	SFC-3L	Reduction	Original	SFC-3L	Reduction
Number of visual tokens	256	102	60.16%	256	204	20.31%
Attention computation (GFLOPs)	51.69	13.42	74.04%	215.42	181.89	15.57%
Prefill latency (ms)	91.68	69.39	24.31%	235.94	200.53	15.01%
KV cache size (MiB)	39.25	20.00	49.04%	80.13	73.63	8.11%

Joint training. Once all layer-specific codecs have been trained, we unfreeze them and optimize them jointly at a reduced learning rate. Joint training allows the codecs to coordinate their representations and context models, improving the overall coding performance of the layered bitstream.

Rate–distortion objective. Both stages use the same rate–distortion objective:

$$\mathcal{L} = \sum_{l=1}^{L} w_l (R_l + \lambda_l D_l). \tag{4}$$

where R_l is the estimated main- and hyperprior latent bitrate, D_l is the mean-squared reconstruction error, and λ_l balances rate and distortion for layer l. During layer-wise training, $w_l = 1$ for the current layer and zero otherwise. During joint training, $w_l = 1$ for all layers.

5 Experiments

5.1 Experimental Settings

Model, datasets, and tasks. Scalable coding requires a single model to support tasks at different semantic granularities and a single dataset to provide annotations for all evaluated tasks. We therefore select InternVL2.5 (Chen et al., 2024b) and COCO 2017 (Lin et al., 2014) to evaluate image captioning, object detection, and instance segmentation, using the InternVL2.5 weights provided by Tang et al. (2025). Instance segmentation is only supported in SFC-3L configuration.

Evaluation metrics. The accuracy of image captioning, object detection, and instance segmentation are measured by BLEU-4, $\mathrm{mAP}_{\mathrm{box}}$, and $\mathrm{mAP}_{\mathrm{mask}}$, respectively. Following feature coding practice, we report bitrate by bits per feature point (BPFP). The rate-accuracy performance is measured by BD-BR (Bjøntegaard, 2001).

Baselines. We compare our method with SOTA large model feature coding methods: LaMoFC Gao et al. (2025b), DT-UFC Gao et al. (2025a), and TOFC Yuan et al. (2026). In addition to feature coding, we also compare our method with image-coding-based LVLM pipelines. In this setting, the most common used JPEG (Wallace, 1992) and Better Portable Graphics (BPG) (Bellard, 2014) are used. Please refer to Appendix A.3 for feature codec training details.

5.2 Comparisons with Feature Coding Baselines

Rate–accuracy (R-A) performance. Figure 5 and Table 1 present the R-A curves and BD-BR results, respectively. Overall, both SFC-2L and SFC-3L outperform all baselines across the three tasks. For image captioning, SFC-2L outperforms SFC-3L, even though both methods encode the same visual tokens in their base layers. We attribute this gap to the additional cross-layer interactions in SFC-3L: during joint training, its base representation must accommodate two enhancement layers, which creates a trade-off between base-layer coding performance and enhancement-layer performance. For object detection, the trend reverses: SFC-3L achieves substantially better R-A performance than SFC-2L, although SFC-2L transmits 20% more visual tokens. This indicates that more information does not necessarily lead to better task performance. This result also supports the rationale for scalable coding: matching the transmitted representation to task's information requirements improves coding efficiency.

Efficiency comparison. Table 2 compares SFC-3L with the original baseline using all visual tokens and shows that its cloud-side efficiency gain depends on the visual-token budget selected for

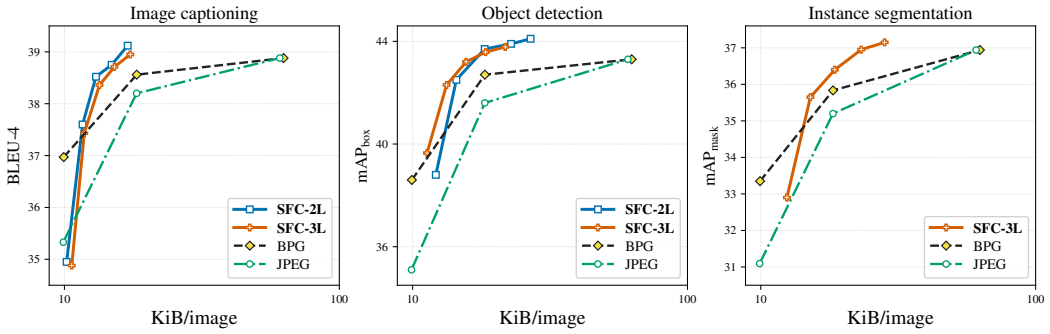


Figure 6: R-A performance comparisons between SFC methods and centralized LVLM pipelines.

each task. Captioning uses only the 102-token base layer, substantially shortening the visual-token sequence and reducing attention computation and memory use. The prefill speedup is more modest because of non-attention operations and fixed execution overhead. Detection retains 204 visual tokens to preserve object-level information, and its longer text prompt further reduces the contribution of visual tokens to the total input. Its efficiency gains are therefore more moderate. The contrast between the two tasks demonstrates the computational scalability of SFC: coarse-grained tasks obtain larger cloud-side savings by terminating at an earlier access point, whereas finer-grained tasks trade part of this efficiency for additional visual information. Instance segmentation is omitted because it uses all visual tokens and therefore provides no input-compaction gain.

5.3 Comparisons with Centralized LVLM Pipelines

We further compare SFC against a centralized pipeline that transmits compressed images and executes the entire LVLM in the cloud to validate the advantage of the SFC-based distributed paradigm. For centralized pipelines, we adopt the two most widely used image codecs, JPEG (Wallace, 1992) and BPG (Bellard, 2014).

Rate–accuracy performance. To unify the bitrate measure, we report the average number of encoded bits per image (KiB/image). The R-A curves are shown in Figure 6. Overall, the SFC-based distributed paradigm outperforms the centralized paradigm. In particular, both SFC methods reach near-uncompressed accuracy at lower rates across all three tasks, which is favorable for applications that require near-lossless performance. Moreover, although JPEG and BPG differ at low rates, both image codecs reach their peak performance at almost the same rate point.

Efficiency comparison. We decompose the LVLM pipeline latency into edge, transmission, and cloud components. For compute transmission latency, we follow the bandwidth settings of TOFC Yuan et al. (2026): 200, 400, and 800 kbps. Results are presented in Figure 7. SFC achieves the lowest pipeline latency across all tasks and bandwidth settings, owing to its superior coding performance. This latency gain becomes more pronounced as the downstream task complexity decreases. Under a bandwidth of 200 kbps, SFC reduces the end-to-end pipeline latency for image captioning by up to 55.6% relative to BPG and 51.3% relative to JPEG. In image captioning, LLM inference is fast because it is a purely text-generative task, so transmission dominates the pipeline latency and the latency gain from coding efficiency is more evident.

5.4 Generalization to Unseen Tasks

To verify the robustness and generalization of SFC, we evaluate semantic segmentation on ADE20K (Zhou et al., 2017) and visual grounding on RefCOCO (Yu et al., 2016). The R-A curves and BD-BR results are presented in Figure 8 and Table 1, respectively. The results show that SFC-2L provides the highest compression performance for semantic segmentation, with a 15.00% saving relative to LaMoFC. For visual grounding, both SFC methods outperform the existing feature coding baselines. These results demonstrate that the proposed SFC methods could be easily generalized to unseen tasks, which is favorable in practical scenarios where the downstream task is unknown.

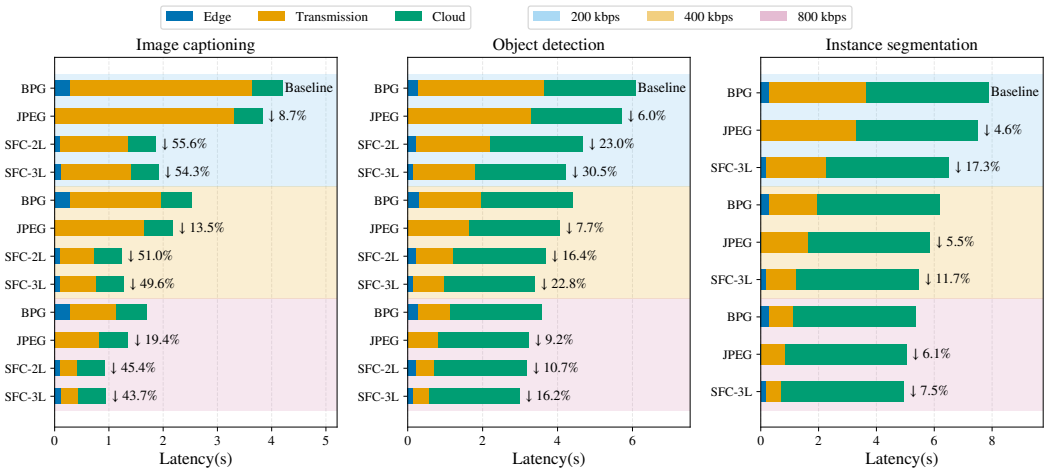


Figure 7: End-to-end latency comparisons between SFC methods and centralized LVLM pipelines. Latency reductions are computed relative to the BPG baseline.

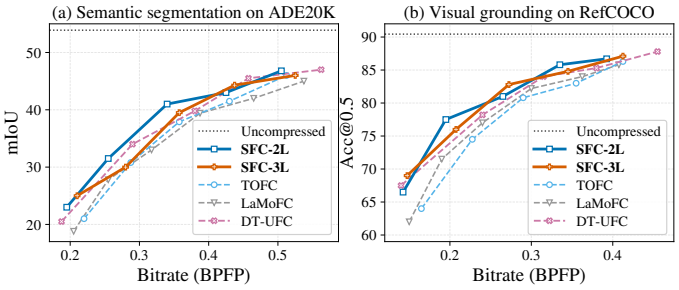


Figure 8: Rate-accuracy comparisons between SFC methods and feature coding baselines on unseen tasks.

Table 3: Ablation studies on CL-CTX and IL-CTX. *Anchor* denotes neither CL-CTX nor IL-CTX is used. The numbers are BD-BR results computed versus *Anchor*.

Context	SFC-2L	SFC-3L	
	Det.	Det.	Seg.
Anchor	0.0	0.0	0.0
CL-CTX	−10.2	−4.8	−7.6
IL-CTX	−17.6	−8.9	−13.1
CL + IL	**−24.1**	**−12.2**	**−18.4**

5.5 Ablation Study

We ablate CL-CTX and IL-CTX by enabling each separately and then jointly, against a hyperprior-only anchor method that disables both. When enabled, CL-CTX is applied to all enhancement layers, whereas IL-CTX is restricted to E_1; all remaining settings are unchanged. Table 3 reports BD-BR results relative to the anchor. Both modules improve coding efficiency, demonstrating the benefit of exploiting the cross-layer and intra-layer dependencies analyzed in Section 3.2. IL-CTX provides higher gains than CL-CTX alone, consistent with the stronger correlations within enhancement layers. Their combination achieves the highest bitrate savings in all settings, indicating that the two context sources are complementary.

6 Conclusion

In this paper, we introduce SFC for distributed LVLM inference, which addresses transmission overhead and cloud-side computation simultaneously. SFC organizes visual tokens into one base layer and two enhancement layers that provide layered information for global, object-level, and pixel-level tasks. CL-CTX and IL-CTX are proposed to improve compression performance by exploiting dependencies across and within these layers. Experiments demonstrate the superiority of SFC through comparisons with existing feature coding baselines, centralized LVLM pipelines, and generalization verification to unseen tasks. Future work will extend the current work to scalable video feature coding.

AI USE STATEMENT

GPT-based generative AI tools assisted with manuscript organization, language polishing, LaTeX editing and code development. All AI-assisted text and code were reviewed and verified by the authors. The authors take full responsibility for all claims, references, code, and reported results.

ETHICS STATEMENT

To the best of our knowledge, this work has no potential ethical issues to disclose.

REPRODUCIBILITY STATEMENT

The method specifies token allocation, context modeling, decoding dependencies, and training objectives. The appendix provides detailed layer configurations, training settings, and rate–distortion parameters to facilitate reproduction.

REFERENCES

Saeed Ranjbar Alvar and Ivan V. Bajić. Pareto-optimal bit allocation for collaborative intelligence. *IEEE Transactions on Image Processing*, 30:3348–3361, 2021. doi: 10.1109/TIP.2021.3060875.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, 2025. doi: 10.1109/CVPR52734.2025.00877.

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.

Fabrice Bellard. BPG image format, 2014. URL `https://bellard.org/bpg/`.

Gisle Bjøntegaard. Calculation of average PSNR differences between RD-curves. Technical Report VCEG-M33, ITU-T Video Coding Experts Group (VCEG), Austin, TX, USA, April 2001.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 46254–46272, 2025.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 19–35, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73004-7.

Qiaoxi Chen, Changsheng Gao, and Dong Liu. End-to-end learned scalable multilayer feature compression for machine vision tasks. In *2024 Data Compression Conference (DCC)*, pp. 550–550, 2024a. doi: 10.1109/DCC58796.2024.00067.

Qiaoxi Chen, Changsheng Gao, Li Li, and Dong Liu. Transform-free feature coding via entropy-constrained vector quantization. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i17.38474.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b.

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*, 29:2230–2243, 2020. doi: 10.1109/TIP.2019.2941660.

Hyomin Choi and Ivan V. Bajić. Latent-space scalability for multi-task collaborative intelligence. In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 3562–3566, 2021. doi: 10.1109/ICIP42928.2021.9506712.

Hyomin Choi and Ivan V. Bajić. Scalable image coding for humans and machines. *IEEE Transactions on Image Processing*, 31:2739–2754, 2022. doi: 10.1109/TIP.2022.3160602.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. MMTok: Multimodal coverage maximization for efficient inference of VLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

Donghui Feng, Fengxi Zhang, Changsheng Gao, Wenhan Yang, Qi Wang, Qunshan Gu, Hongwei Hu, Zhengxue Cheng, and Li Song. Visual token codec: Unleashing spatial redundancy for vit feature coding, 2026.

Changsheng Gao, Zijie Liu, Li Li, Dong Liu, Xiaoyan Sun, and Weisi Lin. Dt-ufc: Universal large model feature coding via peaky-to-balanced distribution transformation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 5198–5207, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3755814.

Changsheng Gao, Yifan Ma, Qiaoxi Chen, Yenan Xu, Dong Liu, and Weisi Lin. Feature coding in the era of large models: Dataset, test conditions, and benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1068–1077, 2025b. doi: 10.1109/ICCV51701.2025.00107.

Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14771–14780, June 2021.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Accelerating vision-language models through dynamic token sparsity. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20477–20486, 2025. doi: 10.1109/ICCV51701.2025.01904.

Cyril Shih-Huan Hsu, Wig Yuan-Cheng Cheng, and Chrysa Papagianni. Progressive semantic communication for efficient edge-cloud vision-language models, 2026.

Yuanyuan Jia, Shunpu Tang, and Qianqian Yang. Covspec: Efficient device-edge co-inference for vision-language models via speculative decoding, 2026.

Yeongwoong Kim, Hyewon Jeong, Janghyun Yu, Younhee Kim, Jooyoung Lee, Se Yoon Jeong, and Hui Yong Kim. End-to-end learnable multi-scale feature compression for vcm. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(5):3156–3167, 2024. doi: 10.1109/TCSVT.2023.3302858.

Yuyang Li, Devika Gumaste, Mehmet Kerem Turkcan, Javad Ghaderi, Gil Zussman, and Zoran Kostic. Distributed vlms: Efficient vision-language processing through cloud-edge collaboration. In *2025 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 280–286, 2025. doi: 10.1109/PerComWorkshops65533.2025.00078.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Lei Liu, Zhihao Hu, Zhenghao Chen, and Dong Xu. Icmh-net: Neural image compression towards both machine vision and human vision. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pp. 8047–8056, New York, NY, USA, 2023a. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3612041.

Tie Liu, Mai Xu, Shengxi Li, Chaoran Chen, Li Yang, and Zhuoyi Lv. Learnt mutual feature compression for machine vision. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023b. doi: 10.1109/ICASSP49357.2023.10094830.

David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 10794–10803, Red Hook, NY, USA, 2018. Curran Associates Inc.

S. Ravi, Daniel Rosenkrantz, and Giri Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42:299–310, 04 1994. doi: 10.1287/opre.42.2.299.

Satoshi Suzuki, Shoichiro Takeda, Motohiro Takagi, Ryuichi Tanida, Hideaki Kimata, and Hayaru Shouno. Deep feature compression using spatio-temporal arrangement toward collaborative intelligent world. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3934–3946, 2022. doi: 10.1109/TCSVT.2021.3107716.

Hao Tang, Chen-Wei Xie, Haiyang Wang, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. Ufo: A unified approach to fine-grained visual perception via open-ended language interface. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 83761–83791. Curran Associates, Inc., 2025. doi: 10.52202/085713-2808.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18992–19001, 2025. doi: 10.1109/CVPR52734.2025.01769.

Hanyue Tu, Li Li, Wengang Zhou, and Houqiang Li. Semantic scalable image compression with cross-layer priors. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, pp. 4044–4052, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386517. doi: 10.1145/3474085.3475533.

G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. doi: 10.1109/30.125072.

Shurun Wang, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. Scalable facial image compression with deep feature reconstruction. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2691–2695, 2019. doi: 10.1109/ICIP.2019.8803255.

Meguru Yamazaki, Yuichiro Kora, Takanori Nakao, Xuying Lei, and Kaoru Yokoo. Deep feature compression using rate-distortion optimization guided autoencoder. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 1216–1220, 2022. doi: 10.1109/ICIP46576.2022.9897877.

Ning Yan, Changsheng Gao, Dong Liu, Houqiang Li, Li Li, and Feng Wu. Sssic: Semantics-to-signal scalable image coding with learned structural representations. *IEEE Transactions on Image Processing*, 30:8939–8954, 2021. doi: 10.1109/TIP.2021.3121131.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, June 2025.

Shengyuan Ye, Bei Ouyang, Liekang Zeng, Tianyi Qian, Xiaowen Chu, Jian Tang, and Xu Chen. Jupiter: Fast and resource-efficient collaborative inference of generative llms on edge devices. In *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*, pp. 1–10, 2025. doi: 10.1109/INFOCOM55648.2025.11044734.

JinYi Yoon, JiHo Lee, Ting He, Nakjung Choi, and Bo Ji. S2m3: Split-and-share multi-modal models for distributed multi-task inference on the edge. In *2025 IEEE 45th International Conference on Distributed Computing Systems (ICDCS)*, pp. 868–878, 2025. doi: 10.1109/ICDCS63083.2025.00089.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 69–85, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46475-6.

Cheng Yuan, Zhening Liu, Jiashu Lv, Jiawei Shao, Yufei Jiang, Jun Zhang, and Xuelong Li. Task-oriented feature compression for multimodal understanding via device-edge co-inference. *IEEE Transactions on Mobile Computing*, 25(4):4762–4775, 2026. doi: 10.1109/TMC.2025.3626724.

Mingjin Zhang, Xiaoming Shen, Jiannong Cao, Zeyang Cui, and Shan Jiang. Edgeshard: Efficient llm inference via collaborative edge computing. *IEEE Internet of Things Journal*, 12(10):13119–13131, 2025. doi: 10.1109/JIOT.2024.3524255.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, 2017. doi: 10.1109/CVPR.2017.544.

A Additional Method Details

A.1 Diversity-Guided Ordering

For nonzero token features x_i, define $v_i = x_i/\|x_i\|_2$ and chord distance

$$d(v_i, v_j) = \|v_i - v_j\|_2 = \sqrt{2(1 - v_i^\top v_j)}. \tag{5}$$

This metric is monotone in cosine dissimilarity, so the two distances give the same greedy ordering with identical initialization and tie-breaking. The metric form makes the diversity guarantee below applicable. Algorithm 1 computes the token ordering used in Section 4.1. Computations across candidates can run in parallel, but selection rounds are sequential. A fixed index-based tie-breaking rule makes the ordering deterministic.

Algorithm 1 Farthest-first visual-token ordering

Require: Unit-normalized features $v_1, \ldots, v_N$, $N \geq 2$
Ensure: Ordered token indices $\pi_1, \ldots, \pi_N$
$\pi_1 \leftarrow \arg\max_i \min_{j \neq i} d(v_i, v_j)$
$S \leftarrow \{\pi_1\}$
$\delta_i \leftarrow d(v_i, v_{\pi_1})$ for all i
for $t = 2, \ldots, N$ **do**
  $\pi_t \leftarrow \arg\max_{i \notin S} \delta_i$
  $S \leftarrow S \cup \{\pi_t\}$
  $\delta_i \leftarrow \min\{\delta_i, d(v_i, v_{\pi_t})\}$ for all $i \notin S$
end for
return π

Metric diversity guarantee. For $2 \leq k \leq N$, let S_k be the first k tokens and let $S_k^\star$ maximize the minimum pairwise distance among all k-token subsets. The metric dispersion bound (Ravi et al., 1994) gives

$$\min_{i \neq j \in S_k} d(v_i, v_j) \geq \frac{1}{2} \min_{i \neq j \in S_k^\star} d(v_i, v_j). \tag{6}$$

To see this, let r_k be the distance from the kth selected token to its nearest predecessor. Greedy insertion distances are nonincreasing, so r_k is the minimum pairwise distance in S_k. After selecting $k-1$ tokens, every candidate is within r_k of one of them. Assign the k optimal tokens to their

nearest members of S_{k-1}; two must share an assigned member. Their distance is at most $2r_k$ by the triangle inequality, proving the bound. This guarantees geometric diversity of prefixes, not task accuracy or semantic disentanglement.

A.2 Layer Configurations

Table 4 specifies the token counts and conditioning information for both codecs. Enhancement contexts always use decoded latents, ensuring that the encoder and decoder share the same information.

Table 4: Token allocation and main-latent entropy-model inputs. ψ_l, c_l, and a_l denote hyperprior, cross-layer, and intra-layer context, respectively.

Model	Layer	Tokens	Cumulative	Anchors	Non-anchors
SFC-2L	B	102	102	ψ_1	ψ_1
	E_1	154	256	ψ_2, c_2	ψ_2, c_2, a_2
SFC-3L	B	102	102	ψ_1	ψ_1
	E_1	102	204	ψ_2, c_2	ψ_2, c_2, a_2
	E_2	52	256	ψ_3, c_3	ψ_3, c_3

A.3 Training Details

We select 20,000 images from COCO train2017 and use the frozen visual encoder and projector to extract visual tokens for training. Similarly, visual tokens extracted from the 5,000 images in COCO val2017 are used for validation. The codecs are trained and validated on the resulting visual-token datasets using the two-stage procedure in Section 4.4. Evaluation follows the three-stage edge–codec–cloud inference pipeline illustrated in Figure 1. Layer-wise training uses the Adam optimizer with a learning rate of 10^{-4} and a batch size of 16; joint fine-tuning uses a learning rate of 5×10^{-5}. Models with different rate–distortion multipliers are trained separately to obtain multiple operating points. SFC-2L uses approximately 40%/60% layer shares and SFC-3L uses 40%/40%/20%. All enhancement layers use CL-CTX, while only E_1 uses IL-CTX. All experiments are run on RTX-3090 GPUs.

Rate–distortion multiplier settings. For each configuration, a global multiplier λ defines a rate point, and the layer-specific multiplier is $\lambda_l = \alpha_l \lambda$, where $\alpha_l = N_l/N$ and $N = \sum_l N_l$. Table 5 lists the layer weights and the five global multipliers used for each configuration.

Table 5: Layer weights and global rate–distortion multipliers for SFC-2L and SFC-3L.

	α_B	α_{E_1}	α_{E_2}	Global λ values
SFC-2L	$\frac{102}{256}$	$\frac{154}{256}$	–	$\{5.0, 7.5, 12.2, 17.1, 21.1\}$
SFC-3L	$\frac{102}{256}$	$\frac{102}{256}$	$\frac{52}{256}$	$\{5.0, 8.4, 12.5, 17.5, 22.4\}$

A.4 Position Signaling and Rate Accounting

Token membership can be transmitted using fixed-length indices on the original patch grid. For $N = 256$, each index requires 8 bits, resulting in 816, 1,632, and 2,048 bits at the 102-, 204-, and 256-token access points, respectively. After normalization by $NC = 256 \times 4096$, the corresponding overheads are only 0.000778, 0.001556, and 0.001953 BPFP. Even the largest overhead is below 0.002 BPFP and is therefore negligible relative to the evaluated feature bitrates. The reported BPFP follows the conventional main-latent-plus-hyperlatent accounting; including the index overhead does not affect the reported comparisons or conclusions.