

A Implementation Details

A.1 Architecture of the control module

The control module described in Section 4.1 is modified from original ControlNet [1]. Key modifications include: (i) Replacing ControlNet’s initial three stride-2 convolutions with a single stride-1 convolution to establish dimensional compatibility with $\hat{\mathbf{z}}$. (ii) Implementing bilateral feature injection through zero-convolutions from both encoder and decoder pathways of the diffusion model (unlike ControlNet’s decoder-only approach) to establish more precise control mechanisms from $\hat{\mathbf{z}}$. Additionally, the extracted word-level tags \mathbf{c} are strategically injected as text prompts into both the control module and the diffusion model to effectively activate relevant generative priors.

A.2 Implementation in ablation studies

Basic setup. As described in Section 5.4, we designed ablation experiments to verify the effectiveness of the SC loss and tag guidance module, along with the hyper-parameter configuration of SC loss. In the experimental setup, training was conducted on the LSDIR dataset with a batch size of 8, a total of 80,000 iterations, and a learning rate of $1e - 5$. The experimental results were rigorously evaluated on the COCO 2017 object detection benchmark dataset.

Semantic Consistency loss with noisy input. In Section 4.2, for SC loss calculation, clean latent features \mathbf{z} and $\hat{\mathbf{z}}$ are utilized as inputs to the diffusion model. Recognizing that diffusion models are typically trained on noisy signals, we conducted a series of comparative experiments (illustrated in Figure 10) to investigate the implications of using noisy latent features \mathbf{z}_t and $\hat{\mathbf{z}}_t$ as inputs, corresponding to results in Section 5.4, “Input noise level for SC loss”. Specifically, in this variant, the same sampled noise is first added to the latent variable \mathbf{z} and the reconstructed latent variable $\hat{\mathbf{z}}$ using the same timestep t sampled from $U(0, t_{\max})$, where t_{\max} is a predefined threshold used to control the maximum noise intensity. A value of $t = 1$ corresponds to the complete 1000-step noising process. These are then separately fed into the trained diffusion model for feature mapping to obtain the corresponding semantic features, which are subsequently aligned.

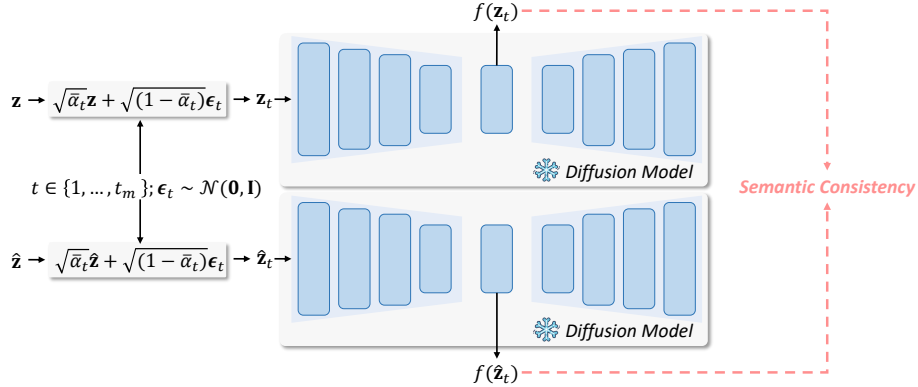


Figure 10: **Illustration of the variant of SC loss calculation.** Noisy latent variable \mathbf{z}_t and $\hat{\mathbf{z}}_t$ are utilized as input into the pre-trained diffusion model.

Bits of Tag Guidance. We employ RAM++ from Recognize Anything [2], which has a maximum default vocabulary of 4585 tags. For simplicity, we utilize fixed-length encoding without entropy coding, requiring 13 bits per tag ID to accommodate a maximum of 8192 tags. Based on our analysis of 500 randomly sampled COCO images, RAM++ predicts an average of 8.7 tags per image under its default settings. Consequently, the average bit overhead for tag guidance amounts to $13 \times 8.7 = 113.1$ bits per image.

A.3 Hardware device

All training and inference experiments are conducted on 4 NVIDIA A100 Tensor Core GPUs.

Table 1: Overview of experimental datasets, tasks, and models.

Dataset	Task Type	Specific Task	Task Model	Backbone
COCO 2017	Traditional Perception	Object Detection	Faster R-CNN	R50-FPN [9]
		Instance Segmentation	Mask R-CNN	
		Pose Estimation	Keypoint R-CNN	
		Panoptic Segmentation	Panoptic-FPN	
Flickr30K	Multi-Modal Retrieval	Image-Text Retrieval Text-Image Retrieval	BEiT-3 [10]	MW-Transformer [11]
RefGTA ADE20K	MLLM Understanding	Referring Expression Open-Set Segmentation	Qwen2.5-VL[12] Ospray [13]	WA-ViT [12] ConvNext [14]

B Experiments

B.1 Details of evaluation protocol

We conduct a comprehensive evaluation from two dimensions: the performance on various intelligent tasks and the perceptual quality of image reconstruction.

Intelligent tasks. As mentioned in Section 5.2, the efficacy and generalization capabilities of our proposed method are rigorously evaluated across a diverse spectrum of intelligent tasks spanning different domains, datasets, task-specific models, and vision backbones. To the best of our knowledge, this is the first comprehensive study to present such extensive experimental validation in the context of machine-oriented image coding. We believe this thorough empirical analysis not only substantiates our contributions but also establishes new benchmarks that may accelerate advancement in this emerging field.

The specific information about datasets and task models is shown in Table 1, where datasets include COCO 2017 [3], Flickr30K¹, RefGTA [4], and ADE20K². The intelligent tasks cover three major categories: traditional perception-based computer vision tasks, multimodal retrieval tasks, and multimodal understanding tasks based on large language models. For task models, traditional perception tasks are evaluated through a series of models in the Detectron2 toolkit; multimodal retrieval tasks are evaluated through the BEiT-3 model; and for multimodal understanding capabilities, we specifically selected two multimodal large language models (MLLMs) based on different backbone networks, processing different data domains, and executing tasks of varying granularities.

This comprehensive experimental framework aims to thoroughly verify the generalizability and superior performance of the Diff-ICMH approach across different data types, datasets, intelligent tasks, and task models. For evaluation metrics, we employ standard measures appropriate to each task: mAP (mean Average Precision) of bounding boxes for object detection, mAP of segmentation masks for instance segmentation, AP of keypoints for pose estimation, PQ (Panoptic Quality) for panoptic segmentation, Recall@1 for multimodal retrieval tasks, and accuracy for referring expression comprehension. The open-set pixel-domain understanding tasks based on MLLMs are evaluated using mIoU for semantic segmentation, mAP for instance segmentation, and PQ for panoptic segmentation.

Image reconstruction. This paper uses three public datasets: Kodak [5], Tecnick [6], and CLIC2020 [7]. During the experiments, images from Tecnick and CLIC2020 datasets are rescaled to 768 pixels on the short side, and samples of size 768×768 are cropped from the center of the images for evaluation, following previous method [8]. Evaluation metrics include PSNR and MS-SSIM for measuring signal fidelity, as well as LPIPS, FID, and DISTS for evaluating perceptual quality.

B.2 Visualization results

Task supporting. Figure 11 presents visualization results comparing our method against VTM-18.2 [15], ELIC [16], and FTIC [17] on instance segmentation tasks. The comparison reveals that

¹<https://hockenmaier.cs.illinois.edu/DenotationGraph/>

²<https://ade20k.csail.mit.edu/>



Figure 11: Visualized results on instance segmentation.

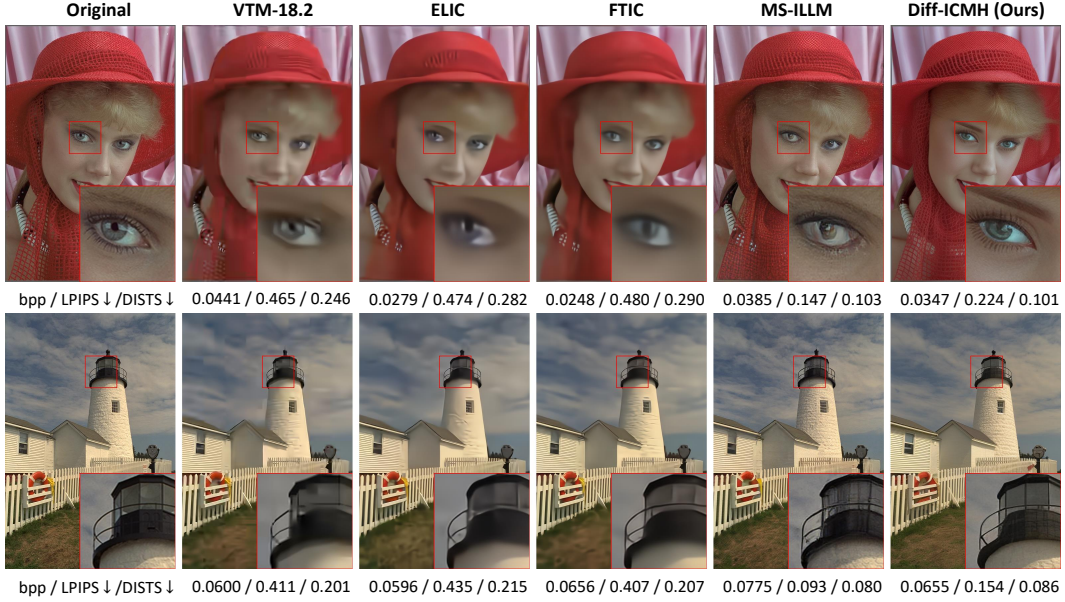


Figure 12: Reconstruction of Diff-ICMH and other compression methods.

VTM-18.2, despite operating at its highest bit rate, fails to correctly identify the two distinct objects within the image. Similarly, ELIC erroneously classifies the “bird” object on the right as a “person,” demonstrating significant recognition inaccuracy. In contrast, only the reconstruction output generated by our Diff-ICMH method successfully preserves the critical semantic information necessary for accurate object identification, thereby effectively supporting the completion of this intelligent task with precision.

Perception-oriented reconstruction. Figure 12 shows the visual comparison of reconstruction quality between Diff-ICMH and other compression methods. From the comparison, it can be observed that methods optimized for signal fidelity (VTM-18.2, ELIC, FTIC) produce reconstructed results with obvious blurring, significantly reducing visual quality. Although MS-ILLM shows some improvement in texture clarity, its reconstructed textures lack authenticity and are accompanied by obvious noise interference. In contrast, our proposed Diff-ICMH demonstrates the best visual realism in reconstruction results while maintaining clear boundary contour features.

B.3 Feature difference analysis

To further evaluate Diff-ICMH’s performance advantages in semantic information protection and intelligent task support, we designed comparative experiments based on feature difference analysis. Specifically, we calculated the differences between feature representations of reconstructed images from various compression methods and those of original images at different layers of a ResNet50 feature extractor (using 1 minus cosine similarity as the metric, where lower values represent higher feature similarity), and plotted the curves of these differences across network depths.

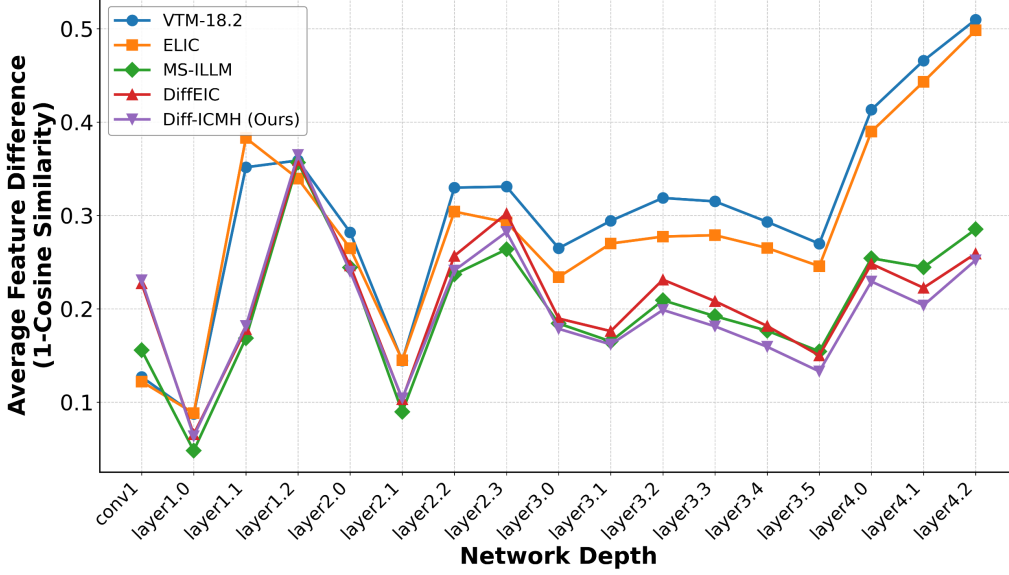


Figure 13: The evolution curves showing feature differences between reconstructed images from Diff-ICMH and other compression methods compared to original images across various layers of the ResNet50 feature extractor. Lower values represent higher feature similarity.

89 The experiments are conducted on the Kodak dataset, analyzing the average results of all images.
90 As shown in Figure 13, the experimental results exhibit significant hierarchical characteristics: in
91 shallow network layers, generative compression methods (MS-ILLM, DiffEIC, Diff-ICMH) show
92 relatively higher difference values, while methods optimized for signal fidelity (VTM-18.2, ELIC)
93 maintain minimal differences. However, as the network hierarchy and semantic level deepen, this
94 pattern changes significantly—the latter methods’ difference values increase rapidly, reaching a peak
95 of approximately 0.5 at the deepest layer, far exceeding generative compression methods.

96 Among generative compression methods, Diff-ICMH’s advantage is demonstrated in its performance
97 from the middle network layers (layer3.0) to the deepest layers (layer4.2), consistently maintaining the
98 lowest feature difference values in this range. This result confirms the superiority of our approach in
99 semantic information protection. More importantly, considering that the feature extractor (ResNet50)
100 used in the experiment and the feature mapping (Stable Diffusion) used in the SC loss belong to
101 different network architectures and different pre-training purpose, this performance advantage also
102 highlights the excellent model generalization capability of our method.

103 B.4 Complexity analysis

104 Table 2 illustrates the encoding and decoding time of different methods. The data reveals that
105 diffusion-based methods (such as DiffEIC and Diff-ICMH) perform well in terms of encoding speed,
106 showing significant advantages compared to traditional VVC methods (13.862 seconds), with DiffEIC
107 taking 0.128 seconds and Diff-ICMH taking 0.232 seconds.

108 However, there is room for improvement in our method’s decoding time. Particularly, when the
109 number of denoising steps increases, the decoding time increases significantly. For instance, Diff-
110 ICMH requires 5.456 seconds and 13.14 seconds for decoding at 20 steps and 50 steps, respectively.
111 The decoding time is considerably longer than traditional methods like VVC (0.066 seconds) and
112 GAN-based methods such as HiFiC (0.038 seconds) and MS-ILLM (0.059 seconds).

113 The main reason for this phenomenon is that the progressive denoising process of diffusion models is
114 inherently an iterative computational process, with each step requiring a complete network forward
115 pass. When the number of steps increases, the computational cost increases linearly. Optimizing
116 encoding and decoding speeds will be an important direction for future work: (i) Investigating more
117 efficient sampling strategies, such as implementing samplers with less steps [18, 19]; (ii) exploring
118 knowledge distillation techniques to distill multi-step denoising networks into lighter single-step

Table 2: Encoding and decoding time on Kodak dataset. (Second)

Method	NFE	Encoding Time	Decoding Time	Hardware
VVC	-	13.862	0.066	13th Core i9-13900K
ELIC	-	0.056	0.081	RTX4090
HiFiC	-	0.038	0.059	RTX4090
MS-ILLM	-	0.038	0.059	RTX4090
PerCo	5	0.080	0.665	A100
PerCo	20	0.080	2.551	A100
DiffEIC	20	0.128	1.964	RTX4090
DiffEIC	50	0.128	4.574	RTX4090
Diff-ICMH	20	0.232	5.456	A100
Diff-ICMH	50	0.232	13.14	A100

or few-step networks [20]; (iii) and optimizing network structures to reduce the computational complexity of each denoising step.

While diffusion-based image methods currently face computational efficiency challenges, this work primarily serves to demonstrate the fundamental potential of this paradigm. We anticipate that ongoing advancements in sampling algorithm efficiency and targeted engineering optimizations will substantially reduce encoding and decoding latency, thereby enhancing the practical utility of diffusion models in this research field.

B.5 Ablation study of distortion loss calculation space

In Equation (4) of the main text, the distortion loss $\mathcal{L}_{\text{dist}}$ are calculated in the VAE latent space:

$$\mathcal{L}_{\text{dist}} = \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2 = \|\mathcal{E}_{\text{VAE}}(\mathbf{x}) - \mathcal{D}_c(\hat{\mathbf{y}})\|_2^2, \quad (1)$$

Here we conduct ablation study of calculating loss in the pixel space:

$$\mathcal{L}_{\text{dist}} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \|\mathbf{x} - \mathcal{D}'_c(\hat{\mathbf{y}})\|_2^2, \quad (2)$$

where \mathcal{D}'_c maintains the same architectural foundation as \mathcal{D}_c but incorporates additional upsampling blocks to reconstruct signals at pixel-level resolution. The reconstructed pixel-space output $\hat{\mathbf{x}}$ is subsequently fed into the control module for generative reconstruction. Figure 14 demonstrates the substantial performance advantage of calculating distortion in the latent space rather than pixel space. This finding confirms that the VAE latent space provides a more compact and perceptually meaningful representation that effectively filters semantically irrelevant information from the pixel domain. Through optimizing fidelity in the latent space, the compressed bitstream effectively prioritizes semantically salient information, thereby achieving superior rate-distortion performance across various downstream machine vision applications.

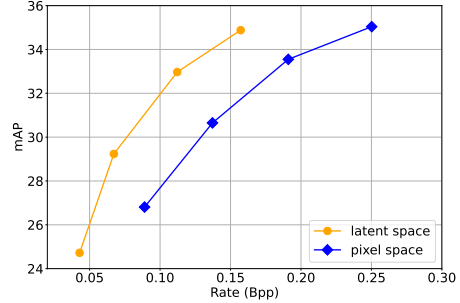


Figure 14: Ablation of distortion loss calculation space.

References

- [1] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.
- [2] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu *et al.*, “Recognize anything: A strong image tagging model,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1724–1732.
- [3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.
- [4] M. Tanaka, T. Itamochi, K. Narioka, I. Sato, Y. Ushiku, and T. Harada, “Generating easy-to-understand referring expressions for target identifications,” in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5794–5803.
- [5] E. Kodak, “Kodak lossless true color image suite (photocd pcd0992),” <http://r0k.us/graphics/kodak>, 1993.
- [6] N. Asuni, A. Giachetti *et al.*, “Testimages: a large-scale archive for testing visual devices and basic image processing algorithms,” in *STAG*, 2014, pp. 63–70.
- [7] G. Toderici, L. Theis, N. Johnston, E. Agustsson, F. Mentzer, J. Ballé, W. Shi, and R. Timofte, “Clic 2020: Challenge on learned image compression,” *Retrieved March*, vol. 29, p. 2021, 2020.
- [8] Z. Li, Y. Zhou, H. Wei, C. Ge, and J. Jiang, “Towards extreme image compression with latent feature guidance and diffusion prior,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.
- [9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.
- [10] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som *et al.*, “Image as a foreign language: Beit pretraining for vision and vision-language tasks,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 175–19 186.
- [11] H. Bao, W. Wang, L. Dong, Q. Liu, O. K. Mohammed, K. Aggarwal, S. Som, S. Piao, and F. Wei, “Vlmo: Unified vision-language pre-training with mixture-of-modality-experts,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 897–32 912, 2022.
- [12] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, “Qwen2.5-vl technical report,” 2025. [Online]. Available: <https://arxiv.org/abs/2502.13923>
- [13] Y. Yuan, W. Li, J. Liu, D. Tang, X. Luo, C. Qin, L. Zhang, and J. Zhu, “Osprey: Pixel understanding with visual instruction tuning,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 202–28 211.
- [14] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.
- [15] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, “Overview of the versatile video coding (vvc) standard and its applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.
- [16] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, “Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5718–5727.
- [17] H. Li, S. Li, W. Dai, C. Li, J. Zou, and H. Xiong, “Frequency-aware transformer for learned image compression,” in *International Conference on Learning Representations*, 2024.
- [18] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 5775–5787, 2022.
- [19] —, “Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models,” *Machine Intelligence Research*, pp. 1–22, 2025.
- [20] T. Salimans and J. Ho, “Progressive distillation for fast sampling of diffusion models,” in *International Conference on Learning Representations*, 2023.