PULSE: UNLOCKING SINGLE-THREAD CPU DECODING FOR PRACTICAL IMAGE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite recent progress in learned image compression, existing methods remain computationally expensive on resource-constrained hardware, particularly CPUs. We introduce PULSE, a practical codec that enables (1) **low-latency decoding on diverse hardware platforms** with an ultra-low-complexity 5.2 kMAC/pixel neural receiver, and (2) **efficient bit-exact entropy coding** with an integer linear CDF predictor and a meta prior. To recover compression performance under this tight budget, we introduce an agentic evolution process guided by heuristic probes that iteratively improves the architecture through human–LLM collaboration. PULSE decodes a 1080p image in 126 ms on **a single CPU thread** while achieving compression performance comparable to HM. After perceptual optimization, PULSE competes with larger perceptual codecs like MS-ILLM.

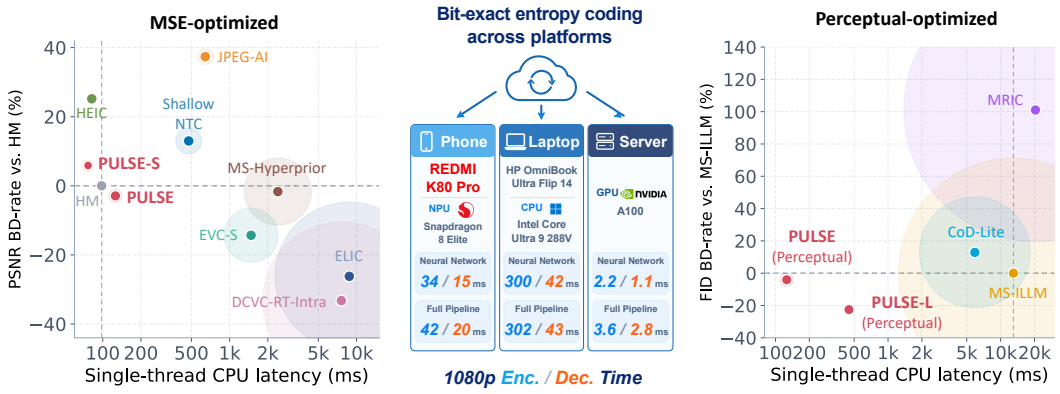


Figure 1: **PULSE targets practical image compression on resource-constrained hardware.** Its 5.2 kMAC/pixel receiver decodes a 1080p image in 126 ms on a single thread of an AMD EPYC 9V84, while enabling bit-exact entropy coding across devices like REDMI K80 Pro. On the CLIC Professional validation set, PULSE matches HM and its perceptual variant competes with larger codecs such as MS-ILLM (Muckley et al., 2023). *Latency is measured for the full decoding pipeline, including network inference and entropy decoding. Circle radius denotes the decoding complexity.*

1 INTRODUCTION

Pioneering learned image compression systems (Ballé et al., 2017; 2018; Minnen et al., 2018) used neural receivers requiring roughly 80 kMAC/pixel. Subsequent work improved rate–distortion performance by scaling receiver complexity to several hundred kMAC/pixel (Cheng et al., 2020; He et al., 2022b; Jiang et al., 2025; Liu et al., 2023), making real-time decoding challenging even on GPUs. To bridge this deployment gap, instance-adaptive codecs reduce receiver complexity to below 5 kMAC/pixel, but shift substantial computation to per-image optimization during encoding (Ladune et al., 2023; Kim et al., 2024). Another line of work pursues efficient end-to-end codecs: EVC reduces receiver complexity to about 110 kMAC/pixel, Shallow NTC to approximately 20 kMAC/pixel, while DCVC-RT reorganizes computation to achieve real-time GPU throughput (Wang et al., 2023; Yang &

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.

Mandt, 2023; Jia et al., 2025). Despite these advances, such systems remain challenging to deploy at scale on resource-constrained edge devices, particularly CPUs.

We propose PULSE to target the missing operating point: an ultra-low-complexity, highly parallel and end-to-end codec, with a sender and a receiver around 45 and 5.2 kMAC/pixel, respectively. We use *a single CPU thread* as a stringent deployment stress test, while retaining bounded-pass entropy dependencies and accelerator-friendly operators so that the same model can benefit from NPUs and GPUs for practical implementation. We make PULSE practical by solving three core questions.

How to design a codec under an ultra-low-complexity budget? Under such a stringent constraint, every channel and operator must be placed where it brings the greatest compression benefit. PICO searches over millions of architectures, incurring substantial candidate-training costs and relying on statically predefined networks (Tatwawadi et al., 2026). We instead introduce an *agentic evolution* process that uses heuristic probes to inform architecture design through human–LLM collaboration. With only seven evolution rounds and 22 training runs, PULSE achieves compression performance comparable to HM on the CLIC Professional validation set. It decodes a 1080p image in 126 ms on a single thread of an AMD EPYC 9V84 CPU.

How to maintain bit-exact entropy coding across devices with minimal loss? Platform-dependent floating-point round-off can change the selected CDF and desynchronize the remaining bitstream. PULSE addresses this with *one integer linear projection* from the hyperlatent to the main-latent CDF bucket, which requires only 1.6 ms for a 1080p image on a single CPU thread. In practice, this compact entropy-control path can execute on CPU for bit-exact entropy coding, while the remaining parts run efficiently on accelerators. With a meta prior to improve hyperlatent coding, this scheme outperforms existing integerization-based methods in rate-distortion loss.

How to effectively improve perceptual quality? We separate perceptual training into two stages: an LPIPS-augmented pre-training stage to establish detail synthesis capability, and a preference-optimization stage that jointly improves pixel fidelity, distribution alignment, and text and facial quality. The optimized model competes with larger perceptual codecs.

Together, these components address key bottlenecks in practical image compression on resource-constrained hardware. As illustrated in Figure 1, PULSE enables practical compression across phones, laptops, and servers with low computational cost. On edge devices like REDMI K80 Pro, it takes 42 ms for encoding and 20 ms for decoding at 1080p. We expect further system-level optimization to improve its efficiency for real-world applications.

2 PULSE ARCHITECTURE

PULSE is an asymmetrical, variable-rate, cross-platform codec designed for ultra-low-complexity decoding. As illustrated in Figure 2, it consists of an analysis transform, a two-step entropy model, and a synthesis transform. All modules are designed with highly parallel computation to enable efficient implementation across different hardware platforms. We build it with depthwise convolution blocks (DCB) with a channel attention layer (Hu et al., 2018), which is used throughout modules.

Analysis transform. Given an image $x \in [0,1]^{3\times H\times W}$, a stride-16 patch embedding produces a feature at $1/16$ spatial resolution. Four DCBs are then applied, followed by a 1×1 projection:

$$y = \mathrm{Encoder}(x) \in \mathbb{R}^{C_y \times H/16 \times W/16}. \tag{1}$$

Entropy model and coding. Unlike the raster-ordered entropy model of Cool-Chic (Ladune et al., 2023), PULSE uses two bounded passes. A hyper encoder (Ballé et al., 2018) maps y to a hyperlatent z at $1/64$ input resolution. The rounded $\hat{z}$ is coded by a QP-conditioned Meta Prior, which selects among a compact bank of factorized priors (Ballé et al., 2017; Jia et al., 2025), and a hyper decoder restores a C_y-channel feature at the resolution of y:

$$z = \mathrm{HyperEncoder}(y) \in \mathbb{R}^{C_z \times H/64 \times W/64}, \qquad h = \mathrm{HyperDecoder}(\hat{z}). \tag{2}$$

Following the dual spatial prior of DCVC-HEM (Li et al., 2022), a spatial–channel checkerboard partitions y into complementary masks m_0 and m_1. The first entropy step predicts a quantization step q and a mean μ_0 from h. After decoding the first part, the second step refines μ_1 from $(h, \hat{y}_0)$, then

$$y_{q_i} = \mathrm{round}(y/q - \mu_i) \odot m_i, \qquad \hat{y}_i = q\,(y_{q_i} + \mu_i) \odot m_i, \qquad \hat{y} = \hat{y}_0 + \hat{y}_1. \tag{3}$$

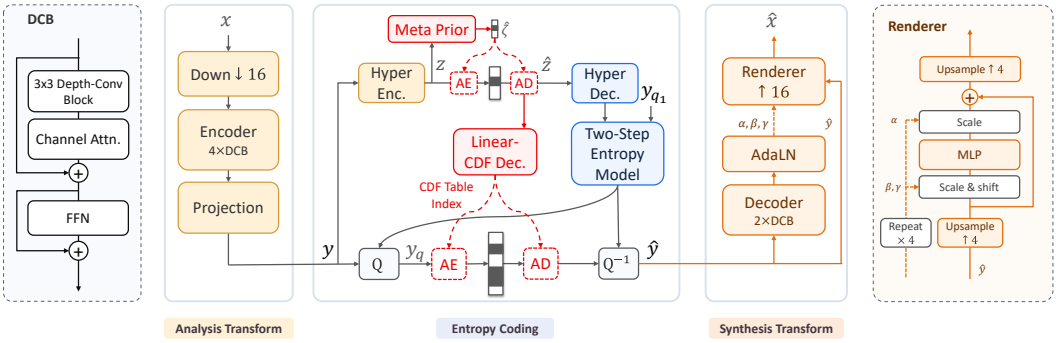


Figure 2: **PULSE architecture.** It consists of an encoder at $1/16$ resolution, a decoupled linear CDF index decoder with a meta prior, a two-step entropy model, and a decoder with an AdaLN-modulated pixel renderer. AE and AD denote arithmetic encoding and decoding, respectively.

The centered symbols are modeled by a discretized zero-mean Gaussian. Scale prediction is independent of the nonlinear mean path: one projection maps $\hat{z}$ directly to a continuous coordinate in a shared Gaussian CDF table. Both passes reuse these scale coordinates, while only their μ_i differ. Section 3 specifies the Meta Prior and deterministic integer implementation.

Synthesis transform. The synthesis transform separates spatial feature processing from patch-wise pixel rendering. The decoder consisting of two DCBs first processes $\hat{y}$ for spatial correlation modeling. In parallel, a content branch upsamples $\hat{y}$ by four to a $1/4$-resolution rendering grid. The low-resolution body generates AdaLN modulation parameters (α, β, γ) (Peebles & Xie, 2023) for a residual pointwise MLP on this grid. Finally, every grid vector is projected to a 4×4 RGB patch and rearranged by PixelShuffle (Shi et al., 2016):

$$\hat{x} = \text{Renderer}(\hat{y}, \text{Decoder}(\hat{y}))\,. \tag{4}$$

3 Bit-Exact Linear CDF Decoding with a Meta Prior

Platform-dependent floating-point round-off is particularly dangerous inside an entropy decoder: a single change in a CDF-table index can desynchronize every subsequent rANS symbol (Duda, 2013). The conventional solution is to integerize the entire entropy model (He et al., 2022a), ensuring deterministic execution of the CPU subgraph from $\hat{z}$ to CDF. However, as shown in Table 1, this introduces substantial CPU overhead and quantization loss under an ultra-low complexity budget. In PULSE, we instead simplify the entropy-control path to the minimum computation required for bit-exact CDF selection while preserving coding efficiency.

Linear scale decoder. We first follow Pärnamaa et al. (2026) to derive the scale directly from $\hat{z}$ using a learned LUT, which is extremely fast but incurs a substantial rate-distortion loss. We then integerize only the independent hyper scale decoder (Ascenso et al., 2023), rather than the entire entropy model, which effectively alleviates both issues. This inspires us to take a step further: *can we significantly simplify the hyper scale decoder while retaining its compression ratio?* Surprisingly, we find that a simple *linear projection* from $\hat{z}$ to the scale provides an effective middle ground. The linear decoder is more robust to quantization, incurring only about 1.5% BD-rate loss at INT8 precision compared to the full hyper scale decoder, while reducing CPU time from 28.1 ms to 1.2 ms.

As shown in Figure 3, CDF selection based on $\log \sigma$ reveals two limitations. Firstly, the $\sigma \to \log \sigma$ path must also execute in INT8, adding 2.9 ms of latency. Secondly, uniform quantization makes $\hat{z}$ a fixed-step additive representation, whereas Gaussian CDF indices are uniformly spaced in log-scale. Consequently, affine scale decoding induces a scale-dependent CDF-index sensitivity, with the same latent step causing larger index changes at small scales and smaller changes at large scales.

Linear CDF decoder. To address this, we propose a simple but effective solution: directly predict the CDF index instead of scale. Specifically, a linear projection predicts a continuous coordinate

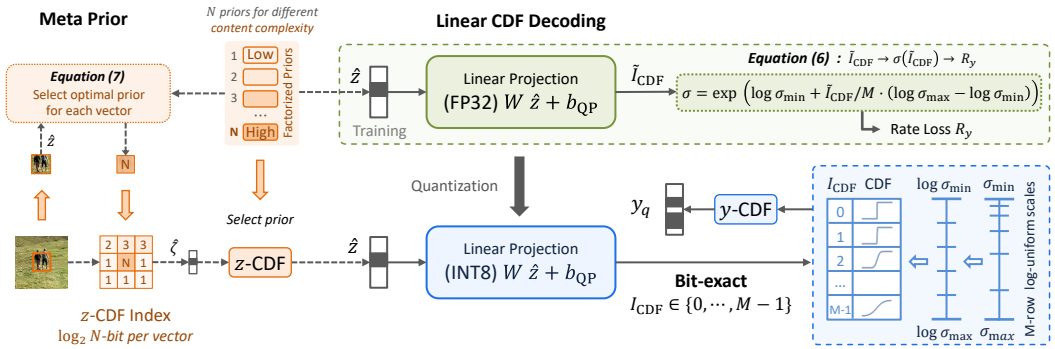


Figure 3: **Bit-exact entropy coding path.** Meta Prior transmits $\hat{\zeta}$ to select CDFs for $\hat{z}$ with different content complexities. An integer linear projection maps $\hat{z}$ to CDF indices for entropy coding of y_q.

$\tilde{I}_{CDF}$ in a shared M-row log-spaced Gaussian CDF table:

$$\tilde{I}_{CDF} = W\hat{z} + b_{\mathrm{QP}}, \qquad I_{\mathrm{CDF}} = \left\lfloor \mathrm{clip}(\tilde{I}_{CDF}, 0, M-1) \right\rfloor. \tag{5}$$

Here b_{QP} is a learned per-QP bias. During training, the bounded coordinate is mapped to

$$\sigma(\tilde{I}_{CDF}) = \exp\left(\log\sigma_{\min} + \tilde{I}_{CDF}/M \cdot (\log\sigma_{\max} - \log\sigma_{\min})\right) \tag{6}$$

for differentiable Gaussian likelihood evaluation. We empirically set $\sigma_{\min} = 0.11$, $\sigma_{\max} = 256$ and $M = 64$. At inference, entropy coding consumes I_{CDF} directly, eliminating the runtime exponential and logarithm. Linear CDF decoding eliminates the $\log\sigma$ computation to reduce CPU time to 1.6 ms, while achieving a lower BD-rate than the INT8 hyper scale decoder on the CLIC test set.

Meta Prior. The extreme simplification of the linear CDF decoder shifts more rate-modeling responsibility onto $\hat{z}$. However, the inflexible factorized prior uses the same prior distribution for $\hat{z}$ across all images and cannot adapt to different image contents. Empirically, we observe that $\hat{z}$ accounts for about half of the bitstream for images with low content complexity. As shown in Table 1, it leads to a substantial degradation on the low-content-complexity (LC) subset.

To address this issue, we introduce a meta prior that enables content-adaptive entropy modeling with minimal additional decoding complexity. Specifically, after training the entire model, we post-train only the factorized prior into a bank of N prior models. Each prior model is fitted to hyperlatent vectors from regions of different content complexity, with other modules fixed. We then allow each $\hat{z}$ vector at a different spatial position to select its own prior model, using an expectation-maximization (EM) procedure to jointly optimize the prior models and their assignments. More details are provided in Appendix B. At inference, for each spatial position r in $\hat{z}$, the encoder evaluates the quantized-CDF coding cost $\ell_{q,k}(\hat{z}_r)$ under every prior bank and transmits the best bank index:

$$k_r^\star = \arg\min_{k\in\{0,\ldots,N-1\}} \ell_{q,k}(\hat{z}_r), \qquad R_z = \sum_r \left[\log_2 N + \ell_{q,k_r^\star}(\hat{z}_r)\right]. \tag{7}$$

A single $\log_2 N$-bit index is shared by all channels at each spatial position, serving as meta-prior information. The decoder reads this metadata and selects the corresponding fixed CDF bank to decode $\hat{z}_r$ without additional neural-network inference. Unlike some prior works that select competing priors for main latents (Brummer & De Vleeschouwer, 2021), meta prior adapts $\hat{z}$ coding to compensate for the reduced capacity of the linear CDF decoder, particularly in low-content-complexity regions. This brings a 9.8% bitrate saving on low-complexity images and a 2.4% saving on the overall set.

In deployment, the linear CDF activation and weights are INT8, accumulation is INT32, and the output is a UINT8 CDF index. Meta Prior bank indices are coded with fixed-length coding, and can be further improved with merge coding.

4 Agentic Evolution with Heuristic Probes

Under an ultra-low complexity budget, every channel and operator matters, making architecture-parameter allocation critical to compression performance. We evolve the neural receiver through four

Table 1: **Ablation of bit-exact entropy-coding designs.** BD-rate is measured on the CLIC test set and a subset of images with low content complexity (LC subset). Latency is measured for a 1080p image on one AMD EPYC 9V84 CPU thread.

Method	CLIC test set		LC subset		Single-thread CPU Time	
	Float	INT8	Float	INT8	$\hat{z} \to \sigma$	$\hat{z} \to$ CDF
Integerize full entropy model	1.5%	5.7%	3.1%	7.8%	85.9 ms	86.0 ms
+ ~~LUT scale decoder~~ (Pärnamaa et al., 2026)	29.7%	29.7%	59.6%	59.6%	0.5 ms	0.5 ms
+ Hyper scale decoder (Ascenso et al., 2023)	**–0.1%**	3.2%	2.1%	6.4%	28.1 ms	29.2 ms
+ Linear scale decoder	4.5%	4.5%	11.8%	12.0%	1.2 ms	4.1 ms
+ Linear CDF decoder	2.4%	2.4%	9.8%	9.9%	N/A	1.6 ms
+ Meta Prior $\to$ **Proposal**	0.0%	**0.04%**	**0.0%**	**0.04%**	N/A	1.6 ms

stages: *probe*, *analyze*, *reallocate*, and *train*, as illustrated in Figure 4. The analysis transform is fixed throughout the process to isolate receiver design.

4.1 Heuristic Representation Probes

Given a trained network, we probe representations at selected interfaces. For layer $l \in \{1, \cdots, L\}$, we examine the input latent a^l with C^l channels. We measure two complementary properties.

Redundancy probe with PCA dimension. The latent a^l may contain redundant channels if its allocated dimension exceeds what is required to preserve end-task performance. We fit a dataset-shared PCA basis and reconstruct it from its first c principal components, denoted by $\Pi_c(a^l)$. We then find the smallest dimension whose relative rate–distortion loss increase is at most $\epsilon = 1\%$:

$$C^l_{\mathrm{PCA}} = \min_c \left\{ c : \mathcal{L}_{\mathrm{RD}}(\Pi_c(a^l)) - \mathcal{L}_{\mathrm{RD}}(a^l) \le \epsilon \cdot \mathcal{L}_{\mathrm{RD}}(a^l) \right\}. \tag{8}$$

The ratio C^l_{PCA}/C^l suggests utilization of the allocated channel subspace. A small ratio indicates potentially reclaimable channel redundancy.

Headroom probe with latent optimization. The PCA probe suggests redundancy but does not assess whether upstream computation produces the best representation the downstream network can exploit. We therefore optimize a^l independently for each image while freezing all downstream parameters. For decoder-side interfaces, the coded symbols are already fixed, so this intervention keeps the rate unchanged and directly measures an oracle PSNR gain:

$$a^{l*} = \arg\min_{a^l} D\big(F_{>l}(a^l), x\big), \qquad G^l_{\mathrm{OPT}} = \mathrm{PSNR}\big(F_{>l}(a^{l*}), x\big) - \mathrm{PSNR}\big(F_{>l}(a^l), x\big). \tag{9}$$

Entropy-side interfaces use the rate–distortion objective, as detailed in Appendix D. A large G^l_{OPT} indicates that the downstream network can benefit from a more informative value at this interface.

4.2 Human–LLM Collaborative Analysis

The LLM agent receives $\{{C_{\mathrm{PCA}}}^l, G^l_{\mathrm{OPT}}\}_{l=1}^L$ together with the network graph, parameter sharing, exact component costs, and the history of previous rounds. With this comprehensive context, the LLM interprets the probe results in the architectural context and proposes fixed-budget changes for empirical validation. The human researcher then reviews the hypotheses, implementation feasibility and budget accounting, and selects candidates for training.

In practice, some bottlenecks cannot be addressed by changing parameters in the predefined search space. Both the LLM agent and the human researcher may therefore introduce a spark: a new architectural mechanism or adjustable factor motivated by the probe evidence. This allows the optimization space itself to evolve as new bottlenecks are discovered.

Discussion: comparison with NAS. Agentic evolution differs from conventional NAS in two respects. First, heuristic probes help prioritize candidate directions before from-scratch training instead of enumerating a large static search space. Second, the human–LLM process is open-ended: probe evidence can motivate new parameters or mechanisms through sparks rather than restricting all candidates to search axes fixed in advance.

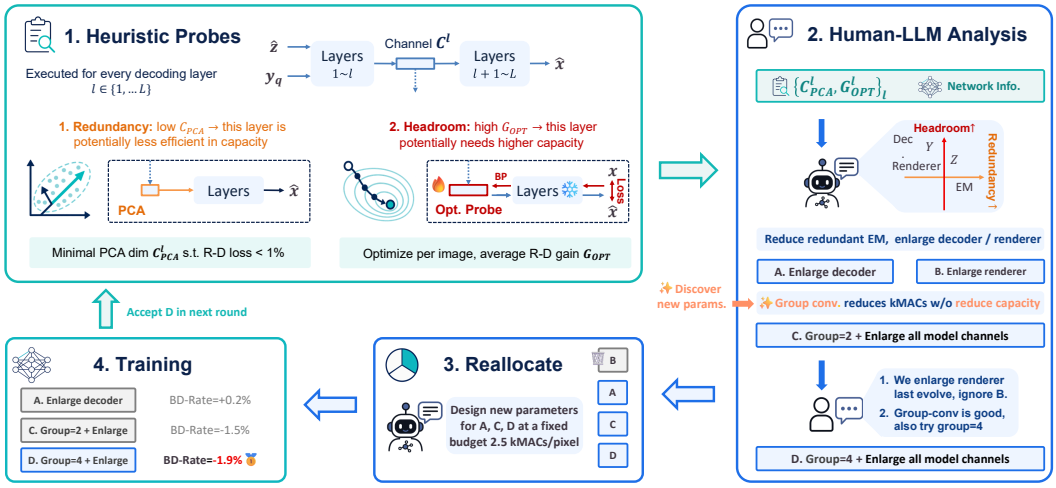


Figure 4: **Agentic evolution**, illustrated for round E4→E5 in the complete trajectory in Figure 10. Layer-wise probes measure redundancy and headroom. A human researcher and an LLM agent interpret the evidence to propose fixed-budget reallocations, and validate them by training.

4.3 Fixed-budget Reallocation and From-Scratch Training

Each selected proposal is profiled before training to ensure that parameter reallocation remains within a fixed complexity budget. The remaining candidates are trained from scratch with the same data, rate range, optimizer, and schedule. Finally, the candidate with the best compression performance is selected as the promotion for the next round.

4.4 Evolution at 2.5 kMAC/pixel and Scaling to Different Sizes

The complete trajectory of seven generations is reported in Appendix D (Figure 10). Across the full trajectory, human–LLM collaboration alternates between reallocating known parameters and introducing new search axes. In detail, the depth and width of different modules are jointly adjusted step by step, and three rounds discover new parameters including the renderer resolution, group convolution and FFN width. The final E7 achieves a 10.2% BD-rate reduction relative to E1. Finally, we adjust the channel dimensions to multiples of 16 for efficient inference of PULSE-S, and then enlarge channels to scale the decoding complexity to 5.2 and 20 kMAC/pixel, resulting in the PULSE and PULSE-L models, respectively.

Discussion: scope and generalization. We study human–LLM co-design for ultra-low-complexity image codecs, rather than task-agnostic architecture search. The workflow is history-dependent, with insights from earlier rounds guiding subsequent decisions. Thus, the contributions of human and LLM agents cannot be cleanly separated post hoc. Within this setting, the workflow enables effective architecture evolution with few training runs. Generalization to other tasks may require task-specific search spaces and objectives, which we leave for future work.

5 Perceptual and Region-of-Interest Optimization

This section introduces the techniques used to improve the subjective reconstruction quality of PULSE. When optimized with MSE alone, the decoder tends to over-smooth uncertain details, resulting in blurry textures and local structures. Inspired by perceptual codecs (Mentzer et al., 2020; Muckley et al., 2023), we adopt a two-stage perceptual optimization process.

Stage I: LPIPS-First Perceptual Pre-Training. The first stage establishes the model's detail synthesis capability by emphasizing the LPIPS loss (Zhang et al., 2018):

$$\mathcal{L}_{\text{Stage I}} = R_y + R_z + \lambda \left[\phi_{\text{MSE}} D_{\text{MSE}} + \phi^{\text{I}}_{\text{LPIPS}} D_{\text{LPIPS}}\right]. \tag{10}$$



Figure 5: **Visual comparison.** PULSE achieves comparable or better visual quality after perceptual and ROI optimization. More results are in Appendix C.4.

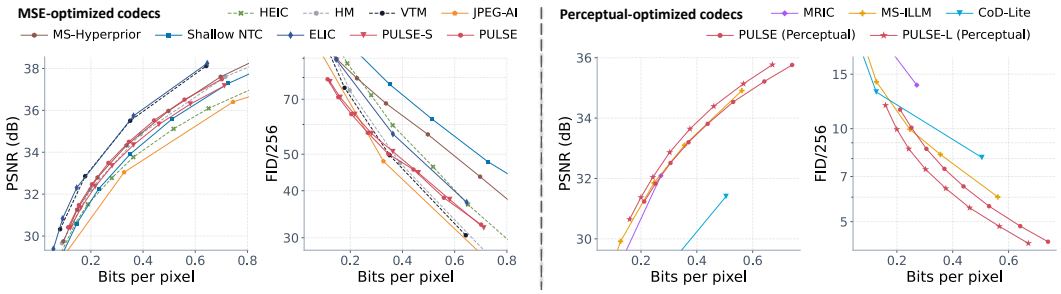


Figure 6: **Rate–distortion curves.** CLIC Professional validation results for MSE-optimized codecs (left) and perceptually optimized codecs (right). More results are in Appendix C.3.

where $\phi^{\mathrm{I}}_{\mathrm{LPIPS}}$ is set substantially higher than in prior work (Muckley et al., 2023) to encourage stronger perceptual detail synthesis.

Stage II: Preference Optimization. The second stage aligns reconstruction quality with perceptual preference criteria. We make three modifications: (1) reduce the LPIPS weight back to mitigate its artifacts and imperfect perceptual alignment; (2) introduce adversarial training with a DINOv2-based discriminator (Oquab et al., 2024) for distribution-level alignment; and (3) introduce region-of-interest (ROI) optimization for semantically important regions, e.g., small text and face regions.

We pre-detect text- and face-rich images and store their corresponding ROI boxes. Within these regions, we increase the MSE and LPIPS weights and introduce semantic feature alignment:

$$\mathcal{L}_{\text{Stage II}} = R_y + R_z + \lambda\big[\phi_{\mathrm{MSE}} D_{\mathrm{MSE}} + \phi^{\mathrm{II}}_{\mathrm{LPIPS}} D_{\mathrm{LPIPS}} + \phi_{\mathrm{GAN}} D_{\mathrm{GAN}} + \phi_{\mathrm{ROI}} D_{\mathrm{ROI}}\big]. \quad (11)$$

$$D_{\mathrm{ROI}} = \phi_{\mathrm{FaceNet}} D_{\mathrm{FaceNet}} + \phi_{\mathrm{CRNN}} D_{\mathrm{CRNN}}, \quad (12)$$

where $\phi^{\mathrm{I}}_{\mathrm{LPIPS}} = 4 \times \phi^{\mathrm{II}}_{\mathrm{LPIPS}}$. FaceNet (Schroff et al., 2015) uses square face crops with contextual regions, while the CRNN-CNN teacher (Shi et al., 2017) uses aspect-preserving text crops. This stage jointly improves pixel fidelity, perceptual quality, and ROI quality, which is verified in Table 8.

6 Experiments

6.1 Experimental Setup

Training. PULSE is optimized using SOAP (Vyas et al., 2025), requiring 15 hours for the MSE version and 36 hours for the perceptual version on a single NVIDIA H100 GPU. More details of training schedules and datasets are in Appendix B. PULSE supports eight quantization parameters in a single model through the module-bank formulation (Jia et al., 2025), and each quantization parameter is trained with its corresponding λ.

Table 2: **Ablation studies.** BD-Rate (↓) is measured in terms of PSNR and FID for the MSE- and perceptual-optimized versions, respectively. Decoder speed is measured on 1080p images on an NPU.

Decoder structure	PSNR	FID	Speed
Progressive Upsampling	1.1%	5.8%	0.9×
Single Low-Res. Decoder	12.0%	18.4%	1.6×
Low-Res. Decoder + Renderer	0.0%	0.0%	1.0×

Architecture search	#Trains ↓	PSNR
No Search	1	0.0%
Hardware-aware NAS	1280	-7.5%
Agentic Evolution	22	-10.2%

Benchmarks and metrics. We use the CLIC Professional validation set (CLIC Organizers, 2020) for comparison. For MSE-optimized compression, we compare with Mean-Scale Hyperprior (Minnen et al., 2018), ELIC (He et al., 2022b), EVC (Wang et al., 2023), DCVC-RT-Intra (Jia et al., 2025), Shallow NTC (Yang & Mandt, 2023), HM-16.25 (Sullivan et al., 2012), and VTM-17.0 (Bross et al., 2021). For perceptual compression, we compare with MRIC (Agustsson et al., 2023), MS-ILLM (Muckley et al., 2023), and CoD-Lite (Jia et al., 2026). HiFiC and PICO comparisons on matched datasets are in the appendix. We report PSNR and patch-based FID (Heusel et al., 2017) (256×256, overlapping) (Mentzer et al., 2020). BD-rate follows Bjøntegaard (2001). We evaluate $\text{bpp} \geq 0.05$, so ultra-low-bitrate results may differ from prior reports.

Complexity. We report the parameters and multiply–accumulate operations exercised by each sender and receiver. Latency is measured at 1088×1920 on a single AMD EPYC 9V84 CPU thread.

6.2 Experimental Results

Comparison results. Figure 6 and Table 3 report comparisons with MSE- and perceptual-optimized codecs. PULSE-S encodes a 1080p image in 920 ms and decodes it in 77 ms, faster than all compared codecs, including software-implemented HEIC. It also achieves a lower PSNR-based BD-rate of $+5.9\%$, compared to $+25.2\%$ for HEIC. PULSE achieves a -2.9% BD-rate relative to HM-16.25 with $9\times$ faster encoding, while competing with the much larger MS-ILLM after perceptual optimization. PULSE-L achieves a -22.5% FID-based BD-rate relative to MS-ILLM with substantially fewer computations. Figure 5 shows text and natural-texture examples for visual comparison, and more examples are in the appendix.

Ablation study. As shown in Table 2, under the same decoding complexity, our rendering-based decoder outperforms the progressive upsampling decoder (Ballé et al., 2017) in both compression efficiency and NPU inference speed. The single low-resolution decoder (Jia et al., 2025) is faster but incurs a relatively large performance loss. Our scheme achieves the best overall trade-off. Compared to hardware-aware NAS (Tatwawadi et al., 2026), the proposed agentic evolution uses fewer training runs to achieve better performance.

6.3 Practical Implementation

In prior sections, we focus on single-thread CPU execution as a pressure test. To assess practical deployment, we evaluate PULSE across diverse platforms, including an Intel Core Ultra 9 288V CPU, NVIDIA A100 and H100 GPUs, and a Qualcomm Snapdragon 8 Elite NPU. Executing the linear CDF decoding on the CPU **verifies bit-exact entropy decoding** across all devices. Figure 1 shows that the low complexity translates to high throughput on capable hardware. On a REDMI K80 Pro with a Snapdragon NPU, the complete encoding and decoding pipelines take 42 ms and 20 ms, respectively, for 1080p images. Overall, these results provide an indication that PULSE's ultra-low-complexity design can translate into practical deployment.

7 Related Work

Efficient neural codecs. Recent work reduces neural decoding complexity through lightweight contexts, networks, and architectures. ELIC reduces serial context (He et al., 2022b), EVC uses mask decay (Wang et al., 2023), FastNIC targets low-complexity compression (Zhang et al., 2025), and DCVC-RT and Shallow NTC reduce operation complexity or depth (Jia et al., 2025; Yang & Mandt, 2023). We instead explicitly target ultra-low-complexity, end-to-end decoding on a single

Table 3: **Complexity and rate–distortion performance.** Time measurements use 1088×1920 inputs on a single-thread CPU. BD-rate is evaluated on the CLIC Professional validation set. VBR denotes variable bitrates within one model, and BE denotes bit-exact entropy coding across platforms.

Method	VBR	BE	Target	Encoding			Decoding			BD-rate ↓	
				Params	kMAC/px	Time	Params	kMAC/px	Time	on PSNR	on FID
HEIC	✓	✓	MSE	libheif + x265		1.4 s	libheif + libde265		80 ms	+25.2%	+2680%
HM-16.25	✓	✓	MSE	HM Encoder		8.8 s	HM Decoder		98 ms	0.0%	+2450%
VTM-17.0	✓	✓	MSE	VTM Encoder		249 s	VTM Decoder		189 ms	-23.4%	+2660%
JPEG-AI	×	✓	MSE	7.1 M	56	2.6 s	4.4 M	8.1	643 ms	+37.4%	+1720%
MS-Hyperprior	×	×	MSE	14 M	112	2.1 s	12 M	109	2.4 s	-1.7%	+3830%
ELIC	×	×	MSE	26 M	335	6.5 s	24 M	332	8.8 s	-26.2%	+3430%
EVC-S	✓	×	MSE	12 M	71	1.4 s	11 M	79	1.4 s	-14.3%	+3040%
DCVC-RT-Intra	✓	×	MSE	31 M	258	5.4 s	36 M	362	7.6 s	-33.2%	+3200%
Shallow NTC	×	×	MSE	23 M	277	5.6 s	10 M	21	476 ms	+13.0%	+5590%
MRIC	×	×	Perc.	74 M	823	16 s	71 M	829	20 s	+46.1%	+101.0%
MS-ILLM	×	×	Perc.	25 M	98	2.2 s	170 M	675	13 s	+38.0%	0.0%
CoD-Lite	×	✓	Perc.	28 M	344	7.0 s	52 M	219	5.9 s	+218.8%	+12.9%
PICO	✓	✓	Perc.	5.6 M	75	2.0 s	4.3 M	46	1.2 s	N/A	N/A‡
PULSE-S	✓	✓	MSE	13 M	45	920 ms	1.0 M	2.7	77 ms	+5.9%	+2530%
PULSE	✓	✓	MSE	13 M	45	920 ms	1.7 M	5.2	126 ms	-2.9%	+2420%
			Perc.	13 M	45	920 ms	1.7 M	5.2	126 ms	+45.5%	-4.1%
PULSE-L	✓	✓	Perc.	15 M	48	972 ms	5.7 M	20	420 ms	+30.8%	-22.5%

‡ PICO complexity and timing use our local reproduction. We compare with its released 300-image reconstructions in Appendix C.2.

CPU thread. Cool-Chic and C3 use tiny image-specific decoders, which require substantial per-image optimization for encoding (Ladune et al., 2023; Kim et al., 2024). In addition, their highly sequential entropy model limits implementation efficiency.

Perceptual neural codecs. Another line targets perceptual quality by modeling richer image details. HiFiC and MS-ILLM enlarge synthesis networks to model realistic texture (Mentzer et al., 2020; Muckley et al., 2023). CoD-Lite enables real-time diffusion-based decoding, while PICO uses hardware-aware NAS to balance perceptual quality and efficiency (Jia et al., 2026; Tatwawadi et al., 2026). However, perceptual compression on resource-constrained hardware remains underexplored.

Cross-platform Compression. Post-training quantization can make probability prediction deterministic across platforms (He et al., 2022a). JPEG AI standardizes integer hyper scale decoder (Ascenso et al., 2023; Tatwawadi et al., 2026). However, their scale decoding still relies on a large network. On devices with only an NPU as an accelerator, this bit-exact computation may fall back to the host CPU, where running such a large network is less efficient. MLVC encodes scales into the hyperlatent using a lookup table, but incurs a considerable coding cost (Pärnamaa et al., 2026). Thus, a simple-to-deploy scheme with low coding loss is still needed.

8 CONCLUSION

PULSE demonstrates that practical learned image compression is possible under an extreme computational budget. With a receiver complexity of 5.2 kMAC/pixel, it decodes a 1080p image in 126 ms on a single CPU thread while achieving HM-comparable compression performance. Agentic evolution recovers performance under this tight budget, while a meta prior and integer linear CDF predictor enable bit-exact entropy coding across platforms. Perceptual optimization further enables it to compete with larger perceptual codecs such as MS-ILLM.

Limitation. Although we deploy PULSE across heterogeneous devices to demonstrate its practical applicability, further optimization is needed for real-world deployment. While ROI processing improves visual quality for practical applications, PULSE is not specifically optimized for screen content, games, or artistic images, and its performance may degrade on these contents. Thus, PULSE should be viewed as an academic prototype rather than a ready-to-use software system.

AI USE STATEMENT

As described in Section 4 and Appendix D, an LLM was used to assist in interpreting representation probes and experimental feedback and in generating candidate architectural hypotheses and fixed-budget modifications. A human researcher reviewed these suggestions for implementation feasibility and budget compliance. Candidate architectures were then trained and evaluated, with subsequent selections based on measured rate–distortion performance. In addition, we used LLMs to assist with manuscript refinement, primarily to improve sentence-level clarity and readability of a human-written draft. The authors are responsible for all final methodological decisions, experimental validation, and the content and claims of this work, including any AI-assisted contributions.

REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1122–1131, 2017. doi: 10.1109/CVPRW.2017.150.

Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi-realism image compression with a conditional generator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

João Ascenso, Elena Alshina, and Touradj Ebrahimi. The JPEG AI standard: Providing efficient human and machine visual data consumption. *IEEE MultiMedia*, 30(1):100–111, 2023. doi: 10.1109/MMUL.2023.3245919.

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

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.

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

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. doi: 10.1109/TCSVT.2021.3101953.

Benoit Brummer and Christophe De Vleeschouwer. End-to-end optimized image compression with competition of prior distributions. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1890–1894. IEEE, 2021.

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7939–7948, 2020. doi: 10.1109/CVPR42600.2020.00796.

CLIC Organizers. CLIC 2020: Challenge on learned image compression. Dataset documentation: `https://www.tensorflow.org/datasets/catalog/clic`, 2020.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2567–2581, 2022. doi: 10.1109/TPAMI.2020.3045810.

Jarek Duda. Asymmetric numeral systems: Entropy coding combining speed of huffman coding with compression rate of arithmetic coding. *arXiv preprint arXiv:1311.2540*, 2013.

Dailan He, Ziming Yang, Yuan Chen, Qi Zhang, Hongwei Qin, and Yan Wang. Post-training quantization for cross-platform learned image compression. *arXiv preprint arXiv:2202.07513*, 2022a.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5718–5727, 2022b. doi: 10.1109/CVPR52688.2022.00563.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018. doi: 10.1109/CVPR.2018.00745.

Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu. Towards practical real-time neural video compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12543–12552, 2025. doi: 10.1109/CVPR52734.2025.01170.

Zhaoyang Jia, Naifu Xue, Zihan Zheng, Jiahao Li, Bin Li, Xiaoyi Zhang, Zongyu Guo, Yuan Zhang, Houqiang Li, and Yan Lu. CoD-Lite: Real-time diffusion-based generative image compression. *arXiv preprint arXiv:2604.12525*, 2026.

Wei Jiang, Jiayu Yang, Yongqi Zhai, Feng Gao, and Ronggang Wang. MLIC++: Linear complexity multi-reference entropy modeling for learned image compression. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 21(5):1–25, 2025. doi: 10.1145/3719011.

Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont. C3: High-performance and low-complexity neural compression from a single image or video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9347–9358, 2024. doi: 10.1109/CVPR52733.2024.00893.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023. doi: 10.1109/ICCV51070.2023.00371.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. doi: 10.1007/s11263-020-01316-z.

Théo Ladune, Pierrick Philippe, Félix Henry, Gordon Clare, and Thomas Leguay. COOL-CHIC: Coordinate-based low complexity hierarchical image codec. In *IEEE/CVF International Conference on Computer Vision*, pp. 13515–13522, 2023. doi: 10.1109/ICCV51070.2023.01243.

Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1503–1511, 2022. doi: 10.1145/3503161.3547845.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1132–1140, 2017. doi: 10.1109/CVPRW.2017.151.

Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14388–14397, 2023. doi: 10.1109/CVPR52729.2023.01383.

Fabian Mentzer, George D. Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

David Minnen, Johannes Ballé, and George D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Matthew J. Muckley, Alaaeldin El-Nouby, Karen Ullrich, Hervé Jégou, and Jakob Verbeek. Improving statistical fidelity for neural image compression with implicit local likelihood models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25426–25443, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Tanel Pärnamaa, Martin Lumiste, Ardi Loot, Evgenii Indenbom, Andrei Znobishchev, and Ando Saabas. MLVC: Multi-platform learned video codec for real-world deployment. *arXiv preprint arXiv:2606.28027*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015. doi: 10.1109/CVPR.2015.7298682.

Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017. doi: 10.1109/TPAMI.2016.2646371.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016. doi: 10.1109/CVPR.2016.207.

Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. doi: 10.1109/TCSVT.2012.2221191.

Kedar Tatwawadi, Parisa Rahimzadeh, Zhanghao Sun, Zhiqi Chen, Ziyun Yang, Sanjay Nair, Divija Hasteer, and Oren Rippel. What matters in practical learned image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12095–12105, 2026.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: Improving and stabilizing shampoo using adam for language modeling. In *International Conference on Learning Representations*, 2025.

Guo-Hua Wang, Jiahao Li, Bin Li, and Yan Lu. EVC: Towards real-time neural image compression with mask decay. In *International Conference on Learning Representations*, 2023.

Yibo Yang and Stephan Mandt. Computationally-efficient neural image compression with shallow decoders. In *IEEE/CVF International Conference on Computer Vision*, pp. 530–540, 2023. doi: 10.1109/ICCV51070.2023.00055.

Haotian Zhang, Yuqi Li, Li Li, and Dong Liu. Learning switchable priors for neural image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(11):11567–11582, 2025. doi: 10.1109/TCSVT.2025.3577134.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

A MODEL DETAILS

This section introduces more detail of implementation of PULSE.

A.1 COMPONENT COMPLEXITY ANALYSIS

Table 4 reports the component accounting of PULSE. The MSE and perceptual models use the same deployed topology and therefore have the same parameter and MAC counts.

Table 4: Component accounting for PULSE.

Component	Sender	Receiver	PULSE		PULSE-S		PULSE-L	
			kMAC/px	Param.	kMAC/px	Param.	kMAC/px	Param.
Analysis transform	✓	×	43.22	12.1M	43.08	11.1M	43.21	12.1M
Hyper encoder	✓	×	0.45	0.30M	0.31	0.19M	0.93	0.66M
Hyper decoder	✓	✓	0.24	0.15M	0.16	0.09M	0.44	0.32M
Entropy model step-1	✓	✓	0.58	0.17M	0.48	0.13M	1.35	0.41M
Entropy model step-2	✓	✓	0.74	0.21M	0.61	0.16M	1.67	0.50M
Linear-CDF predictor	✓	✓	0.12	0.50M	0.08	0.33M	0.18	0.74M
Synthesis transform	×	✓	3.53	0.71M	1.43	0.26M	16.34	3.70M
Neural sender	✓	×	45.35	13.5M	44.73	12.0M	47.78	14.8M
Neural receiver	×	✓	5.21	1.74M	2.77	0.98M	19.98	5.69M

A.2 LATENCY ANALYSIS

Table 5 breaks down the end-to-end CPU latency of PULSE. Neural inference dominates the runtime, while non-neural processing takes only 7 ms for either encoding or decoding. PULSE-S retains the same encoding latency but lowers end-to-end decoding latency from 126 ms to 77 ms, a 38.9% reduction. PULSE-L is slower due to increased complexity.

Table 5: Single-thread CPU end-to-end latency of PULSE. *Others* includes non-neural processing, such as entropy coding and CDF indexing.

Model	Pipeline	Neural inference	Others	Total
PULSE-S	Encoding	913 ms	7 ms	920 ms
	Decoding	70 ms	7 ms	77 ms
PULSE	Encoding	913 ms	7 ms	920 ms
	Decoding	119 ms	7 ms	126 ms
PULSE-L	Encoding	963 ms	9 ms	972 ms
	Decoding	411 ms	9 ms	420 ms

A.3 CROSS-DEVICE ENTROPY CODING VALIDATION

As shown in Figure 1, we validate the linear CDF decoded index and latent-symbol decoding paths on three platforms: (1) a REDMI K80 Pro phone using the Snapdragon 8 Elite NPU, (2) an HP OmniBook Ultra Flip 14 laptop using only its Intel Core Ultra 9 288V CPU, and (3) a server using an NVIDIA A100 GPU. Although the laptop also includes an NPU, we disable NPU acceleration to characterize CPU-only performance. For all devices, we deploy entropy coding and linear CDF decoding on there CPUs. Table 6 reports zero CDF-index and decoded-symbol mismatches.

B TRAINING DETAILS

This section details the training process of PULSE.

Table 6: Cross-device entropy coding bit-exact validation. Mismatch counts both CDF-index and decoded-symbol mismatches.

Environment	Platform	Neural backend	CPU backend	Mismatch
Phone	REDMI K80 Pro	NPU: Snapdragon 8 Elite	Snapdragon 8 Elite	**0**
Laptop	HP OmniBook Ultra Flip 14	CPU: Intel Core Ultra 9 288V	Intel Core Ultra 9 288V	**0**
Server	GPU server	GPU: NVIDIA A100	AMD EPYC 9V84	**0**

B.1 Main Training Process

Table 7 summarizes the complete schedules. During training, we normalize the input images to $[-1, 1]$. When training on 512 low-resolution crops, we use the a subset of 4.8 M randomly sampled images from SAM-1B (Kirillov et al., 2023) and Open Images (Kuznetsova et al., 2020). When training on 1024 high-resolution crops, we use 4,035 images: 800 DIV2K images (Agustsson & Timofte, 2017), 2,650 Flickr2K images (Lim et al., 2017), and 585 CLIC training images (CLIC Organizers, 2020). The ROI stages use the same data as 1024 high-resolution crops, augmented with offline extracted face and text boxes.

Table 7: Training schedules.

Model	Stage	Crop	Steps	Batch	λ range	Learning-rate schedule
MSE	LR training	512	500k	8	24–672	10^{-3} (200k) $\rightarrow$ 10^{-4} (200k) $\rightarrow$ 10^{-5} (100k)
	HR training	1024	50k	8	24–672	10^{-4} (40k) $\rightarrow$ 10^{-5} (10k)
Perceptual	Stage I LR	512	300k	16	0.5–12.5	10^{-3} (200k) $\rightarrow$ 10^{-4} (100k)
	Stage I HR	1024	50k	16	0.5–12.5	10^{-4} (40k) $\rightarrow$ 10^{-5} (10k)
	Stage II	1024	20k	32	3–25	10^{-4} (10k) $\rightarrow$ 10^{-5} (10k)

B.2 More Details of Perceptual and ROI Optimization

Hyperparameters. For perceptual optimization, we set $\phi_{\text{MSE}} = 5$ and define $D_{\text{LPIPS}} = D_{\text{LPIPS-VGG}} + 0.5D_{\text{LPIPS-Alex}}$. We use $\phi^{\text{I}}_{\text{LPIPS}} = 1$ and $\phi^{\text{II}}_{\text{LPIPS}} = 0.25$. In Stage II, we set $\phi_{\text{GAN}} = 0.005$, $\phi_{\text{ROI}} = 0.0125$, $\phi_{\text{FaceNet}} = 1$, and $\phi_{\text{CRNN}} = 2$. We additionally apply ROI-restricted MSE and LPIPS supervision, each with an additional weight of 0.125.

GAN Training. For latent-conditioned adversarial training, we extract features from blocks 3, 6, 9, and 12 of a frozen DINOv2 ViT-B/14 backbone. At each block, the detached latent $\hat{y}$ is resized and projected, then concatenated with the image features for discrimination. We use a hinge discriminator loss and the negative mean fake-image logit as D_{GAN}.

Effectiveness of Perceptual Training Designs. As shown in Table 8, we examine the effectiveness of each design. LPIPS-first perceptual pre-training assigns a larger weight to the LPIPS loss, encouraging stronger perceptual detail synthesis, with consistent improvements in LPIPS-VGG, FID, and regional reconstruction quality. Adversarial training with a DINO-based projected GAN loss plays a key role in improving perceptual realism: removing the GAN loss substantially degrades FID by 49.8%. ROI training primarily benefits perceptually important regions, with its removal causing large degradations of 37.7% and 20.8% in face and text PSNR, respectively, while having a relatively limited effect on the overall metrics. These designs provide complementary benefits, and their combination achieves the best overall perceptual quality.

Table 8: Ablation study on perceptual training designs. BD-Rate is measured.

Training variant	PSNR	LPIPS-VGG	FID	Face PSNR	Text PSNR
Stage I w/o high LPIPS weight	-0.4%	+6.0%	+5.7%	+6.1%	+1.6%
Stage II w/o GAN Loss	–3.6%	–6.4%	+49.8%	–4.1%	–4.5%
Stage II w/o ROI	+0.2%	–2.4%	–3.7%	+37.7%	+20.8%
Proposed perceptual optimization	0.0%	0.0%	0.0%	0.0%	0.0%

B.3 META PRIOR POST-TRAINING

After main training, we expand the factorized prior into $N = 64$ banks for each QP and fit them in two stages. All other codec parameters remain fixed during this procedure.

Complexity-stratified initialization. For each hyperlatent location, we compute three luminance statistics from the corresponding input region: mean absolute gradient, local standard deviation, and coarse-scale standard deviation interpolated to the hyperlatent grid. Each statistic is normalized by its 75th percentile over the fitting set and clipped to $[0, 4]$. Their weighted sum, with weights $(0.5, 0.3, 0.2)$, defines the complexity score. We partition these locations into N quantile bins, copy the pretrained QP-specific factorized prior into every bank, and fit each bank to the hyperlatent vectors in its bin. This initializes banks specialized to different local content complexities, rather than assigning one bank to an entire image.

Spatially adaptive hard-EM refinement. We replace the complexity-based assignments with position-wise minimum-cost assignments. In detail, we adopt an expectation-maximization (EM) procedure to jointly optimize the prior models and their assignment. The hard E-step selects

$$k^{\star}_{q,r} = \arg \min_{k \in \{1,\dots,N\}} \left[-\sum_{c=1}^{Z} \log_2 p_{q,k,c}(\hat{z}_{q,r,c}) \right], \tag{13}$$

where $p_{q,k,c}$ is the discrete probability mass of channel c in bank k at QP q. The M-step refits each bank by minimizing its normalized cross-entropy on the channel-wise symbol histograms of its currently assigned vectors. Thus, different positions in the same image can use different banks, regardless of their initial complexity bins.

We fit Meta Prior on hyperlatents extracted from 512×512 crops of 4,035 high-resolution images. Fitting starts with a 500-step initialization, followed by six 300-step refinements, totaling 2,300 optimizer steps per QP. Each fitting stage uses a learning rate of 10^{-2} with cosine decay to 2×10^{-4}. Optimization is full-batch over the aggregated hyperlatent histograms for each QP.

C EVALUATION DETAILS

This section details the training process of PULSE.

C.1 TIME MEASUREMENT

Latency is measured on a single CPU thread at 1088×1920, without hardware codec acceleration. Neural inference uses batch size one and one inference stream.

HEIC. We use libheif 1.17.6 with x265 3.5 for encoding and libde265 1.0.15 for decoding. Encoding uses 8-bit YCbCr 4:2:0, `preset=slow`, and `tune=psnr`, with additional codec worker threads disabled. We also evaluated YCbCr 4:4:4, but observed worse rate–distortion performance and lower coding speed, possibly reflecting less effective optimization for 4:4:4 in this implementation. The benchmark operates on in-memory RGB inputs and bitstreams. Disk I/O is excluded.

HM & VTM. We use HM-16.25 and VTM-17.0 reference implementations with 8-bit YUV 4:4:4 and intra-only coding. Their timings use the native YUV interface and do not include RGB conversion. To reduce the effect of fixed command-line initialization overhead, we measure identical 5-frame and 15-frame runs and estimating the per-picture latency as $(T_{15} - T_5)/10$.

Neural codecs. For compared neural codec, we evaluate the supported combinations of BF16, FP16, and FP32 precision with OpenVINO and PyTorch, and report the fastest measured configuration under the same single-thread constraint. Model loading, graph conversion and compilation, and warm-up are excluded from steady-state inference timing. Entropy coding is included in the reported codec latency, using the same rANS implementation as in PULSE for consistent timing.

C.2 BASELINE IMPLEMENTATIONS

For all methods, we use their officially released implementations, with the following exceptions.

For **HiFiC** (Mentzer et al., 2020), we were unable to obtain usable official pretrained models during our experiments in September 2026. We therefore use the officially published rate–distortion measurements released with TensorFlow Compression.[1].

For **MS-Hyperprior** (Minnen et al., 2018), **ELIC** (He et al., 2022b), and **MRIC** (Agustsson et al., 2023), our measurements use third-party implementations and their pretrained models. Specifically, we use CompressAI's `mbt2018-mean` model for MS-Hyperprior, VincentChandelier's reimplementation for ELIC,[2] and Nikolai10's TensorFlow reimplementation for MRIC.[3]

For **PICO** (Tatwawadi et al., 2026), we reproduce their architecture based on the published specifications for complexity and latency measurements. PICO does not release an implementation, but provides reconstructions for a 300-image subset of the CLIC 2020 test set (CLIC Organizers, 2020). We therefore use the same subset for comparison in Figure 7 and Table 9. PULSE-L is $2.1\times$ and $2.9\times$ faster for encoding and decoding, respectively, while achieving better BD-Rate on PSNR and FID.

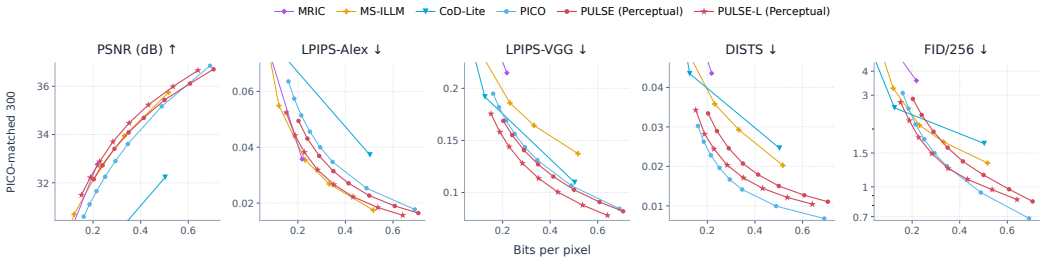


Figure 7: Rate–distortion curves on the 300-image CLIC test subset released by PICO.

Table 9: BD-rate comparison on PICO-matched 300-image CLIC test subset. Anchor: MS-ILLM.

Method	PSNR	LPIPS-Alex	LPIPS-VGG	DISTS	FID
MRIC	-1.2%	+9.7%	+58.5%	+51.1%	+98.4%
MS-ILLM	0%	0%	0%	0%	0%
CoD-Lite	+160%	+109.4%	-39.7%	-1.3%	+9.4%
PICO	+18.2%	+50.7%	-30.6%	-52.3%	-6%
PULSE (Perceptual)	**+2.2%**	**+31.1%**	**-38.2%**	**-29.3%**	**+12.7%**
PULSE-L (Perceptual)	**-7.1%**	**+8.9%**	**-48.9%**	**-45.2%**	**-18.2%**

C.3 Comparison on More Benchmarks

In Figures 8, 9 and Tables 11–15, we present rate-distortion curves and BD-Rates across more metrics and benchmarks. We evaluate PSNR, LPIPS with VGG and AlexNet, DISTS (Ding et al., 2022), and FID on Kodak, Tecnick, DIV2K, CLIC Professional validation, and CLIC test sets. We compute FID using overlapping 64×64 patches for Kodak and overlapping 256×256 patches for all other benchmarks. Across all benchmarks and metrics, PULSE achieves competitive performance with substantially lower complexity.

C.4 Additional Visual Examples

Figure 11 and Figure 12 provide additional visual comparison examples on text and natural content. We do not include facial comparison to avoid personally identifiable information.

[1] `https://github.com/tensorflow/compression/blob/master/models/hific/data.csv`

[2] `https://github.com/VincentChandelier/ELiC-ReImplemetation`

[3] `https://github.com/Nikolai10/MRIC`

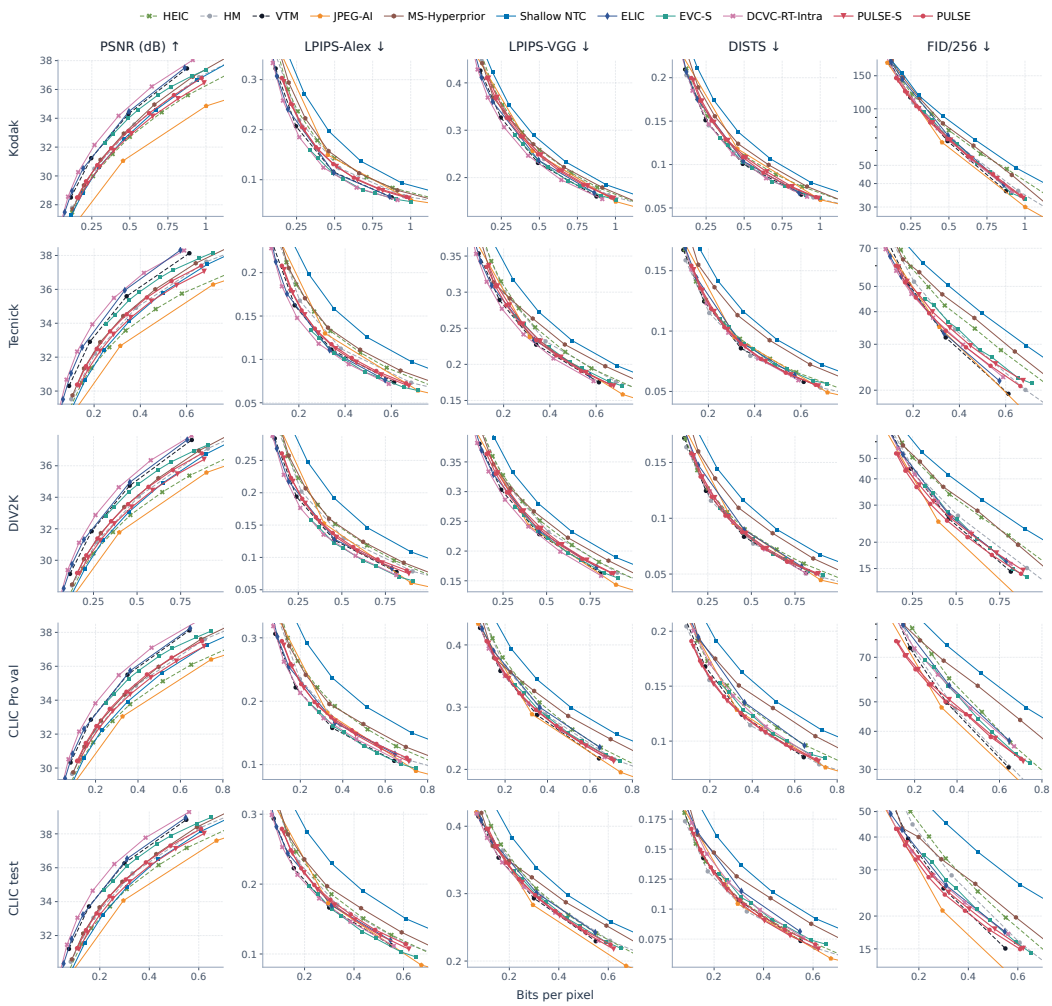


Figure 8: Rate-distortion curves on more benchmarks for MSE-optimized codecs.

D AGENTIC EVOLUTION PROTOCOL

All E1–E7 generations follow the same human–LLM agentic research loop described in the paper.

Table 10: **Illustrations on each evolution.**

Transition	Main change	Motivation
E1→E2	Move rendering to a $1/4$-resolution grid	relieve the narrow full-resolution renderer
E2→E3	Replace four narrow blocks with one wider block	concentrate low-resolution capacity
E3→E4	Increase main and hyper latent capacity	address pressure in shared representations
E4→E5	Introduce grouped pointwise convolution	reclaim entropy-model computation
E5→E6	Reallocate internal width to the main latent	strengthen shared latent capacity
E6→E7	Reduce FFN channels and increase decoder depth	improve nonlinear decoder processing
E7→stop	Train three further candidates	none improves E7

D.1 IMPLEMENTATION DETAILS

For LLM agent, we use `gpt-5.6-sol` with high reasoning effort. The model receives the probe summaries, network graph, parameter-sharing constraints, component-level MAC accounting, and the complete history of earlier rounds. Each interaction produces two to four candidate suggestions.

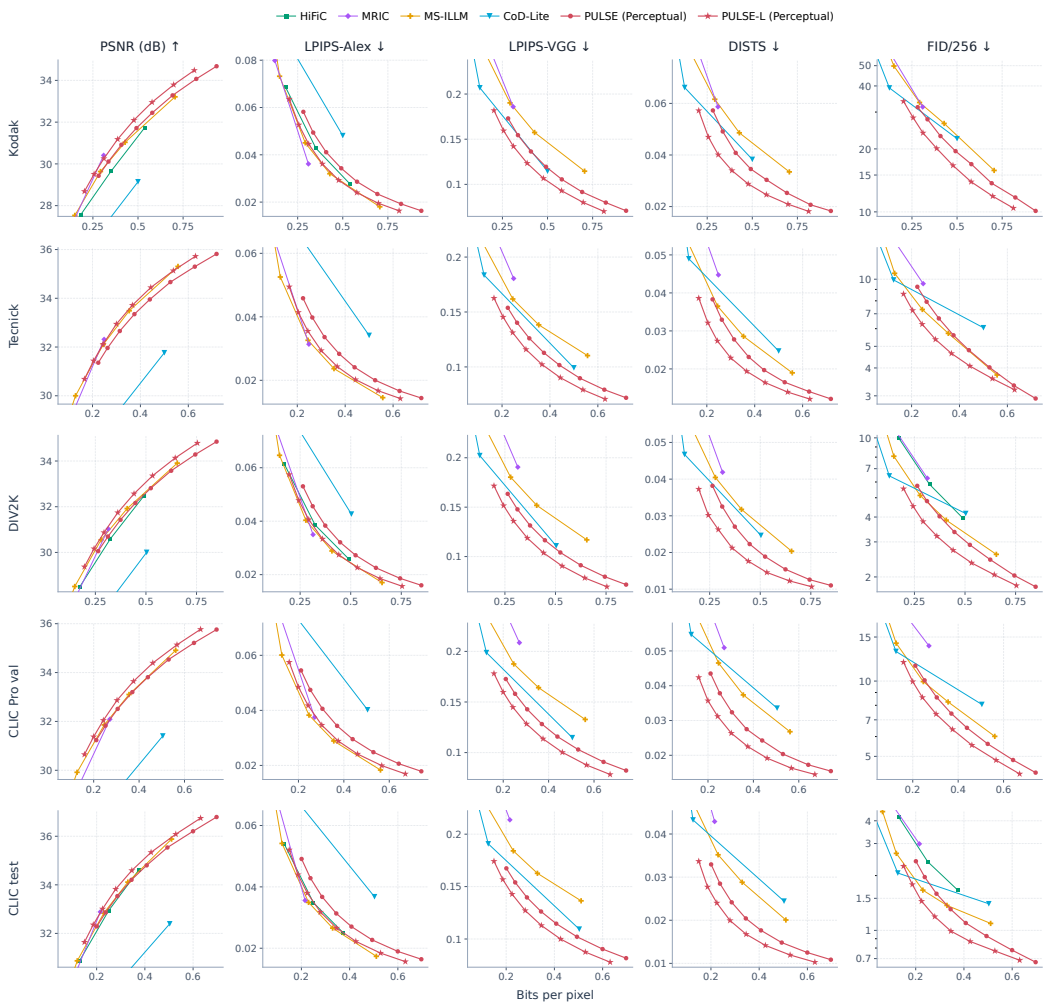


Figure 9: Rate-distortion curves on more benchmarks for perceptual-optimized codecs.



Figure 10: Full agentic evolution process and final parameters at different scales.

A human researcher checks their interpretation, implementation feasibility, and budget accounting, then fixes three architectures for full training in every round.

Table 11: **BD-Rate comparison on PSNR.** Anchor: HM-16.25.

Method	Kodak	Tecnick	DIV2K	CLIC pro val	CLIC test
HEIC	+14.3%	+28.4%	+21.1%	+25.2%	+16.9%
HM-16.25	0%	0%	0%	0%	0%
VTM-17.0	-21.3%	-23.8%	-22.8%	-23.4%	-23%
JPEG-AI	+54.3%	+41.2%	+33%	+37.4%	+33.2%
MS-Hyperprior	-2.2%	-4.6%	-3.3%	-1.7%	-2%
Shallow NTC	+6.3%	+9.9%	+9.3%	+13%	+11.9%
ELIC	-25.1%	-32%	-27.6%	-26.2%	-26.2%
EVC-S	-12.6%	-14.3%	-13.5%	-14.3%	-13.2%
DCVC-RT-Intra	-31.1%	-36.4%	-33.3%	-33.2%	-33.7%
PULSE-S	**+9.9%**	**+7.7%**	**+6.5%**	**+5.9%**	**+9.4%**
PULSE	**+1.2%**	**-2.1%**	**-2.4%**	**-2.9%**	**+0.3%**
HiFiC	+61.9%	—	+37.3%	—	+45.1%
MRIC	+22.2%	+18.3%	+22.3%	+46.1%	+29.1%
MS-ILLM	+39.6%	+23%	+23.3%	+38%	+38.7%
CoD-Lite	+167.9%	+162.2%	+145.8%	+218.8%	+232.8%
PULSE (Perceptual)	+38.8%	+42.4%	+33.7%	+45.5%	+49.6%
PULSE-L (Perceptual)	+24.4%	+26%	+19.4%	+30.8%	+34.5%

Table 12: **BD-Rate comparison on LPIPS-Alex.** Anchor: MS-ILLM.

Method	Kodak	Tecnick	DIV2K	CLIC pro val	CLIC test
HEIC	+890%	+820%	+900%	+1420%	+1330%
HM-16.25	+840%	+800%	+880%	+1340%	+1450%
VTM-17.0	+740%	+690%	+790%	+1140%	+1250%
JPEG-AI	+830%	+700%	+760%	+1130%	+1120%
MS-Hyperprior	+1010%	+920%	+1000%	+1630%	+1640%
Shallow NTC	+1530%	+1300%	+1540%	+2410%	+2360%
ELIC	+770%	+600%	+760%	+1100%	+1110%
EVC-S	+530%	+710%	+700%	+1320%	+1460%
DCVC-RT-Intra	+710%	+550%	+680%	+970%	+980%
PULSE-S	+850%	+770%	+880%	+1410%	+1500%
PULSE	+860%	+770%	+890%	+1410%	+1520%
HiFiC	+12%	—	+9.4%	—	+7.8%
MRIC	-8.7%	+7.7%	+6.6%	+23.1%	+10.8%
MS-ILLM	0%	0%	0%	0%	0%
CoD-Lite	+61.2%	+98.7%	+70.3%	+96.8%	+111.5%
PULSE (Perceptual)	**+25.2%**	**+31.7%**	**+25.6%**	**+32.2%**	**+31.9%**
PULSE-L (Perceptual)	**+4.9%**	**+10.4%**	**+5.8%**	**+9.9%**	**+9.4%**

All three candidates are trained from scratch with the same data, QP range, optimizer, and schedule. The human researcher reviews the measured rate–distortion curves and promotes the best improving candidate. If no candidate improves the current parent, the search terminates.

D.2 Full Evolution Trajectory

As shown in Figure 10, the reported search contains one initial model training, seven rounds with three trained candidates per round, and several finalized model trainig at different scale. Six rounds promote E2 through E7. In the final round after E7, all three candidates underperform E7, so the search stops. Table 10 records the changes and motivations at high level.

Table 13: **BD-Rate comparison on LPIPS-VGG.** Anchor: MS-ILLM.

Method	Kodak	Tecnick	DIV2K	CLIC pro val	CLIC test
HEIC	+330%	+340%	+290%	+520%	+530%
HM-16.25	+300%	+330%	+270%	+490%	+560%
VTM-17.0	+260%	+270%	+230%	+480%	+540%
JPEG-AI	+270%	+240%	+200%	+360%	+370%
MS-Hyperprior	+300%	+350%	+290%	+570%	+580%
Shallow NTC	+460%	+540%	+450%	+950%	+1000%
ELIC	+310%	+280%	+270%	+590%	+600%
EVC-S	+180%	+260%	+190%	+470%	+530%
DCVC-RT-Intra	+270%	+250%	+230%	+500%	+510%
PULSE-S	+310%	+300%	+270%	+500%	+570%
PULSE	+300%	+280%	+250%	+510%	+580%
HiFiC	—	—	—	—	—
MRIC	+8.9%	+37.8%	+39.3%	+61.1%	+60.1%
MS-ILLM	0%	0%	0%	0%	0%
CoD-Lite	-39.3%	-26.6%	-34.8%	-36.9%	-38.3%
PULSE (Perceptual)	**-24.7%**	**-25.3%**	**-28.6%**	**-35.8%**	**-38.1%**
PULSE-L (Perceptual)	**-38.8%**	**-37%**	**-41%**	**-47.5%**	**-48.9%**

Table 14: **BD-Rate comparison on DISTS.** Anchor: MS-ILLM.

Method	Kodak	Tecnick	DIV2K	CLIC pro val	CLIC test
HEIC	+550%	+590%	+570%	+1070%	+920%
HM-16.25	+480%	+560%	+500%	+960%	+960%
VTM-17.0	+570%	+630%	+610%	+1160%	+1130%
JPEG-AI	+560%	+550%	+540%	+910%	+860%
MS-Hyperprior	+680%	+890%	+790%	+1500%	+1410%
Shallow NTC	+980%	+1090%	+1070%	+2090%	+1920%
ELIC	+670%	+680%	+710%	+1290%	+1190%
EVC-S	+420%	+650%	+560%	+1270%	+1330%
DCVC-RT-Intra	+630%	+620%	+660%	+1200%	+1090%
PULSE-S	+640%	+680%	+660%	+1130%	+1160%
PULSE	+630%	+670%	+630%	+1160%	+1210%
HiFiC	—	—	—	—	—
MRIC	+11.4%	+47.5%	+33.3%	+40.3%	+51.2%
MS-ILLM	0%	0%	0%	0%	0%
CoD-Lite	-35.3%	+2.6%	-26.4%	-7%	+1.2%
PULSE (Perceptual)	**-22.6%**	**-12.5%**	**-22.4%**	**-28.8%**	**-29.6%**
PULSE-L (Perceptual)	**-43%**	**-32.4%**	**-42.8%**	**-47.1%**	**-45.6%**

Table 15: **BD-Rate comparison on FID.** Anchor: MS-ILLM.

Method	Kodak	Tecnick	DIV2K	CLIC pro val	CLIC test
HEIC	+770%	+1770%	+1940%	+2680%	+4820%
HM-16.25	+630%	+1630%	+1700%	+2450%	+4890%
VTM-17.0	+730%	+1480%	+1700%	+2660%	+3820%
JPEG-AI	+530%	+1140%	+1130%	+1720%	+2540%
MS-Hyperprior	+800%	+2060%	+2120%	+3830%	+4980%
Shallow NTC	+1010%	+2800%	+2990%	+5590%	+8750%
ELIC	+970%	+1360%	+1750%	+3430%	+3820%
EVC-S	+480%	+1820%	+1620%	+3040%	+4890%
DCVC-RT-Intra	+840%	+1330%	+1700%	+3200%	+3510%
PULSE-S	+630%	+1650%	+1600%	+2530%	+4010%
PULSE	+640%	+1520%	+1480%	+2420%	+3700%
HiFiC	—	—	+42.8%	—	+85.1%
MRIC	+8.1%	+63.8%	+55.6%	+101%	+107.1%
MS-ILLM	0%	0%	0%	0%	0%
CoD-Lite	-26.4%	+17.2%	-6.4%	+12.9%	+14.2%
PULSE (Perceptual)	**-16.7%**	**+12%**	**-1.7%**	**-4.1%**	**+17.8%**
PULSE-L (Perceptual)	**-33.3%**	**-17.9%**	**-27.3%**	**-22.5%**	**-17.2%**



Figure 11: More visual examples on text contents.

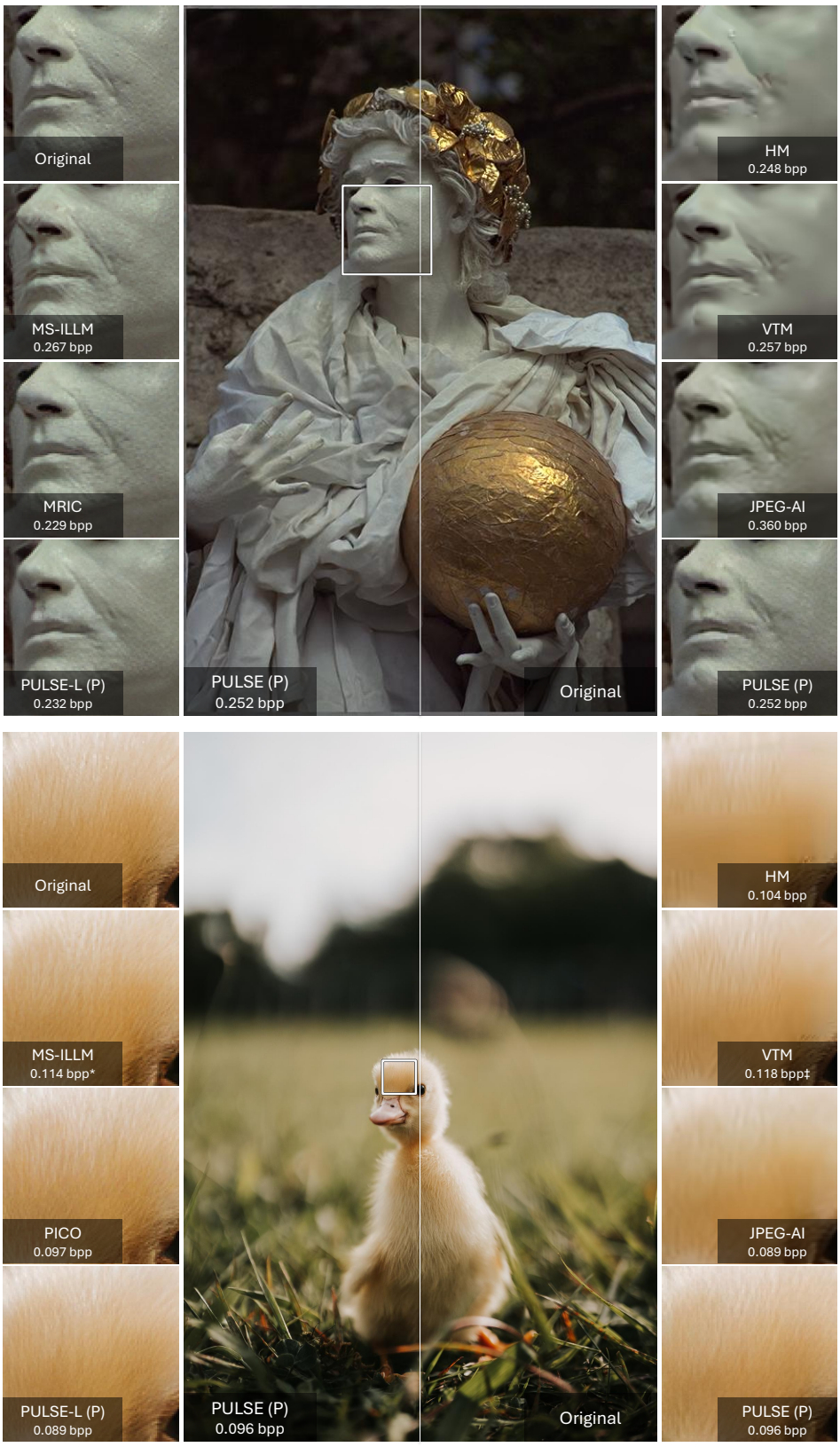


Figure 12: More visual examples on natural contents.