SCALING CONTEXT FOR UNIFIED VARIABLE-RATE AND VARIABLE-COMPLEXITY IMAGE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned image compression (LIC) increasingly requires flexible rate–distortion–complexity (R–D–C) control for heterogeneous deployment. We introduce *context scaling*, a complementary scaling dimension that varies autoregressive context modeling within a fixed codec, rather than changing model capacity. By adjusting the context-modeling schedule, a single codec can jointly support variable rate and variable complexity while reusing the same transform output across complexity levels. Interestingly, we find that the benefit of context scaling is strongly input dependent: some images saturate with coarse context modeling, whereas others continue to benefit from finer-grained conditioning. Based on this observation, we introduce an adaptive context schedule that assigns finer context modeling to inputs with larger expected gains under a given complexity budget. Experiments on standard benchmarks show that context scaling provides effective R–D–C scalability, while adaptive scheduling consistently outperforms fixed schedules at matched average complexity. These results establish context modeling as a controllable complexity dimension orthogonal to model-capacity scaling, and reveal input-dependent context allocation as an effective mechanism for improving compression efficiency under constrained computation.

1 INTRODUCTION

Learned image compression (LIC) (Ballé et al., 2018; Minnen et al., 2018; Cheng et al., 2020; He et al., 2022; Li et al., 2025b; Cong et al., 2026; Zhang et al., 2026) has made substantial progress in rate–distortion performance, yet practical deployment still poses important challenges. Two representative requirements are support for varying network bandwidths and devices with heterogeneous computational capabilities. Without such flexibility, multiple models must be separately trained, stored, and maintained for different bitrate and complexity requirements. Variable-rate compression (Choi et al., 2019; Chen & Ma, 2020; Tong et al., 2023; Arezki et al., 2026; Ta et al., 2026) has been extensively studied, whereas variable-complexity compression (Guo et al., 2023; Li et al., 2025a; Wang et al., 2026b) has recently attracted increasing attention. Existing approaches typically achieve complexity adaptation by varying active network capacity or execution paths. Some couple complexity with bitrate, as changing the active capacity can also alter the latent representation and its corresponding bitrate (Yang et al., 2021; Li et al., 2025a), while others decouple the two but primarily scale computation in the transform or reconstruction modules (Guo et al., 2023).

A complementary scaling paradigm has recently emerged in large language models and generative models, where performance can be improved by scaling the inference trajectory (e.g., through iterative reasoning or denoising) without increasing model capacity (Snell et al., 2025; Ma et al., 2025). This raises a natural question for learned compression: *can compression performance be similarly scaled by varying the inference trajectory within a fixed codec?* In LIC, entropy modeling provides a promising opportunity for such scaling. Autoregressive entropy models (Han et al., 2024) estimate conditional distributions of latents by progressively incorporating already decoded information, forming a sequential context-modeling process. From an information-theoretic perspective (Cover et al., 1991), richer conditioning can reduce the conditional uncertainty of the variables being coded and enable more accurate probability estimation. Varying the number of context steps therefore changes both the available conditioning context and the granularity of conditional modeling, while incurring different levels of sequential computation. This makes context modeling a direct and controllable dimension for scaling compression performance within a fixed codec.

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.

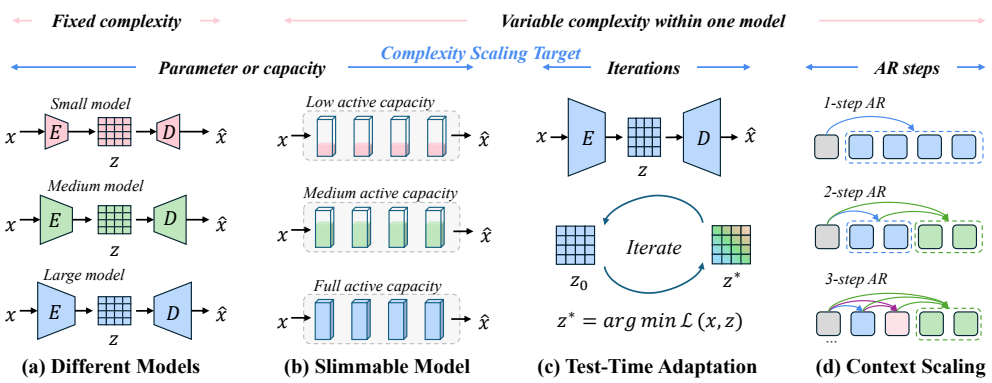


Figure 1: Comparison of complexity-scaling paradigms in LIC. Existing approaches vary model parameters, active capacity, or iterative optimization, whereas our *context scaling* adjusts the autoregressive context schedule within a unified codec, providing a complementary dimension for complexity scaling.

In this work, we introduce *context scaling*, which varies the autoregressive context-modeling schedule within a unified codec to control inference complexity. Since complexity scaling is confined to entropy modeling, transform computation can be shared across different complexity levels, avoiding the complexity-specific transform passes required by slimmable approaches (Yang et al., 2021). This property is particularly attractive for content distribution and multi-profile encoding.

A key observation is that the benefit of context scaling varies substantially across inputs. Some images obtain most of their achievable gain with only a few context steps, whereas others continue to benefit from finer-grained context modeling. Consequently, applying the same context schedule to all images can be inefficient under a limited average complexity budget, as computation may be over-allocated to saturated inputs and under-allocated to those with larger potential gains. We therefore introduce an adaptive context schedule that assigns finer-grained context modeling to inputs expected to benefit most under a given average complexity budget.

Experiments on standard benchmarks demonstrate that context scaling consistently improves rate–distortion performance and enables a unified codec to span a broad range of R–D–C operating points. Moreover, the proposed adaptive context schedule effectively exploits input-dependent scaling behavior, consistently outperforming fixed schedules under matched average complexity.

Our main contributions are summarized as follows:

- We introduce *context scaling* for learned image compression, establishing autoregressive context modeling as a controllable complexity dimension that is orthogonal to model-capacity scaling and can be combined with it to further improve rate–distortion performance.
- We develop a unified variable-rate and variable-complexity codec that jointly controls the rate condition and context schedule, covering a broad range of R–D–C operating points while sharing the same transform computation across complexity levels.
- We reveal strong input-dependent context-scaling behavior and introduce an adaptive context schedule that assigns finer-grained context modeling to inputs with larger expected gains under a given complexity budget. The expected gain is estimated from hyperprior information already available during coding, requiring no additional signaling overhead.

2 Related Work

Model and inference scaling. Scaling laws have revealed predictable performance gains with increased model size, data, and training compute across language (Kaplan et al., 2020; Hoffmann et al., 2022), vision (Zhai et al., 2022), and generative models (Peebles & Xie, 2023). Beyond parameter scaling, recent generative modeling methods also treat the inference trajectory as a controllable computational dimension. For example, DDIM (Song et al., 2021) varies the denoising process to trade

sequential computation for generation quality. In LIC, Li et al. (2025c) scale model size from 0.1 billion to 1 billion parameters and observe systematic performance gains with model size and training compute. Orthogonal to parameter scaling, we scale the conditional entropy-modeling process by varying the autoregressive context schedule, providing a complementary complexity dimension within one codec.

Complexity-scalable coding. Existing complexity-scalable codecs mainly vary network capacity or execution paths. Slimmable approaches, such as SlimCAE (Yang et al., 2021) and One-is-All (Li et al., 2025a), expose multiple operating points by activating subnetworks of different widths, which can couple complexity with bitrate as the active representation capacity changes. CBANet (Guo et al., 2023) instead decouples bitrate and complexity through separate adaptation modules. Other methods derive complexity-specific codecs from a shared high-capacity model or supernet via knowledge distillation, e.g., EVC (Wang et al., 2023) and CSLIC (Wang et al., 2026b), but still require different transform configurations across complexity levels.

A complementary direction performs content-adaptive computation within an image. AdaNIC (Tao et al., 2023) and RDC optimization (Gao et al., 2024; Hang et al., 2025) allocate different transform capacities or computation to different spatial regions according to image content. Test-time adaptation provides another complexity axis by optimizing latent representations and/or model parameters for each input (Yang et al., 2020; Chen et al., 2025), but typically relies on iterative gradient-based optimization, which may be unsuitable for deployment in certain scenarios.

3 Context Scaling: Conditional Modeling and Scaling Behavior

We first establish autoregressive (AR) context modeling as a scalable computational dimension for LIC. We show that varying the context schedule changes the conditional factorization used for entropy coding without modifying model capacity, and then study its empirical scaling behavior alongside conventional parameter scaling. Our analysis reveals a clear contrast between global and individual scaling behavior: context computation follows a systematic scaling trend at the dataset level, while its marginal benefit varies substantially across individual images.

Scalable autoregressive context modeling. Let $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ denote the quantized main latent and hyper-latent, respectively. Given an AR schedule with S sequential steps, we partition $\hat{\mathbf{y}}$ into ordered groups $\{\mathcal{G}_t^{(S)}\}_{t=1}^{S}$, where $\mathcal{G}_{<t}^{(S)} = \bigcup_{\tau<t} \mathcal{G}_\tau^{(S)}$ contains the latents decoded before step t. Assuming parallel probability prediction within each group, the entropy model factorizes as

$$p_\theta^{(S)}(\hat{\mathbf{y}} \mid \hat{\mathbf{z}}) = \prod_{t=1}^{S} \prod_{j \in \mathcal{G}_t^{(S)}} p_\theta\left(\hat{y}_j \mid \hat{\mathbf{z}}, \hat{\mathbf{y}}_{\mathcal{G}_{<t}^{(S)}}\right). \tag{1}$$

The context schedule therefore determines the conditional factorization of the latent distribution: coarser schedules predict larger groups in parallel with less available context, whereas finer schedules expose more previously decoded latents to subsequent probability prediction. From an information-theoretic perspective, richer conditioning can reduce conditional uncertainty and thus provide a stronger basis for entropy estimation. Consequently, finer context modeling can improve coding efficiency at the cost of additional sequential computation. Importantly, varying S does not change the model parameters or transform; it changes the context-conditioned entropy-modeling process by controlling how extensively dependencies among latent variables are exploited. We refer to this computation–compression trade-off as *context scaling*.

Global scaling behavior. We next examine whether context scaling yields a systematic trend comparable to conventional parameter scaling. Fig. 2(a) shows the joint BD-rate landscape over model size N and AR context steps S. Performance improves consistently along both axes, suggesting that model capacity and context computation form two distinct scaling dimensions, while the iso-KMAC contours illustrate their trade-off under a shared computational budget.

To characterize the two scaling axes, we fit the empirical performance with the same asymptotic power-law form

$$\mathrm{BDR}(u) = b_u + a_u u^{-\alpha_u}, \qquad u \in \{N, S\}, \tag{2}$$

where BDR denotes BD-rate, b_u is the fitted asymptote, and a_u and α_u control the magnitude and rate of scaling, respectively. The parameters are fitted independently for the two scaling axes.

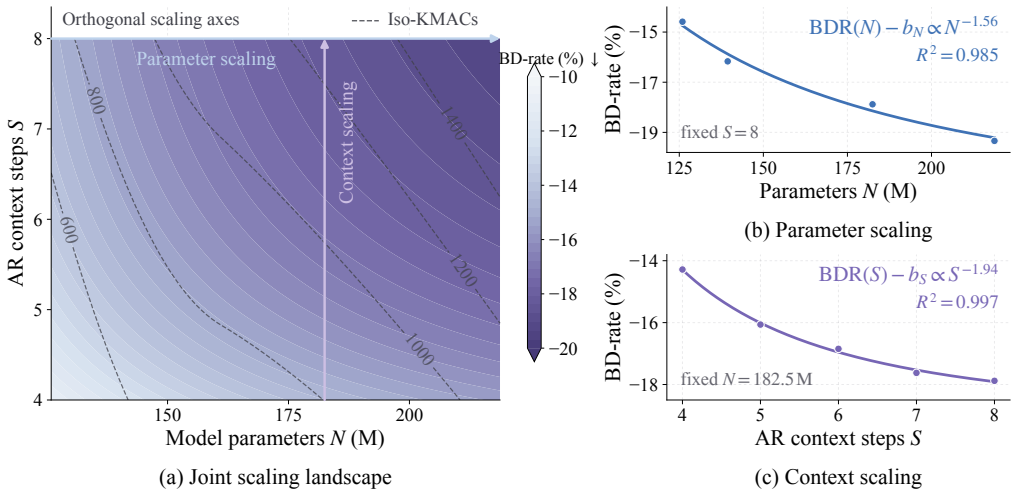


Figure 2: Global scaling behavior along parameter and context dimensions. (a) Joint BD-rate landscape over model size N and AR context steps S, with dashed iso-KMAC contours. (b–c) Parameter scaling at fixed $S = 8$ and context scaling at fixed $N = 182.5$M, respectively, both well approximated by the same asymptotic power-law form. Similar fits are observed across other settings.

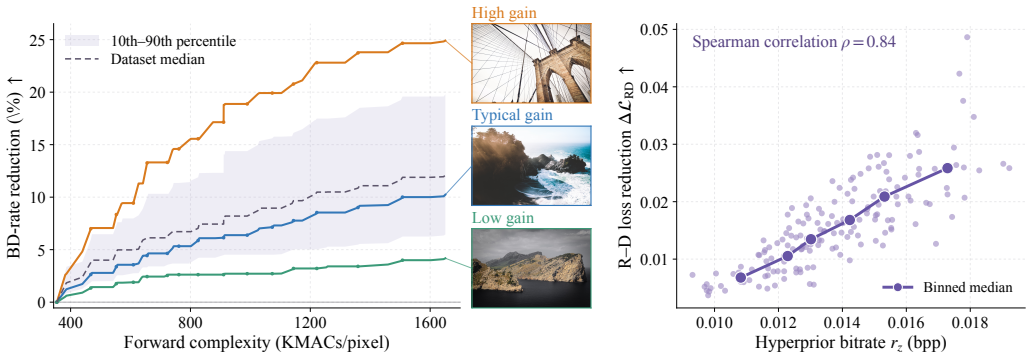
At fixed $S = 8$, parameter scaling is well fit by the asymptotic form in Eq. 2 with $\alpha_N = 1.56$ and a coefficient of determination of $R^2 = 0.985$ (Fig. 2(b)). Likewise, at a fixed model size of $N = 182.5$M, varying only the context schedule yields $\alpha_S = 1.94$ and $R^2 = 0.997$ (Fig. 2(c)). Similar empirical fits are observed across other model sizes and context schedules, with additional results provided in Appendix C. These results establish context computation as a complementary scaling axis: parameter scaling increases model capacity, whereas context scaling refines conditional entropy modeling within a fixed model. The two can therefore be combined to improve rate–distortion performance under different computational budgets.

Input-dependent scaling and predictability. The dataset-level scaling trend conceals substantial variation across individual images. Fig. 3(a) shows the per-image BD-rate gain relative to the lowest-complexity context configuration as the context schedule becomes finer. Although the average gain increases with context computation, the image-wise distribution also broadens: some images saturate after only a few context steps, whereas others continue to benefit from finer-grained context modeling. Representative high-, median-, and low-gain examples further illustrate these distinct scaling behaviors.

The variation is strongly associated with side information already available from the hyperprior. As shown in Fig. 3(b), the realized hyperprior bitrate exhibits a Spearman rank correlation of $\rho = 0.84$ with the context-scaling gain, with higher hyperprior rates generally corresponding to larger gains from finer-grained context modeling. Intuitively, a larger hyperprior rate suggests that more side information is required to characterize the statistics of the main latent, making it a useful empirical indicator of latent modeling difficulty. Since the hyperprior is available before autoregressive coding of the main latent, its bitrate provides a natural signal for estimating the expected scaling benefit before selecting the context schedule. Together with the global scaling behavior, these results reveal a clear contrast between systematic dataset-level scaling and substantial image-wise variation, motivating an adaptive context schedule under a given complexity budget.

4 Unified Codec with Adaptive Context Scaling

The observations in Sec. 3 establish context modeling as a controllable scaling dimension, while revealing substantial variation in its image-wise benefit. Building on these findings, we develop a unified codec that jointly supports variable rate and variable context complexity, and further adapts the context schedule to each input under a prescribed complexity budget.



(a) Image-wise variation

(b) Gain predictability

Figure 3: Input-dependent context scaling and its predictability. (a) Per-image BD-rate gains relative to the lowest-complexity context configuration exhibit substantial variation as the context schedule becomes finer. The solid curve denotes the median and the shaded region the 10th–90th percentiles; representative images illustrate different scaling behaviors. (b) The context-scaling gain is strongly correlated with hyperprior bitrate (Spearman $\rho = 0.84$), making it a useful early indicator of the benefit from finer-grained context modeling.

4.1 Unified Rate– and Context–Conditioned Codec

We construct a single codec that supports multiple rate and context-complexity operating points. Let $q \in \mathcal{Q}$ denote the rate condition and $c \in \mathcal{C}$ index a context schedule. The former controls the rate–distortion operating point, while the latter determines the granularity of autoregressive entropy modeling.

Rate and context conditioning. Both conditions are represented by learned embeddings and injected through AdaLN-based modulation (Peebles & Xie, 2023). For variable-rate coding, we follow common practice and apply the rate condition to the transform modules (Duan et al., 2024; Jia et al., 2025). In contrast, the context condition is applied to the entropy-modeling blocks. For a fixed q, all context schedules reuse the same transform outputs $(\mathbf{y}, \mathbf{z})$, while varying c changes the context-conditioned distribution parameters used for quantization and entropy coding of $\mathbf{y}$. This allows the transformed representation to be reused across complexity levels without recomputing the analysis transform.

During training, rate–context pairs (q, c) are jointly sampled from the supported configuration space. The shared codec is optimized with

$$\mathcal{L} = \mathbb{E}_{q,c}\left[R(\hat{\mathbf{y}}_{q,c}, \hat{\mathbf{z}}_q; q, c) + \lambda_q D(\mathbf{x}, \hat{\mathbf{x}}_{q,c})\right]. \tag{3}$$

where R denotes the expected coding rate of the quantized latents and varies with the context schedule through the conditional factorization of $\hat{\mathbf{y}}$. Since c does not affect the transform representation, it can be selected after the hyper-latent is obtained but before autoregressive coding of $\hat{\mathbf{y}}$, which enables the adaptive strategy introduced below.

Nested context schedules. We construct multiple context-complexity levels from a common finest partition of $\hat{\mathbf{y}}$. Starting from the maximum-step schedule, lower-complexity configurations are obtained by progressively merging adjacent groups, yielding a nested family of context schedules. Fewer groups increase parallelism and reduce sequential computation, while finer partitions provide more decoded context for subsequent probability prediction.

To preserve the most useful conditional structure, the merging is non-uniform. Motivated by prior observations that fine-grained context modeling is particularly effective in earlier coding stages (He et al., 2022; Han et al., 2024), we retain early group boundaries and preferentially merge later groups. This design provides controllable context complexity within a shared entropy model and, together with the rate condition, enables a single codec to span a broad range of R–D–C operating points.

4.2 Adaptive Context Scaling under a Complexity Budget

The image-wise analysis in Sec. 3 shows that the marginal gain from finer context modeling varies substantially across inputs. A uniform context schedule is therefore inefficient under a limited average complexity budget. We formulate context scheduling as a budgeted allocation problem and use hyperprior information to adapt the schedule online.

Budgeted context allocation. For image $\mathbf{x}_i$ at rate condition q, let $J_i(c;q) = R_i(c;q) + \lambda_q D_i(c;q)$ denote the R–D cost under context schedule c, and let $C(c)$ denote its computational complexity. For a stream of M images with target average complexity B, the ideal allocation is

$$\min_{\{c_i\}_{i=1}^{M}} \quad \frac{1}{M}\sum_{i=1}^{M} J_i(c_i;q) \qquad \text{s.t.} \quad \frac{1}{M}\sum_{i=1}^{M} C(c_i) \le B, \qquad c_i \in \mathcal{C}. \tag{4}$$

If all candidate-wise R–D costs were known, Eq. 4 would yield an offline oracle allocation. In online coding, however, the schedule must be selected before its coding gain is observed.

Hyperprior-guided gain estimation. We use the correlation identified in Fig. 3 to predict the benefit of finer context modeling from the hyperprior bitrate. Relative to the lowest-complexity schedule c_0, we define the R–D gain of candidate c as $\Delta_i(c;q) = J_i(c_0;q) - J_i(c;q)$.

On a held-out calibration set, we estimate $\widehat{\Delta}(c \mid r^z, q)$, the expected gain of schedule c conditioned on the realized hyperprior bitrate r^z and rate condition q. We partition calibration samples into equal-count quantile bins according to r^z and store the mean gain of each candidate within each bin. This yields a lightweight piecewise-constant predictor requiring no additional model inference. Since r^z is available after hyperprior coding and before autoregressive coding of $\hat{\mathbf{y}}$, the gain can be estimated before selecting the context schedule. More details are provided in Appendix A.

Streaming adaptive context. For the t-th incoming image, we obtain its hyperprior bitrate r_t^z and select

$$c_t^* = \arg\max_{c\in\mathcal{C}} \left[\widehat{\Delta}(c \mid r_t^z, q) - \mu_t C(c)\right], \tag{5}$$

where μ_t is a dynamic complexity penalty. This is a Lagrangian form of the budgeted objective: larger μ_t favors cheaper schedules, whereas smaller μ_t allows additional computation when the predicted gain is sufficiently large.

For each rate and target budget, we first calibrate a base penalty on held-out data such that the average selected complexity approaches B. During streaming, we track the accumulated complexity debt

$$d_t = \sum_{\tau<t} \left[C(c_\tau) - B\right], \tag{6}$$

and adjust the penalty accordingly. A positive debt indicates that the stream has exceeded its target budget and increases the penalty on expensive schedules, while a negative debt permits more computation on subsequent inputs. We use a clipped linear correction to the calibrated penalty; the exact update rule and hyperparameters are given in Appendix A.

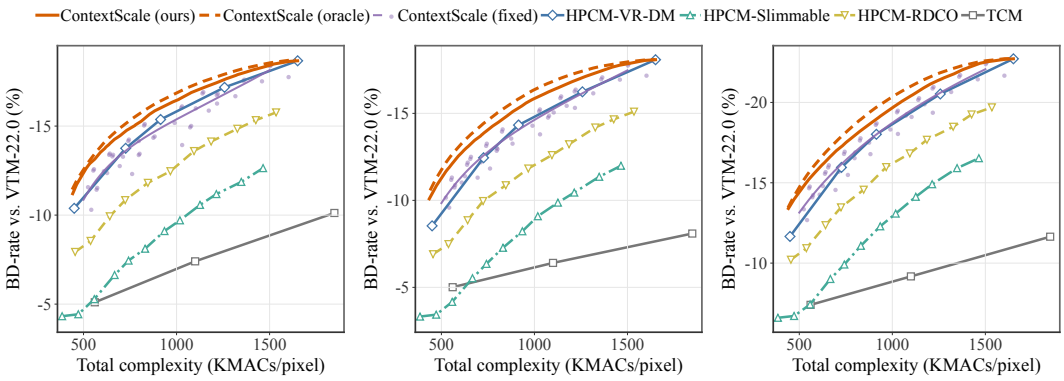
Algorithm 1 **Streaming Adaptive Context**

```
# image x_t, rate q, budget B

# Shared latent representation
y_t, z_t ← Analysis Transform(x_t, q)
r_t^z ← EncodeHyperprior(z_t)                          Eq. 16

# Adaptive context selection
Δ̂_t ← LookupGain(r_t^z, q)                             Eq. 21
μ_t ← BudgetController(B, d_t)                         Eq. 28
c_t^* ← arg max_{c∈C} [Δ̂_t(c) − μ_t C(c)]               Eq. 5

# Context coding & budget update
B_t^y ← ContextEncode(y_t, c_t^*)
d_{t+1} ← d_t + C(c_t^*) − B                           Eq. 26
```

Overall, the proposed framework combines unified rate–complexity control with adaptive context scheduling, assigning finer context modeling to images with larger predicted gains while maintaining a prescribed average complexity budget. This exploits image-wise scaling variation without evaluating multiple context configurations.

5 Experiments

We conduct extensive experiments to evaluate ContextScale in terms of rate–distortion performance, complexity scalability, and adaptive context allocation. Additional implementation details and experiments are provided in Appendix.



All HPCM-based variants follow the same backbone family and variable-rate training protocol. *Oracle* selects image-wise context configurations using measured R–D gains, whereas *Fixed* applies the same context schedule to all images. HPCM-VR-DM (Li et al., 2025c), HPCM-Slimmable (Guo et al., 2023), and HPCM-RDCO (Gao et al., 2024) scale model size, active capacity, and spatial context coverage, respectively. TCM (Liu et al., 2023) provides fixed-complexity reference points.

Figure 4: Rate–distortion–complexity comparison with variable-complexity methods.

Backbone and complexity range. We instantiate ContextScale on HPCM (Li et al., 2025b), a recent high-performance autoregressive context-modeling framework, and augment it with the rate- and context-conditioning modules described in Sec. 4.1. Each ContextScale model independently supports variable rate and variable context complexity by changing the context schedule while keeping model capacity and transform computation fixed. To further study the complementarity between parameter and context scaling, we train four model sizes, which together with multiple context schedules cover average complexities from approximately 400 to 1700 kMACs/pixel, with the target average complexity controlled at 1 kMAC/pixel granularity. For fair comparison, we also reproduce representative variable-complexity methods using the same HPCM backbone whenever applicable.

Training. We train ContextScale on MLIC-Train-100K (Jiang et al., 2025) using Adam (Kingma & Ba, 2015). The main stage runs for 300 epochs with random 256×256 crops, a batch size of 32, and an initial learning rate of 2×10^{-4}, reduced to 2×10^{-5} after 200 epochs. We then fine-tune for 50 epochs using 512×512 crops and a batch size of 8, with the learning rate reduced from 1×10^{-5} to 5×10^{-6} during this stage. Rate conditions and autoregressive context schedules are jointly sampled during training so that each model supports both variable rate and variable complexity. All evaluations use exponential moving average (EMA) weights with a decay of 0.999.

Evaluation. We evaluate on Kodak (Franzen, 1999), Tecnick (Asuni & Giachetti, 2014), and CLIC Pro. Validation (Toderici et al., 2020), covering a broad range of image content and resolutions. Kodak contains 24 images at 768×512, Tecnick contains 100 images at 1200×1200, and CLIC Pro. Validation contains 41 images at approximately 2K resolution. We report PSNR BD-rate (Bjøntegaard, 2001) using VTM 22.0 (Joint Video Experts Team (JVET), 2023) as the anchor.

5.1 Rate–Distortion–Complexity Trade-offs

Fig. 4 and Table 1 summarize the rate–distortion–complexity (R–D–C) performance from variable- and fixed-complexity perspectives, respectively. Across Kodak, CLIC, and Tecnick, ContextScale consistently improves compression performance as context complexity increases, confirming the controllable scaling behavior established in Sec. 3.

Comparison with variable-complexity methods. We compare ContextScale with representative variable-complexity strategies. HPCM-Slimmable (Guo et al., 2023) varies the active network capacity to provide different complexity levels, whereas HPCM-RDCO (Gao et al., 2024) adapts computation spatially by allocating context modeling to different latent regions. Within each model size, ContextScale instead varies the autoregressive context schedule while keeping the transform and model capacity unchanged. As shown in Fig. 4, this leads to a consistently stronger R–D–C trade-off across a broad range of complexity budgets.

Table 1: Rate–distortion–complexity comparison with fixed-complexity methods.

Model	Complexity ↓			PSNR BD-Rate (%) ↓			
	kMACs /pixel	Enc. (ms)	Dec. (ms)	Kodak	CLIC	Tecnick	Avg.
Compute budget: $<$ 600 *kMACs/pixel*							
QARV (TPAMI'24) (Duan et al., 2024)♯	380.48	73.2	71.4	-1.92	-1.20	-5.74	-2.95
TIC (DCC'22) (Lu et al., 2022)	407.83	286.8	307.0	-2.03	-3.42	-4.55	-3.33
EVC_LM (ICLR'23) (Wang et al., 2023)	421.52	19.8	10.8	2.38	4.04	2.78	3.07
DCVC-UF-Intra (CVPR'26) (Li et al., 2026)♯	455.54	16.4	12.5	-11.10	-10.03	-13.85	-11.66
EVC_LL (ICLR'23) (Wang et al., 2023)	555.30	23.5	16.3	1.12	2.57	1.10	1.60
TCM-Small (CVPR'23) (Liu et al., 2023)	560.56	99.5	107.2	-5.10	-5.01	-7.40	-5.84
Baseline-450 (HPCM-VR-DM)♯	**450.00**	**49.2**	**48.8**	**-10.38**	**-8.54**	**-11.66**	**-10.19**
ContextScale♯♢	**450.00**	**49.7**	**48.8**	**-11.40**	**-10.39**	**-13.56**	**-11.78**
Compute budget: 600–1200 kMACs/pixel							
HPCM-Base (ICCV'25) (Li et al., 2025b)	905.77	60.2	60.0	-14.32	-13.85	-17.83	-15.33
MSINN (TMM'25) (Tu et al., 2025)♯	1025.88	129.5	155.5	-3.28	-0.93	-2.63	-2.28
FTIC (ICLR'24) (Li et al., 2024)	1095.60	$>$1000	$>$1000	-12.97	-9.57	-15.03	-12.52
TCM-Middle (CVPR'23) (Liu et al., 2023)	1099.87	138.1	155.2	-7.40	-6.41	-9.17	-7.66
CCA (NeurIPS'24) (Han et al., 2024)	1142.83	129.9	111.4	-12.01	-10.81	-14.78	-12.54
MLIC++ (ICMLW'23) (Jiang & Wang, 2023)	1153.24	118.4	166.2	-13.34	-13.10	-17.75	-14.73
Baseline-900 (HPCM-VR-DM)♯	**900.00**	**60.1**	**60.0**	**-15.25**	**-14.18**	**-17.86**	**-15.76**
ContextScale♯♢	**900.00**	**60.5**	**59.9**	**-15.94**	**-15.21**	**-18.84**	**-16.66**
Compute budget: $>$ 1200 *kMACs/pixel*							
HPCM-Large (ICCV'25) (Li et al., 2025b)	1248.58	70.9	70.0	-18.38	-18.07	-21.89	-19.45
GLIC (CVPR'26) (Chen et al., 2026)	1343.07	224.9	160.2	-17.72	-17.82	-22.46	-19.33
TCM-Large (CVPR'23) (Liu et al., 2023)	1848.26	183.2	167.7	-10.12	-8.09	-11.64	-9.95
WeConvene (ECCV'24) (Fu et al., 2024)	2343.57	223.9	223.5	-6.69	-5.79	-8.64	-7.04
Baseline-1350 (HPCM-VR-DM)♯	**1350.00**	**72.8**	**72.3**	**-17.53**	**-16.67**	**-21.04**	**-18.41**
ContextScale♯♢	**1350.00**	**69.9**	**69.2**	**-17.96**	**-17.40**	**-21.88**	**-19.08**

BD-rate is computed relative to VTM-22.0. ♯and ♢denote the model supporting variable-rate and variable-complexity operation, respectively. HPCM-VR-DM uses a separately sized model for each computation budget. Encoding/decoding latency is measured on images at a resolution of 512×768. Blue shading denotes our implementations, and underlining marks the best runtime or BD-rate within each computation group.

Adaptive context scheduling further improves over the corresponding fixed schedule at matched average complexity, confirming that exploiting image-wise variation is more effective than assigning the same context complexity to every input. Its performance remains close to the oracle allocation, indicating that the hyperprior-based predictor captures much of the variation in context-scaling gain without evaluating multiple candidate schedules.

Comparison with fixed-complexity methods. Table 1 further shows that ContextScale remains competitive with strong fixed-complexity codecs across a broad computational range. Although our model adopts HPCM (Li et al., 2025b) as the backbone, introducing variable-rate coding into a unified model results in an inherent $\sim$2% BD-rate loss compared with the original fixed-rate HPCM, consistent with prior observations (Wang et al., 2026a; Ta et al., 2026). Despite this disadvantage, ContextScale achieves the best average BD-rate in the low- and medium-complexity groups and remains close to HPCM and GLIC (Chen et al., 2026) at high complexity. At matched budgets, ContextScale consistently improves its HPCM-VR-DM baseline, with larger gains in the lower-complexity regime where context computation is more limited. These results show that context scaling can effectively improve compression efficiency while retaining the flexibility of unified variable-rate and variable-complexity coding.

5.2 Ablation Studies

We examine the robustness of gain calibration, the effectiveness of online budget control, and the transferability of ContextScale across context models in Table 2. We further analyze the relation between computational complexity and actual runtime in Fig. 5.

Calibration robustness. Different calibration sources lead to nearly identical performance, with the average BD-rate varying by only 0.08 percentage points. This indicates that the relation between

Table 2: Ablations of calibration, budget control, and context-model generalization.

(a) Calibration: BDR (%)	
DIV2K	-16.02
MLIC	-15.98
DIV2K+MLIC	**-16.06**

(b) Budget Dev.: $\lvert\Delta C\rvert$ (%)	
Static*	3.15
Online	2.60
Global†	**0.02**

(c) CHARM: BDR (%)	
Baseline-VR	-9.77
+ ContextScale	-11.04
+ ContextScale (Oracle)‡	-12.63

BDR denotes the average PSNR BD-rate over Kodak, CLIC, and Tecnick, with VTM-22.0 as the anchor. For (a), each calibration setting uses 100 sampled images. * Static selects the context schedule independently for each image using the calibrated gain predictor, without online budget feedback. † Global jointly assigns context schedules over the complete image sequence using the estimated r^z–gain relation to match the target average complexity. ‡ Oracle uses the measured R–D gain of every candidate schedule for each image and selects the best allocation under the same average complexity budget.

hyperprior bitrate and context-scaling gain is not strongly dependent on the calibration distribution. We therefore use a single calibrated gain predictor throughout the main experiments.

Online budget control. Without feedback, static selection can accumulate complexity deviation over a finite image stream. The proposed online controller reduces this deviation by correcting accumulated over- or under-computation as coding proceeds. Global allocation achieves nearly exact budget matching, but requires access to the entire sequence in advance. These results show that gain prediction and budget control play complementary roles: the former determines which inputs are more likely to benefit from finer context modeling, while the latter keeps the stream-level average complexity close to the target.

Generalization to another context model. Replacing HPCM with the channel-wise autoregressive model CHARM (Minnen & Singh, 2020) yields a similar improvement from ContextScale, while oracle allocation provides a further upper bound. This suggests that the benefit of adaptive context scaling is not tied to the specific context structure of HPCM, but extends to a different autoregressive entropy model.

Context latency. Fig. 5 compares the complexity and measured context-modeling latency across different context schedules. Runtime generally follows the complexity trend, while finer-grained schedules incur additional latency due to reduced parallelism in autoregressive coding. Table 1 also shows that the adaptive scheduler introduces negligible overhead, requiring only a lightweight gain lookup and budget update.

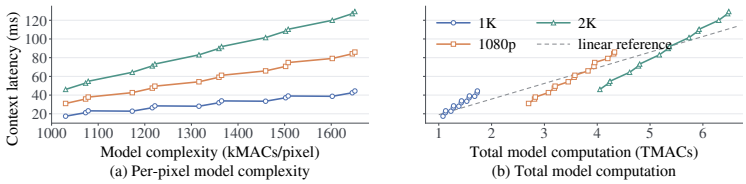


Figure 5: Context latency across context schedules and resolutions.

6 Conclusion and Future Work

Conclusion. We introduced *context scaling* as a complementary complexity dimension for learned image compression, enabling a unified codec to vary autoregressive context modeling without changing model capacity. Since different complexity levels reuse the same analysis transform, the framework is particularly suitable for content distribution and heterogeneous deployment. Our analysis reveals a clear contrast between systematic dataset-level scaling and substantial image-wise variation in R–D gains, motivating an adaptive context schedule that allocates finer context modeling to inputs with larger expected benefits under a given complexity budget.

Future work. The current design performs adaptation at the image level and uses a lightweight calibration model to estimate context-scaling gains. Combining context scaling with spatially adaptive or slimmable computation could further extend the controllable complexity range and improve R–D performance, while more accurate gain prediction may narrow the remaining gap to oracle allocation. More broadly, the pronounced variation in R–D gain across images suggests that different inputs may have fundamentally different demands for context modeling, a phenomenon that we only begin to characterize in this work. Further understanding the factors behind this variation may provide a promising path toward overcoming current performance limitations in learned image compression.

AI USE STATEMENT

Generative AI tools were used only as auxiliary tools for manuscript preparation and presentation, including language polishing, grammar and spelling checks, formatting checks, and organizing numerical results already produced by our experimental pipeline into tables and summaries. They were not used for research ideation, problem formulation, hypothesis development, methodology or experimental design, or interpretation of experimental results.

ETHICS STATEMENT

This work focuses on learned image compression and does not involve human subject studies or the collection of personal or sensitive data by the authors. As with general-purpose image processing and compression techniques, potential ethical considerations may arise from specific downstream applications and the data being processed. Such deployments should comply with applicable requirements regarding privacy, copyright, security, and responsible data use.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, training and evaluation protocols, datasets, implementation environment, and complexity measurement in the main paper and appendix. For reproduced baselines, we document the corresponding implementations, configurations, and evaluation settings used in our comparisons. Upon publication, we will release the source code and configuration files to facilitate independent reproduction and future research.

REFERENCES

Bouzid Arezki, Anissa Mokraoui, and Fangchen Feng. Efficient variable-rate state-space model for image compression with channel-wise entropy. In *2026 IEEE International Conference on Image Processing (ICIP)*, pp. 1–6. IEEE, 2026.

Nicola Asuni and Andrea Giachetti. TESTIMAGES: A large-scale archive for testing visual devices and basic image processing algorithms. In *STAG: Smart Tools & Apps for Graphics*. The Eurographics Association, 2014.

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, Austin, TX, USA, April 2001. URL https://www.itu.int/wftp3/av-arch/video-site/0104_Aus/VCEG-M33.doc.

Kecheng Chen, Pingping Zhang, Tiexin Qin, Shiqi Wang, Hong Yan, and Haoliang Li. Test-time adaptation for image compression with distribution regularization. In *International Conference on Learning Representations*, 2025.

Tong Chen and Zhan Ma. Variable bitrate image compression with quality scaling factors. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2163–2167. IEEE, 2020.

Yunuo Chen, Bing He, Zezheng Lyu, Hongwei Hu, Qunshan Gu, Yuan Tian, and Guo Lu. Adaptive learned image compression with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12150–12161, 2026.

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

Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Variable rate deep image compression with a conditional autoencoder. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3146–3154. IEEE, 2019.

Wuyang Cong, Junqi Shi, Ming Lu, Xu Zhang, and Zhan Ma. Taming hierarchical image coding optimization: A spectral regularization perspective. In *The Fourteenth International Conference on Learning Representations*, volume 1, 2026.

Thomas M Cover, Joy A Thomas, and John Kieffer. *Elements of information theory*, volume 2. wiley New York, 1991.

Zhihao Duan, Ming Lu, Jack Ma, Yuning Huang, Zhan Ma, and Fengqing Zhu. Qarv: Quantization-aware resnet vae for lossy image compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):436–450, 2024.

Rich Franzen. Kodak lossless true color image suite, 1999. URL https://r0k.us/graphics/kodak/.

Haisheng Fu, Jie Liang, Zhenman Fang, Jingning Han, Feng Liang, and Guohe Zhang. Weconvene: Learned image compression with wavelet-domain convolution and entropy model. In *Computer Vision – ECCV 2024*, pp. 37–53, 2024.

Yixin Gao, Runsen Feng, Zongyu Guo, and Zhibo Chen. Exploring the rate-distortion-complexity optimization in neural image compression. *Journal of Visual Communication and Image Representation*, 105:104294, 2024.

Jinyang Guo, Dong Xu, and Guo Lu. Cbanet: Toward complexity and bitrate adaptive deep image compression using a single network. *IEEE Transactions on Image Processing*, 32:2049–2062, 2023.

Minghao Han, Shiyin Jiang, Shengxi Li, Xin Deng, Mai Xu, Ce Zhu, and Shuhang Gu. Causal context adjustment loss for learned image compression. *Advances in Neural Information Processing Systems*, 37:133231–133253, 2024.

Xinyu Hang, Ziqing Ge, Hongfei Fan, Chuanmin Jia, Siwei Ma, and Wen Gao. Rate-distortion-complexity optimized framework for multi-model image compression. *IEEE Transactions on Image Processing*, 2025.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *Advances in Neural Information Processing Systems*, 35:30016–30030, 2022.

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

Wei Jiang and Ronggang Wang. Mlic++: Linear complexity multi-reference entropy modeling for learned image compression. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023.

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), 2025.

Joint Video Experts Team (JVET). VVC test model (VTM), version 22.0, 2023. URL https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-22.0. Reference software.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Chao Li, Tianyi Li, Fanyang Meng, Qingyu Mao, Youneng Bao, Yonghong Tian, and Yongsheng Liang. One is all: A unified rate-distortion-complexity framework for learned image compression under energy concentration criteria. *IEEE Transactions on Multimedia*, 27:3992–4007, 2025a.

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

Jiahao Li, Wenxuan Xie, Zhaoyang Jia, Bin Li, Zongyu Guo, Xiaoyi Zhang, and Yan Lu. Ultra-fast neural video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41311–41321, 2026.

Yuqi Li, Haotian Zhang, Li Li, and Dong Liu. Learned image compression with hierarchical progressive context modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18834–18843, 2025b.

Yuqi Li, Haotian Zhang, Li Li, Dong Liu, and Feng Wu. Scaling learned image compression models up to 1 billion. *arXiv preprint arXiv:2508.09075*, 2025c.

Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 14388–14397. IEEE, 2023.

Ming Lu, Peiyao Guo, Huiqing Shi, Chuntong Cao, and Zhan Ma. Transformer-based image compression. In *2022 Data Compression Conference (DCC)*, pp. 80, 2022.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2523–2534, 2025.

David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 3339–3343. IEEE, 2020.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Calvin-Khang Ta, Praneet Singh, Tong Shao, and Peng Yin. Mixcompress: Mixture of experts for variable rate learned image compression. In *European Conference on Computer Vision*, pp. 285–302. Springer, 2026.

Lvfang Tao, Wei Gao, Ge Li, and Chenhao Zhang. AdaNIC: Towards practical neural image compression via dynamic transform routing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16879–16888, 2023.

George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020). In *CVPR*, 2020.

Kedeng Tong, Yaojun Wu, Yue Li, Kai Zhang, Li Zhang, and Xin Jin. Qvrf: A quantization-error-aware variable rate framework for learned image compression. In *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 1310–1314. IEEE, 2023.

Hanyue Tu, Siqi Wu, Li Li, Wengang Zhou, and Houqiang Li. Multi-scale invertible neural network for wide-range variable-rate learned image compression. *IEEE Transactions on Multimedia*, 27: 8592–8602, 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.

Ran Wang, Yongqiang Wang, Heming Sun, and Jiro Katto. Variable-rate learned image compression using parameter efficient fine-tuning and trainable quantization step-size. In *2026 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3856–3860. IEEE, 2026a.

Shen Wang, Zhengxue Cheng, Donghui Feng, Qi Wang, Qunshan Gu, Li Song, and Wenjun Zhang. Distilling complexity-scalable learned image compression models via neural architecture search. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026b.

Fei Yang, Luis Herranz, Yongmei Cheng, and Mikhail G Mozerov. Slimmable compressive autoencoders for practical neural image compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4996–5005. IEEE, 2021.

Yibo Yang, Robert Bamler, and Stephan Mandt. Improving inference for neural image compression. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12104–12113, 2022.

Yichi Zhang, Yuning Huang, and Fengqing Zhu. Qarv++: An improved hierarchical vae for learned image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

A Details of Adaptive Context Scaling

This section provides additional details of the adaptive context-scaling framework introduced in Sec. 4.2. We first summarize the underlying compression pipeline and define the R–D gain used for context selection. We then describe hyperprior-guided gain calibration, the budget-aware streaming controller, and the offline oracle used for analysis.

A.1 Compression Preliminaries

We first summarize the compression and entropy-modeling pipeline underlying ContextScale. Following the standard hyperprior-based formulation, an input image $\mathbf{x}_i$ at rate condition q is transformed into a main latent representation

$$\mathbf{y}_i = g_a(\mathbf{x}_i; q), \tag{7}$$

and a hyper-analysis transform further produces

$$\mathbf{z}_i = h_a(\mathbf{y}_i; q). \tag{8}$$

The quantized hyper-latent $\hat{\mathbf{z}}_i$ is entropy coded first, and the hyper-synthesis transform generates side information

$$\mathbf{h}_i = h_s(\hat{\mathbf{z}}_i) \tag{9}$$

for conditional entropy modeling of the main latent.

For a context configuration $c \in \mathcal{C}$, let $\{\mathcal{G}_t^{(c)}\}_{t=1}^{S_c}$ denote the corresponding ordered partition of $\mathbf{y}_i$. At coding step t, the entropy model predicts the distribution parameters of the symbols in $\mathcal{G}_t^{(c)}$ from the hyperprior feature and previously decoded latent groups:

$$(\boldsymbol{\mu}_{i,t}^{(c)}, \boldsymbol{\sigma}_{i,t}^{(c)}) = f_\theta\left(\mathbf{h}_i,\, \hat{\mathbf{y}}_{i,\mathcal{G}_{<t}^{(c)}};\, c\right), \tag{10}$$

where $\boldsymbol{\mu}_{i,t}^{(c)}$ and $\boldsymbol{\sigma}_{i,t}^{(c)}$ denote the predicted mean and scale parameters, respectively.

Following mean-centered entropy coding, each latent element $y_{i,j}$ is represented by the integer symbol

$$s_{i,j} = \mathrm{round}\left(y_{i,j} - \mu_{i,j}^{(c)}\right), \qquad \hat{y}_{i,j} = s_{i,j} + \mu_{i,j}^{(c)}. \tag{11}$$

Let $F_\theta(\cdot; \sigma_{i,j}^{(c)})$ denote the CDF of the corresponding zero-centered conditional distribution. The probability mass assigned to $s_{i,j}$ is

$$p_{i,j}^{(c)} = F_\theta\left(s_{i,j} + \frac{1}{2}; \sigma_{i,j}^{(c)}\right) - F_\theta\left(s_{i,j} - \frac{1}{2}; \sigma_{i,j}^{(c)}\right). \tag{12}$$

The conditional rate of the main latent can therefore be estimated by

$$R_i^y(c; q) = -\frac{1}{H_i W_i} \sum_j \log_2 p_{i,j}^{(c)}. \tag{13}$$

Similarly, the hyper-latent contributes a rate $R_i^z(q)$ under its hyperprior entropy model, giving the total likelihood-estimated rate

$$R_i(c; q) = R_i^y(c; q) + R_i^z(q). \tag{14}$$

For actual entropy coding, we denote the resulting main-latent and hyper-latent bitstreams by $\mathcal{B}_i^y(c)$ and $\mathcal{B}_i^z$, respectively. Their realized bitrate is

$$r_i(c; q) = \frac{8\left(|\mathcal{B}_i^y(c)| + |\mathcal{B}_i^z|\right)}{H_i W_i}. \tag{15}$$

The realized hyperprior bitrate is correspondingly

$$r_i^z = \frac{8|\mathcal{B}_i^z|}{H_i W_i} \tag{16}$$

Since the hyperprior is entropy coded before the main latent, r_i^z is available when the context configuration is selected and is used as the input statistic for adaptive scheduling.

A.2 R–D Loss and Context-Scaling Gain

For image $\mathbf{x}_i$ at rate condition q, we define the R–D loss under context configuration c as

$$J_i(c;q) = R_i(c;q) + \lambda_q D_i(c;q), \tag{17}$$

where $D_i(c;q)$ denotes the distortion and λ_q is the rate-dependent Lagrange multiplier.

Let $c_0 \in \mathcal{C}$ denote the lowest-complexity context configuration. The context-scaling gain of candidate c is defined relative to c_0 as

$$\Delta_i(c;q) = J_i(c_0;q) - J_i(c;q). \tag{18}$$

A larger $\Delta_i(c;q)$ indicates a larger reduction in R–D loss from using configuration c.

The rate-dependent multiplier follows the variable-rate training objective and is interpolated logarithmically:

$$\lambda_q = \exp\left[\log \lambda_{\min} + \frac{q}{q_{\max}} \left(\log \lambda_{\max} - \log \lambda_{\min}\right)\right], \tag{19}$$

where $\lambda_{\min} = 0.001$, $\lambda_{\max} = 0.05$, and $q_{\max} = 62$.

A.3 Hyperprior-Guided Gain Calibration

As shown in Sec. 3, the hyperprior bitrate is strongly correlated with the image-wise benefit of finer context modeling. We therefore estimate the expected context-scaling gain from r_i^z.

For each rate condition q, the calibration samples are sorted according to r_i^z and partitioned into K approximately equal-count quantile bins $\{\mathcal{I}_{q,k}\}_{k=1}^{K}$. For context configuration c, the expected gain associated with bin k is estimated by

$$\widehat{\Delta}_{q,c,k} = \frac{1}{|\mathcal{I}_{q,k}|} \sum_{i \in \mathcal{I}_{q,k}} \Delta_i(c;q). \tag{20}$$

Given a hyperprior bitrate r^z, let $k(r^z)$ denote its corresponding quantile bin. The predicted gain of configuration c is then

$$\widehat{\Delta}(c \mid r^z, q) = \widehat{\Delta}_{q,c,k(r^z)}. \tag{21}$$

We use $K = 6$ bins in all experiments. Values outside the calibrated bitrate range are assigned to the nearest boundary bin. The resulting estimator is a piecewise-constant lookup table indexed by the rate condition, hyperprior-bitrate bin, and context configuration, requiring no additional model inference.

A.4 Budget-Aware Streaming Controller

Let $C(c)$ denote the computational complexity of context configuration c, and let B denote the target average complexity. For the t-th incoming image, the adaptive policy selects

$$c_t^* = \arg\max_{c \in \mathcal{C}} \left[\widehat{\Delta}(c \mid r_t^z, q) - \mu_t C(c)\right], \tag{22}$$

where $\mu_t \geq 0$ is the complexity penalty. This Lagrangian form balances the predicted R–D gain against the computational cost of each candidate.

Base complexity penalty. For each rate condition q and target budget B, we first calibrate a base penalty μ_{base} on a held-out calibration set $\mathcal{D}_{\text{cal}}$. For a fixed penalty μ, define

$$c_i(\mu) = \arg\max_{c \in \mathcal{C}} \left[\widehat{\Delta}(c \mid r_i^z, q) - \mu C(c)\right]. \tag{23}$$

The corresponding average complexity on the calibration set is

$$\bar{C}(\mu) = \frac{1}{|\mathcal{D}_{\text{cal}}|} \sum_{i \in \mathcal{D}_{\text{cal}}} C\big(c_i(\mu)\big). \tag{24}$$

We choose

$$\mu_{\text{base}} = \arg\min_{\mu \geq 0} \left| \bar{C}(\mu) - B \right|. \tag{25}$$

Because $\bar{C}(\mu)$ is non-increasing with respect to μ, the base penalty can be determined efficiently by binary search.

Online budget correction. Although μ_{base} matches the target budget on the calibration distribution, a finite input stream may accumulate over- or under-computation. Before processing image t, we define the accumulated complexity debt as

$$d_t = \sum_{\tau=1}^{t-1} \left[C(c_\tau^*) - B \right]. \tag{26}$$

Positive d_t indicates that the accumulated computation exceeds the target, whereas negative d_t indicates unused budget.

The normalized debt is

$$\rho_t = \text{clip}\left(\frac{d_t}{BH}, -4, 4 \right), \tag{27}$$

where H controls the effective feedback horizon. We adjust the complexity penalty according to

$$\mu_t = \left[\mu_{\text{base}} + \alpha \max\left(\mu_{\text{base}}, \mu_{\min} \right) \rho_t \right]_+, \tag{28}$$

where $[a]_+ = \max(a, 0)$. We use $H = 32$, $\alpha = 0.05$, and $\mu_{\min} = 10^{-7}$ in all experiments.

After selecting c_t^*, the complexity state is updated as

$$d_{t+1} = d_t + C(c_t^*) - B. \tag{29}$$

Accumulated over-computation therefore increases the penalty applied to subsequent images, while unused budget decreases it. The correction is computed from the fixed calibrated value μ_{base} and the current debt at each step, rather than recursively updating the previous penalty.

A.5 Oracle Reference

We additionally report an offline oracle to characterize the potential of input-adaptive context allocation when the true gain of every candidate configuration is available. Given a set of M images and target average complexity B, the oracle solves

$$\begin{aligned} \max_{\{c_i\}_{i=1}^M} \quad & \sum_{i=1}^{M} \Delta_i(c_i; q) \\ \text{s.t.} \quad & \sum_{i=1}^{M} C(c_i) \leq MB, \qquad c_i \in \mathcal{C}. \end{aligned} \tag{30}$$

Since one context configuration is selected for each image, this is a multiple-choice knapsack problem. We solve it by dynamic programming over the finite candidate set and attainable cumulative complexity states.

The oracle is used only as a reference for the achievable benefit of adaptive context allocation under the same candidate set and complexity budget. In contrast, the streaming policy makes each decision from the predicted gain and current budget state without observing the gains of unselected configurations.

B Additional Experimental Details

B.1 Implementation.

All experiments are conducted on two NVIDIA RTX 4090 GPUs. For compared methods, we recompute BD-rate using their reported rate–distortion points against our VTM-22.0 anchor to ensure a consistent evaluation protocol. When the required rate–distortion results are not available, we reproduce the corresponding open-source implementations and evaluate them under the same protocol. Model complexity is measured using `fvcore.nn.FlopCountAnalysis` and normalized by the number of input pixels, reported in kMACs/pixel.

B.2 VTM-22.0 Anchor Reproduction

We use VTM-22.0 (Joint Video Experts Team (JVET), 2023) with the All-Intra configuration `encoder_intra_vtm.cfg` as the conventional-codec anchor. The evaluation pipeline consists of RGB-to-YUV conversion, VTM encoding and decoding, and conversion of the reconstructed frame back to RGB. Each image is processed independently as a single frame at its original resolution.

RGB-to-YUV conversion. The original RGB image is converted to planar 8-bit YUV 4:4:4 using FFmpeg. For an image with spatial resolution $W \times H$, we use

```
ffmpeg -y \
    -i {input.png} \
    -pix_fmt yuv444p \
    -f rawvideo \
    {input.yuv}
```

The resulting raw YUV frame is directly used as the input to VTM. We use 4:4:4 chroma sampling to avoid additional chroma subsampling during the comparison.

VTM encoding. Each image is encoded with VTM-22.0 using its All-Intra configuration. A representative command is

```
EncoderAppStatic \
    -c {VTM_ROOT}/cfg/encoder_intra_vtm.cfg \
    --InputFile={input.yuv} \
    --BitstreamFile={bitstream.vvc} \
    --SourceWidth={width} \
    --SourceHeight={height} \
    --FrameRate=1 \
    --FramesToBeEncoded=1 \
    --QP={QP} \
    --InputChromaFormat=444 \
    --ChromaFormatIDC=444 \
    --InputBitDepth=8 \
    --InternalBitDepth=10 \
    --OutputBitDepth=8 \
    --ConformanceWindowMode=1
```

Both the input and coded chroma formats are set to 4:4:4. The input and reconstructed samples are 8-bit, while VTM uses 10-bit internal processing. Automatic conformance-window handling is enabled. We evaluate the following 26 QP values:

$$\mathcal{Q}_{\mathrm{VTM}} = \{20, 21, \ldots, 40\} \cup \{42, 44, 46, 48, 50\}. \tag{31}$$

VTM decoding. The encoded bitstream is decoded using the corresponding VTM-22.0 decoder:

```
DecoderAppStatic \
    --BitstreamFile={bitstream.vvc} \
    --ReconFile={decoded.yuv}
```

RGB reconstruction. The decoded 8-bit YUV 4:4:4 frame is then converted back to RGB using FFmpeg:

```
ffmpeg -y \
    -f rawvideo \
    -pix_fmt yuv444p \
    -video_size {width}x{height} \
    -i {decoded.yuv} \
    -frames:v 1 \
    {reconstructed.png}
```

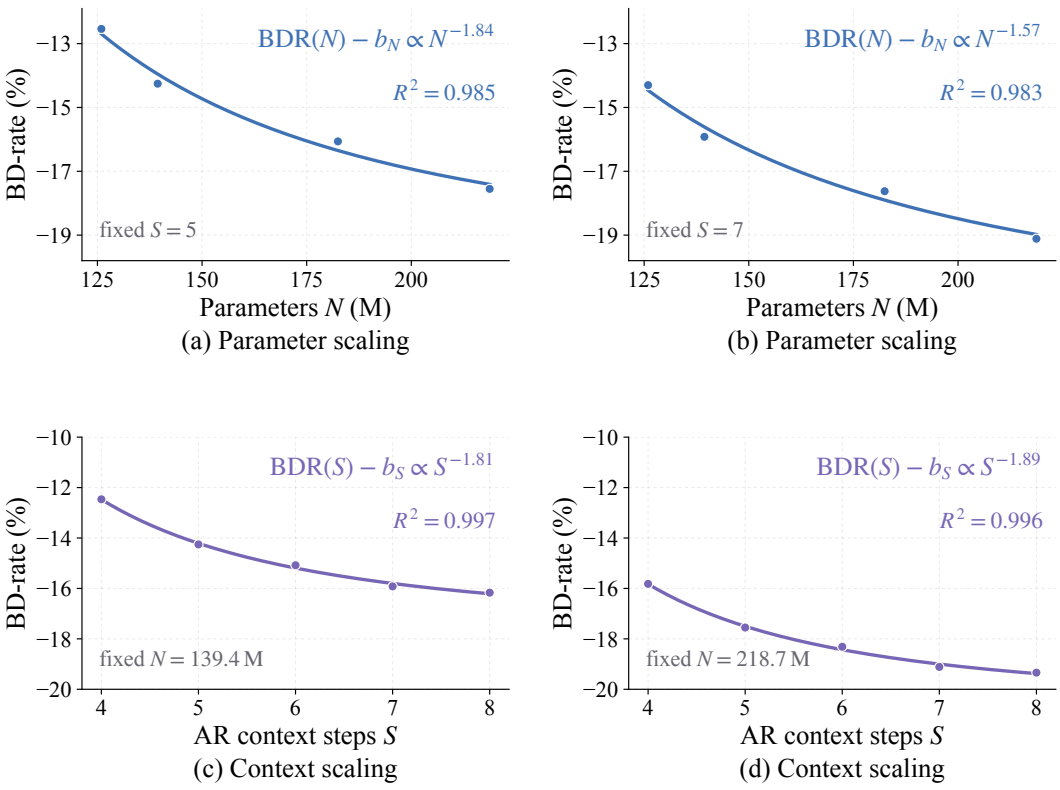


Figure 6: Additional power-law fits across parameter and context dimensions. (a–b) Parameter scaling at fixed context schedules $S = 5$ and $S = 7$. (c–d) Context scaling at fixed model sizes $N = 139.4$M and $N = 218.7$M. The asymptotic power-law form consistently provides a strong fit across different operating settings.

C Additional Scaling Behavior

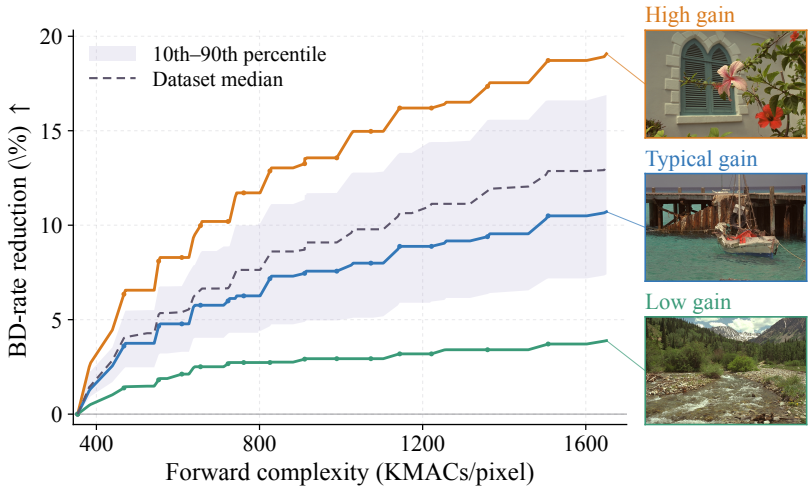
We provide additional empirical fits to examine whether the scaling trends observed in Fig. 2 remain consistent across different operating settings. Fig. 6(a–b) shows parameter scaling with the context schedule fixed at $S = 5$ and $S = 7$, respectively, while Fig. 6(c–d) evaluates context scaling at fixed model sizes of $N = 139.4$M and $N = 218.7$M.

Across all settings, the empirical BD-rate curves remain well described by the asymptotic power-law form introduced in Sec. 3. The parameter-scaling fits achieve $R^2 = 0.985$ and 0.983, while the context-scaling fits achieve $R^2 = 0.997$ and 0.996. These results confirm that the scaling behavior reported in the main paper is stable across different context schedules and model capacities.

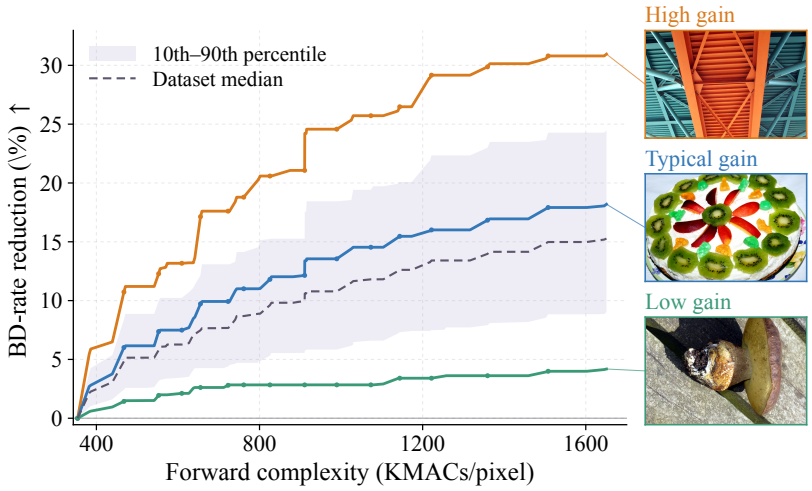
D Additional Image-Wise Scaling Results

We further examine the image-wise variation of context scaling on Kodak and Tecnick. As shown in Fig. 7, the same input-dependent scaling trend consistently appears on Kodak and Tecnick: the median gain increases with context complexity, while the distribution broadens as different images benefit to markedly different extents.

The representative examples further illustrate this variation. High-gain images continue to benefit from finer context modeling over a wide complexity range, whereas low-gain images improve only modestly and tend to saturate earlier. The effect is particularly pronounced on Tecnick, where both the overall gain and the spread across images are larger. These results support the input-dependent scaling behavior observed in the main paper and motivate allocating context computation according to the expected gain of each image.



(a) Kodak



(b) Tecnick

Figure 7: Image-wise context-scaling behavior on Kodak and Tecnick. Across both datasets, the median BD-rate reduction increases with context complexity while the gain distribution broadens, revealing substantial variation in the benefit of additional context computation across images. Representative high-, typical-, and low-gain examples further illustrate the distinct scaling behaviors.

E Additional Gain-Predictability Results

We further examine the relation between hyperprior bitrate and context-scaling gain under different model capacities, rate conditions, and context schedules. As shown in Fig. 8, the positive association observed in the main paper remains consistent across all tested settings, with Spearman correlations ranging from $\rho = 0.75$ to 0.85.

Varying the model size mainly changes the strength of the correlation, while the overall monotonic trend is preserved. The relation is particularly stable across rate conditions and context schedules, where the binned median consistently increases with hyperprior bitrate. These results indicate that the predictive value of the hyperprior bitrate is robust to changes in the operating point and context configuration, supporting its use as a lightweight input statistic for adaptive context allocation.

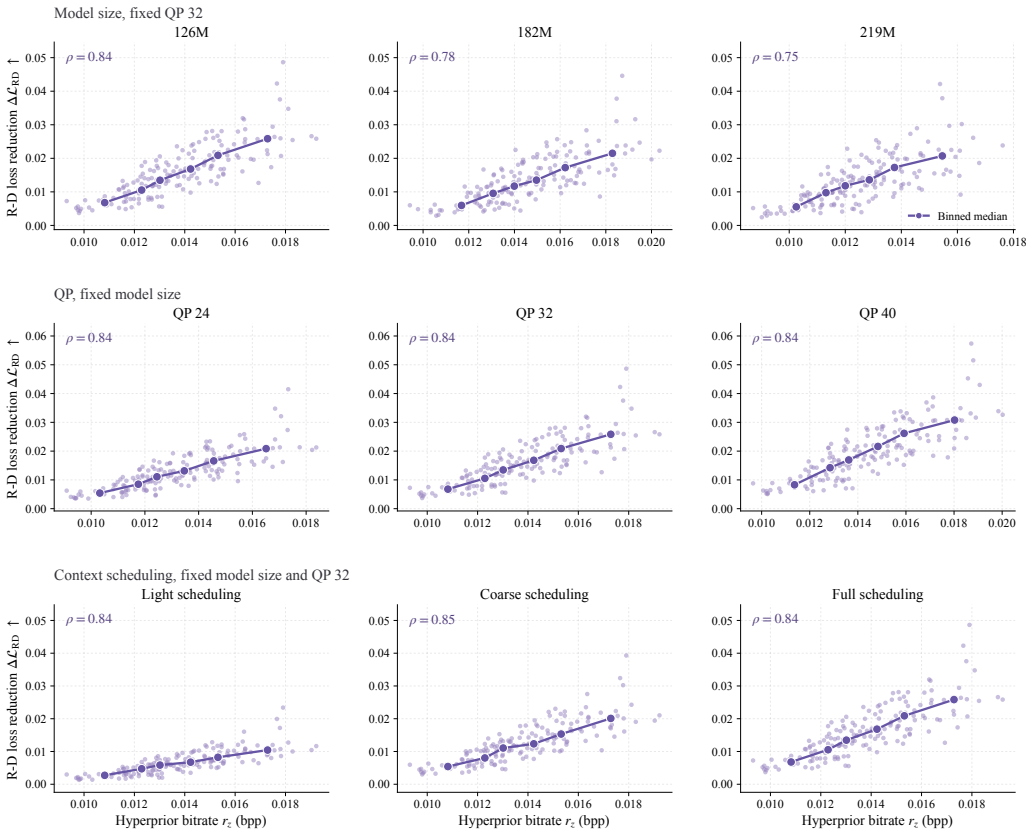


Figure 8: Gain predictability across different operating settings. Top: varying model size at fixed QP 32. Middle: varying QP at a fixed model size. Bottom: varying context schedules at a fixed model size and QP 32. Each panel shows the relation between realized hyperprior bitrate r^z and R–D loss reduction, together with the binned median and Spearman rank correlation ρ. The positive correlation remains consistent across model capacities, rate conditions, and context schedules.

F Adaptive Context-Schedule Allocation

We further analyze how the adaptive policy distributes context schedules under different complexity budgets. Fig. 9 shows the fraction of images assigned to each schedule on Kodak, Tecnick, and CLIC at two representative rate conditions.

As the target average complexity increases, the allocation progressively shifts from coarse to finer context schedules. At intermediate budgets, multiple schedules are used simultaneously rather than collapsing to a single operating point, indicating that the controller allocates computation selectively across images. The overall transition is consistent across datasets and rate conditions, while the detailed schedule proportions vary with image content and bitrate. This behavior is consistent with the image-wise variation observed in Sec. 3: a fixed average budget is realized through heterogeneous context allocations across inputs.

G Additional Ablation Studies

We further evaluate the sensitivity of the adaptive scheduler to the number of hyperprior-bitrate bins, controller strength, and feedback horizon. Across all tested settings, the BD-rate variation remains below 0.1 percentage points, indicating that the scheduler is robust to these hyperparameters. We therefore adopt 6 hyperprior-bitrate bins, controller strength $\alpha = 0.05$, and feedback horizon $H = 32$ as the default configuration throughout our experiments.

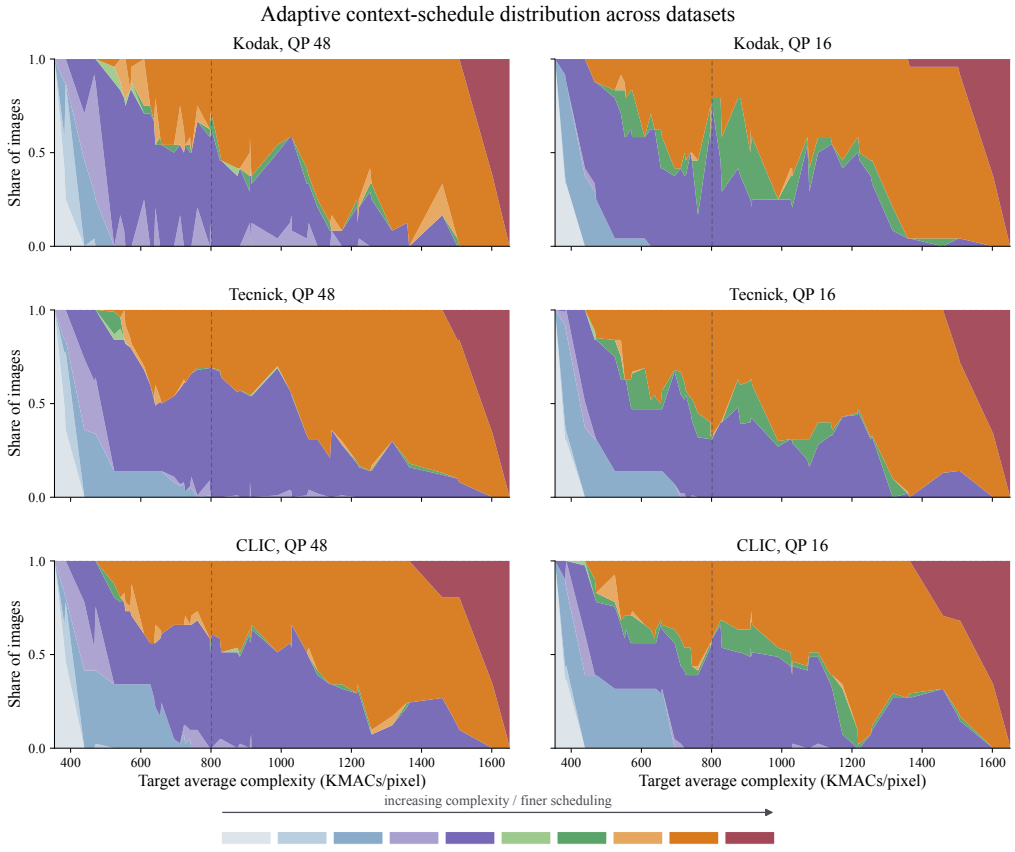


Figure 9: Adaptive context-schedule distribution across datasets and rate conditions. Each area denotes the fraction of images assigned to a context schedule at a given target average complexity, with schedules ordered from coarser to finer context modeling. As the budget increases, the allocation gradually shifts toward finer schedules, while multiple schedules coexist over a broad intermediate range.