CACHETOK: ONE BINARY STATE PER SAMPLED IMAGE TOKEN, READ BY EVERY LAYER

Anonymous authors
Paper under double-blind review

ABSTRACT

Block-autoregressive image generators such as VAR, Infinity, and PAR sample an image in a few blocks of tokens, yet every block attends to all earlier tokens, whose keys and values the transformer stores separately at each layer: Infinity-2B keeps 3.1 GiB of them for one 1024^2 image. Existing compressors for these models choose which tokens to keep, while later scales keep reading the oldest ones. We locate the redundancy along depth. Once a block is sampled, the per-layer keys and values of its tokens are views of one fixed content, and one linear subspace shared by all layers captures most of their energy. CacheTok writes each sampled token once, as b sign bits per key–value head computed from its final hidden state and its token embedding, and every layer decodes its own keys and values from these bits through two linear readers; the losses of later blocks train the writer through a straight-through estimator. We prove that a shared state never needs more width than per-layer states and that exact training becomes sequential over blocks, a cost that grows with the number of blocks and stays small for next-scale generators with 10–13 blocks. On VAR-d16, PAR-XL4x, and Infinity-2B, CacheTok stores $512\times$, $1152\times$, and $1024\times$ less visual history than matched full caches, changes FID by at most 0.03 and GenEval by -0.23 points, and doubles the batch capacity of Infinity-2B on one 80 GB GPU.

1 INTRODUCTION

Autoregressive image generation has moved from one token per step to one block of tokens per step. VAR predicts a whole resolution scale at each step (Tian et al., 2024), Infinity extends next-scale prediction to bitwise tokens at 1024^2 (Han et al., 2025), and PAR predicts groups of spatially distant tokens in parallel (Wang et al., 2025). Infinity-2B draws a 1024^2 image in 13 steps that sample 10,521 token positions. The number of steps drops, but the attention history does not: each block attends to every token sampled before it, and the transformer stores the key and value of each such token separately at every one of its L layers. Before its last scale, Infinity-2B holds 3.1 GiB of visual keys and values per image, enough to limit one 80 GB A100 to 11 images (Appendix E).

Recent accelerators select cached tokens by head or scale, prune and restore late-scale features, or sparsify attention reads (Qin et al., 2026; Li et al., 2025b; Xu et al., 2026; Guo et al., 2025; Li et al., 2026b). Retention varies across heads and layers; sparse reads can leave the full resident cache intact. Coarse-to-fine generation makes deletion costly, because the coarse scales fix the layout that the fine scales refine. Masking direct reads of all blocks older than the previous one costs the full-cache Infinity-2B 2.66 GenEval and 3.14 DPG points, most on counting and color attribution (Section 5.4). Old tokens stay in use until the last scale.

We therefore turn to the other factor of the cache size, its depth. Once a block is sampled, its tokens are final, and the key and value of each of its positions at every layer are deterministic functions of the same inputs: the condition and the tokens sampled before the block. The L per-layer entries of a position are L functions of one input, and the measured spectra show that they share most of their directions. For Infinity-2B, a single 256-dimensional linear subspace shared by all 32 layers captures about 86% of the energy of their historical keys and values, while the best split of the same 256 dimensions into layer-private subspaces captures about 13% (Figure 2). A sampled token needs one state, and each layer needs its own way of reading it.

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.

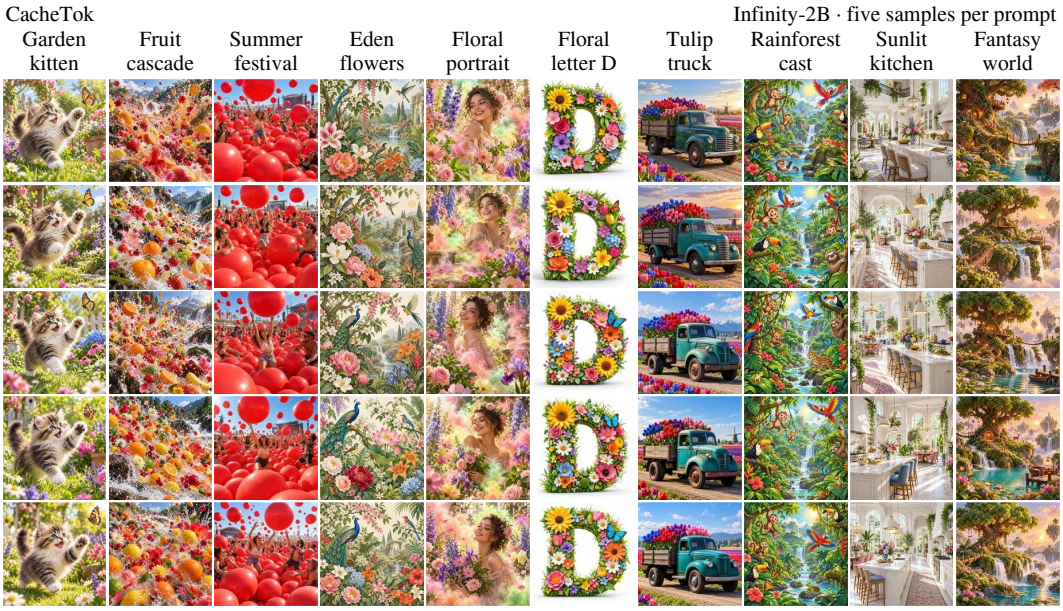


Figure 1: **Infinity-2B samples generated with a 128-bit CacheTok state per token and KV head.** Each column shares one text condition; the stored history of earlier scales is $1024\times$ smaller than the per-layer cache of the same model.

CacheTok (Figure 3) makes this state explicit. After block s is sampled, a writer combines the final-layer hidden state at each new position with the embedding of the token drawn there and stores b sign bits per KV head, with b equal to the head dimension by default. Every layer decodes keys and values from these bits through its own pair of linear readers and attends to the decoded history within the same softmax as the current block. The hidden state records the prediction context, the embedding records the drawn token, the per-layer readers let each layer retrieve different features from the same bits, and the sign fixes the storage at one bit per coordinate. Because the state serves only later blocks, their prediction losses train the writer without a teacher cache.

Writing after the full-depth pass lengthens training. The first layer of block $s{+}1$ waits for the last layer of block s, so the critical path of teacher-forced training grows from L to KL layer stages for K blocks (Theorem 2). The cost grows with the number of steps: 10 and 13 blocks for VAR-d16 and Infinity-2B, 147 for PAR-XL4x, and one per token for a token-level generator.

Our contributions are (i) measurements on three generators showing that their visual history is redundant across depth while old positions remain in use; (ii) CacheTok, a history written once per sampled token in binary form, read by every layer through its own readers, and trained by the losses of later blocks; (iii) two exact analyses, of the width a shared state needs and of the training span a late write imposes; and (iv) matched experiments on three primary and four larger backbones against six history-reduction methods. On VAR-d16, PAR-XL4x, and Infinity-2B (samples in Figure 1), CacheTok stores 512–$1152\times$ fewer bytes of visual history, changes FID by at most 0.03 and GenEval by -0.23 points, and doubles the batch capacity of Infinity-2B.

2 Related Work

Block-autoregressive image generation and its history. Raster-scan generators such as LlamaGen predict one image token per step (Sun et al., 2024a), so a high-resolution image takes thousands of sequential steps. MAR cuts this number by predicting random sets of continuous tokens (Li et al., 2024c). VAR predicts one resolution scale per step (Tian et al., 2024), Infinity scales next-scale prediction to 1024^2 with bitwise tokens (Han et al., 2025), HART adds continuous residual tokens (Tang et al., 2025), and PAR predicts groups of distant positions in parallel (Wang et al., 2025). The causal block generators studied here attend densely to earlier tokens, motivating history reduction and sparse attention. HACK allocates the cache by attention head, ScaleKV by scale and layer, and AMS-KV condenses early scales (Qin et al., 2026; Li et al., 2025b; Xu et al., 2026). FastVAR,

SkipVAR, SparseVAR, FasterVAR, and ToProVAR prune tokens or skip computation at late scales (Guo et al., 2025; Li et al., 2025a; Chen et al., 2025; Li et al., 2026a; Chen et al., 2026), SparVAR sparsifies the reads of a resident cache (Li et al., 2026b), and MVAR retrains the generator with scale and spatial Markovian conditioning (Zhang et al., 2026). These methods use token selection, sparse reads, feature reuse, or shorter conditioning histories. Their policies can vary across heads and layers; pruning may reconstruct omitted features, and sparse attention can retain the full cache. Since coarse scales fix the layout that fine scales refine, old positions can remain useful. CacheTok retains a shared state at every sampled position and learns separate layer readouts.

Reducing the per-layer key–value cache. Language models met the same memory limit earlier, with methods for three factors of the cache. Eviction keeps tokens with high attention scores or at fixed anchor positions (Zhang et al., 2023; Xiao et al., 2024; Li et al., 2024d; Ge et al., 2024), which inherits the position trade-off above. Per-layer compression reduces stored feature dimensions or quantizes attention states: grouped-query attention shares keys and values among heads (Ainslie et al., 2023), KIVI quantizes them to two bits (Liu et al., 2024b), Palu caches low-rank projections of a pretrained model (Chang et al., 2025), multi-head latent attention caches one compressed latent per layer (DeepSeek-AI, 2024), and Transformer-VQ learns vector-quantized keys (Lingle, 2024). Depth sharing lets layers reuse each other's cache: CLA shares keys and values between adjacent layers, MiniCache merges similar states of neighboring layers, YOCO computes one global cache for its upper half, LCKV shares top-layer keys and values outside its warmup layers, GoldFinch pairs a compressed cache with a recurrent backbone, and CLLA shares latents with layer-specific projections and optional quantization (Brandon et al., 2024; Liu et al., 2024a; Sun et al., 2024b; Wu & Tu, 2024; Goldstein et al., 2024; Yang et al., 2024). CLLA already provides distinct layer readouts of shared latents; CacheTok instead writes a binary state from the final hidden state and sampled token. A state written after the final layer makes each token wait for its predecessor's full forward pass, serializing training over thousands of tokens; LCKV approximates its training for this reason. Next-scale generators take 10 to 13 steps per image, letting CacheTok write its state after sampling and train through later block losses with distinct layer readouts.

3 Where the Visual History Is Redundant

3.1 Block-autoregressive generation and its cache

A generator produces an image from a condition c, a class label or a text prompt, in K blocks. Block s holds N_s token positions with sampled symbols $R_s = (r_{s,1}, \dots, r_{s,N_s})$, and the model factorizes

$$p_\theta(R_{1:K} \mid c) = \prod_{s=1}^{K} p_\theta\big(R_s \mid R_{<s}, c\big). \tag{1}$$

Each backbone keeps its own sampling rule inside a block (a scale, the bits of a token, or a group of positions), and CacheTok changes only how $R_{<s}$ is represented for later blocks. Before block s there are $N_{<s} = \sum_{u<s} N_u$ historical positions. A backbone with L layers and H KV heads of dimension d_h stores $M_{\mathrm{KV}} = 2N_{<s}LHd_h\beta$ bytes of visual keys and values per image and guidance branch, with β bytes per scalar. For Infinity-2B at 1024^2, $N_{<13} = 6425$, $L = 32$, $H = 16$, and $d_h = 128$, so two BF16 guidance branches hold 3212.5 MiB.

3.2 Old positions are in use, while depth is redundant

M_{KV} is the product of three factors: the positions $N_{<s}$, the layers L, and the $2Hd_h\beta$ bytes that one layer stores per position. Token-selection methods reduce the first factor, yet masking the old positions of the full-cache Infinity-2B costs 2.66 GenEval points (Section 5.4); we keep $N_{<s}$ intact and ask how much new information the layer factor adds.

Fix one image, one KV head, and the history before the final block, with N positions. Let $\boldsymbol{C}_\ell = [\tilde{\boldsymbol{K}}_\ell \mid \boldsymbol{V}_\ell] \in \mathbb{R}^{N\times 2d_h}$ stack the historical keys, taken before key normalization and rotary position, and the values of layer ℓ, and let $\boldsymbol{C} = [\boldsymbol{C}_1 \mid \cdots \mid \boldsymbol{C}_L]$. A *shared* linear state stores $\boldsymbol{S} \in \mathbb{R}^{N\times m}$ and reconstructs every layer as $\boldsymbol{C}_\ell \approx \boldsymbol{S}\boldsymbol{T}_\ell$. *Private* states store $\boldsymbol{S}_\ell \in \mathbb{R}^{N\times m_\ell}$ for each layer with $\sum_\ell m_\ell \le m$. Both are charged m stored coordinates per position.

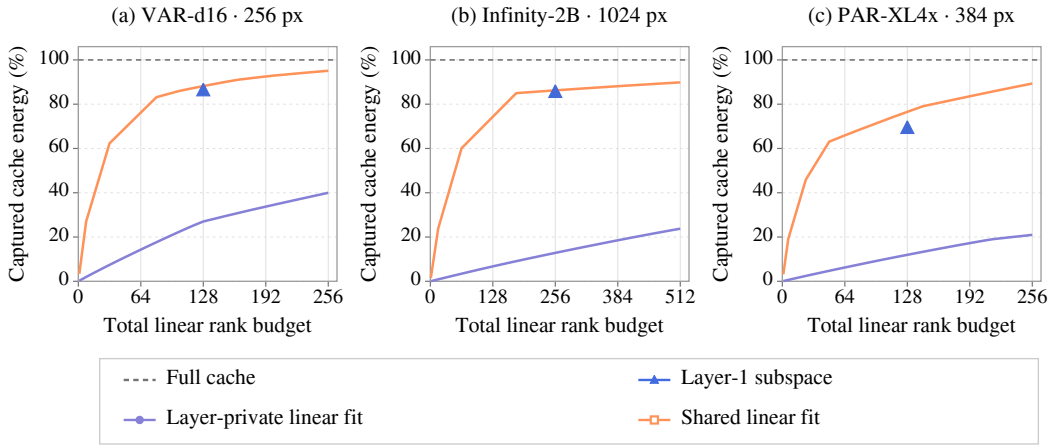


Figure 2: **One subspace shared by all layers captures most of the per-layer cache.** Captured energy of the historical keys and values of all layers against the total number of stored linear coordinates (Full-KV caches before the final block, three seeds). Orange: one shared subspace; purple: optimally allocated layer-private subspaces; triangles: the column space of layer 1 (Appendix D.6).

Theorem 1 (Shared width; proof in Appendix B.1)**.** *Exact reconstruction needs width* $\operatorname{rank}(\boldsymbol{C})$ *with a shared state and total width* $\sum_\ell \operatorname{rank}(\boldsymbol{C}_\ell)$ *with private states. For every budget* m*, the least squared error of a shared state,* $E_{\mathrm{sh}}(m) = \sum_{j>m} \sigma_j(\boldsymbol{C})^2$*, is at most that of private states,* $E_{\mathrm{pr}}(m)$*.*

The ratio $\Gamma = \sum_\ell \operatorname{rank}(\boldsymbol{C}_\ell)/\operatorname{rank}(\boldsymbol{C}) \in [1, L]$ measures how many times the per-layer caches store the same directions, from 1 for independent layer subspaces to L for one common subspace. The theorem concerns subspaces, so layers can share one state even when their key vectors at the same position are orthogonal (Appendix B.2); MiniCache instead merges similar states and retains dissimilar ones (Liu et al., 2024a).

Figure 2 evaluates both errors on the Full-KV caches of the three generators. With 256 stored coordinates in total, one shared subspace captures about 86% of the historical key–value energy of Infinity-2B across its 32 layers, whereas the best private split captures about 13%, and VAR-d16 and PAR-XL4x show the same gap. The layer factor of M_{KV} thus repeats a small set of directions that each layer combines in its own way. Three requirements follow: keep every position readable, store one state per position for all layers, and give each layer its own readout. Section 4 meets them with one binary writer and readers learned across histories.

4 CacheTok: Write Once, Read at Every Layer

CacheTok replaces the per-layer visual cache with one binary state per sampled position and KV head (Figure 3). Block s runs the native L layers with ordinary keys and values for its own tokens and reads the states $\boldsymbol{Z}_{<s}$ of all earlier blocks; after its tokens are sampled, the writer appends $\boldsymbol{Z}_s$. The three parts below decide what is written, how each layer reads it, and how it is learned.

4.1 What to Write, and When

The state must serve every layer of every later block, so it has to describe a position completely at the moment it is stored. A native key or value at layer ℓ is computed while the model predicts the block, from the hidden state of that layer, before the tokens are sampled. It belongs to one layer, and it cannot contain the token drawn at its own position: two different tokens drawn from the same predicted distribution give the same native keys at that position. CacheTok writes after sampling, when two complementary descriptions of each position are available. The final-layer hidden state $\boldsymbol{x}_i \in \mathbb{R}^d$ aggregates the context of all layers and produced the logits at position i; the native embedding $\boldsymbol{e}(r_i) \in \mathbb{R}^d$ identifies the token r_i drawn there. For KV head g, the writer stores

$$\boldsymbol{z}_{i,g} = \tfrac{1}{\sqrt{b}}\,\operatorname{sgn}\big(\boldsymbol{y}_{i,g}\big), \qquad \boldsymbol{y}_{i,g} = \operatorname{RMSNorm}\big(\boldsymbol{x}_i + \boldsymbol{e}(r_i)\big)\,\boldsymbol{W}_g, \tag{2}$$

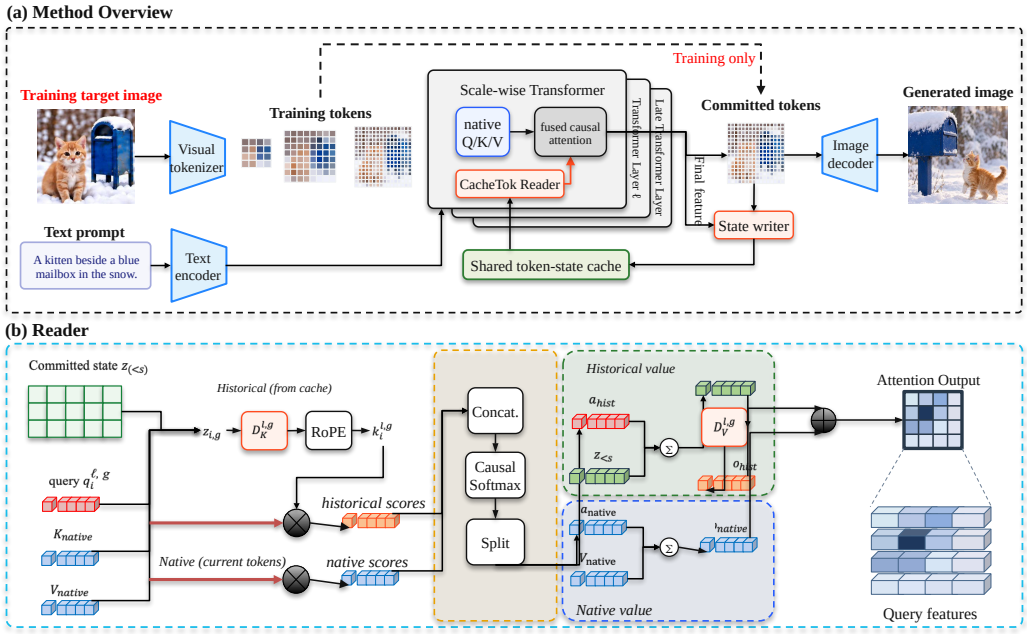


Figure 3: **CacheTok writes one state per sampled token and every layer reads it.** (a) After a block is predicted, the writer combines the final hidden features with the sampled tokens (targets during training) and appends packed bits that only later blocks read. (b) Each layer decodes historical keys, applies the native rotary position, and normalizes historical and current scores in one softmax; historical values are summed in the b state coordinates and decoded once.

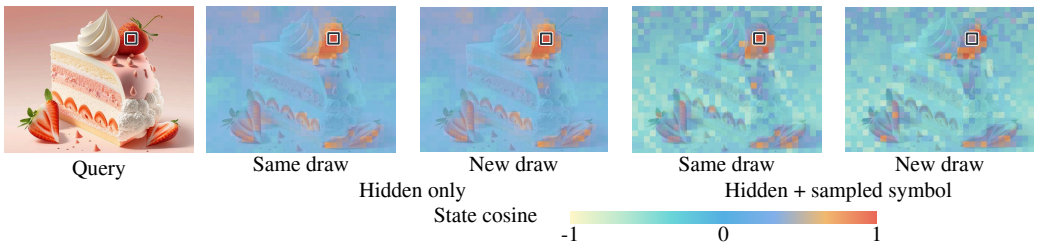


Figure 4: **The sampled token changes the state while the prediction context stays fixed.** Cosine similarity of each state to the marked query for the same draw and a second draw from the same distribution; only the write that includes the token separates the two draws.

with a bias-free projection $\boldsymbol{W}_g \in \mathbb{R}^{d\times b}$ and $\mathrm{sgn}(0) = +1$. RMSNorm, with a learned gain, sets the scale of the summed input. The sign makes the storage exactly one bit per coordinate, and the factor $1/\sqrt{b}$ gives every stored vector unit norm, so the readers always receive inputs of the same scale. States trained in BF16 at the same width reach the same quality (Table 3), so one bit per coordinate keeps what later blocks use.

Figure 4 shows the effect of the token input on one image: with the context fixed, two draws give identical hidden-only states and different states once the token enters the write. During training, r_i is the teacher-forced target token; during sampling, it is the drawn token. In both cases $\boldsymbol{Z}_s$ is written after all logits of block s exist, so no prediction reads a state derived from its own target. With 32-bit packing and b a multiple of 32, block s adds $N_s Hb/8$ bytes per guidance branch, independent of L. At $b = d_h$, the history is $32L$ times smaller than BF16 keys and values.

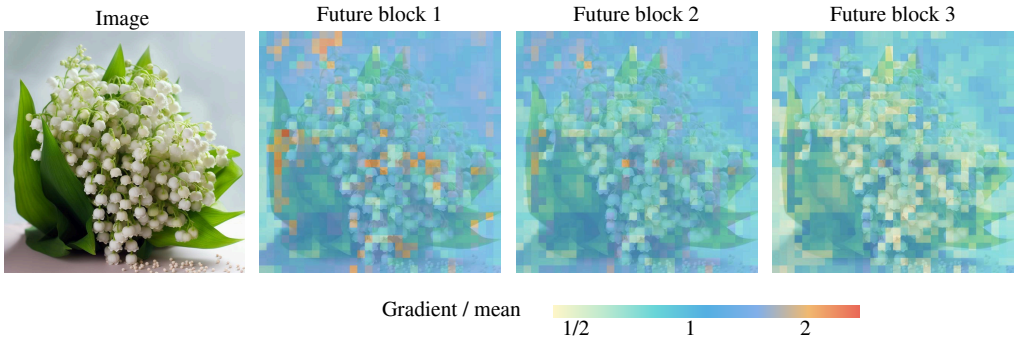


Figure 5: **Different later losses emphasize different written positions.** Gradient norm of three later-block losses with respect to one earlier state, as $\log_2$ ratio to the pooled mean.

4.2 How each layer reads the shared state

Layers issue different queries and need different value features, so one set of historical keys and values cannot serve them all. Cross-layer methods such as CLA and LCKV give several layers identical keys and values (Brandon et al., 2024; Wu & Tu, 2024). CacheTok follows Theorem 1: the state stores the shared subspace, and each layer applies its own linear map. For layer ℓ and head g, with learned $\boldsymbol{D}_K^{\ell,g}, \boldsymbol{D}_V^{\ell,g} \in \mathbb{R}^{b\times d_h}$,

$$\boldsymbol{k}_i^{\ell,g} = \mathrm{Pos}_i^{\ell}\big(\mathrm{Norm}_K^{\ell}(\boldsymbol{z}_{i,g}\boldsymbol{D}_K^{\ell,g})\big), \qquad \boldsymbol{v}_i^{\ell,g} = \boldsymbol{z}_{i,g}\boldsymbol{D}_V^{\ell,g}, \tag{3}$$

where Norm_K^{ℓ} and Pos_i^{ℓ} are the backbone's key normalization and positional transform, or the identity when absent. Decoding comes first, so a historical key enters attention exactly like a native key, with the rotary angle of position i applied at read time.

Historical and current entries compete in one softmax. For a query at layer ℓ, let $\mathcal{I}_{\mathrm{hist}}$ contain the visible historical positions and $\mathcal{I}_{\mathrm{nat}}$ the native self-attention entries of the current block, with weights α_i from a single softmax over $\mathcal{I}_{\mathrm{hist}} \cup \mathcal{I}_{\mathrm{nat}}$. Since every historical value uses the same reader,

$$\boldsymbol{o}^{\ell,g} = \Big(\sum_{i\in\mathcal{I}_{\mathrm{hist}}} \alpha_i\, \boldsymbol{z}_{i,g}\Big)\boldsymbol{D}_V^{\ell,g} + \sum_{j\in\mathcal{I}_{\mathrm{nat}}} \alpha_j\, \boldsymbol{v}_j^{\ell,g}. \tag{4}$$

The historical sum is accumulated in b coordinates and decoded once per query, and the shared normalizer keeps the native attention function. Keys are decoded and rotated one tile at a time in on-chip memory with shared online-softmax statistics, following FlashAttention (Dao, 2024), so the decoded history never reaches device memory (Appendix A.2). Each layer reads b bits per historical position and head in place of $2d_h$ BF16 values, $32\times$ fewer bytes at $b = d_h$, and spends bd_h multiply–adds to decode each historical key.

4.3 How the state is learned

The state of block s affects no loss of block s; it matters only through later predictions. One option is to distill a teacher by making $\boldsymbol{Z}_s$ reproduce the Full-KV keys and values of every layer. That target is the set of L private caches that Section 3.2 found redundant, and it ties quality to a cache the model no longer uses. CacheTok trains the writer, the readers, and the backbone jointly on the backbone's own loss, with the history read from states:

$$\mathcal{L} = \sum_{s=1}^{K} \mathcal{L}_{\mathrm{nat}}\big(R_s \mid \boldsymbol{Z}_{<s}, c\big), \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_s} = \sum_{t=s+1}^{K} \frac{\partial \mathcal{L}_{\mathrm{nat}}(R_t \mid \boldsymbol{Z}_{<t}, c)}{\partial \boldsymbol{Z}_s}. \tag{5}$$

Here $\mathcal{L}_{\mathrm{nat}}$ is the backbone's categorical loss with its native token or bit weighting. The gradient of a state collects the losses of all later blocks, through direct reads and intermediate blocks, and these losses weight the written positions differently (Figure 5), so the write has to serve several later predictions at once. The sign has zero derivative almost everywhere, so the backward pass uses the straight-through surrogate $\partial \boldsymbol{z}_{i,g}/\partial \boldsymbol{y}_{i,g} := b^{-1/2}\boldsymbol{I}$ and the forward pass keeps hard signs (Appendix A.3). The later-block losses alone decide what the b bits store.

Table 1: Quality at the default state width (three-seed means). Upper: CacheTok and its matched Full-KV control; lower: six token-selection methods on the same backbone. P/R: precision/recall; GE: GenEval (0–100); bold: best per panel.

Matched continuation: VAR-d16 / PAR-XL4x / Infinity-2B; three training seeds

Method	VAR-d16, 256^2					PAR-XL4x, 384^2					Infinity-2B, 1024^2			
	ImageNet / ADM					ImageNet / ADM					GE	DPG	MJHQ	HPS
	FID ↓	sFID ↓	IS ↑	P ↑	R ↑	FID ↓	sFID ↓	IS ↑	P ↑	R ↑	Score ↑	Score ↑	FID ↓	Score ↑
Full-KV	**3.55**	6.43	274.40	**0.8403**	**0.5125**	**2.63**	**5.96**	**259.73**	0.8206	0.5585	**69.2**	**83.47**	**9.82**	**30.48**
CacheTok	3.58	**6.40**	**275.07**	0.8387	0.5106	2.65	6.00	259.20	**0.8217**	**0.5587**	68.9	83.31	9.85	**30.48**

Matched external comparison: VAR-d16, 256^2; Infinity-2B, 1024^2

Method	Mechanism	VAR-d16	Infinity-2B			
		FID ↓	GenEval ↑	DPG ↑	MJHQ FID ↓	HPS ↑
Full-KV	Reference	**3.55**	**69.2**	**83.47**	**9.82**	**30.48**
HACK (Qin et al., 2026)	Head selection	3.64	68.4	83.00	10.01	30.39
ScaleKV (Li et al., 2025b)	Scale budgets	3.71	67.7	82.45	10.17	30.29
FastVAR (Guo et al., 2025)	Prune / restore	3.83	67.4	82.55	10.46	30.12
AMS-KV (Xu et al., 2026)	KV condensation	3.60	68.6	83.25	9.91	30.46
SparVAR (Li et al., 2026b)	Sparse reads	3.57	68.8	83.40	9.93	30.45
ToProVAR-F (Chen et al., 2026)	Entropy pruning	3.56	68.9	83.34	9.84	**30.48**
CacheTok	Binary history	3.58	68.9	83.31	9.85	**30.48**

The price is a longer training dependency. Treat each block-layer pair as a unit-time stage, and let layer ℓ read, from every earlier block, a state produced at depth p_ℓ ($p_\ell = 0$: ready at the input).

Theorem 2 (Training span; proof in Appendix B.4)**.** *For $K \geq 2$ blocks, the minimum length of a schedule of the forward dependency graph is*

$$T_{\text{span}} = L + (K-1)\,\Delta, \qquad \Delta = \max\Big(0,\ \max_{\ell}\,(p_\ell - \ell + 1)\Big). \tag{6}$$

A native cache is produced at the input of the layer that reads it, $p_\ell = \ell - 1$, so $\Delta = 0$ and teacher forcing processes all blocks in one pass of L stages. CacheTok writes at $p_\ell = L$, so $\Delta = L$ and $T_{\text{span}} = KL$, which is 160, 5292, and 416 stages for VAR-d16, PAR-XL4x, and Infinity-2B (Table 2). Positions inside a block stay parallel, so the cost grows with the number of blocks, and gradient checkpointing lowers activation memory along the same path. Sampling runs the blocks in order anyway and keeps the native K backbone calls; each guidance branch writes its own history.

5 Experiments

Setup. We fine-tune three public generators with CacheTok from their released checkpoints: VAR-d16 at 256^2 (class-conditional, 10 scales), PAR-XL4x at 384^2 (class-conditional, 147 position groups), and Infinity-2B at 1024^2 (text-conditional, 13 bitwise scales), with $b = d_h$, that is, 64, 64, and 128 bits (Tian et al., 2024; Wang et al., 2025; Han et al., 2025). A Full-KV control receives the same initialization, data, and 10,000 updates, and six history-reduction methods run on this backbone (Appendix C). The class-conditional models are evaluated with ImageNet ADM metrics, and Infinity-2B with GenEval, DPG-Bench, MJHQ-30K FID, and HPSv2.1 (Ghosh et al., 2023; Hu et al., 2024; Li et al., 2024b; Wu et al., 2023). Tables report means over training seeds 17, 29, and 43, and paired differences use a t interval with two degrees of freedom.

5.1 Quality with a 512–1152× smaller history

In the upper panel of Table 1, FID changes by $+0.03$ on VAR-d16 and $+0.02$ on PAR-XL4x, sFID by at most 0.04, IS by at most 0.7, and precision and recall by at most 0.002. On Infinity-2B, GenEval changes by -0.23 points with a paired 95% interval of $[-1.81, 1.35]$, DPG by -0.16, MJHQ FID by $+0.03$, and HPSv2.1 stays at 30.48; the extension to four larger backbones shows the same pattern (Appendix F).

In the lower panel, methods that keep most positions readable (SparVAR, AMS-KV, ToProVAR-F) stay within 0.02 FID and 0.3 GenEval points of CacheTok, while methods that delete more lose more: ScaleKV, which retains 10% of the cache, reaches FID 3.71 and GenEval 67.7, and FastVAR 3.83 and 67.4 (Section 5.4 isolates this trend).

The width sweep in Figure 19 (Appendix D.6) shows quality rising steeply up to $b = d_h$ and changing little beyond it: Infinity-2B gains 4.20 GenEval points from 64 to 128 bits (95% CI $[3.01, 5.39]$) and 0.26 points from 128 to 256 bits ($[-0.53, 1.05]$), and the shared energy of Infinity-2B in Figure 2 levels off in the same range. We therefore use $b = d_h$.

5.2 Memory, speed, and batch capacity

Table 2 counts the persistent visual history in bytes. One d_h-bit state in place of BF16 keys and values at L layers gives the predicted $32L$ reduction, $512\times$, $1152\times$, and $1024\times$; the history of one Infinity-2B image falls from 3212.5 to 3.1 MiB. The costs are 3.1–21.0 million added parameters, shared by all requests, and the longer training span in the last column.

Table 2: Visual history (MiB, one image, two branches), added parameters (M), and training span (layer stages).

Backbone	Full KV	Binary	Factor	New M	Span
VAR-d16	53.000	0.104	512×	3.147	160
PAR-XL4x	201.094	0.175	1152×	7.538	5292
Infinity-2B	3212.500	3.137	1024×	20.974	416

At batch one, CacheTok generates an Infinity-2B image in 4.62 s against 4.85 s for Full-KV ($1.05\times$), since most GPU time goes to the current scale, text encoding, and image decoding; the history matters once memory limits the batch. At a common batch of ten, Infinity-2B throughput rises from 0.255 to 0.372 images/s ($1.46\times$), as each layer reads $32\times$ fewer historical bytes; VAR-d16 and PAR-XL4x gain $1.07\times$ and $1.14\times$ at their common batches. The analytic capacity of one 80 GB A100 grows from 11 to 22 Infinity-2B images and stops there, because the 2.7 GiB of temporary tensors of the final 64×64 scale then dominate per-image memory (Appendix E). Figure 24 (Appendix E) adds the token-selection methods, which run up to $2\times$ faster at batch one and keep at least a tenth of the history.

5.3 What each design choice contributes

Table 3 changes one part at a time; every change except continuous training lowers quality. *Both writer inputs carry information.* Removing the hidden state raises VAR FID to 4.28 and lowers Infinity GenEval to 64.3; removing the token embedding gives 3.97 and 65.8. The embedding matters more for PAR (FID 3.42 without it, 3.13 without the hidden state), which samples at temperature 1 without truncation, while VAR truncates to top-$p = 0.96$ and Infinity samples at temperature 0.5. A broader sampling distribution leaves more of the drawn token unexplained by its context (Figure 4).

Table 3: Design controls (three-seed means); $b = 64$ for VAR/PAR, 128 for Infinity-2B. GE: GenEval.

Variant	VAR	PAR	Infinity-2B	
	FID ↓	FID ↓	GE ↑	DPG ↑
Default	**3.58**	**2.65**	**68.9**	**83.31**
No symbol input	3.97	3.42	65.8	79.86
No hidden input	4.28	3.13	64.3	78.62
Tied readers	4.77	4.46	62.2	76.26
Stop state gradient	5.34	4.09	61.2	75.76
Train continuous	**3.56**	**2.62**	**68.8**	**83.54**
Sign off at inference	5.21	5.87	58.6	72.48

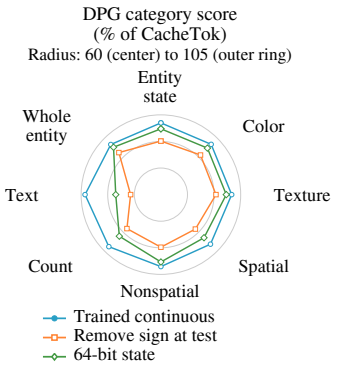


Figure 6: **Readers depend on state training.** DPG scores, % of the binary default.

Layers need their own readers. Tying $\boldsymbol{D}_K^{\ell,g}$ and $\boldsymbol{D}_V^{\ell,g}$ across layers raises VAR FID to 4.77 and lowers GenEval to 62.2, although the tied model still writes full-rank, balanced states (Figure 21): each layer loses its own view (Theorem 1).

The write must be learned from later blocks. Stopping the gradient into $\boldsymbol{Z}_s$ freezes the writer at its initialization and is the most damaging training change on VAR and Infinity (VAR FID 5.34, GenEval 61.2), although the frozen writer also produces full-rank, balanced states. The future losses select which directions the b bits keep.

The readers adapt to the bits. Removing the sign only at inference drops GenEval to 58.6 and raises PAR FID to 5.87, because the readers were trained on unit-norm sign patterns and the pre-sign activations change the direction and norm of the decoded keys (Figure 16). Figure 6 resolves this by DPG category. The BF16-trained state stays near the binary

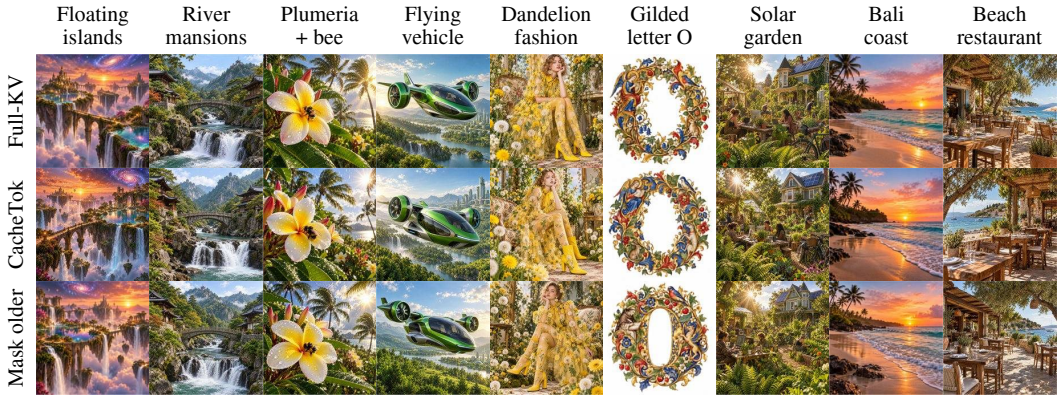


Figure 7: **Complete outputs with old positions masked.** Full-KV, CacheTok, and CacheTok with reads of blocks $u < s - 1$ masked; weights and stored states are fixed.

default in every category, inference-only sign removal lowers all eight, most in text and counting, and a 64-bit state mostly lies in between.

5.4 Why every position is kept

To test whether later blocks use old positions, we mask all direct reads of blocks older than $s - 1$ at inference, with weights and stored states fixed (Table 4). Both models degrade by similar amounts: FID rises by 0.72 and 0.85 with Full-KV and by 0.58 and 1.07 with CacheTok on VAR and PAR, and Infinity-2B GenEval falls by 2.66 and 2.59 points and DPG by 3.14 and 3.83 points; the paired intervals of all CacheTok changes exclude zero. Deleting old positions pays this loss or part of it; CacheTok keeps them all for 3.1 MiB per Infinity-2B image.

Table 4: Masking reads of blocks older than $s-1$, weights and states fixed (three-seed means).

Method	VAR	PAR	Infinity-2B	
	FID ↓	FID ↓	GE ↑	DPG ↑
Intact generation				
Full-KV	**3.55**	**2.63**	**69.16**	**83.47**
CacheTok	3.58	2.65	68.94	83.31
Old-position masking: masked minus intact				
Full-KV	+0.72	+0.85	-2.66	-3.14
CacheTok	+0.58	+1.07	-2.59	-3.83

Per GenEval task (Figure 15), single-object success hardly changes, while the multi-object judgments lose 2.7–4.7 points in both models: coarse scales fix where objects are, and fine scales read that placement again. Masking the b bits of CacheTok costs as much as masking the full per-layer entries, so later blocks read these bits as actively as the full cache. The methods that delete the most history in Table 1, ScaleKV and FastVAR, also lose the most GenEval points (1.5 and 1.8). In complete outputs (Figure 7), each masked sample keeps the subject and layout of its prompt and changes in local structure.

The mask and CacheTok shrink the history in two different ways. The mask keeps only the previous block (169 of 424 historical positions in VAR-d16, 2304 of 6425 in Infinity-2B), removes 60–64% of the positions, and costs 0.6–0.7 FID and about 2.6 GenEval points. CacheTok removes 99.8% or more of the stored bytes, keeps every position, and costs 0.03 FID and 0.23 GenEval points (Table 1). Removing bytes per position costs far less quality than removing positions, which confirms the choice of Section 3.2.

6 Conclusion

In block-autoregressive image generation, the history of a sampled block is fixed content that every later block and every layer reads. CacheTok writes it once per position as a binary state computed from the final hidden state and the drawn token, and every layer reads it through its own linear readers. The cache spectra, design controls, and masking test show why one state suffices, why each part of the write and read is needed, and why every position is kept. At matched quality on three generators, and with simplified normalization on four larger ones, the visual history becomes 512–1152× smaller; the price is a training critical path that grows with the number of steps.

AI USE STATEMENT

AI tools assisted research ideation, literature search, mathematical checks, figure preparation, and drafting and polishing paper sections. The authors reviewed all AI-assisted content and take responsibility for the final text, methodology, results, and conclusions.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.298/.

William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan-Kelley. Reducing Transformer Key-Value Cache Size with Cross-Layer Attention. In *Advances in Neural Information Processing Systems*, 2024. URL https://doi.org/10.52202/079017-2758.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S. Abdelfattah, and Kai-Chiang Wu. Palu: KV-Cache Compression with Low-Rank Projection. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2407.21118.

Jiayu Chen, Ruoyu Lin, Zihao Zheng, Jingxin Li, Maoliang Li, Guojie Luo, and Xiang Chen. ToProVAR: Efficient Visual Autoregressive Modeling via Tri-Dimensional Entropy-Aware Semantic Analysis and Sparsity Optimization. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.22948.

Zhuokun Chen, Jugang Fan, Zhuowei Yu, Bohan Zhuang, and Mingkui Tan. Frequency-Aware Autoregressive Modeling for Efficient High-Resolution Image Synthesis. *arXiv preprint arXiv:2507.20454*, 2025. URL https://arxiv.org/abs/2507.20454.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.08691.

DeepSeek-AI. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv preprint arXiv:2405.04434*, 2024. URL https://arxiv.org/abs/2405.04434.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.01801.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://doi.org/10.52202/075280-2270.

Daniel Goldstein, Fares Obeid, Eric Alcaide, Guangyu Song, and Eugene Cheah. GoldFinch: High Performance RWKV/Transformer Hybrid with Linear Pre-Fill and Extreme KV-Cache Compression. *arXiv preprint arXiv:2407.12077*, 2024. URL https://arxiv.org/abs/2407.12077.

Hang Guo, Yawei Li, Taolin Zhang, Jiangshan Wang, Tao Dai, Shu-Tao Xia, and Luca Benini. FastVAR: Linear Visual Autoregressive Modeling via Cached Token Pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2503.23367.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling Bitwise AutoRegressive Modeling for High-Resolution Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2412.04431.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip Diffusion Models with LLM for Enhanced Semantic Alignment. *arXiv preprint arXiv:2403.05135*, 2024. URL https://arxiv.org/abs/2403.05135.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A Comprehensive Benchmark for Open-world Compositional Text-to-image Generation. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2307.06350v2.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench++: An Enhanced and Comprehensive Benchmark for Compositional Text-to-image Generation. *arXiv preprint arXiv:2307.06350v3*, 2025. URL https://arxiv.org/abs/2307.06350v3.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. GenAI-Bench: Evaluating and Improving Compositional Text-to-Visual Generation. *arXiv preprint arXiv:2406.13743*, 2024a. URL https://arxiv.org/abs/2406.13743.

Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three Insights towards Enhancing Aesthetic Quality in Text-to-Image Generation. *arXiv preprint arXiv:2402.17245*, 2024b. URL https://arxiv.org/abs/2402.17245.

Jiajun Li, Yue Ma, Xinyu Zhang, Qingyan Wei, Songhua Liu, and Linfeng Zhang. SkipVAR: Accelerating Visual Autoregressive Modeling via Adaptive Frequency-Aware Skipping. *arXiv preprint arXiv:2506.08908*, 2025a. URL https://arxiv.org/abs/2506.08908.

Kunjun Li, Zigeng Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Memory-Efficient Visual Autoregressive Modeling with Scale-Aware KV Cache Compression. In *Advances in Neural Information Processing Systems 38*, 2025b. URL https://arxiv.org/abs/2505.19602.

Senmao Li, Kai Wang, Salman Khan, Fahad Shahbaz Khan, Jian Yang, and Yaxing Wang. FasterVAR: Plug-and-Play Acceleration for Visual Autoregressive Models. In *International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2512.16483.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive Image Generation without Vector Quantization. In *Advances in Neural Information Processing Systems*, 2024c. URL https://arxiv.org/abs/2406.11838.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM Knows What You are Looking for Before Generation. In *Advances in Neural Information Processing Systems*, 2024d. URL https://arxiv.org/abs/2404.14469.

Zekun Li, Ning Wang, Tongxin Bai, Changwang Mei, Peisong Wang, Shuang Qiu, and Jian Cheng. SparVAR: Exploring Sparsity in Visual AutoRegressive Modeling for Training-Free Acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b. URL https://arxiv.org/abs/2602.04361.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating Text-to-Visual Generation with Image-to-Text Generation. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2404.01291.

Lucas D. Lingle. Transformer-VQ: Linear-Time Transformers via Vector Quantization. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16354.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. MiniCache: KV Cache Compression in Depth Dimension for Large Language Models. In *Advances in Neural Information Processing Systems*, 2024a. URL https://doi.org/10.52202/079017-4443.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. In *International Conference on Machine Learning*, 2024b. URL https://arxiv.org/abs/2402.02750.

Ziran Qin, Youru Lv, Mingbao Lin, Hang Guo, Zeren Zhang, Danping Zou, and Weiyao Lin. Head-Aware KV Cache Compression for Efficient Visual Autoregressive Modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. URL https://arxiv.org/abs/2504.09261.

Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, Jifeng Dai, Yu Qiao, Limin Wang, and Hongsheng Li. JourneyDB: A Benchmark for Generative Image Understanding. In *Advances in Neural Information Processing Systems*, 2023. URL https://doi.org/10.52202/075280-2161.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive Model Beats Diffusion: Llama for Scalable Image Generation. *arXiv preprint arXiv:2406.06525*, 2024a. URL https://arxiv.org/abs/2406.06525.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. You Only Cache Once: Decoder-Decoder Architectures for Language Models. In *Advances in Neural Information Processing Systems*, 2024b. URL https://doi.org/10.52202/079017-0235.

Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. HART: Efficient Visual Generation with Hybrid Autoregressive Transformer. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10812.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.02905.

Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized Autoregressive Visual Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2412.15119.

Haoyi Wu and Kewei Tu. Layer-Condensed KV Cache for Efficient Inference of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL https://doi.org/10.18653/v1/2024.acl-long.602.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis. *arXiv preprint arXiv:2306.09341*, 2023. URL https://arxiv.org/abs/2306.09341.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.17453.

Yi Xin, Juncheng Yan, Qi Qin, Zhen Li, Dongyang Liu, Shicheng Li, Victor Shea-Jay Huang, Yupeng Zhou, Renrui Zhang, Le Zhuo, Tiancheng Han, Xiaoqing Sun, Siqi Luo, Mengmeng Wang, Bin Fu, Yuewen Cao, Hongsheng Li, Guangtao Zhai, Xiaohong Liu, Yu Qiao, and Peng Gao. Lumina-mGPT 2.0: Stand-Alone AutoRegressive Image Modeling. *arXiv preprint arXiv:2507.17801*, 2025. URL https://arxiv.org/abs/2507.17801.

Boxun Xu, Yu Wang, Zihu Wang, and Peng Li. AMS-KV: Adaptive KV Caching in Multi-Scale Visual Autoregressive Transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. URL https://arxiv.org/abs/2511.16047.

Zhen Yang, J. N. Han, Kan Wu, Ruobing Xie, An Wang, Xingwu Sun, and Zhanhui Kang. Lossless KV Cache Compression to 2%. *arXiv preprint arXiv:2410.15252*, 2024. URL https://arxiv.org/abs/2410.15252.

Jinhua Zhang, Wei Long, Minghao Han, Weiyi You, and Shuhang Gu. MVAR: Visual Autoregressive Modeling with Scale and Spatial Markovian Conditioning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2505.12742.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.14048.

A Implementation Details

Notation. Blocks are indexed by $s, t, u \in \{1, \dots, K\}$, layers by $\ell \in \{1, \dots, L\}$, KV heads by $g \in \{1, \dots, H\}$, and positions by i, j. Bold lowercase letters denote row vectors ($\boldsymbol{x}_i$, $\boldsymbol{z}_{i,g}$, $\boldsymbol{k}_i^{\ell,g}$), bold uppercase letters denote matrices ($\boldsymbol{W}_g$, $\boldsymbol{D}_K^{\ell,g}$, $\boldsymbol{Z}_s$), and calligraphic letters denote index sets ($\mathcal{I}_{\text{hist}}$, $\mathcal{I}_{\text{nat}}$). d is the model width, d_h the head dimension, and b the number of state bits per head.

A.1 State ownership and the generation loop

Each native visual prediction position owns one state per KV head and guidance branch. Ownership follows the original position, including its block, spatial index, and any native sub-position identity, so a repeated bit pattern neither removes a position nor pools two positions. States are stored in append order, and block offsets with the native positional schedule determine their addresses; this metadata is outside the bit payload. Text self-attention entries, text cross-attention state, class-conditioning buffers, and other backbone-specific persistent tensors keep their native form and are counted separately in whole-request memory.

At block s, generation (i) builds the native block input from the condition and the tokens sampled so far, (ii) evaluates all L layers with $\boldsymbol{Z}_{<s}$ as visual history and ordinary temporary keys and values for the current block, (iii) forms the native output distribution and samples R_s, and (iv) evaluates Eq. (2) and appends the packed signs if $s < K$. The temporary keys and values of the current block are then released. The writer adds one projection per position and no second pass through the backbone, so the K backbone calls of the native sampler remain, with writer and reader operations inside and between them.

Figure 8 shows why each guidance branch stores its own history. Both branches use the same sampled tokens and matched readers, but their writer inputs contain different hidden states. The decoded keys of the conditional and unconditional branches share their coarse layout and differ in local detail at both readers. The storage accounting of Table 2 therefore counts two branches, and each branch owns its states as described above.

A.2 Joint attention with a shared online normalizer

For a query $\boldsymbol{q}$ at layer ℓ, let $a_i = \tau_\ell \langle \boldsymbol{q}, \boldsymbol{k}_i \rangle + \kappa_i$ be the native scaled logit with additive mask κ_i. The weights normalize over all allowed historical and current entries,

$$\alpha_i = \frac{\exp(a_i)}{\sum_{j \in \mathcal{I}_{\text{hist}} \cup \mathcal{I}_{\text{nat}}} \exp(a_j)}, \qquad \boldsymbol{o} = \sum_{i \in \mathcal{I}_{\text{hist}}} \alpha_i \boldsymbol{v}_i + \sum_{j \in \mathcal{I}_{\text{nat}}} \alpha_j \boldsymbol{v}_j. \tag{7}$$

Masked positions are excluded from the denominator, and attention dropout, when the backbone uses it, follows its native rule during training. Under grouped-query attention, each query head uses the reader of its KV head and computes its own weights; the evaluated backbones use multi-head attention.

A 32-bit word stores 32 signs of one head and position. When b is not a multiple of 32, the unused tail bits are masked and storage is rounded up per vector; all widths in this paper are multiples of

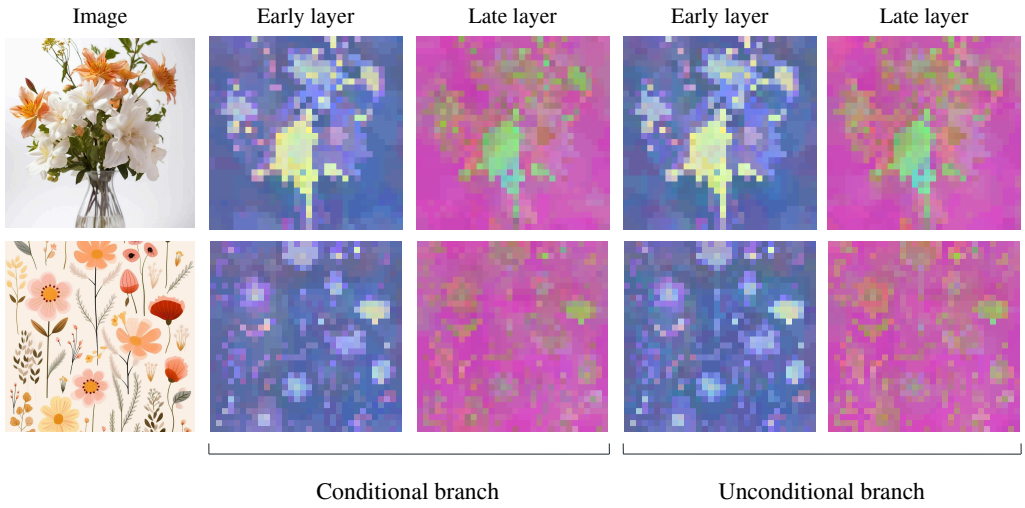


Figure 8: **Shared sampled tokens leave the guidance-branch states distinct.** Two MJHQ-30K plant samples: a white-flower vase (plants/$bc9adb26\ldots$) and a floral-pattern image (viewer row 24008). Each row shows RGB and key-PCA readouts from an early and a late reader in the conditional and unconditional branches. Both branches use the same token embedding $\boldsymbol{e}(r_i)$ and matched readers, with their own hidden inputs to the writer in Eq. (2). All eight key maps use one pooled PCA basis and common component ranges.

32. Unpacking maps a bit to $\pm 1/\sqrt{b}$ with no learned scale or zero point. Keys are decoded before native key normalization and rotary position, and values use Eq. (4) because one reader serves all historical positions of a layer. Reader parameters are model weights shared by all requests.

Online normalizer. At inference, each query keeps a running maximum logit $a_{\max}$, a denominator ν, a historical accumulator $\boldsymbol{\mu}_{\text{hist}} \in \mathbb{R}^b$, and a native accumulator $\boldsymbol{\mu}_{\text{nat}} \in \mathbb{R}^{d_h}$, initialized to $-\infty$ and zeros. For a nonempty tile $\mathcal{J}$ of allowed entries, the kernel decodes the historical keys, forms their logits, and sets $a'_{\max} = \max(a_{\max}, \max_{i\in\mathcal{J}} a_i)$, $\rho = \exp(a_{\max} - a'_{\max})$, and $w_i = \exp(a_i - a'_{\max})$. It then updates

$$\nu' = \rho\,\nu + \sum_{i\in\mathcal{J}} w_i, \qquad \boldsymbol{\mu}'_{\text{hist}} = \rho\,\boldsymbol{\mu}_{\text{hist}} + \sum_{i\in\mathcal{J}\cap\mathcal{I}_{\text{hist}}} w_i\,\boldsymbol{z}_{i,g}, \qquad \boldsymbol{\mu}'_{\text{nat}} = \rho\,\boldsymbol{\mu}_{\text{nat}} + \sum_{j\in\mathcal{J}\cap\mathcal{I}_{\text{nat}}} w_j\,\boldsymbol{v}_j. \tag{8}$$

For a query with at least one allowed entry, the output is $(\boldsymbol{\mu}_{\text{hist}}\boldsymbol{D}_V^{\ell,g} + \boldsymbol{\mu}_{\text{nat}})/\nu$. The shared rescaling reproduces the joint softmax of Eq. (7) exactly in real arithmetic; floating-point results depend on the reduction order. Decoded key tiles are discarded after use, and decoded historical values are never formed. Appendix B.3 bounds how errors in decoded keys and values change the output of one attention operation.

A.3 Straight-through backward pass

Under teacher forcing, the target tokens are fixed data, so no gradient passes through sampling. Gradients do pass through every later use of a state, including indirect paths through intermediate hidden states and writes. Let $\boldsymbol{G}_s$ be the accumulated adjoint of $\boldsymbol{Z}_s$ from all consumers, and let $\boldsymbol{U}_s$ stack the normalized writer inputs $\mathrm{RMSNorm}(\boldsymbol{x}_i + \boldsymbol{e}(r_i))$ of block s. With the surrogate of Section 4.3, the writer adjoints are

$$\bar{\boldsymbol{Y}}_s = b^{-1/2}\boldsymbol{G}_s, \qquad \bar{\boldsymbol{U}}_s = \bar{\boldsymbol{Y}}_s\boldsymbol{W}^\top, \qquad \bar{\boldsymbol{W}} = \sum_{s=1}^{K-1} \boldsymbol{U}_s^\top \bar{\boldsymbol{Y}}_s, \tag{9}$$

where $\boldsymbol{W} = [\boldsymbol{W}_1 \mid \cdots \mid \boldsymbol{W}_H]$ and $\bar{\boldsymbol{Y}}_s$ is flattened over heads. The shared writer accumulates contributions from every consumed write, and the RMSNorm backward pass sends $\bar{\boldsymbol{U}}_s$ to both $\boldsymbol{x}_i$ and $\boldsymbol{e}(r_i)$.

An integer tensor cannot carry floating-point autograd by itself. The implementation therefore exposes floating-point read outputs to the autograd engine and keeps signs packed inside a joint write–read operation. The backward pass replays or retrieves the writer inputs, differentiates the native reader operations, accumulates $\boldsymbol{G}_s$, and applies Eq. (9). Checkpointing trades recomputation for activation memory and keeps the dependency graph unchanged. Training memory additionally includes saved activations, adjoints, and recomputation workspaces.

A.4 Parameter and memory accounting

The writer has dHb projection parameters and d RMSNorm gains, and each of the L layers has H key and value readers with bd_h parameters each. The added parameters and the packed visual payload in bytes are

$$n_{\mathrm{add}} = dHb + 2LHbd_h + d, \qquad M_{\mathrm{bin}} = 4\, B_{\mathrm{img}}\, C_{\mathrm{cfg}}\, NH\lceil b/32 \rceil, \tag{10}$$

where N is the peak number of historical positions, B_{img} the image batch size, and C_{cfg} the number of stored guidance branches. The parameter count grows with L while the request state does not, so a depth-independent history comes with depth-dependent readers. Current-block tensors, persistent conditioning, allocator reservation, and kernel workspaces are added at their execution peaks in Appendix E.

B Proofs and Auxiliary Results

B.1 Shared and private linear states

Fix one history and one KV head. For each layer, $\boldsymbol{C}_\ell \in \mathbb{R}^{N\times 2d_h}$ concatenates the target keys before native key processing and the target values. A shared linear state factors all targets as $\boldsymbol{C}_\ell = \boldsymbol{S}\boldsymbol{T}_\ell$ with $\boldsymbol{S} \in \mathbb{R}^{N\times m}$ and real readers $\boldsymbol{T}_\ell$. A private state uses $\boldsymbol{C}_\ell = \boldsymbol{S}_\ell \boldsymbol{T}'_\ell$ with widths m_ℓ charged separately. All matrices of this oracle are chosen for the fixed history; CacheTok instead learns binary states and readers across histories.

Lemma B.1 (Exact width)**.** *For a matrix $\boldsymbol{X}$, the minimum inner dimension of a factorization $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B}$ is* $\mathrm{rank}(\boldsymbol{X})$.

Proof. A product $\boldsymbol{A}\boldsymbol{B}$ has rank at most its inner dimension, which gives the lower bound. Conversely, let the columns of $\boldsymbol{A}$ be a basis of the column space of $\boldsymbol{X}$ and let the columns of $\boldsymbol{B}$ hold the coordinates of the columns of $\boldsymbol{X}$ in that basis; this uses $\mathrm{rank}(\boldsymbol{X})$ coordinates. Applied to $\boldsymbol{C} = [\boldsymbol{C}_1 \mid \cdots \mid \boldsymbol{C}_L]$, the argument gives the shared width; applied to each $\boldsymbol{C}_\ell$, it gives the private widths, whose sum is charged because a private state serves only its own layer. □

Lemma B.2 (Optimal approximation at fixed width)**.** *For real factors with inner dimension at most m,*

$$\min_{\boldsymbol{A},\boldsymbol{B}} \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}\|_F^2 = \sum_{j>m} \sigma_j(\boldsymbol{X})^2, \tag{11}$$

where singular values beyond the rank are zero.

Proof. For a fixed column space $\mathcal{U}$, the orthogonal projection $\boldsymbol{P}_{\mathcal{U}}\boldsymbol{X}$ is the closest matrix with columns in $\mathcal{U}$, and it captures the squared norm $\mathrm{tr}(\boldsymbol{P}_{\mathcal{U}}\boldsymbol{X}\boldsymbol{X}^\top)$. In an eigenbasis of $\boldsymbol{X}\boldsymbol{X}^\top$ this trace equals $\sum_j \lambda_j w_j$ with $0 \le w_j \le 1$ and $\sum_j w_j \le m$, which is at most $\sum_{j\le m} \lambda_j$, attained by the top eigenvectors. Subtracting from $\|\boldsymbol{X}\|_F^2$ gives the singular-value tail, and a factorization of the projected matrix attains it by Lemma B.1. □

Proof of Theorem 1. The optimal errors are

$$E_{\mathrm{sh}}(m) = \sum_{j>m} \sigma_j(\boldsymbol{C})^2, \qquad E_{\mathrm{pr}}(m) = \min_{m_\ell \in \mathbb{N}_0,\ \sum_\ell m_\ell \le m} \sum_{\ell=1}^{L} \sum_{j>m_\ell} \sigma_j(\boldsymbol{C}_\ell)^2. \tag{12}$$

The exact widths follow from Lemma B.1. Concatenating the shared readers gives one rank-m approximation of $\boldsymbol{C}$, so Lemma B.2 gives E_{sh}. For a fixed private allocation $(m_1, \ldots, m_L)$, each

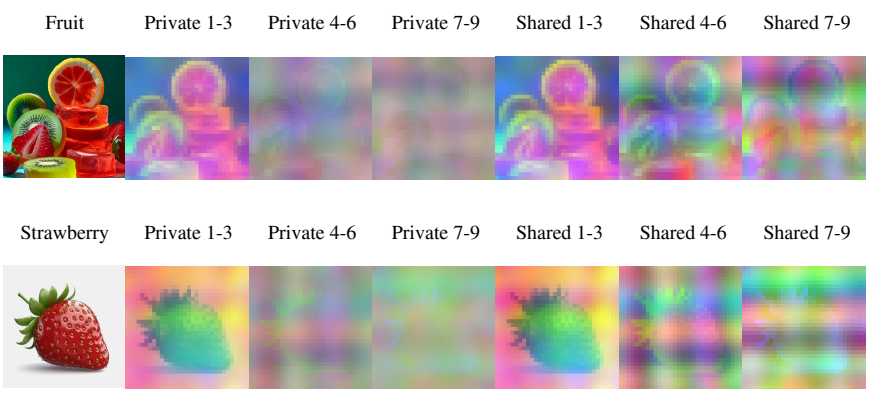


Figure 9: **Equal total rank yields different spatial reconstructions.** MJHQ-30K fruit slices (food/*bce*8671*d*...) and strawberry (food/5*fcf*208*e*...). For a four-layer target, shared and optimally allocated layer-private fits both use total rank $m = 16$. Columns encode principal components 1–3, 4–6, and 7–9 of the reconstructed target, with PCA axes from the full target and shared component ranges within each sample; colors are fitted per sample.

layer attains its own singular-value tail, and minimizing over allocations gives E_{pr}. Finally, every private solution is a shared solution: concatenate $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_L$ and let each layer's reader select its own coordinates. The shared width is $\sum_\ell m_\ell \le m$, with the same outputs and error. The shared feasible set therefore contains every private solution at equal total width, which proves $E_{\mathrm{sh}}(m) \le E_{\mathrm{pr}}(m)$. □

When $\boldsymbol{C} \neq \boldsymbol{0}$, the reuse ratio of Section 3.2,

$$\Gamma = \frac{\sum_\ell \operatorname{rank}(\boldsymbol{C}_\ell)}{\operatorname{rank}(\boldsymbol{C})}, \qquad 1 \le \Gamma \le L, \tag{13}$$

is bounded below because the dimension of a sum of column spaces is at most the sum of their dimensions, and above because each layer's column space lies in the joint space. $\Gamma = L$ requires all layer column spaces to equal one nonzero joint space, and $\Gamma = 1$ holds exactly when the sum is direct.

Proposition B.1 (Restriction to an existing source)**.** *If the shared state is fixed to $\boldsymbol{C}_j$, the minimum error of linear readers of that state is*

$$\min_{\boldsymbol{T}} \|\boldsymbol{C} - \boldsymbol{C}_j \boldsymbol{T}\|_F^2 = \|(\boldsymbol{I} - \boldsymbol{P}_j)\boldsymbol{C}\|_F^2, \qquad \boldsymbol{P}_j = \boldsymbol{C}_j \boldsymbol{C}_j^\dagger. \tag{14}$$

Proof. Every column of $\boldsymbol{C}_j \boldsymbol{T}$ lies in $\operatorname{col}(\boldsymbol{C}_j)$. The best approximation of each column of $\boldsymbol{C}$ is its orthogonal projection onto that space, attained by $\boldsymbol{T} = \boldsymbol{C}_j^\dagger \boldsymbol{C}$; summing the residual norms gives the expression. □

A learned state can span directions absent from any single layer, which reuse schemes that feed one layer's cache to others cannot. The triangles in Figure 2 evaluate Eq. (14) for $j = 1$.

Figure 9 shows the same budget difference in image space. At equal total rank $m = 16$ on a four-layer target, the shared fit keeps object structure in principal components 4–9, where the private fit is blurred; Eq. (12) gives the corresponding squared errors.

B.2 Binary separation and a misleading similarity diagnostic

Proposition B.2 (A binary family with depth-factor reuse)**.** *For $1 \le b \le \min(N, 2d_h)$, there exist binary states $\boldsymbol{Z} \in \{\pm 1/\sqrt{b}\}^{N \times b}$ and fixed readers $\boldsymbol{T}_\ell \in \mathbb{R}^{b \times 2d_h}$ such that every $\boldsymbol{Z}\boldsymbol{T}_\ell$ and their concatenation have rank b. Over the full family of binary $\boldsymbol{Z}$, lossless fixed-length private binary codes need at least LNb bits in total, whereas a shared code uses Nb bits.*

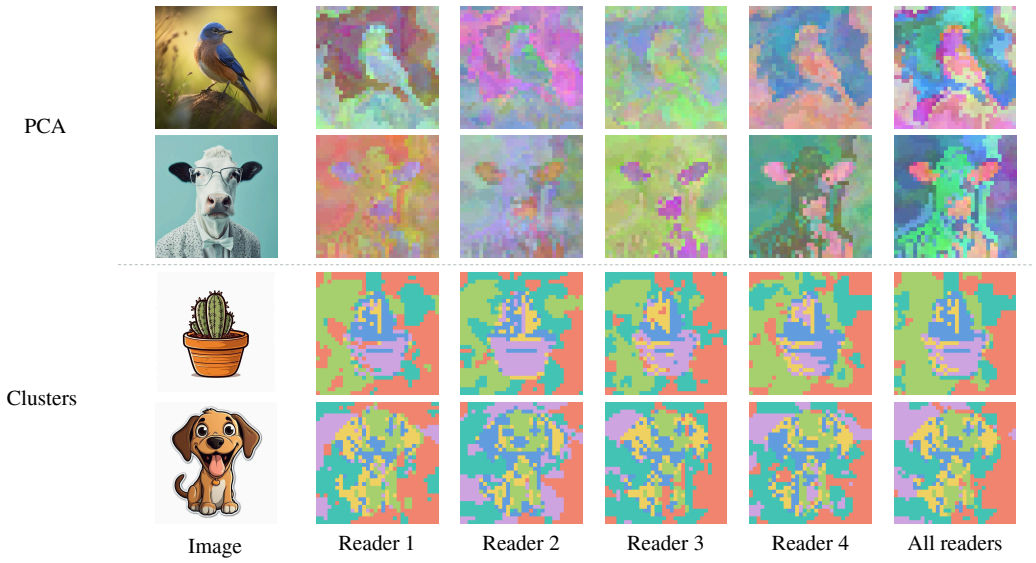


Figure 10: **One stored state supports distinct spatial readouts.** Four MJHQ-30K samples: bluebird (animals/0109396c...), cow (viewer row 0), cactus (row 24001), and dog (row 6). Each sample has one state $\boldsymbol{Z}$, and the four key readers produce $\boldsymbol{K}_\ell = \boldsymbol{Z}\boldsymbol{D}_K^\ell$. The top rows show PCA with a common basis across the individual readouts; the concatenated "All readers" view has its own PCA fit. The bottom rows show six-cluster assignments matched to the joint clustering by spatial overlap; cluster colors are unordered labels.

Proof of the rank statement. Choose each $\boldsymbol{T}_\ell$ with full row rank. A binary $\boldsymbol{Z}$ of full column rank exists: its first b rows can use the matrix with entries $+1$ for $i \le j$ and -1 otherwise, scaled by $1/\sqrt{b}$. Consecutive row differences of this matrix are twice the standard basis rows, so its determinant is 2^{b-1} and it has full rank; arbitrary binary rows complete $\boldsymbol{Z}$ when $N > b$. Since each $\boldsymbol{T}_\ell$ has a right inverse, $\mathrm{rank}(\boldsymbol{Z}\boldsymbol{T}_\ell) = \mathrm{rank}(\boldsymbol{Z}) = b$, and the same argument applies to the concatenation. □

Proof of the bit-count statement. Keep these readers fixed and range over all 2^{Nb} binary matrices $\boldsymbol{Z}$, including rank-deficient ones. The map $\boldsymbol{Z} \mapsto \boldsymbol{Z}\boldsymbol{T}_\ell$ is injective, since multiplying by a right inverse of $\boldsymbol{T}_\ell$ recovers $\boldsymbol{Z}$. Each layer must therefore distinguish 2^{Nb} targets. A private decoder sees only its own fixed-length code and uses a fixed decoding rule, and a code of q_ℓ bits has at most 2^{q_ℓ} values, so losslessness requires $q_\ell \ge Nb$ and the total is at least LNb bits. Storing the signs of $\boldsymbol{Z}$ once uses Nb bits and serves every reader. □

Low row-wise cosine similarity between layers does not prevent this reuse. Let $b \le \min(N, 2d_h)$ be even, let $\boldsymbol{Z}$ have full column rank, and let $\boldsymbol{O}$ be block-diagonal with $90°$ rotations, so $\boldsymbol{O}^\top = -\boldsymbol{O}$ and $\boldsymbol{O}$ is invertible. Define $\boldsymbol{C}_1 = [\boldsymbol{Z} \mid \boldsymbol{0}]$ and $\boldsymbol{C}_2 = [\boldsymbol{Z}\boldsymbol{O} \mid \boldsymbol{0}]$, padded with zeros to $2d_h$ columns. For every row $\boldsymbol{z}$ of $\boldsymbol{Z}$, $\langle \boldsymbol{z}, \boldsymbol{z}\boldsymbol{O}\rangle = \boldsymbol{z}\boldsymbol{O}^\top\boldsymbol{z}^\top = 0$, yet

$$\mathrm{col}(\boldsymbol{C}_1) = \mathrm{col}(\boldsymbol{C}_2) = \mathrm{col}(\boldsymbol{Z}), \qquad \mathrm{rank}([\boldsymbol{C}_1 \mid \boldsymbol{C}_2]) = b. \tag{15}$$

Both layers are served exactly by one b-coordinate state while their corresponding rows are orthogonal. The joint singular-value tail measures the error at a shared budget directly, and cross-layer cosine similarity does not.

Figure 10 shows how one stored state supports different layers. Four key readers decode the same state $\boldsymbol{Z}$ of each sample into maps with different principal components and different six-way clusterings, so the layers group the image differently from the same bits. This is the empirical counterpart of the construction above: the shared subspace is stored once, and each layer projects it onto its own directions, which is the behavior that tied readers remove in Table 3.

B.3 A LOCAL ATTENTION STABILITY BOUND

Lemma B.3 (Softmax perturbation). *For finite logit vectors $\boldsymbol{a}, \hat{\boldsymbol{a}}$ on the same allowed positions,*

$$\|\mathrm{softmax}(\boldsymbol{a}) - \mathrm{softmax}(\hat{\boldsymbol{a}})\|_1 \le \|\boldsymbol{a} - \hat{\boldsymbol{a}}\|_\infty. \tag{16}$$

Proof. For $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{a})$, the Jacobian is $\mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$. For a direction $\boldsymbol{u}$ with $\|\boldsymbol{u}\|_\infty \le 1$, the image has ℓ_1 norm $\sum_i p_i|u_i - \mathbb{E}_{\boldsymbol{p}} u| \le \sqrt{\mathrm{Var}_{\boldsymbol{p}}(u)} \le 1$ by Cauchy–Schwarz and $\mathbb{E}_{\boldsymbol{p}} u^2 \le 1$. Integrating the Jacobian along the segment from $\boldsymbol{a}$ to $\hat{\boldsymbol{a}}$ proves the bound. □

Proposition B.3 (Fixed-query attention error). *Fix the query, the allowed positions, and all non-historical entries, without attention dropout. Let δ_a bound the absolute change of each historical logit, δ_V the Euclidean error of each historical value, and $V_{\max}$ the norm of every original value. Then*

$$\|\boldsymbol{o} - \hat{\boldsymbol{o}}\|_2 \le \delta_V + V_{\max}\,\delta_a \le \delta_V + V_{\max}\,|\tau|\,\|\boldsymbol{q}\|_2\,\delta_K, \tag{17}$$

where δ_K bounds the key error after native key normalization and rotary position.

Proof. Add and subtract $\sum_i \hat{\alpha}_i \boldsymbol{v}_i$. The value term is at most $\sum_{i \in \mathcal{I}_{\mathrm{hist}}} \hat{\alpha}_i \delta_V \le \delta_V$, and the weight term is at most $V_{\max}\|\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}\|_1$, bounded by Lemma B.3. Each changed logit equals $\tau\langle \boldsymbol{q}, \boldsymbol{k}_i - \hat{\boldsymbol{k}}_i\rangle$, and Cauchy–Schwarz gives the second inequality. □

The bound measures key errors after normalization and rotation. For RMS normalization $\mathrm{N}(\boldsymbol{h}) = \boldsymbol{\gamma} \odot \boldsymbol{h}/\sqrt{\|\boldsymbol{h}\|_2^2/d_h + \epsilon}$ with $\epsilon > 0$, the Jacobian norm is at most $\|\boldsymbol{\gamma}\|_\infty/\sqrt{\epsilon}$: before multiplication by $\mathrm{diag}(\boldsymbol{\gamma})$, the radial eigenvalue is $\epsilon/(\|\boldsymbol{h}\|_2^2/d_h + \epsilon)^{3/2}$ and the tangential eigenvalues equal the reciprocal denominator. An orthogonal rotation preserves norms. Such bounds relate pre-normalization errors to native key errors and become loose when ϵ is small. The proposition covers one attention operation; during generation, errors also change later queries and sampled tokens.

B.4 EXACT SPAN OF THE BLOCK DEPENDENCY GRAPH

Let $F_{s,\ell}$ be a unit-time stage for block s and layer ℓ. Within a block, edges connect consecutive layers. For every $u < s$, an edge connects F_{u,p_ℓ} to $F_{s,\ell}$ when layer ℓ reads a state produced at depth $p_\ell \in \{1, \dots, L\}$; a producer with $p_\ell = 0$ is ready at time zero, and when several intermediates are read, p_ℓ is the latest one. Blocks are causally ordered, so a cross-block edge from a deep layer to a shallow one creates no cycle. All positions of a block-layer operation form one stage, and writer, communication, and sampling costs are excluded.

Upper bound in Theorem 2. Assign completion time $t_{s,\ell} = \ell + (s-1)\Delta$. Consecutive layers within a block finish one unit apart. For a cross-block producer of positive depth,

$$t_{s,\ell} - t_{u,p_\ell} = \ell - p_\ell + (s-u)\Delta \ge \ell - p_\ell + \Delta \ge 1, \tag{18}$$

because $\Delta \ge p_\ell - \ell + 1$; producers ready at the input satisfy the schedule as well. Every edge is respected, and the last completion time is $L + (K-1)\Delta$. □

Lower bound in Theorem 2. If $\Delta = 0$, the L stages of one block already form a path of length L. Otherwise choose $\ell^\star$ with $\Delta = p_{\ell^\star} - \ell^\star + 1 > 0$ and write $p^\star = p_{\ell^\star}$. Follow block one to depth $p^\star$, cross to depth $\ell^\star$ of block two, continue to $p^\star$, and repeat through block K; after the last crossing, finish at depth L. Since $p^\star \ge \ell^\star$, every within-block segment exists, and the path contains

$$p^\star + (K-1)(p^\star - \ell^\star + 1) + (L - p^\star) = L + (K-1)\Delta \tag{19}$$

stages. Every schedule takes at least this long, which together with the upper bound proves the span. □

For $K \ge 2$, a span of L stages is possible exactly when $p_\ell \le \ell - 1$ for every consumer, which holds for a native cache formed from the input of each layer. CacheTok has $p_\ell = L$, and its first layer waits for the full-depth write of the preceding block, which gives KL. A writer of uniform cost t_{w} per consumed block boundary gives $KL + (K-1)t_{\mathrm{w}}$ in the same model, with the final unused write

excluded. Unequal layer runtimes and finite hardware call for weighted analysis or measurement, so KL counts dependent stages, and GPU time is measured in Appendix E.

The reverse graph has the same longest path, because reversing a directed path preserves its number of vertices, so the surrogate backward pass keeps the same dependency constraint. Detaching an earlier state removes edges and changes the training computation, while recomputing a state keeps the edges and changes only work and saved activations. Checkpointing therefore reduces training memory and keeps the KL span, and an iterative or truncated approximation is a different training procedure from the exact graph. Appendix B.5 shows how writing one state per group of layers trades stored bits against this span.

B.5 A STORAGE–SPAN TRADEOFF FOR GROUPED PRODUCERS

Proposition B.4 (Post-group production)**.** *Partition the L layers into J nonempty groups $\mathcal{G}_1, \ldots, \mathcal{G}_J$, and let every layer in $\mathcal{G}_j$ read an earlier block's group state produced after depth $\max \mathcal{G}_j$. Then $\Delta \geq \lceil L/J \rceil$, with equality for balanced contiguous groups. If every group stores b bits for each of N positions and H heads, the payload $M = JNHb$ bits satisfies*

$$M\,(T_{\mathrm{span}} - L) \geq NHbL(K-1). \tag{20}$$

Proof. Within $\mathcal{G}_j$, the earliest consumer has lag $\max \mathcal{G}_j - \min \mathcal{G}_j + 1 \geq |\mathcal{G}_j|$, and some group has at least $\lceil L/J \rceil$ layers. Balanced contiguous groups attain the bound because each interval length equals the group size. Multiplying $T_{\mathrm{span}} - L = (K-1)\Delta$ by M and using $J\lceil L/J \rceil \geq L$ gives the product bound. □

The product bound describes post-group producers with equal-width group states. Native pre-layer caches form a different family: $J = L$ still writes after each layer and has $\Delta = 1$, whereas a pre-layer producer has $\Delta = 0$. Within the post-group family, fewer groups store less and lengthen the span, and CacheTok is the extreme $J = 1$, which stores the least and has the span KL of Theorem 2.

C EVALUATION PROTOCOL AND COMPARISON SETTINGS

C.1 MODEL GEOMETRY

VAR-d16 predicts class-conditional scales at 256^2. PAR-XL4x predicts position groups at 384^2 and applies its native 256^2 evaluation resizing. Infinity-2B performs text-conditional bitwise next-scale generation at 1024^2 and 512^2. The block schedule of each backbone sets both the write boundaries and the history lengths, and Table 5 lists the geometry used for storage accounting. The VAR scale sides are $(1, 2, 3, 4, 5, 6, 8, 10, 13, 16)$. Infinity uses $(1, 2, 4, 6, 8, 12, 16, 20, 24, 32, 40, 48, 64)$ at 1024^2 and the prefix ending at 32 at 512^2. PAR has four non-visual conditioning positions, which are accounted for separately from its visual history.

Table 5: Native geometry for exact accounting. N_{all} includes the final block; N_{hist} excludes its unused write. b_0 is the default state width. All rows use multi-head attention. Infinity resolutions have separately continued checkpoints.

Configuration	L	d	H	d_h	K	N_{all}	N_{hist}	b_0
VAR-d16, 256^2	16	1024	16	64	10	680	424	64
PAR-XL4x, 384^2	36	1280	20	64	147	576	572	64
Infinity-2B, 1024^2	32	2048	16	128	13	10521	6425	128
Infinity-2B, 512^2	32	2048	16	128	10	2521	1497	128

C.2 CONTINUATION AND SAMPLING

Each model starts from its public checkpoint. The tokenizer and the conditioning encoder stay fixed, and the backbone, writer, and readers train jointly; Full-KV receives the same data and update budget. Each run uses 10,000 updates at global batch 32 with seeds 17, 29, and 43. AdamW uses moments $(0.9, 0.95)$, weight decay 0.05, gradient clipping at 1, 500 warmup updates, and cosine decay to one tenth of the peak learning rate. Training uses four A100 80GB SXM4 GPUs in BF16,

with data parallelism for VAR and PAR and parameter sharding for Infinity. Infinity's bit self-correction flip is disabled.

The learning-rate candidates are $\{5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}\}$ for VAR and PAR and $\{10^{-5}, 2 \times 10^{-5}, 4 \times 10^{-5}\}$ for Infinity, selected by held-out native prediction loss. Each candidate is compared after 1000 updates at seed 17, and the selected candidate continues from that point. Model selection evaluates update zero and every 1000 updates, and earlier checkpoints break ties.

ImageNet reserves ten validation and two calibration images per class from its training split. Infinity uses JourneyDB (Sun et al., 2023), with 4096 validation and 1024 calibration prompts selected by deterministic hashing and separated from training and evaluation prompts. The 512^2 and 1024^2 continuations start from the same public initialization and train separately; within a resolution, guided and single-branch evaluations use the same model.

VAR samples with CFG 1.5, top-$k = 900$, and top-$p = 0.96$. PAR uses CFG 1.5, temperature 1, top-$k = 0$, and top-$p = 1$, which disables truncation. Infinity uses CFG 3, temperature 0.5, and the original evaluation prompts. The single-branch CFG values are 0, 1, and 1. Native MVAR uses the released unstarred d16 model, CFG 2.7, top-$k = 1200$, top-$p = 0.99$, and its native scale and spatial conditioning; its three repeats vary sampling seeds with fixed weights.

C.3 External-method adaptations

The methods in Table 6 run on the same continued backbone within each family. HACK protects the required initial and structural positions. ScaleKV maps its last-five-scale and layer allocations to each native schedule. AMS-KV uses the stated threshold and capacity endpoints. SparVAR uses sparse reads over a resident cache. Persistent storage is counted from these allocations and the retained tensors.

Table 6: Matched plug-in settings on VAR-d16 and Infinity-2B. ToProVAR-F is the equation-based variant in Appendix C.3.

Method	Configuration
HACK (Qin et al., 2026)	Delete 50% on VAR / 70% on Infinity. Protect first five and structural positions; final Infinity budget 4101.
ScaleKV (Li et al., 2025b)	Retain 10%; mean budget 43 / 650 (VAR / Infinity). Last-five-scale, depth-weighted allocation.
FastVAR (Guo et al., 2025)	Last-four pruning 40/50/100/100%. Pivotal tokens and cached restoration; skip final two scales.
AMS-KV (Xu et al., 2026)	Condense first two scales; $\theta = -0.007$. Capacities 174/430 (VAR), 2309/6405 (Infinity).
SparVAR (Li et al., 2026b)	No scale skipping. Full resident KV, sparse reads, selection and compensation metadata.
ToProVAR-F (Chen et al., 2026)	$\tau = 0.4$, $\beta = 1$, endpoints 0/1. Sum-normalized entropy; restore previous-scale outputs.

ToProVAR-F. This equation-based variant instantiates Equations 3–6 of Chen et al. (2026) with $\tau = 0.4$, $\beta = 1$, and pruning-probability endpoints 0 and 1. Token entropy averages native attention entropy across heads, and scale selection also averages across layers. Pruning starts at the first scale where at least 40% of tokens have entropy below the scale mean, or after the last scale if none qualifies. For each layer, the entropy matrix has heads as rows and current tokens as columns; with its two leading singular values σ_1, σ_2, the method sets $\gamma = \exp[-(\sigma_1/\sigma_2 - 1)]$, with $\gamma = 0$ when $\sigma_2 = 0$, and retains one-token stages. The drop probability of token i is $(s/K)\,\gamma\, e_i / \sum_j e_j$, where e_i is its entropy; a zero entropy sum retains all tokens. Bernoulli retention keeps at least the most salient token, omitted positions receive bilinearly resized hidden outputs of the previous scale, and the drop probabilities sum to at most one per layer and scale. The row reports this variant with its own timing and storage.

C.4 Benchmark aggregation and independent units

ImageNet evaluates 50,000 generated images per seed with balanced classes and the ADM procedure for FID, sFID, Inception Score, precision, and recall. GenEval uses all 553 prompts with four images each (2212 images). Its six tag sizes are 80, 99, 80, 94, 100, and 100 for single object, two objects, counting, colors, position, and color attribution; each tag uses the official success rule, and the overall score averages the six tags equally on a 0–100 scale.

DPG-Bench uses 1065 prompts with four images each, giving 14,392 questions per image set and 57,568 image–question answers in total. The overall score applies the evaluator's ordered single-pass dependency correction to each image, then averages images within prompts and prompts within the benchmark. Category scores pool raw yes/no answers per category, and all images and questions of one prompt stay together during resampling. For one image per prompt, the category question

counts are: whole entity 4741, state 947, part 512, color 1993, texture 1653, other attribute 903, size 242, shape 229, spatial relation 2434, non-spatial relation 159, count 200, text 50, and global 329. Each count is multiplied by four for the full evaluation, so the category average in Figure 13 differs from the dependency-corrected overall DPG score by construction.

T2I-CompBench v1 evaluates six sets of 300 prompts with ten images per prompt. MJHQ-30K evaluates 3000 images in each of ten categories at 1024^2, and the overall FID pools feature moments over all 30,000 images. HPSv2.1 scores 3200 prompts, divided equally among anime, concept art, paintings, and photos.

Tables display means. For uncertainty in the text and figures, seed results ϕ_1, ϕ_2, ϕ_3 use an SD denominator of 2; paired intervals form $\delta_k = \phi_k - \phi'_k$ and report $\bar{\delta} \pm 4.303\, s_\delta/\sqrt{3}$. Conditional bootstrap intervals use 10,000 paired resamples at each fixed training seed, stratified by GenEval tag or HPS style and clustered by DPG prompt. Training seeds are the independent units for learned models, and native fixed-checkpoint comparisons use sampling repeats.

C.5 Intervention definitions

The writer-input controls remove either the token embedding $\boldsymbol{e}(r_i)$ ("No symbol input" or "No token input") or the final hidden state $\boldsymbol{x}_i$ ("No hidden input" or "No top feature") before RMSNorm in Eq. (2). Tied readers share one key reader and one value reader per head across layers. The state-gradient control applies a stop-gradient to $\boldsymbol{Z}_s$ at every later read, so the writer receives no task gradient while the readers and backbone train through their remaining paths. These controls are independent continuations under the same training protocol.

The trained continuous control replaces $\mathrm{sgn}(\boldsymbol{y}_{i,g})/\sqrt{b}$ by $\boldsymbol{y}_{i,g}/\sqrt{b}$ throughout training and evaluation, at the default width and in BF16. Inference-only sign removal applies the same change to a binary-trained model with fixed weights. The low-width binary control uses $b = 64$ for Infinity. The older-address intervention holds weights and allocated states fixed and masks reads from all blocks $u < s - 1$ at every layer, and each masked condition is paired with its unmasked evaluation under the same sampling seeds.

Figure 11 shows complete outputs of the writer-input controls; the older-position mask is shown in Figure 7. The writer-input controls keep the requested objects and change local texture and lighting, the detail that the category scores in Figure 13 resolve.

D Additional Task and Mechanism Analyses

D.1 Object and compositional judgments

Table 7: Complete subtask profiles for Infinity-2B, 1024^2, CFG 3; three-seed means. C. attr. denotes color attribution; GenEval has six object-level tasks; CompBench v1 reports its six evaluators separately. Bold compares the two interfaces within each task.

Method	GenEval (0–100) ↑						T2I-CompBench v1 (native scale) ↑					
	Single	Two	Count	Colors	Position	C. attr.	Color	Shape	Texture	Spatial	Nonsp.	Complex
Full-KV	**98.23**	**79.55**	**67.60**	**87.85**	**24.67**	57.08	0.7940	**0.5150**	0.6810	0.2870	**0.3082**	**0.3790**
CacheTok	97.81	**79.55**	**67.60**	87.23	23.92	**57.50**	**0.7960**	0.5120	**0.6820**	**0.2890**	0.3080	0.3780

Table 7 breaks the Infinity-2B results into the six GenEval tasks and the six evaluators of T2I-CompBench v1 (Huang et al., 2023). Single-object scores stay near 98 and position scores near 24 for both interfaces, so CacheTok keeps the backbone's profile of strong single-object accuracy and weak spatial alignment. The paired position change is -0.75 points with a 95% interval of $[-4.48, 2.98]$. The CompBench changes vary in sign across the six evaluators and stay within their paired intervals.

Figure 12 resolves each task average into prompt-level losses. For every prompt, the loss is the Full-KV score minus the method score, and each curve sorts its prompts from the largest loss to the smallest and plots the running mean. The first few percent of prompts carry losses several times larger than the task mean for every method, and the ranking of methods changes with the included fraction. The right endpoint of each curve equals the task-level difference.



Figure 11: **Writer-input controls under two detailed text conditions.** Infinity-2B outputs for Full-KV, default CacheTok, no token input, and no top hidden feature, with four outputs per condition arranged as a 2×2 group. The omitted inputs are $\boldsymbol{e}(r_i)$ and $\boldsymbol{x}_i$, respectively.

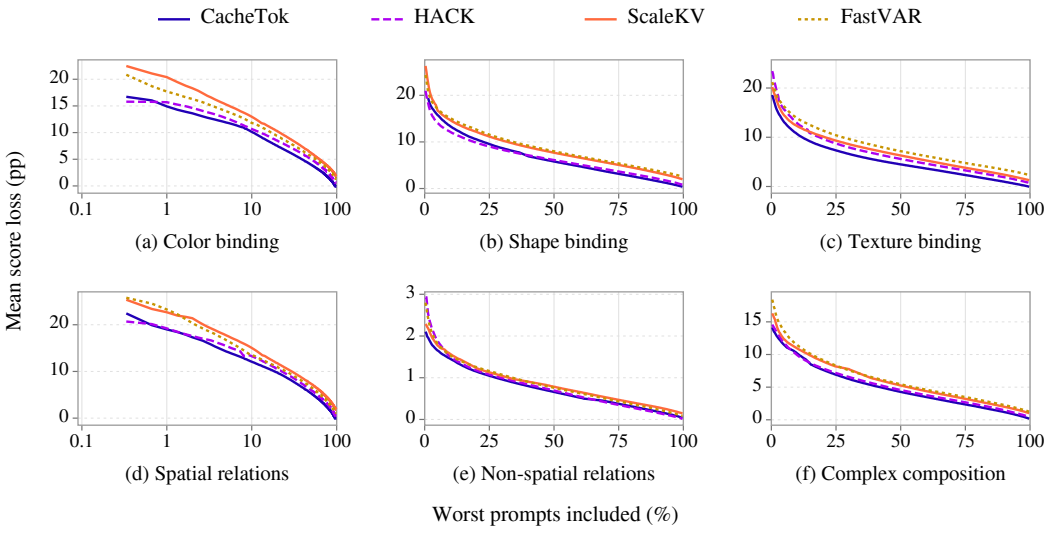


Figure 12: **Small average changes coexist with larger losses on a subset of prompts.** Infinity-2B, 1024^2, CFG 3, on all six T2I-CompBench tasks (300 prompts and ten images each per task and seed). For each prompt, loss is 100 times the Full-KV score minus the method score, averaged over three paired seeds. Each curve sorts its own losses from largest to smallest and plots their cumulative mean over all 300 inclusion points; the right endpoint is the full-task mean difference. A shared horizontal coordinate denotes the same included fraction, and the selected prompts can differ by method.

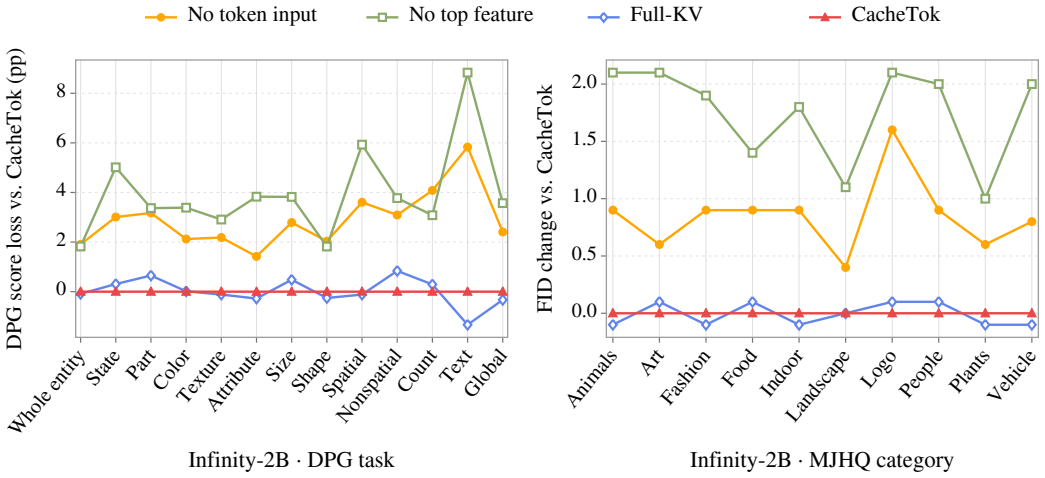
D.2 WRITER INPUTS AND CATEGORY-LEVEL EFFECTS

Table 8 adds distributional fidelity and preference to the Infinity-2B controls of Table 3, from the same continued models. Removing either writer input raises MJHQ FID and lowers HPSv2.1, and the ordering of the controls matches the alignment metrics: removing the hidden state costs more than removing the token embedding, tied readers and the stopped state gradient cost more still, and inference-only sign removal costs most.

Table 8: Distributional and preference-score checks for the Infinity-2B ablations in Table 3; three-seed means at 1024^2.

Variant	MJHQ FID ↓	HPSv2.1 ↑
Default	**9.85**	**30.48**
No symbol input	10.67	30.20
No hidden input	11.56	29.89
Tied readers	12.24	29.53
Stop state gradient	12.61	29.54
Train continuous	**9.83**	**30.48**
Sign off at inference	13.48	29.05

Figure 13 resolves the two writer-input controls by category. Removing the hidden state produces the larger change in ten of the thirteen DPG categories and in every MJHQ category; in whole entity, shape, and count, removing the token input costs as much or more, and the text category of DPG shows the largest change under both removals. Among MJHQ categories, logo images, which also contain lettering, show the largest change from removing the token embedding. Rendered text depends on the exact tokens and on their context, so it needs both inputs of the writer.



Infinity-2B · DPG task Infinity-2B · MJHQ category

Figure 13: **Sampled tokens and full-depth context both shape the stored state.** Infinity-2B, 1024^2, CFG 3, $b = 128$, three seeds. Left: CacheTok minus control DPG score in percentage points for all thirteen categories, from 1065×4 images and the category question counts of Appendix C.4. Right: control minus CacheTok FID for each MJHQ category (3000 images per category per seed). Positive values indicate deterioration. "No token input" removes $\boldsymbol{e}(r_i)$; "No top feature" removes $\boldsymbol{x}_i$.

Settings of Figure 4 and Figure 5. Figure 4 uses the MJHQ-30K sample food/$fcd86a64\ldots$ with the query at normalized image coordinates $(0.672, 0.297)$. The writer and the hidden input are fixed, and two draws from the same predicted distribution are written with and without the token input. At location i, the same-draw map is $\langle \boldsymbol{z}_i^{(0)}, \boldsymbol{z}_q^{(0)} \rangle$ and the new-draw map is $\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_q^{(0)} \rangle$; states have unit norm, so these are cosine similarities on a common $[-1, 1]$ scale, and RGB blending only locates the object. A hidden-only write gives identical states for both draws by construction.

Figure 5 uses plants/$2fde0af2\ldots$ and differentiates three later-block losses with respect to the same earlier state. The maps show $\log_2(\|\partial \mathcal{L}_t / \partial \boldsymbol{z}_i\|_2 / \bar{\eta})$ with one mean $\bar{\eta}$ and one color range pooled over the three maps, so a warm location has a larger gradient norm than the pooled mean; the straight-through factor $1/\sqrt{b}$ cancels in this ratio. The nearest later block concentrates its gradient on a few positions in the flower cluster and along the leaf edges, at up to about twice the pooled mean, while the third later block spreads a lower gradient more evenly over the image.

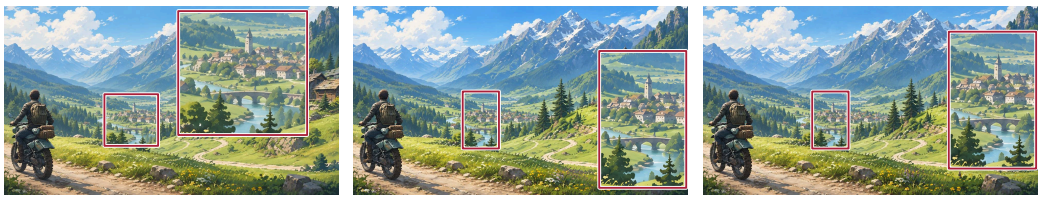


Figure 14: **Older-position access at corresponding scene locations.** Infinity-2B (1024^2, CFG 3, $b = 128$): earlier scene, intact completion, and completion with blocks $u < s - 1$ masked. The mask keeps the allocated states and model weights. Insets share a source-coordinate center and $2.4\times$ magnification; compare the common boxed region.

D.3 Older positions in generated images

Figure 15 resolves the GenEval masking loss of Table 4 into its six tasks, and the mean of the six task losses equals that GenEval change. Single-object success hardly changes and single-object color changes by about 2 points or less, while the four judgments that involve several objects lose between 2.7 and 4.7 points in both models. Full-KV loses more on counting (4.69 against 3.33 points) and CacheTok more on color attribution (4.00 against 3.08), so the two models rely on old positions for different judgments.

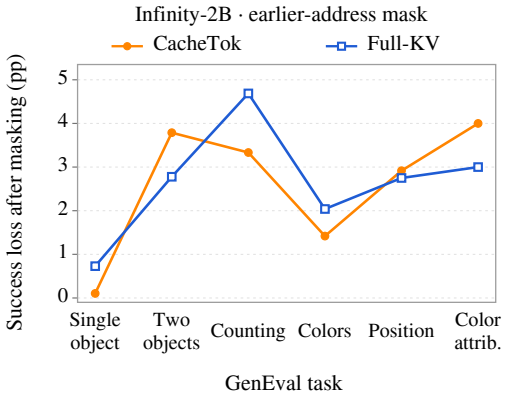


Figure 15: **Use of old positions varies by task.** Intact minus masked GenEval success (pp), Infinity-2B, three paired seeds; blocks $u < s - 1$ masked with weights and states fixed.

Figure 14 shows the masking intervention of Section 5.4 on one Infinity-2B scene, from MJHQ-30K landscape/c3bff0b9bf73948d98346f9b770ccb6351130fc3 with the prompt "studio ghiblie style, man riding old motorbike through the mountains, village in a valley in the distance, stunning scenery, happy vibes, positive vibes, serene, clear sunny day". With the previous block still readable, the global layout survives the mask, while the village in the valley changes its local structure. The change concentrates on small objects whose placement depends on the coarse layout, the same judgments that lose most under the mask in Figure 15. The insets inspect the same source-coordinate region in all three panels. Because the weights and the stored states are identical in the two completions, their difference isolates the contribution of direct reads of the old blocks. Figure 7 in Section 5.4 shows complete outputs for nine prompts.

D.4 Continuous training and inference-only sign removal

Training with continuous states and removing the sign at inference are different interventions. Figure 6 in Section 5.3 compares them on eight DPG categories, each normalized by the default binary model; radial values are 100 times the three-seed category mean divided by the default mean, with 60% at the center and 105% at the outer ring, from 1065×4 DPG images per seed. The continuous model stays near the binary default in every category, so one bit per coordinate matches a BF16 coordinate at this width. Inference-only sign removal lowers every category, because readers trained on sign patterns receive pre-sign activations of a different magnitude. The 64-bit binary control lies between the two in most categories and isolates the effect of width from the effect of precision. The categories are whole entity, color, texture, spatial relation, non-spatial relation, count, text, and entity state, with the raw-answer counts of Appendix C.4.

Continuous and binary states use the same 128 coordinates, and BF16 storage uses sixteen bits per coordinate. Figure 16 shows the decoded keys behind the sign-removal loss. A binary state has

unit norm, whereas its decoded key has a learned norm pattern; removing the sign changes both the principal directions and the norm map that the fixed reader produces.

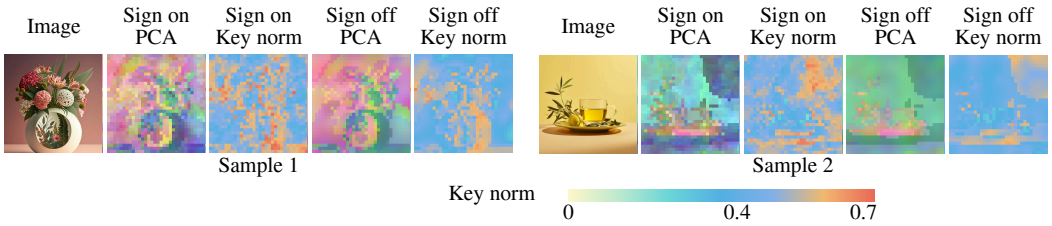


Figure 16: **Inference-only sign removal changes decoded-key structure and magnitude.** Two MJHQ-30K samples: a tropical-flower vase (plants/845999$d0\ldots$) and an olive-oil image (food, viewer row 9009). Each group shows RGB, binary-key PCA and norm, then sign-removed-key PCA and norm. Writer pre-activations $\boldsymbol{Y}$ and the reader $\boldsymbol{D}_K$ are fixed: $\boldsymbol{K}_{\mathrm{b}} = \mathrm{sgn}(\boldsymbol{Y})\boldsymbol{D}_K/\sqrt{b}$ and $\boldsymbol{K}_{\mathrm{c}} = \boldsymbol{Y}\boldsymbol{D}_K/\sqrt{b}$, shown before native key normalization and rotary position. PCA uses one basis and common component ranges across both samples and conditions, and all norm maps share a range starting at zero.

D.5 Guidance, resolution, and native checkpoints

Table 9: Guidance and resolution checks, three-seed means. GenEval uses 0–100. Guided and single-branch sampling share a checkpoint within each resolution; the two Infinity resolutions are continued separately.

Method	VAR-d16 256^2, CFG 0	PAR-XL4x 384^2, CFG 1	Infinity-2B, 1024^2 Single branch, CFG 1		Infinity-2B, 512^2 Guided, CFG 3		Infinity-2B, 512^2 Single branch, CFG 1	
	FID ↓	FID ↓	GenEval ↑	DPG ↑	GenEval ↑	DPG ↑	GenEval ↑	DPG ↑
Full-KV	8.26	**5.43**	63.81	**79.48**	66.40	**81.94**	**60.90**	76.39
CacheTok	**8.23**	5.47	**64.13**	79.33	**66.43**	81.47	60.84	**76.73**

Table 9 varies guidance and resolution. At 512^2 with guidance, Infinity GenEval changes by $+0.03$ points and DPG by -0.47 points, with a DPG 95% interval of $[-1.66, 0.72]$. Without guidance, the class-conditional models change FID by -0.03 (VAR) and $+0.04$ (PAR), and the signs of the differences vary with metric and setting. Each resolution uses its own continued model, and the guidance settings reuse it.

Table 10 separates matched interfaces from native checkpoints. Its first two blocks repeat the ADM endpoints of Table 1, and the third compares the released VAR-d16 with MVAR-d16 (Zhang et al., 2026), each under its own weights, conditioning, and sampler. MVAR reaches a lower FID (3.04) by changing the conditioning of the architecture, which makes it context for the matched study; the three repeats of these rows vary generation seeds with fixed weights.

Table 10: ImageNet ADM means. Matched rows reuse Table 1; native VAR/MVAR use three sampling repeats of fixed checkpoints, with native MVAR settings (Zhang et al., 2026). P/R retain 0–1.

Method	FID ↓	sFID ↓	IS ↑	P ↑	R ↑
VAR-d16 / 256^2, matched					
Full-KV	**3.55**	6.43	274.40	**0.8403**	**0.5125**
CacheTok	3.58	**6.40**	**275.07**	0.8387	0.5106
PAR-XL4x / 384^2, matched					
Full-KV	**2.63**	**5.96**	**259.73**	0.8206	0.5585
CacheTok	2.65	6.00	259.20	**0.8217**	**0.5587**
Native checkpoints / 256^2					
VAR-d16	3.55	6.46	274.10	0.8380	**0.5165**
MVAR-d16 (Zhang et al., 2026)	**3.04**	**6.29**	**284.70**	**0.8508**	0.5077

Figure 17 compares Full-KV and CacheTok samples of the two class-conditional backbones for eight ImageNet classes under the same class conditions, and Figure 18 shows three CacheTok samples for each of eight further classes.

D.6 Width budget and shared linear directions

The width sweep in Figure 19 uses 32, 64, and 128 bits for VAR and PAR and 64, 128, and 256 bits for Infinity, with the same continuation protocol at each width. Infinity GenEval is 64.74, 68.94, and 69.19: the $64 \to 128$ gain is 4.20 points (95% CI $[3.01, 5.39]$) and the $128 \to 256$ gain is 0.26 points ($[-0.53, 1.05]$). Each doubling doubles both the stored width and the writer and reader

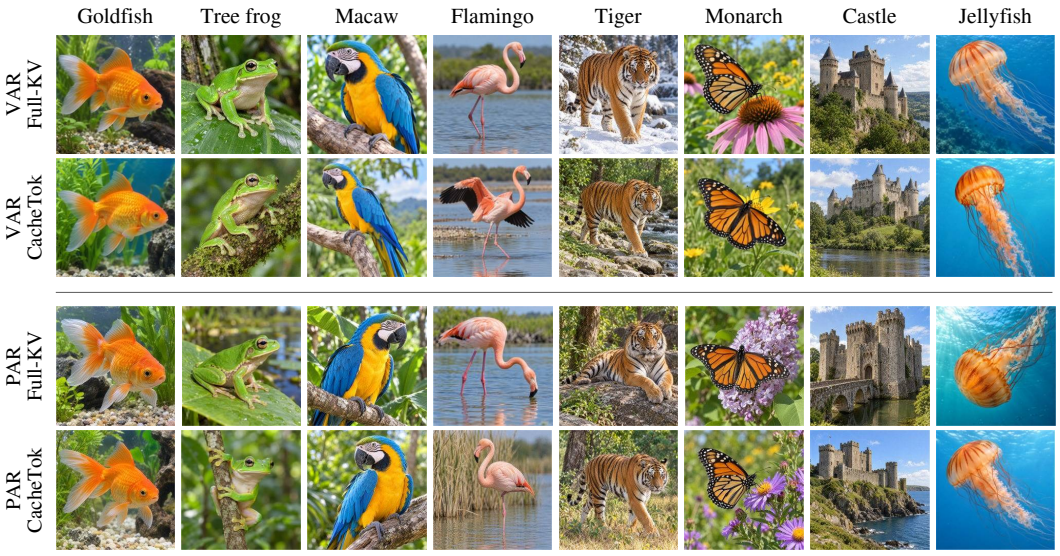


Figure 17: **Class-conditional outputs with the full cache and with CacheTok.** Eight ImageNet conditions for VAR-d16 at 256^2 (upper pair of rows) and PAR-XL4x at 384^2 (lower pair). Within each backbone, the rows show Full-KV and CacheTok at the default 64-bit width under the same class conditions; distributional quality is measured in Table 1.

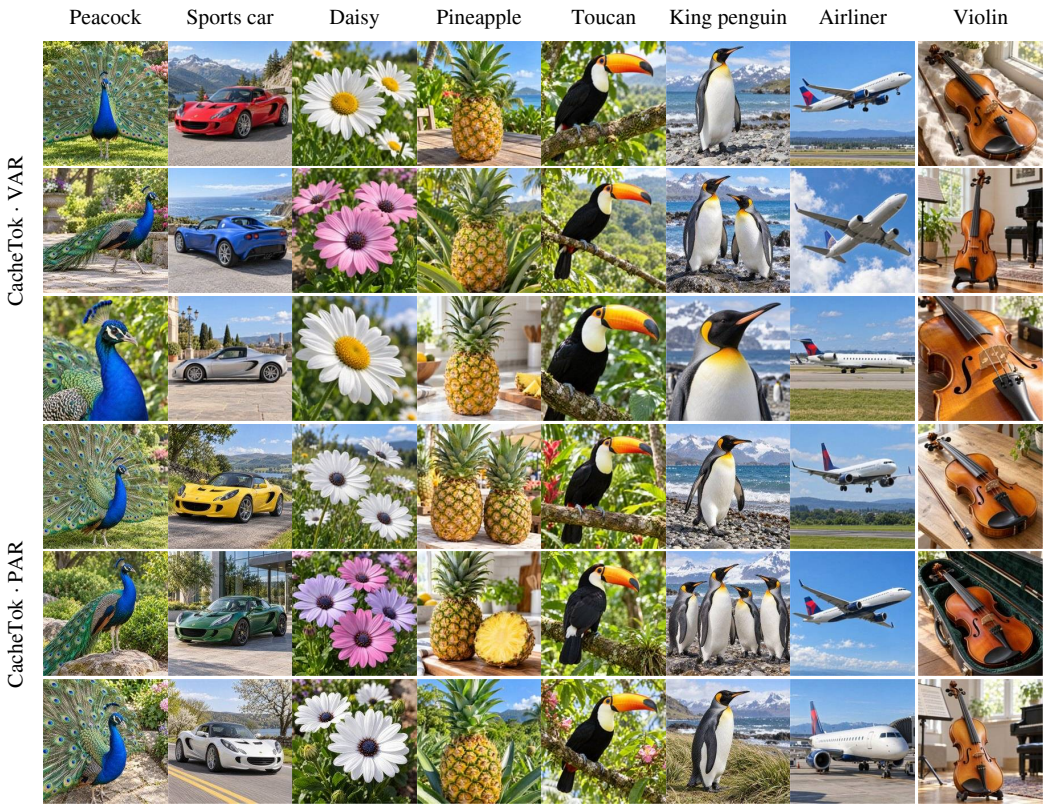


Figure 18: **CacheTok outputs for eight further ImageNet classes.** Three VAR-d16 samples in the upper block and three PAR-XL4x samples in the lower block, all with the default 64-bit state width.

parameters. BF16 states trained at the default width (triangles in Figure 19) match the binary states with 16× the payload, so quality is set by the number of coordinates.

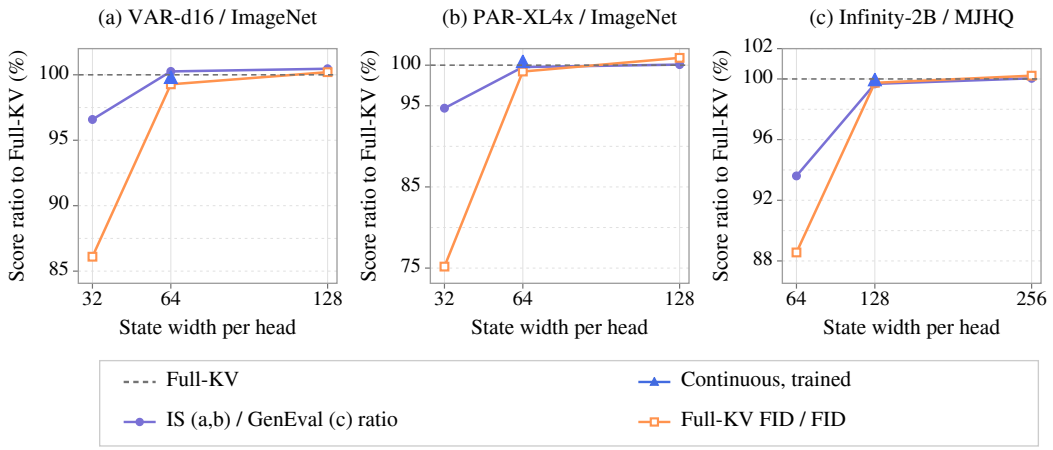


Figure 19: **Quality saturates at a state width equal to the head dimension.** Purple: $100\,\mathrm{IS}/\mathrm{IS}_{\text{Full-KV}}$ in (a,b) and the GenEval ratio in (c); orange: $100\,\mathrm{FID}_{\text{Full-KV}}/\mathrm{FID}$; higher is better, from three-seed means on ImageNet (50,000 images per seed), GenEval (553×4 images), and MJHQ-30K. Triangles: BF16 states trained at the default width, with $16\times$ the binary payload.

Reference matrices and analytic fits. For each head, $\boldsymbol{C}_\ell = [\tilde{\boldsymbol{K}}_\ell \mid \boldsymbol{V}_\ell]$ uses historical Full-KV targets before key normalization and rotary position. The conditional-branch summaries use the history immediately before the final block, with $N = 424$, 572, and 6425 positions for VAR, PAR, and Infinity, over 50,000 ImageNet images per class-conditional model and 4260 DPG images for Infinity per seed. Each head is analyzed separately; head and image summaries are averaged within a seed and then across the three seeds, and all layers share the same histories. Shared fitting keeps the leading singular directions of $\boldsymbol{C}$, and private fitting allocates a common total budget across layers by keeping the largest squared singular values of their combined lists, following Eq. (12). Captured energy is $100(1 - E/\|\boldsymbol{C}\|_F^2)$ and reconstruction MSE is $E/(2NLd_h)$. For the layer-1 source, $\boldsymbol{P}_1 = \boldsymbol{C}_1\boldsymbol{C}_1^\dagger$ captures $100\|\boldsymbol{P}_1\boldsymbol{C}\|_F^2/\|\boldsymbol{C}\|_F^2$ at budget $\mathrm{rank}(\boldsymbol{C}_1)$.

Figure 20 gives three further views. Panel (a) shows that one 128-direction subspace covers about 70% of the energy of every one of the 32 Infinity layers, while four private directions per layer cover about 5%; the shared gain is uniform across depth. Panel (b) shows that the shared MSE falls from about 0.26 to 0.09 as the budget grows from 64 to 256, while the private MSE stays near 0.5. Panel (c) pairs the shared captured energy of VAR and PAR at each width with the FID of the trained binary model at the same width: higher energy goes with lower FID.

D.7 State diagnostics

This section examines two properties of the stored bits: which statistics of the bits relate to task quality (Figure 21), and how the states of different samples of one prompt compare (Figure 23).

Figure 21 pairs task losses with statistics of the stored state. All widths and controls keep bit entropy near 0.9 bits and full state rank; the tied-reader and stopped-gradient controls lose about 4 to 10 GenEval points on counting, colors, and position with the same statistics. Balanced, linearly independent bits are therefore a property of every writer, and the quality of a state depends on which directions the trained writer and readers select. The key-reader rank ratio of 0.5 at $b = 256$ reflects the maximum rank 128 of a 256×128 reader.

The task losses in the left panel separate the controls. Narrowing the state to 64 bits costs counting 7.2 and position 4.6 points but color only 1.8: a smaller state keeps the appearance of each object and loses the number and placement of objects, which depend on relations among many positions. Stopping the state gradient costs position most (9.9 points). An untrained writer keeps b directions chosen at initialization, and position judgments depend on the layout stored in old blocks (Section 5.4). Tied readers cost all three tasks 5.5 to 8.2 points, because every layer then retrieves the same keys. Removing the top feature costs more than removing the token input on all three tasks,

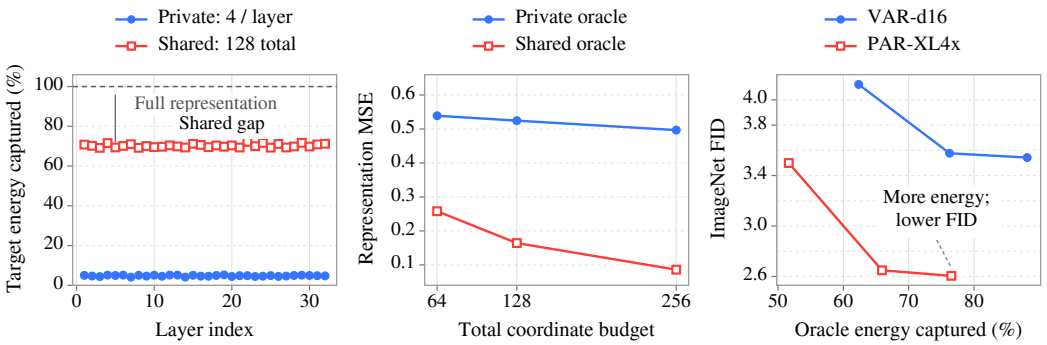


Figure 20: **Shared directions cover all layers under a common total budget.** (a) Infinity-2B conditional DPG histories, three seeds, all 32 layers: a shared 128-direction subspace against four private directions per layer (128 in total). Coverage is projected layer energy divided by that layer's full energy. (b) Analytic minimum reconstruction MSE of shared and optimally allocated private fits at total budgets 64, 128, and 256. (c) VAR and PAR shared captured energy paired with ImageNet FID at the same three widths; the fit is linear and real-valued, and the generation model is binary and trained.

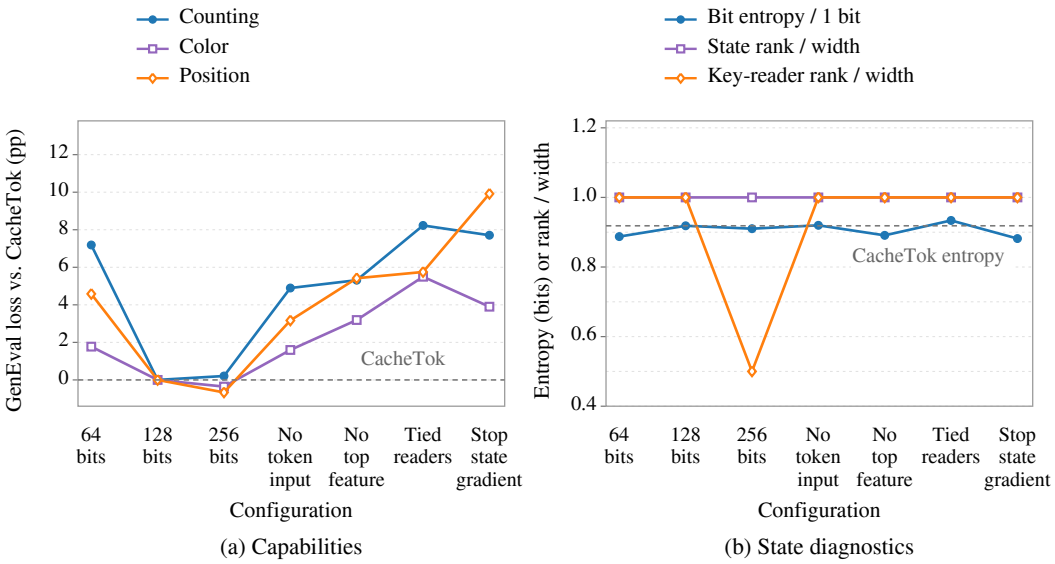


Figure 21: **Balanced bits and full numerical rank coexist with task losses.** Infinity-2B, 1024^2, CFG 3, three seeds, GenEval with 553×4 images per seed. Left: paired CacheTok-minus-variant losses for counting, colors, and position. Right: conditional-history marginal bit entropy, state rank divided by b, and key-reader rank divided by b. The default has $b = 128$, and the input, reader, and state-gradient controls keep that width. The dashed lines mark the default task loss and entropy.

the ordering of Table 3. In the right panel, all these configurations keep bit entropy between 0.88 and 0.94 bits, so none of the losses comes from collapsed or unused bits.

The statistics are defined as follows. For bit coordinate j of a head, let p_j be its positive-sign fraction over the 6425 retained positions. Marginal entropy averages $-p_j \log_2 p_j - (1-p_j)\log_2(1-p_j)$ with $0 \log 0 = 0$. State rank is the rank of the $N \times b$ signed-state matrix, and key-reader rank applies to each $b \times d_h$ reader. Singular values above $\epsilon_{64}\,\sigma_1$ times the larger matrix dimension, after conversion to float64, count toward the numerical rank, and head and layer summaries are averaged within a seed before averaging seeds.



Figure 22: **Repeated outputs under the same text condition.** Infinity-2B with Full-KV above and default CacheTok below. Each of two prompts, a Huichol-beaded jaguar in a jungle and flowers with bubbles in a fish tank, has four samples per interface; sample numbers are display indices.

Figure 23 compares the stored states of four samples drawn for one prompt. For each position of a fixed 32×32 address grid, the heatmap averages the cosine similarity of the unit-norm states over the six sample pairs. Agreement is highest in the upper part of the grid, where the four samples share the sky and mountain layout, and lower in the water and foreground, where their details differ. The state therefore records the content of each sample at each address, beyond what the prompt fixes.

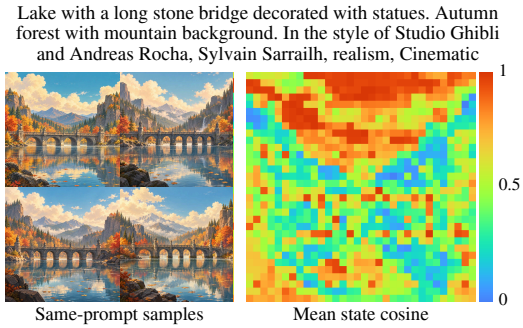


Figure 23: **Same prompt, different samples.** Four outputs share one condition. The map shows the mean cosine similarity of stored states at matching positions over six sample pairs; the color bar applies to the whole map.

This observation bounds a possible shortcut. If the state at an address were determined by the prompt, a server could compute it once per prompt and share it across requests. The low agreement in the foreground shows that the state follows the sampled tokens: the writer receives $\boldsymbol{e}(r_i)$ and the hidden state that the earlier samples produced, so two samples of one prompt diverge as soon as their tokens diverge. The persistent history is therefore a per-request quantity, and the $1024\times$ reduction in Table 2 applies to each request separately. The same property explains why each guidance branch keeps its own history in Figure 8: agreement of the sampled tokens does not make the writer inputs agree. For generation, the map shows that the binary state keeps sample-specific detail exactly where samples differ, in the foreground regions that the later scales refine.

Figure 22 shows complete outputs of repeated sampling for two prompts under both interfaces. Repeated samples of one prompt differ in composition and detail under Full-KV and under CacheTok alike, so the binary history keeps the sample diversity of the backbone, consistent with the per-address state agreement in Figure 23.

E Inference Timing and Batch Capacity

Timing conditions. Table 11 uses one NVIDIA A100 80GB SXM4, BF16, native resolution, and two guidance branches. End-to-end timing covers condition encoding, every visual block, sampling, state reads and writes, and image decoding; model loading, compilation, saving, and evaluation lie outside this window. Each comparison uses three independent sessions with ten warmup batches and thirty timed batches per session, and throughput divides completed images by elapsed time. The common batches are 134 for VAR, 146 for PAR, and 10 for Infinity, identical for both methods.

Exact payload and analytic capacity. Table 2 computes the packed visual bytes from Eq. (10). The last column of Table 11 is the analytic batch capacity $B_{\mathrm{cap}} = \lfloor (79 - M_{\mathrm{fix}})/(M_{\mathrm{req}} + M_{\mathrm{tmp}}) \rfloor$, with all terms in GiB: fixed reservation M_{fix}, persistent request memory per image M_{req}, and temporary allocation per image M_{tmp}. The Full-KV inputs are $M_{\mathrm{fix}} = 2.75$, 4.4, and 11.4 and $M_{\mathrm{tmp}} = 0.515$, 0.310, and 2.7 for VAR, PAR, and Infinity. CacheTok adds its BF16 writer and reader weights to

M_{fix} and replaces the visual keys and values in M_{req} by its packed payload; Infinity includes its maximum native text allocation. For Infinity-2B, the per-image visual history of 3.1 GiB in M_{req} drops to 3.1 MiB, and M_{tmp}, the temporary tensors of the final 64×64 scale, becomes the dominant per-image term. This is why the capacity doubles from 11 to 22 images and grows no further.

Table 11: A100 80GB SXM4, BF16, two guidance branches. Latency is batch-one end-to-end seconds; throughput uses the listed common batch. Capacity is the analytic number of images that fit in 79 GiB.

Method	Latency ↓	Images/s ↑	Capacity ↑
VAR-d16 / 256^2; common batch 134			
Full-KV	**0.234**	6.640	134
CacheTok	0.238	**7.100**	**147**
PAR-XL4x / 384^2; common batch 146			
Full-KV	5.340	5.040	146
CacheTok	**5.120**	**5.730**	**237**
Infinity-2B / 1024^2; common batch 10			
Full-KV	4.850	0.255	11
CacheTok	**4.620**	**0.372**	**22**

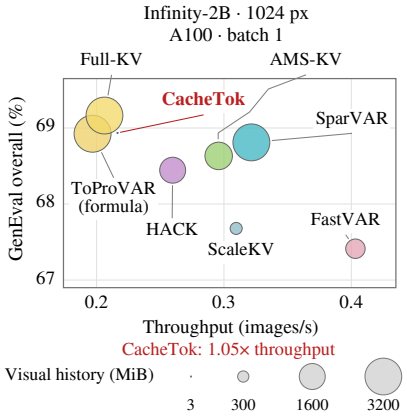


Figure 24: **Quality, throughput, and stored history.** Infinity-2B, 1024^2, CFG 3, one A100, batch one; bubble area is persistent visual MiB.

Quality against cost. Figure 24 places every method of Table 1 on GenEval, batch-one throughput, and persistent visual history for Infinity-2B. CacheTok is $1.05\times$ faster than Full-KV at this batch and stores $1024\times$ less history. The token-selection methods that run faster at batch one skip or sparsify attention work at late scales, and each of them keeps between one tenth and all of the full history, so their bubbles stay large. Methods at the top of the GenEval axis are those that keep most positions readable, in line with the masking results of Section 5.4. The throughput gains at the common batches follow the size of the history each image carries: $1.07\times$ for VAR-d16 with 53 MiB of Full-KV history per image, $1.14\times$ for PAR-XL4x with 201 MiB, and $1.46\times$ for Infinity-2B with 3212.5 MiB (Table 2). The larger the history, the larger the share of attention time spent reading it, and the more a $32\times$ smaller read saves. SparVAR keeps the full resident history, so its bubble matches Full-KV.

Batch-one latency. PAR-XL4x reads the history in each of its 147 blocks, so the $32\times$ smaller read already shortens batch-one latency from 5.34 to 5.12 s. VAR-d16 reads a history of 53 MiB in 10 blocks; there the decoding arithmetic of the readers outweighs the saved bytes, and batch-one latency rises slightly from 0.234 to 0.238 s. Infinity-2B lies between the two, with 4.85 against 4.62 s. Once the batch is large enough for memory to matter, all three backbones run faster with CacheTok (Table 11).

F Larger-Backbone and Multi-Benchmark Extension

F.1 Separate continuation protocol

This study extends the quality evaluation to four larger backbones under its own protocol, and its results are kept separate from the three-seed primary study. The class-conditional settings are VAR-d30 at 256^2 and VAR-d36-s at 512^2 (Tian et al., 2024); the text-conditional settings are Infinity-8B at 1024^2 (Han et al., 2025) and Lumina-mGPT 2.0 7B at 768^2 (Xin et al., 2025). Each CacheTok configuration uses $b = d_h$. Lumina generates one token per step, so it tests the history interface on a token-level generator, where exact training has the longest span.

The extension configuration differs from the primary model in two places: its writer uses RMS normalization without a learned gain, and its decoded historical keys use $\boldsymbol{k}_i^{\ell,g} = \mathrm{Pos}_i^{\ell}(\boldsymbol{z}_{i,g}\boldsymbol{D}_K^{\ell,g})$ without key normalization. We label these rows "CacheTok (ext.)".

All matched learned arms use seeds 17, 29, 43, 71, and 101, 40,000 updates, and effective batch 256. AdamW has moments $(0.9, 0.95)$, $\epsilon = 10^{-8}$, and weight decay 0.05; existing parameters use learning rate 2×10^{-5} and new parameters 10^{-4}, with 1000 warmup updates, BF16 computation, and FP32 optimizer states. The reported checkpoint is update 40,000. ImageNet train supplies the class-conditional continuation and JourneyDB train supplies image–caption pairs. ImageNet reserves ten

validation and two calibration examples per class, and JourneyDB reserves 10,000 validation and 2000 calibration source groups before continuation. The initial Full-KV rows are five generation repeats of one fixed checkpoint. Matched arms share sampling, data order, and initialization, and training-free plug-ins run on the five trained Full-KV parents.

VAR uses CFG 1.5, top-$k = 900$, top-$p = 0.96$, and no smoothing. Infinity uses CFG 3, temperature 1, one sample per bit, and unmodified prompts, without prompt rewriting. Lumina uses CFG 4, top-$k = 4096$, temperature 1, and its native top-p, without Jacobi decoding, prompt thinking, or best-of-N selection. Means are compared within each configuration.

Table 12: Extended ImageNet study: five-seed means under the separate 40,000-update protocol in Appendix F.1. CacheTok (ext.) uses the extension normalization choices described there. Initial Full-KV samples the fixed public checkpoint. Bold compares matched rows.

Method	ImageNet / VAR-d30, 256^2				ImageNet / VAR-d36-s, 512^2			
	FID ↓	IS ↑	Precision ↑	Recall ↑	FID ↓	IS ↑	Precision ↑	Recall ↑
Full-KV (initial)	1.99	302.72	0.8084	0.6012	2.64	302.82	0.8178	0.5957
Full-KV (matched)	**1.93**	303.77	0.8138	0.6042	**2.60**	**306.37**	0.8202	0.5974
HACK (50%) (Qin et al., 2026)	2.01	296.55	0.8069	0.6103	2.64	300.32	0.8181	0.6036
HACK (70%) (Qin et al., 2026)	2.71	271.18	0.7836	**0.6163**	3.09	284.67	0.7997	**0.6120**
AMS-KV (Xu et al., 2026)	2.02	**304.86**	**0.8204**	0.5884	2.69	305.67	**0.8210**	0.5906
CacheTok (ext.)	1.96	304.09	0.8125	0.6058	2.61	304.47	0.8193	0.5990

ImageNet fidelity and coverage. ImageNet uses 50,000 balanced-class samples per seed and resolution-specific ADM references. Table 12 compares the extension with matched Full-KV, two HACK settings (Qin et al., 2026), and AMS-KV (Xu et al., 2026). CacheTok (ext.) reaches FID 1.96 against 1.93 on VAR-d30 and 2.61 against 2.60 on VAR-d36-s, and keeps Inception Score, precision, and recall within the range of the matched rows. HACK at 70% deletion loses 0.78 FID on VAR-d30, which again shows the cost of removing positions.

F.2 Five text-to-image benchmarks

The suite comprises MJHQ-30K (Li et al., 2024b), GenEval (Ghosh et al., 2023), DPG-Bench (Hu et al., 2024), T2I-CompBench++ (Huang et al., 2025), and GenAI-Bench-1600 (Lin et al., 2024; Li et al., 2024a). MJHQ uses all 30,000 prompts, pooled FID, and CLIP ViT-B/32 cosine similarity times 100, at each model's own resolution. GenEval uses 553×4 images and an equal average over the six official tasks. DPG generates 1065×4 images; the pinned evaluator scores 1064 effective prompts after its structural exclusion, with ordered single-pass dependency correction.

CompBench++ uses all eight official tasks, each with 300 prompts and ten images per prompt (24,000 images per seed), and it differs from the six-task CompBench v1 of Table 7. Color and Complex appear in the two-backbone summary of Table 13, and Table 14 lists all eight tasks for every method. GenAI-Bench generates 1600 images and uses the pinned `clip-flant5-xxl` VQAScore (Lin et al., 2024); its tagged union contains 1593 scored inputs (722 basic and 871 advanced). CompBench++ and GenAI scores keep their native scales.

Table 13: Extended text-to-image study, five-seed means. All scores except FID are higher-is-better. GE denotes GenEval (0–100); DPG uses its native percentage scale. CompBench++ Color/Complex and GenAI-Bench all retain native scores. Bold compares matched rows within each backbone. Initial rows are fixed-checkpoint references.

Method	Infinity-8B, 1024^2							Lumina-mGPT 2.0 7B, 768^2						
	MJHQ-30K		GE	DPG	CompBench++		GenAI	MJHQ-30K		GE	DPG	CompBench++		GenAI
	FID ↓	CLIP	All ↑	All ↑	Color	Cmplx	All ↑	FID ↓	CLIP	All ↑	All ↑	Color	Cmplx	All ↑
Full-KV (initial)	8.76	28.55	83.00	86.58	0.821	0.421	0.745	10.32	28.14	68.87	84.19	0.734	0.372	0.720
Full-KV (matched)	**8.60**	28.57	**83.78**	86.87	**0.826**	0.426	**0.750**	**10.15**	**28.18**	**69.76**	**84.53**	**0.741**	0.378	**0.725**
CacheTok (ext.)	8.65	**28.60**	**83.78**	**86.98**	0.824	**0.431**	0.749	10.19	28.14	**69.76**	84.39	0.738	**0.380**	0.723

Larger text-conditional backbones. Full-KV and CacheTok (ext.) reach the same displayed GenEval within each backbone, 83.78 on Infinity-8B and 69.76 on Lumina, and differ by at most 0.14 DPG points. Full-KV keeps a lead of at most 0.05 in MJHQ FID and 0.002 in GenAI-Bench on both backbones, and CacheTok (ext.) leads in CompBench++ Complex on both.

F.3 COMPLETE EXTERNAL-METHOD PANEL

Table 14 uses the same Infinity-8B checkpoint family, sampling settings, and five-benchmark suite for its upper block. HACK keeps its 70% deletion setting, ScaleKV its 10% retention setting, and SparVAR its no-scale-skip setting (Qin et al., 2026; Li et al., 2025b; 2026b). AMS-KV, FastVAR, SparseVAR, ToProVAR, and FasterVAR use the corresponding extension configurations (Xu et al., 2026; Guo et al., 2025; Chen et al., 2025; 2026; Li et al., 2026a). SparseVAR is a different method from SparVAR, and this ToProVAR row uses the extension configuration, distinct from ToProVAR-F in Appendix C.3. The native SkipVAR rows use independently trained decision rules at thresholds 0.88, 0.86, and 0.84 (Li et al., 2025a). Bold marks the best mean within each block, and timing and capacity are reported for the three primary backbones in Appendix E.

Table 14: Complete primary endpoint panel for the extended Infinity-8B comparison: five-seed means over all tasks. All eight CompBench++ tasks are reported separately. GE is GenEval (0–100); CLIP is higher-is-better. Method references and settings appear in Appendix F.3. Bold marks the best displayed mean, including ties, within each training-regime block. Native SkipVAR rows use their own trained decision rules.

Method	MJHQ		GE	DPG	T2I-CompBench++								GenAI
	FID ↓	CLIP	All ↑	All ↑	Color	Shape	Tex.	2D	3D	Num.	Nonsp.	Cmplx	All ↑
Infinity-8B / 1024^2: matched continuation; all CompBench++ endpoints higher is better													
Full-KV	8.60	28.57	83.78	86.87	0.826	0.598	0.700	0.352	**0.449**	0.597	0.319	0.426	0.750
HACK (70%) (Qin et al., 2026)	**8.59**	**28.70**	82.80	86.45	0.826	0.592	0.690	0.343	0.439	0.587	0.318	0.418	0.745
ScaleKV (10%) (Li et al., 2025b)	8.69	28.57	83.03	86.63	0.824	0.594	0.693	0.347	0.441	0.594	0.319	0.422	0.747
AMS-KV (Xu et al., 2026)	8.77	28.54	83.24	86.63	0.825	0.599	0.695	0.347	0.441	0.593	0.318	0.424	0.747
FastVAR (Guo et al., 2025)	8.86	28.45	82.79	86.42	0.819	0.590	0.674	0.351	0.434	0.590	0.318	0.419	0.744
SparseVAR	8.69	28.56	83.20	86.76	0.820	0.594	0.690	0.355	0.441	0.597	0.319	0.424	0.747
ToProVAR (Chen et al., 2026)	**8.59**	28.61	**84.04**	87.02	**0.831**	0.601	0.695	**0.363**	0.443	0.598	**0.320**	**0.431**	**0.751**
FasterVAR (Li et al., 2026a)	8.80	28.54	83.51	86.83	0.827	0.597	0.690	0.356	0.443	0.595	0.319	0.427	0.748
SparVAR (no skip) (Li et al., 2026b)	8.62	28.59	83.80	**87.08**	0.827	**0.602**	**0.702**	0.356	0.445	**0.602**	**0.320**	0.429	0.750
CacheTok (ext.)	8.65	28.60	83.78	86.98	0.824	0.597	0.699	0.360	0.443	0.592	0.318	**0.431**	0.749
Infinity-8B / 1024^2: separately trained native SkipVAR decision rules; thresholds shown													
SkipVAR (0.88) (Li et al., 2025a)	**8.90**	**28.53**	**82.77**	**86.48**	**0.818**	**0.592**	**0.684**	**0.347**	**0.442**	**0.596**	**0.319**	**0.419**	**0.745**
SkipVAR (0.86) (Li et al., 2025a)	9.05	28.46	82.16	86.23	0.816	0.586	0.678	0.344	0.432	0.582	0.318	0.414	0.742
SkipVAR (0.84) (Li et al., 2025a)	9.32	28.36	81.23	85.70	0.808	0.575	0.659	0.333	0.421	0.571	0.316	0.402	0.734