

Peer Review

Review of: "CAT: Content-Adaptive Image Tokenization"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The paper argues that an image tokenizer should not spend the same number of latent tokens on every image. A picture of a zebra on grass and a chart full of small labels are hard in different ways, and pixel-level measures such as JPEG size, MSE, and LPIPS get this backwards: they rate high-contrast textures as complex and underrate text and faces (Section 3.1.2, Figure 3). The authors therefore predict complexity from text alone. InstructBLIP writes a caption and answers two yes/no questions (text present? faces present?), Llama 3 70B Instruct turns this description into a score from 1 to 9, and two thresholds map the score to 8x, 16x, or 32x compression. A single nested VAE with a shared middle block and skip connections with channel matching (Section 3.3) then encodes each image at its assigned ratio.

The claimed outcomes are three. For reconstruction, CAT matches a fixed 16x tokenizer on natural images while using fewer tokens and beats it clearly on faces and charts (Table 1). For generation, a DiT-XL/2 trained on CAT latents reaches FID 4.56 on 512×512 class-conditional ImageNet against 4.78 for the fixed 16x tokenizer at equal training compute, with 18.5% higher inference throughput (Table 4, Section 5.2). And by forcing the token count at inference, the user can control the complexity of the generated image (Table 6, Figure 5).

2. Whether the conclusion holds on the data

I think the core idea is genuinely good, and the experimental design is careful where it matters most. The authors built the right control: the same nested architecture trained with JPEG size as the complexity signal, with thresholds (38761, 65837) chosen so that the training distribution of ratios is identical to CAT's (Appendix 8.3). That isolates the value of the caption-based score from the value of the architecture, and Table 1 shows the caption score winning on every dataset and metric against that baseline. The

fixed-ratio baselines share data, backbone, and training schedule (Appendix 8.3). The motivation study in Section 3.1.1 is also concrete: 28.3% of 41K COCO images lose less than 0.001 MSE going from 8x to 32x. I also appreciated Table 8, where the authors show LDM VAE numbers that are partly better than their own. I checked the arithmetic behind the token counts because several headline claims rest on it. The "Latent Dim" column of Table 2 is the square root of the mean token count, not the mean side length. For ImageNet with CAT (Table 2: 6% / 49% / 45%), the mean token count is $0.06 \times 4096 + 0.49 \times 1024 + 0.45 \times 256 = 862.72$, and $\sqrt{862.72} = 29.37$, which matches the reported 29.32 up to rounding of the percentages. The same holds for COCO (1016.32, $\sqrt{} = 31.88$ vs. 31.87), CelebA (1546.24, $\sqrt{} = 39.32$ vs. 39.29), and ChartQA (3973.12, $\sqrt{} = 63.03$ vs. 63.02). The "16% fewer tokens" on ImageNet in Section 1 follows: $29.32^2/32^2 = 859.7/1024 = 0.84$. So the numbers are internally consistent, and they follow the token-weighted average of Eq. 2. The Table 2 caption, though, calls the column the "average spatial dimension" r/f , which a reader will take as the mean side length (26.72 for ImageNet). A pointer to Eq. 2 in the caption would prevent that reading.

The same check changes how I read the CelebA and ChartQA gains. Section 1 reports that CAT reduces rFID "by 12% on CelebA and 39% on ChartQA relative to fixed-ratio baselines." From Table 1 (rFID, CelebA: 2.25 vs 1.97; ChartQA: 8.67 vs 5.27), I get $(2.25-1.97)/2.25 = 12.4\%$ and $(8.67-5.27)/8.67 = 39.2\%$, so the percentages are correct. But on these test sets, CAT is not a 16x tokenizer. By the Table 2 counts, it spends $1546/1024 = 1.51$ times the tokens of the fixed 16x model on CelebA and $3973/1024 = 3.88$ times on ChartQA. This is exactly what the method is designed to do, and it is a fair result. It is not a comparison at equal budget, however, and the text presents it as if it were. The claim in Section 4.2 that "On ChartQA, CAT even surpasses the fixed 8x baseline" is true for rFID (5.27 vs 8.21) but not for LPIPS (0.021 vs 0.019) or PSNR (36.45 vs 36.98), while CAT uses about 97% of the 8x budget there (3973 of 4096 tokens).

On ImageNet, the text says CAT has "only a slight drop in rFID" (Section 4.2). Table 1 gives 0.46 for CAT and 0.38 for fixed 16x, a relative increase of 21%. At 16% fewer tokens, that may still be a good trade, but I would call it a trade rather than a slight drop.

Section 4.3 states that thresholds (4,7) "produce better reconstruction metrics across all datasets." Table 3 does not quite say this: the ImageNet rFID is 0.46 for (4,7) and 0.43 for (2,8). The (2,8) setting is also at 16.5x rather than 16.0x (Table 3, column Average), so the two rows are close to, but not exactly at, equal budget. I checked that 16.5x is what the token-based average gives: $0.005 \times 4096 + 0.895 \times 1024 + 0.10 \times 256 = 962.56$ tokens, $512/\sqrt{962.56} = 16.5$. The arithmetic mean of the ratios would be 17.56, so the "Average" column is again token-weighted, as Eq. 2 defines it. It would help to say so in the Table 3 caption.

For generation, the FID gap in Table 4 is 4.56 vs 4.78 (row Adaptive CAT vs. row Fixed 16x, column FID). This is one training run per configuration, with no seed variation and no interval. A 0.22 difference in FID-50K is in the range where different seeds of the same DiT can disagree, so I would like to see at least two more seeds for the two top rows. The sFID gap (10.55 vs 11.81) and precision (0.75 vs 0.72) point the same way, and recall is tied at 0.49, which makes the overall picture plausible but not yet settled.

The throughput figure is consistent with the token counts. Mean inference tokens are 216 against 256 (Section 5.2); $256/216 = 1.185$, which is the 18.5% claimed, and $216/256 = 0.84$ is close to the 0.82 relative FLOPs in Table 4. What I could not find is how variable-length sequences are batched at inference. If samples are grouped by length, throughput scales roughly with tokens; if they are padded, the gain shrinks. The same question applies to training, where "same flops" is enforced by a fixed token batch of 262,144 (Section 5.1). Attention cost grows with the square of sequence length, so equal tokens do not mean equal FLOPs, and CAT's shorter sequences likely cost somewhat less. That favors CAT slightly.

Section 5.2 says CAT with $c = 8$ "obtains the best FID across all experiments we perform" (Table 7, 4.38). Table 6 reports 4.12 for "CAT 8x," i.e., when all images are forced to 1024 tokens. The text should either exclude forced-ratio runs from that claim explicitly or revise it. Relatedly, Table 6 shows that FID improves monotonically with more tokens (5.83 at 32x, 5.02 at 16x, 4.12 at 8x). That is expected, but it raises a question the paper does not address: how much of the Table 4 advantage comes from the 8x images in the adaptive mix, and would a DiT on the fixed 8x tokenizer at the same token budget beat both? There is no fixed 8x row in Table 4.

Finally, the evaluator. Section 3.2 reports an exact agreement of 62.39% between the caption-based ratio and the maximum acceptable ratio on COCO. I would add the tolerance τ used here and a reference point: what agreement do JPEG size or the majority class achieve under the same thresholds? Without that, 62.39% is hard to interpret. The correlations themselves appear only in Figure 2 (right), and the text refers to "Table 2" for them twice (Section 3.1.2 and the last paragraph of Section 3.2), while Table 2 on Qeios shows test-set ratio distributions.

Overall, the main claims are supported in direction. The size of the gains on faces and charts is partly a matter of budget, and the generation result needs replication across seeds.

3. Reproducibility

The architecture and training settings are given in unusual detail. Appendix 8.1 lists the diffusers-style VAE configuration, 8.2 the optimizer, schedule, and all five loss weights, 9.2 the DiT and DDPM settings

(1000 training steps, squaredcos_cap_v2, 250 inference steps, cfg 1.5, 10% label dropout). The Hugging Face identifiers of both language models are named: Salesforce/instructblip-vicuna-7b and meta-llama/Meta-Llama-3-70B-Instruct (Appendix 7).

The most important piece is missing on Qeios. Appendix 7, Step 2 ends with "the prompt:" and nothing follows. As rendered, the complexity prompt, with its scoring factors and per-score examples described in Section 3.2, is not in the article. Since the whole method depends on this prompt, I would ask the authors to check the Qeios version and restore it.

Decoding parameters for both models are not given: temperature, sampling vs. greedy, maximum tokens, and how the integer score is parsed from Llama's output. I could not find whether the score is stable across repeated calls on the same caption. A small test (say, 1,000 images scored three times) would show how often an image flips between ratios.

The training data are 380M licensed Shutterstock images (Section 4.1), which outside readers cannot obtain. That is understandable, but it means the reconstruction results cannot be reproduced as such. The distribution of LLM scores on that set is also not shown, and the choice of (4,7) depends on it. I found no statement about code or weights. Seeds are not reported for any run.

For ImageNet generation, training captions come from InstructBLIP, while inference uses the template "this is an image of [label]" (Section 5.1). I could not tell how the face/text questions are answered at inference, or what share of classes lands in each ratio.

Compute is stated: 64 A100s for 1M steps for each tokenizer, 16 H100s for 400K steps for each DiT.

4. What specifically to fix

1. **Appendix 7, Step 2.** Restore the full Llama 3 prompt, which is absent on Qeios, and add decoding parameters and the score-parsing rule.
2. **Section 3.1.2 and Section 3.2, final paragraph.** Both cite "Table 2" for Pearson correlations; the values are in Figure 2 (right). Correct the reference and give the correlation coefficients as numbers in the text, including the one for the caption score.
3. **Section 3.2, the 62.39% agreement.** State τ , and add the agreement of the JPEG-size rule and of a constant 16x prediction under the same setup.
4. **Table 2 and Table 3 captions.** Refer to Eq. 2 and say that "Latent Dim" and "Average" are token-weighted (Latent Dim = $\sqrt{(\text{mean token count})}$); my checks above reproduce both tables only under

that definition, while "average spatial dimension" suggests a plain mean.

5. **Section 1, paragraph on reconstruction, and Section 4.2.** Next to the 12% (CelebA) and 39% (ChartQA) rFID reductions, report the token budget actually used, about 1.5× and 3.9× the fixed 16x model by Table 2. A token-matched comparison would help a lot, for example against the fixed 8x row for ChartQA.
6. **Section 4.2, "CAT even surpasses the fixed 8x baseline".** Limit this to rFID; on LPIPS and PSNR (Table 1, ChartQA) the 8x baseline is better.
7. **Section 4.2, "only a slight drop in rFID" on ImageNet.** 0.38 → 0.46 is a 21% relative increase. I would describe it as a trade against 16% fewer tokens.
8. **Section 4.3, "Better reconstruction metrics across all datasets"** does not match Table 3, ImageNet column (0.46 vs 0.43). Please qualify.
9. **Table 4.** Run at least two more seeds for Fixed 16x and CAT and report the mean and spread of FID.
10. Add a DiT on the fixed 8x tokenizer to **Table 4**, even at reduced steps. It shows how much of the gain the 8x share of the mix carries.
11. **Section 5.2, "best FID across all experiments".** Table 6 has 4.12 for forced 8x. Revise the sentence or restrict its scope.
12. **Section 5.1.** Describe how variable-length sequences are batched in training and at inference, and how throughput was measured (batch size, hardware, padding).
13. Short one: report the distribution of LLM scores 1 to 9 on the Shutterstock training set, so that the (4,7) and (2,8) choices in Section 4.1 can be seen.
14. **Section 1, last paragraph.** The closing sentence presents CAT as a key step toward efficient image modeling. The evidence is one resolution (512), one average budget (16x), and one generation benchmark; I would tone this down.
15. Typos: "emprical" (Section 4.1), "trainingcompression" (Section 4.2), "existing work all require" (Section 1).

5. Recommendation

Minor revisions. The idea of assigning tokenizer capacity from a text description is new, practical for text-to-image pipelines, and tested against the right JPEG-based control. The numbers I could recompute are consistent. What needs work is the framing of the budget in the CelebA and ChartQA gains, seed replication for the Table 4 FID gap, and the missing scoring prompt in Appendix 7. None of this requires new methodology.

Declarations

Potential competing interests: No potential competing interests to declare.