# CAT: Content-Adaptive Image Tokenization

**Junhong Shen**[*] Carnegie Mellon University
**Kushal Tirumala** FAIR at Meta
**Michihiro Yasunaga** FAIR at Meta
**Ishan Misra** FAIR at Meta
**Luke Zettlemoyer** FAIR at Meta
**Lili Yu**[†] FAIR at Meta
**Chunting Zhou**[†] FAIR at Meta

## Abstract

Most existing image tokenizers encode images into a fixed number of tokens or patches, overlooking the inherent variability in image complexity and introducing unnecessary computate overhead for simpler images. To address this, we propose **C**ontent-**A**daptive **T**okenizer (**CAT**), which dynamically adjusts representation capacity based on the image content and encodes simpler images into fewer tokens. We design (1) a *caption-based evaluation system* that leverages LLMs to predict content complexity and determine the optimal compression ratio for an image, and (2) a novel *nested VAE architecture* that performs variable-rate compression in a single model. Trained on images with varying complexity, CAT achieves an average of **15% reduction in rFID** across **seven** detail-rich datasets containing text, humans, and complex textures. On natural image datasets like ImageNet and COCO, it **reduces token usage by 18%** while maintaining high-fidelity reconstructions. We further evaluate CAT on two downstream tasks. For image classification, CAT consistently improves top-1 accuracy across **five** datasets spanning diverse domains. For image generation, it **boosts training throughput by 23%** on ImageNet, leading to more efficient learning and improved FIDs over fixed-token baselines.

## 1 Introduction

Image tokenizers compress high-resolution images into low-dimensional latent representations, enabling compact and semantically meaningful inputs for downstream tasks such as generation and classification [1–8]. Despite their effectiveness, most existing tokenizers operate at a fixed compression ratio, producing latent representations of uniform length regardless of the image's content. However, natural images exhibit significant variability in complexity, from sparse scenes to densely textured ones, suggesting that fixed-length representations can be both inefficient and suboptimal.

Classical codecs such as JPEG [9] implicitly exploit this variation: when fixing the quality level, they produce different file sizes for images with different frequency characteristics. In contrast, fixed-ratio tokenizers may under-compress simple images, wasting compute on redundant information, or over-compressing complex ones, losing important details. These problems become more prominent when tokenizers are used in large-scale generative pipelines or as feature extractors in downstream tasks.

Several recent works explore dynamic token representation during inference [10, 11]. However, these methods typically require access to the *input image*—an assumption incompatible with many practical use cases. For example, in image generation with latent diffusion models [12], only the user's text prompt is available, and the number of latent tokens, which significantly influences generation quality, must be specified in advance. Moreover, these methods do not adapt tokenizer *training* to image complexity, missing an opportunity to optimize for both content and downstream utility.

[*]Work done during internship at Meta. [†]Joint senior author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

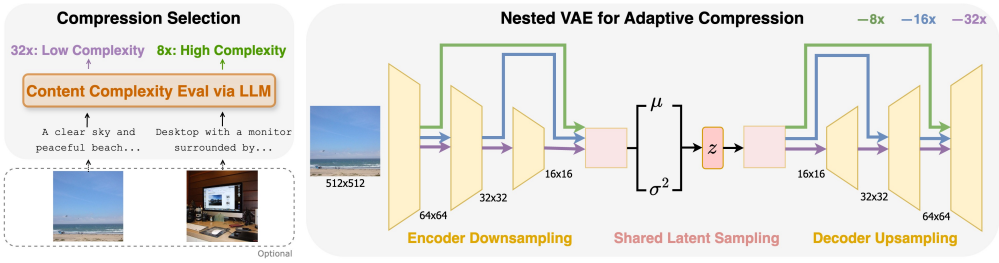



Figure 1: **Content-Adaptive Tokenization.** CAT uses an LLM to determine the compression ratio from the image's text description and uses a nested VAE to generate latent features by dynamically routing the input.

In this work, we introduce **Content-Adaptive Tokenizer (CAT)**, a novel approach that dynamically adjusts representation capacity based on image complexity. CAT combines a *caption-driven complexity evaluator* with a *nested autoencoder architecture* to produce variable-length latent features in a single forward pass (Figure 1). Specifically, the evaluator uses large language models (LLMs) to predict the optimal compression ratio from textual descriptions. It analyzes the image's caption and answers perception-oriented questions (e.g., *"are there human faces or text?"*) to produce an interpretable complexity score. Based on this score, we assign one of three compression ratios to the image: 8x, 16x, or 32x. Empirical results (Section 3.2) show that this system is robust across different LLMs and caption styles, providing a general mechanism for content-aware adaptation.

To support variable-length representations, we design a nested variational autoencoder (VAE) architecture that routes intermediate encoder features to a shared latent block for generating Gaussian parameters of different shapes, enabling latent codes at multiple spatial scales. This design allows us to train a *single* model that supports multiple compression levels while maintaining architectural efficiency.

We train CAT on a diverse set of images using LLM-evaluated compression ratios and conduct extensive evaluations across nine datasets, covering natural scenes (ImageNet [13], COCO [14]), human faces (CelebA [15]), and detail-heavy domains such as text (ChartQA [16], GTSRB [17], SVHN [18]), textures, and satellite imagery. On large-scale natural images, CAT preserves high reconstruction quality while **reducing token usage by 18%** compared to fix-token baselines. On complex images, CAT achieves significantly better reconstruction quiality, **improving the rFID by 12% on CelebA, 17% on GTSRB, 20% on SVHN, and 39% on ChartQA** relative to fixed-token baselines. We also benchmark CAT on two critical downstream tasks:

- **Image classification:** CAT achieves the **highest linear probing accuracy** compared to all fixed-token baselines across five challenging datasets where prior work has shown that zero-shot models perform poorly [19]. This highlights the quality of our content-adaptive latent representations. Besides, CAT consistently improves performance in full fine-tuning settings.
- **Text-to-Image Generation:** We integrate CAT into Latent Diffusion Transformers (DiTs) [20]. On class-conditional ImageNet generation, CAT **increases the training throughput by 23%**, thus achieving **better FIDs** than all fixed-ratio tokenizers trained under the same FLOPs. We note that CAT allows users to specify the desired token count at inference time, enabling a flexible trade-off between computational cost and output quality, with more tokens typically yielding higher fidelity.

In summary, we propose CAT, an efficient and effective image tokenizer that enables content-adaptive compression through an LLM-based evaluator and a nested VAE architecture. To the best of our knowledge, this is the first work to combine language-guided tokenization with adaptive representation, showing both performance and efficiency gain in image reconstruction, classification, and generation.

## 2 Related Work

**Image tokenization.** Existing tokenizers use diverse architectures and encoding schemes. Continuous tokenizers often utilize the VAE architecture [2] to generate Gaussian distributions for sampling continuous latent features. Discrete tokenizers like VQ-VAE [21], RQ-VAE [22], MoVQ [23], MAGVIT-v2 [3], and FSQ [7] use quantization techniques to convert latent representations into tokens. VQ-GAN [1], ViT-VQGAN [24], and Efficient-VQGAN [25] further built on adversarial training to improve performance. Beyond methods that tokenize images into 2D grids, 1D tokenizers such as TiTok [8] are proposed to enhance efficiency. While CAT is designed as a continuous 2D tokenizer, the proposed adaptive image encoding scheme can be applied to discrete and 1D tokenizers.

**Adaptive compression.** Traditional codecs like JPEG [9] for images and H.264 [26] for videos apply varying levels of compression based on the input media, producing files of different sizes. In deep learning, patch dropout [27, 28], patch merging [29–32], and sequence packing [33] are proposed for Vision Transformers [34]. Quadformer [35] uses mixed-resolution patches to vary token count. However, these methods are tailored for visual understanding tasks and cannot be used for generation. A few recent works such as VAR [36] study multi-scale tokenization for generation. Nonetheless, these works are not adaptive to image content.

Adaptive tokenizers for image generation remain relatively underexplored. ElasticTok [10] employs random masking to drop the tail tokens during training. ALIT [11] iteratively distills 2D tokens into 1D to reduce the token count. DQ-VAE [37] leverages information-density for dynamic representation. However, all of these methods (1) require an input image to determine the token count, limiting their use in generation settings where only text is available; and (2) overlook image complexity during training. In contrast, we enable adaptive compression directly from textual descriptions without observing the image. We also explicitly train the tokenizer with complexity-aware supervision. A concurrent work, TexTok [38], explores language-guided tokenization by supplying the caption embeddings to the VAE. However, it is not designed for adaptive representation.

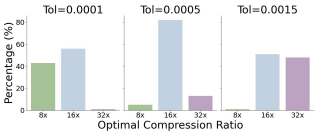
**Multi-scale network design.** Our work is also related to designing neural networks for multi-scale feature extraction. Inspired by U-Net [39] and Matryoshka networks [40–42], we incorporate skip connections into the VAE to support multi-ratio compression in a single forward pass. Parallel work explores transformer-based multi-scale architectures [43–49]. To the best of our knowledge, our nested design offers the simplest yet effective solution for generating multi-scale latents via VAEs without additional architectural or computational overhead, while achieving strong results (Section 4).

## 3 Method

In this section, we present CAT for adaptive image tokenization. We begin by motivating our caption-based evaluator for image complexity estimation. Then, we describe the nested VAE architecture.

### 3.1 Proof of Concept

**How much can we actually compress?** A key question in this work is to determine how much an image can be compressed without significant loss of quality. To explore this, we study the reconstruction performance of existing tokenizers under various compression ratios. We take the open-source LDM tokenizers [2] with 8x, 16x and 32x compression ratios and compute their reconstruction mean squared error (MSE) on 41K COCO 2014 [14] images with resolution 512. We find that for 28.3% of the images, 32x compression results in less than a 0.001 MSE increase compared to 8x, while reducing the token count by a factor of 16. We also compute the best MSE among all compression ratios for each image and determine the maximum acceptable compression ratio under a tolerance $\tau$ ($\arg\max_{ratio} \text{MSE}_{ratio} - \text{MSE}_{best} < \tau$). Fig. 2 (left) shows that 56% of the images can be compressed at least to 16x with negligible (0.0001) increase in MSE[3]. That is, a large portion of natural images can be compressed more aggressively to save compute.




| Metric | Pearson $r$ |
|---|---|
| JPEG | 0.31 |
| MSE | 0.36 |
| LPIPS | 0.23 |
| Caption | **0.55** |

Figure 2: **Left:** Max acceptable compression ratios for different $\tau$. **Right:** Correlation with max acceptable compression ratio for $\tau = 0.0015$.

On the other hand, our visual inspection reveals that images with fine-grained elements like text have significantly worse reconstruction quality at 32x compression compared to 8x compression (e.g., see row 3 and 4 in Figure 3). This suggests that more tokens are required to accurately reconstruct low-level details. The above results provide strong motivation for developing an adaptive tokenizer.

**Limitations of existing complexity metrics.** Next, we want to identify a metric for predicting an image's optimal compression ratio. We explore two existing options: (1) metrics produced by traditional codecs, such as the JPEG file size; (2) metrics based on pretrained VAEs, such as the reconstruction MSE and LPIPS distance [50]. We use Stable Diffusion's sd-vae-ft-mse [51] for this

[2] LDM [12] released a series of VAE tokenizers with diverse compression ratios and trained in a controlled setting. Most other tokenizers, such as `stabilityai/sd-vae-ft-mse`, only have one compressed ratio.

[3] The average MSE across all images for 8x LDM VAE is 0.0039, so a 0.0001 tolerance should be acceptable.

Figure 3: **Existing metrics can misjudge image complexity.** Text-heavy images that are difficult to difficulty to reconstruct (note the distortion in the bottom two rows) are considered as easy by existing metrics.

analysis. We compute these metrics on COCO and analyze their correlation with the maximum acceptable compression ratio under 0.0015 tolerance. However, Table 2 shows that the Pearson $r$'s are relatively low. Statistically, these metrics are not highly correlated with an image's complexity.

We also manually inspect images with large JPEG sizes and MSEs. We note that images featuring repetitive patterns, such as grass, forests, and animals like giraffes and zebras consistently show high complexity metrics. Indeed, JPEG compression can be inefficient for images with sharp edges and high contrast. A single-pixel shift in a zebra image can toggle pixel values between black and white, significantly increasing the reconstruction error. However, as Figure 3 (left) show, large JPEG sizes or MSEs do not always notably affect visual quality. For example, we can easily recognize the zebra and may not perceive the differences resulting from various compression ratios.

Conversely, images with small JPEG sizes and low MSEs can have poor fidelity. For example, as Figure 3 (right) shows, distortions in images containing elements sensitive to human perception, such as text, numbers, and human faces, can drastically reduce visual quality. Despite this, these images have low reconstruction errors since the critical elements occupy only small portions of the images.

Thus, existing metrics fail to capture details crucial to human perception. In contrast to the predicted complexity, we actually want to use a large compression ratio for zebra images, and a small ratio for the phone images. Beyond this, all considered metrics require images as input and cannot be used for text-to-image generation tasks, where no image is available. Given all these limitations, we seek a new complexity metric that is independent of pixel data and aligns with human perception. We note that it is impractical to use the tolerance-based compression ratio in Figure 2 as the complexity metric because it requires at least three model calls to get the MSEs, making it computationally expensive.

### 3.2 Complexity Evaluation via Captions and LLMs

Image generation typically involves users providing a prompt that describes the desired image content. Inspired by this use case, we propose to use the text description of an image to evaluate its complexity.

We propose a 3-stage evaluation pipeline: (1) obtaining the text description, (2) prompting a LLM for a complexity score, and (3) classifying the score into a compression ratio. The text description includes both the image caption and answers to a set of pre-defined queries of the form "*Are there [obj]?*" For example, *obj* $\in \{$*human faces*, *text*$\}$ can be used to align with human perceptions. During training, when image data are available, we use a vision language model (VLM) to generate the captions and answers. During inference, users need to provide the necessary textual information.

We prompt a language model with the text description to generate an integer score ranging from 1 to 10, where higher scores indicate greater complexity. To ensure scoring consistency, we design a detailed rubric that instructs the model to consider factors including semantic complexity (objects, scenes), visual complexity (color, texture), and perceptual complexity. We also provide in-context examples for each score. The complete prompt is provided in Appendix B.

Based on the complexity score, each image is classified into one of three compression ratios: 8x, 16x, or 32x, with higher complexity scores corresponding to lower ratios. We choose these options as they are widely used in existing tokenizers [12] and provide meaningful variation in token counts. Then, we implement a thresholding scheme to divide the scores into three intervals: $[1, a]$, $(a, b]$, and $(b, 10]$, where $a < b \in \mathbb{Z}^+$ are selected to achieve an average compression ratio of 16x across all training data to enable fair comparison with fixed 16x tokenizers. Formally, denote the training distribution as $\mathcal{X}$, input resolution as $r$, the compression ratio of an image $x \in \mathcal{X}$ as

$f(x) \in \{f_1 = 8, f_2 = 16, f_3 = 32\}$, the target average compression ratio as $\bar{f} := 16$. We set $a, b$ by:

$$\mathbb{E}_{x \in \mathcal{X}}[\frac{r^2}{f(x)^2}] \approx \sum_{x \in \mathcal{D}} p\big(f(x)\big) \frac{r^2}{f(x)^2} \approx \frac{r^2}{\bar{f}^2} \tag{1}$$

While multiple threshold configurations can achieve the target average compression ratio, our experiments in Section 4.2 demonstrate that a diverse distribution of ratios yields better performance. For the specific training dataset and thresholds used in our experiments, see Section 4.

**Robustness testing and bias mitigation.** To evaluate the robustness of our caption-based complexity evaluator, particularly under different model choices and caption styles, we conduct ablation studies on diverse datasets such as ImageNet [13], CelebA [15] and EuroSAT [52]. We compare two pipelines: (1) a unified pipeline using LLaVA1.5 7B [53] for both captioning and scoring in a single pass, and (2) a separated pipeline using InstructBLIP [54] for captioning and either Qwen2.5 7B [55] or LLaMA3.1 8B Instruct [56] for scoring. Additionally, within the unified pipeline, we test variants that explicitly prompt the model to produce longer versus shorter captions. As shown in Appendix Table 10, all configurations result in similar compression distributions. We attribute this robustness to (1) our carefully designed scoring rubrics and in-context examples, and (2) the LLM's ability to infer content complexity is more influenced by the semantic content of the caption than its wording. These findings suggest that our evaluator is robust to model choices and captioning variations. Given its simplicity and efficiency, we adopt the unified pipeline in all subsequent experiments.

**High correlation with visual complexity.** We verify that our caption score provides reliable estimates of the optimal compression ratio. Similar to Section 3.1, we compute the correlation between complexity scores and maximum acceptable compression ratios on COCO. Our metric achieves the highest Pearson $r$ among all options (Table 2) and 62.39% exact agreement on compression ratio selection. Manual inspection also confirms that perceptually challenging images receive high complexity scores. We note that during development, we also tested other caption-derived metrics such as caption length, but it does not perform as well as LLM-based scoring (Appendix Table 9).

**Minimal captioning overhead.** When text information is available, our pipeline requires only a single LLM call to obtain the complexity score. During training, where only images are provided, we generate the caption, query responses, and complexity score within a single inference pass, keeping the evaluation overhead minimal. Given the efficiency of modern inference engines like vLLM [57], the cost of complexity evaluation is negligible compared to the overall compute required for training the tokenizer and downstream models.

### 3.3 Nested VAE for Adaptive Comprssion

To reduce the tokenizer's training and storage costs, we introduce a nested structure to the standard VAE architecture [2] to enable multiple compression ratios within a single model. In standard VAE architecture, the encoder consists of multiple downsampling blocks followed by an attention-based middle block. The decoder consists of an attention-based middle block followed by upsampling blocks. This symmetrical design resembles U-Net [39] and Matryoshka networks [40] for multi-scale feature extraction. Inspired by these works, we leverage the intermediate outputs of the downsampling blocks to enable adaptive compression (Figure 1). We describe the proposed architecture below.

**Skip connection with channel matching.** Denote the feature shape under the largest compression ratio as $(c_3, \frac{r}{f_3}, \frac{r}{f_3})$, where $c_3$ is the channel dimension. We observe that, in the standard VAE encoder, the spatial dimension of the intermediate outputs from the downsampling blocks decreases by a factor of 2 with each additional block. This means that the output of the second-to-last downsampling block has shape $(c_2, \frac{r}{f_2}, \frac{r}{f_2})$, and the output of the third-to-last downsampling block has shape $(c_1, \frac{r}{f_1}, \frac{r}{f_1})$. Then, a natural thought is to directly route these intermediate outputs to the middle block to generate latent features. However, since the channel dimensions of these intermediate outputs vary, we incorporate ResNet blocks [58] for channel matching. Let the latent channel dimension of the VAE be $c$. Applying channel matching enables us to transform intermediate features of shape $(c_n, \frac{r}{f_n}, \frac{r}{f_n})$ to $(c, \frac{r}{f_n}, \frac{r}{f_n})$ for $n = 1, 2, 3$. This will be the shape of the latent parameters.

For decoder, we similarly add skip connection with channel matching and use the decoder middle block's output as the input to the corresponding upsampling block, i.e., for compression ratio $f_n$, we bypass the first $n - 1$ upsampling blocks so the final output has the same size as the original image.

**Shared mean/variance parametrization.** Features after channel matching are directed to the middle block to generate the latent distribution parameters. In CAT, we share the middle block for all compression ratios to maintain scale consistency of the parameterized mean and variance. The convolutional design of the middle block allows it to process inputs of varying spatial dimensions, as long as the channel dimension is aligned. Thus, for all $n \in \{1, 2, 3\}$, the mean $\mu_n$, variance $\sigma_n^2$, and samples $z_n$ of the Gaussian distribution all have shape $(c, \frac{r}{f_n}, \frac{r}{f_n})$, i.e., the input compressed at $f_n$.

**Increasing parameter allocation for shared modules.** Images assigned larger compression ratios do not go through the later downsampling blocks and are directed straight to the middle block. The middle block is thus tasked with handling multi-scale features. To improve its capacity, we allocate more parameters to the middle block by increasing the number of attention layers.

### 3.4 Training

While existing adaptive tokenizers like ElasticTok [10] do not consider the varying complexity of training data, we explicitly incorporate content complexity into training to learn feature extraction at different granularity. For each training example, we first obtain the compression ratio from the LLM. Then, the image is processed only by the layers dedicated to its compression ratio.

Similar to prior works [2, 1], we use a joint objective that minimizes reconstruction error, Kullback-Leibler (KL) divergence, and perceptual loss. Specifically, we use $\mathcal{L}_1$ loss for pixel-wise reconstruction. To encourage the encoder output $z$ towards a normal distribution, KL-regularization is added: $\mathcal{L}_{\text{KL}}(z) := \mathbb{KL}(q_\theta(z|x) \| p(z))$, where $\theta$ is the encoder parameters and $p(z) \sim \mathcal{N}(0, \mathbf{I})$. The perceptual loss consists of the LPIPS similarity [50] and a loss based on the internal features of the MoCo v2 model [59]. Beyond these, we train our tokenizer in an adversarial manner [60] using a patch-based discriminator $\psi$, which adds a GAN loss $\mathcal{L}_{\text{GAN}}(x, \hat{x}, \psi)$. Thus, our overall objective is:

$$\mathcal{L} = \min_{\theta} \max_{\psi} \mathbb{E}_{x \in \mathcal{X}} \Big[ \mathcal{L}_{\text{rec}}(x, \hat{x}) + \beta \mathcal{L}_{\text{KL}}(z) + \gamma \mathcal{L}_{\text{perc}}(\hat{x}) + \delta \mathcal{L}_{\text{GAN}}(x, \hat{x}, \psi) \Big], \tag{2}$$

where $\beta, \gamma, \delta$ are loss weights. We provide our training code in the supplementary material.

## 4 Image Reconstruction

We first evaluate CAT's reconstruction quality and analyze various design choices via ablation studies.

**Training details.** We use a nested VAE with six downsampling blocks and 187M parameters. For our main results (Table 3), we set the latent channel $c$ to 16. We study its effect as an ablation study in Section 4.2. For training data, we use 380M licensed Shutterstock images with 512x512 resolution. After obtaining the complexity scores, we find that two sets of thresholds, $(a, b) \in \{(4, 7), (2, 8)\}$, both achieve an average compression ratio of 16x. We select $(4, 7)$ for our main experiments because it leads to a more diverse distribution and better empirical results (see ablation studies in Section 4.2). All models including the baselines are trained using a global batch size of 512 on 64 NVIDIA A100 GPUs for 1M steps. Other architecture and training details can be found in Appendix E.

**Evaluation datasets.** We use four representative datasets: COCO [14] and ImageNet [13] for natural images, CelebA [15] and ChartQA [16] for perceptually challenging images. Table 1 shows the compression ratio distributions.

Table 1: **Test data distribution.** CAT applies larger compression to natural images and smaller ratios to CelebA and ChartQA.

| Datasets | Method | 8x | 16x | 32x | Avg Latent Dim | Avg Rate |
|---|---|---|---|---|---|---|
| COCO | **CAT (Ours)** | 9% | 54% | 37% | 31.87 | 16.07x |
| | JPEG | 10% | 54% | 36% | 32.43 | 15.79x |
| ImageNet | **CAT (Ours)** | 6% | 49% | 45% | 29.32 | 17.46x |
| | JPEG | 9% | 49% | 42% | 31.24 | 16.39x |
| CelebA | **CAT (Ours)** | 17% | 83% | 0% | 39.29 | 13.03x |
| | JPEG | 0% | 0% | 100% | 16 | 32x |
| ChartQA | **CAT (Ours)** | 96% | 4% | 0% | 63.02 | 8.12x |
| | JPEG | 0% | 3% | 97% | 16.61 | 30.82x |

**Baselines.** We compare against fixed-token baselines that use the same VAE architecture but without the nested structure. We further study the effect of caption complexity by training a nested VAE using JPEG size as the complexity metric. All baselines have average 16x compression. To our knowledge, none of existing adaptive tokenizers (e.g., ElasticTok, ALIT) report quantitative results on the datasets we use, so we do not compare with them. ALIT only shows an rFID of 8.03 on ImageNet100. See Appendix E.3 for more baseline details.

### 4.1 Main Results

**Better reconstruction for complex images, higher efficiency for natural images.** Table 3 presents the reconstruction FID (rFID), LPIPS, and PSNR [61] on four datasets. Comparing CAT with the

Table 3: **Reconstruction results.** All models have latent channel $c = 16$.

| Average Compression | | | COCO | | | ImageNet | | | CelebA | | | ChartQA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | rFID↓ | LPIPS↓ | PSNR ↑ | rFID↓ | LPIPS↓ | PSNR ↑ | rFID↓ | LPIPS↓ | PSNR ↑ | rFID↓ | LPIPS↓ | PSNR ↑ |
| 8 | Fixed | 8x | 0.48 | 0.10 | 30.95 | 0.24 | 0.095 | 33.86 | 1.86 | 0.028 | 45.36 | 8.21 | 0.019 | 36.98 |
| 16 | Fixed | 16x | 0.66 | 0.16 | 29.79 | **0.38** | **0.15** | 30.45 | 2.25 | 0.059 | 41.84 | 8.67 | 0.029 | 33.48 |
| | Adaptive | JPEG | 0.72 | 0.17 | 30.11 | 0.51 | 0.16 | 30.61 | 6.57 | 0.14 | 36.47 | 10.17 | 0.048 | 31.54 |
| | Adaptive | **CAT (Ours)** | **0.65** | **0.15** | **30.19** | 0.46 | **0.15** | **30.62** | **1.97** | **0.051** | **42.43** | **5.27** | **0.021** | **36.45** |
| 32 | Fixed | 32x | 1.18 | 0.26 | 26.93 | 0.81 | 0.25 | 27.48 | 6.10 | 0.16 | 36.35 | 10.79 | 0.045 | 30.99 |

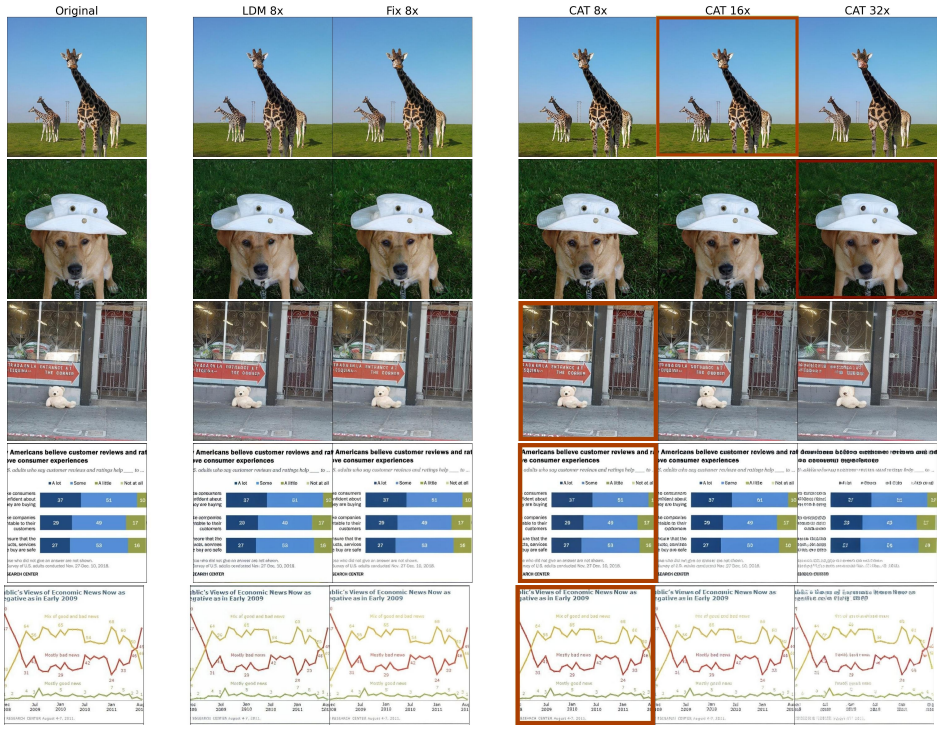



Figure 4: **Qualitative reconstruction examples.** We highlight the compression ratio selected by our caption complexity in red. On simpler images (top two rows), adjusting the compression ratio does not significantly affect quality. On more complex images (bottom three rows), the impact is substantial.

fixed 16x baseline, our method significantly outperforms the baselines across all metrics on CelebA and ChartQA, **improving the rFID by 12% on CelebA and 39% on ChartQA**. Our **ChartQA rFID even surpasses the fixed 8x baseline**, likely because the LLM evaluator can effectively identify rich visual details in these datasets and assign lower compression ratios accordingly (Table 1). On COCO and ImageNet, CAT generally outperforms the baselines, with only a slight drop in rFID on ImageNet. However, On ImageNet, CAT achieves an average compression ratio of 17.46x, which means we use **18% fewer tokens** to represent the dataset compared to the 16x baseline,

Table 2: ImageNet-512 reconstruction.

| ImageNet | rFID |
|---|---|
| MaskGIT-VQGAN [62] | 1.97 |
| TiTok-B-128 [8] | 1.52 |
| LFQ [3] | 1.22 |
| TexTok [38] | 0.73 |
| LDM [12] | 0.53 |
| CAT | 0.46 |

Figure 4 shows qualitative examples of progressive reconstruction quality as we manually increase the token count and reduce the compression ratio to represent each image. We highlight the compression ratio predicted by CAT in red. Different visual inputs need different ideal compression ratios. Natural images with fewer objects and simpler patterns can be accurately reconstructed at 32x, whereas complex images with visual details require lower compression. Thus, the caption-based CAT reconstruction has comparable quality to the fixed 16x baseline on natural images but surpasses it on text-heavy images. We include more visualization in Appendix E.4. We also report CAT's performance with uniform 16x compression in Appendix E.5. Lastly, while most prior works do not evaluate on diverse datasets like ChartQA, we include a side-by-side comparison with popular fixed-ratio tokenizers on ImageNet in Table 2. This table is for reference only due to differing training setups (we refer the readers to Table 3 for fully comparable baselines). Nonetheless, the results highlight that CAT achieves competitive performance by adapting compression ratios to image content.

**Caption complexity outperforms JPEG metric.** We further compare CAT against training the same adaptive architecture but using JPEG size as the complexity metric. Table 3 shows that **CAT achieves better rFID, LPIPS, and PSNR across all datasets**. While both tokenizers have similar training compression ratio distribution, the test-time compression ratio distribution varies significantly

Table 4: **Ablation on thresholds.** The $(4, 7)$-setting with more diverse training distribution achieves generally better performance.

| $(a, b)$ | COCO | ImageNet | CelebA | ChartQA |
|---|---|---|---|---|
| $(4, 7)$ | **0.65** | 0.46 | **1.97** | **5.27** |
| $(2, 8)$ | 0.67 | **0.43** | 2.58 | 7.70 |

Table 5: **Ablation on latent channel.** Increasing latent channel $c$ improves rFID across all four evaluated datasets.

| $c$ | COCO | ImageNet | CelebA | ChartQA |
|---|---|---|---|---|
| 4 | 1.66 | 1.10 | 5.83 | 9.13 |
| 8 | 1.03 | 0.60 | 4.54 | 7.95 |
| 16 | **0.65** | **0.46** | **1.97** | **5.27** |

Table 6: **More reconstruction and classification results.** CAT achieves better reconstruction and classification results.

| | DTD | EuroSAT | GTSRB | SUN397 | SVHN |
|---|---|---|---|---|---|
| *Reconstruction FID (↓)* | | | | | |
| LDM VAE 16x | 7.86 | 7.04 | 1.22 | 1.95 | 1.76 |
| Fix 16x | 7.29 | 6.26 | 1.38 | 1.95 | 2.11 |
| **CAT (Ours)** | **7.23** | **5.45** | **1.14** | **1.93** | **1.69** |
| *Linear Probing Top-1 Accuracy (%, ↑)* | | | | | |
| LDM VAE 16x | 53.81 | 75.38 | 70.29 | 63.70 | 64.87 |
| Fix 16x | 50.42 | 78.08 | 70.07 | 62.92 | 65.07 |
| **CAT (Ours)** | **54.51** | **78.21** | **71.28** | **64.16** | **66.39** |
| *Fine-Tuning Top-1 Accuracy (%, ↑)* | | | | | |
| LDM VAE 16x | **75.91** | 97.00 | 90.61 | **79.89** | 83.05 |
| Fix 16x | 71.96 | 92.46 | 95.07 | 78.77 | 85.93 |
| **CAT (Ours)** | 74.11 | **98.00** | **95.32** | 78.45 | **86.13** |

(Table 1). Notably, since JPEG size often cannot capture perceptually important factors (see discussion in Section 3.1), nearly all images in CelebA and ChartQA are assigned 32x compression. Thus, CAT significantly outperforms JPEG on these two datasets, showing the effectiveness of caption-based metric and LLM evaluation in determining image intrinsic complexity.

### 4.2 Ablation Studies

**Benefits of diverse compression ratios.** We explore several design choices for our tokenizer. First, we study how the distribution of compression ratios affects overall reconstruction. To achieve an average compression ratio of 16, we could set the thresholds $(a, b)$ to either $(4, 7)$ or $(2, 8)$. $(4, 7)$ yields a more diverse distribution of compression ratios, whereas $(2, 8)$ results in a concentrated distribution similar to a fixed 16x tokenizer, making it less interesting. Table 4 shows that $(4, 7)$ produces better reconstruction metrics across all datasets due to the diversity in compression ratios ensures that all parts of the model are fully trained. Hence, we adopt $(4, 7)$ as the thresholds for CAT.

**Effect of latent channel dimension.** We also vary the latent channel $c$ to study its effect. Table 5 shows that a larger $c$ improves reconstruction results. However, similar to previous work [12, 63], we observe a reconstruction-generation trade-off: while increasing $c$ is beneficial for reconstruction, it does not necessarily improve second-stage generative results. We elaborate more on this in Section 6.

## 5 Classification on Diverse Datasets

While CAT demonstrates strong reconstruction performance on COCO, ImageNet, CelebA, and ChartQA, these datasets represent only a subset of image domains. To thoroughly assess CAT's capabilities in diverse domains and tasks beyond reconstruction, we follow OpenCLIP [19] and test CAT's classification performance on five datasets that feature details beyond text and human faces: DTD [64] (textures), EuroSAT [52] (satellite images), GTSRB [17] (traffic signs), SUN397 [65] (indoor and outdoor scenes), and SVHN [18] (street numbers). We examine two settings: (1) linear probing, where we freeze the encoder and train only a linear layer on top of the latent features; and (2) fine-tuning both the encoder and classification head. See Appendix F for experiment details.

As shown in Table 6, CAT consistently outperforms fixed-ratio baselines across both settings, achieving the **best top-1 accuracy on all five datasets in the linear probing setup**. This highlights the quality and generalizability of CAT's latent representations, which transfer effectively to downstream classification tasks even without fine-tuning. In addition, CAT maintains strong reconstruction performance on these datasets, **improving the rFID by 13% on EuroSAT, 17% on GTSRB, and 20% on SVHN** (Table 6, top rows). These results further confirm that our caption-based complexity metric supports learning representations that balance both compression quality and downstream utility.

## 6 Image Generation

Lastly, we evaluate CAT on text-to-image generation to show that (1) its adaptive design does not compromise generation quality; (2) by using fewer tokens to represent the training data, CAT actually enables more efficient learning, leading to stronger generative models under the same compute budget.

**Setup.** Given CAT's ability to produce variable-length token sequences, we integrate it with Diffusion Transformer (DiT) [20], which handles adaptive token representations naturally. DiT

Table 7: **ImageNet-512 generation results**. All tokenizers have average 16x compression ratio. "rFLOPs" means relative FLOPs.

| | | FID↓ | sFID↓ | IS↑ | Precision↑ | Recall↑ | Eval rFLOPs↓ |
|---|---|---|---|---|---|---|---|
| Fixed | DiT + LDM VAE | 10.03 | 16.88 | 114.84 | 0.65 | **0.50** | 1× |
| | DiT + Fixed 16x | 4.78 | 11.81 | 187.47 | 0.72 | 0.49 | 1× |
| Adaptive | DQ-Transformer [37] | 5.11 | - | 178.2 | - | - | - |
| | **DiT + CAT (Ours)** | **4.56** | **10.55** | **191.09** | **0.75** | 0.49 | **0.82**× |

Table 8: Larger channel $c$ is not always better for generation.

| $c$ | FID↓ | sFID↓ | IS↑ | Precision↑ | Recall↑ |
|---|---|---|---|---|---|
| 4 | 5.12 | 11.12 | 152.39 | 0.72 | 0.48 |
| 8 | **4.38** | **10.31** | 181.03 | **0.76** | 0.48 |
| 16 | 4.56 | 10.55 | **191.09** | 0.75 | **0.49** |

takes the noised latent features as input, applies patching for downsampling, and uses a transformer architecture to predict the added noise. Following [20], we work with class-conditional generation on ImageNet-512, leveraging DiT-XL with a patch size of 2.

During training, each image is processed through the CAT pipeline to determine its compression ratio, resulting in variable-length latent representations tailored to its complexity. That is, *each example can have different number of latent tokens*. For inference, we obtain a class-level compression ratio by providing a textual description of the form "*this is an image of [label]*" to the LLM evaluator, which predicts the appropriate compression ratio for that class. For example, if the evaluator suggests a 16x compression, we generate $(\frac{512}{16 \cdot 2})^2 = 256$ tokens, where the 2 in the denominator accounts for DiT's patching. The generated tokens are then decoded by CAT to reconstruct 512x512 images. Following [20], we report FID [66], Sliding FID [67], Inception Score [68], precision and recall [69] on 50K images generated with 250 DDPM steps and classifier-free guidance [70]. We note that we use the same compression ratio per class predicted by LLMs mainly for benchmarking purpose. In reality, CAT allows users to flexibly specify the desired token count at inference time, as we will show later.

**Baselines.** We compare against DiT-XL paired with the open-source 16x LDM VAE and the fixed 16x tokenizer we trained ourselves. All models are trained on 16 NVIDIA H100 GPUs for 400K steps, using a global token batch size of 262,144, which is equivalent to 1024 images at 16x compression. See Appendix G for more implementation and baselinedetails.

As shown in Table 7, DiT-CAT achieves the **best FID, sFID, IS, and precision** among all DiT baselines trained with the same computational resources. We attribute this performance to two factors. First, using fewer tokens for simpler images improves processing efficiency, allowing for more extensive training within the same computational budget. In fact, the average token count per *training* image for DiT-CAT is 197.44. Compared to the 256 tokens used by fixed 16x tokenizers, CAT achieves a **23% reduction in token count**, allowing the model to process more images within the same training budget. Second, adaptively allocating representation capacity also enables more effective modeling of complex images, as richer visual details are better preserved through the use of additional tokens.

As discussed earlier, we use our LLM evaluator to obtain the generation token count using pre-defined captions mainly because we want to automate the evaluation process. In reality, users can flexibly set the token count based on their computational budget. To explore this, we manually vary the token count during generation with DiT-CAT and observe that FID-50K improves from 5.83 (64 tokens) to 5.02 (256 tokens) and 4.12 (1024 tokens), confirming that more tokens lead to higher-quality images. Qualitative examples in Figure 5 further support this observation. Thus, CAT enables a controllable trade-off between efficiency and generation quality. For more visualization, see Appendix G.4.

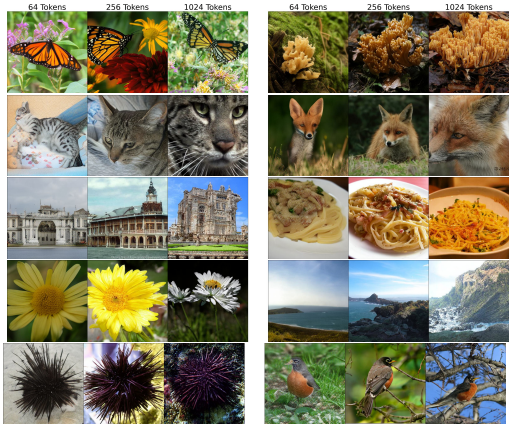


Figure 5: Increasing token count (left→right) for CAT leads to better image quality and higher complexity.

Lastly, recall that we trained tokenizers with different latent channels in Section 4.1. Table 8 shows the generation performance. While larger $c$ is better for reconstruction, it does not benefit generation. CAT with $c = 8$ achieves the best FID across all experiments. This observation agrees with prior work [12] and highlights the importance of choosing an appropriate $c$. We leave exploring $c$'s impact on downstream task for future work.

## 7 Conclusion

We propose an adaptive image tokenizer, CAT, that allocates different number of tokens to images based on the content complexity derived from text description. Our experiments show that CAT improves the quality and efficiency of image representation on a variety of downstream tasks.

**Limitations and Future Work.** Nested VAE is a natural and effective extension of the VAE architecture but is constrained to predefined compression ratios that scale by factors of 2. An intriguing direction for future work would be to enable more flexible compression ratios by transitioning to transformer backbones and dynamically adjusting token counts. Besides, images contain diverse global and local information. While CAT addresses global complexity by increasing token allocation for images with intricate details, further efficiency improvements could come through local tokenization—allocating more tokens specifically to detailed regions while reducing tokens for simpler areas. Besides, an ideal pipeline would enable LLMs to automatically identify perception-critical elements without relying on predefined queries. This would be an important next step for our work.

## References


[1] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2020.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[3] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024. URL `https://arxiv.org/abs/2310.05737`.

[4] Junhong Shen, Mikhail Khodak, and Ameet Talwalkar. Efficient architecture search for diverse tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[5] Renbo Tu, Nicholas Roberts, Mikhail Khodak, Junhong Shen, Frederic Sala, and Ameet Talwalkar. NAS-bench-360: Benchmarking neural architecture search on diverse tasks. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2022.

[6] Luyao Yuan, Dongruo Zhou, Junhong Shen, Jingdong Gao, Jeffrey L Chen, Quanquan Gu, Ying Nian Wu, and Song-Chun Zhu. Iterative teacher-aware learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29231–29245. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/f48c04ffab49ff0e5d1176244fdfb65c-Paper.pdf`.

[7] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple, 2023. URL `https://arxiv.org/abs/2309.15505`.

[8] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *arxiv: 2406.07550*, 2024.

[9] G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. doi: 10.1109/30.125072.

[10] Wilson Yan, Matei Zaharia, Volodymyr Mnih, Pieter Abbeel, Aleksandra Faust, and Hao Liu. Elastictok: Adaptive tokenization for image and video. *arXiv preprint*, 2024.

[11] Shivam Duggal, Phillip Isola, Antonio Torralba, and William T. Freeman. Adaptive length image tokenization via recurrent allocation, 2024. URL `https://arxiv.org/abs/2411.02393`.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL `https://arxiv.org/abs/1405.0312`.

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[16] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, 2022. URL `https://arxiv.org/abs/2203.10244`.

[17] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pages 1453–1460, 2011. doi: 10.1109/IJCNN.2011.6033395.

[18] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[19] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773. If you use this software, please cite it as below.

[20] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[21] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL https://arxiv.org/abs/1711.00937.

[22] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11513–11522, 2022. URL https://api.semanticscholar.org/CorpusID:247244535.

[23] Chuanxia Zheng, Long Tung Vuong, Jianfei Cai, and Dinh Q. Phung. Movq: Modulating quantized vectors for high-fidelity image generation. *ArXiv*, abs/2209.09002, 2022. URL https://api.semanticscholar.org/CorpusID:252367709.

[24] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *ArXiv*, abs/2110.04627, 2021. URL https://api.semanticscholar.org/CorpusID:238582653.

[25] Shiyue Cao, Yueqin Yin, Lianghua Huang, Yu Liu, Xin Zhao, Deli Zhao, and Kaiqi Huang. Efficient-vqgan: Towards high-resolution image generation with efficient vision transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7334–7343, 2023. URL https://api.semanticscholar.org/CorpusID:263830997.

[26] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7): 560–576, 2003. doi: 10.1109/TCSVT.2003.815165.

[27] Lei Chen, Zhan Tong, Yibing Song, Gangshan Wu, and Limin Wang. Efficient video action detection with token dropout and context refinement, 2023. URL https://arxiv.org/abs/2304.08451.

[28] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[29] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[30] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

[31] Junhong Shen and Lin F. Yang. Theoretically principled deep rl acceleration via nearest neighbor function approximation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35 (11):9558–9566, May 2021. doi: 10.1609/aaai.v35i11.17151. URL https://ojs.aaai.org/index.php/AAAI/article/view/17151.

[32] Junhong Shen, Atishay Jain, Zedian Xiao, Ishan Amlekar, Mouad Hadji, Aaron Podolny, and Ameet Talwalkar. Scribeagent: Towards specialized web agents using production-scale workflow data, 2024. URL https://arxiv.org/abs/2411.15004.

[33] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M. Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey A. Gritsenko, Mario Luvci'c, and Neil Houlsby. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. *ArXiv*, abs/2307.06304, 2023. URL `https://api.semanticscholar.org/CorpusID:259837358`.

[34] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL `https://arxiv.org/abs/2010.11929`.

[35] Tomer Ronen, Omer Levy, and Avram Golbert. Vision transformers with mixed-resolution tokenization, 2023. URL `https://arxiv.org/abs/2304.00287`.

[36] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction, 2024. URL `https://arxiv.org/abs/2404.02905`.

[37] Mengqi Huang, Zhendong Mao, Zhuowei Chen, and Yongdong Zhang. Towards accurate image coding: Improved autoregressive image generation with dynamic vector quantization. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22596–22605, 2023. URL `https://api.semanticscholar.org/CorpusID:258823089`.

[38] Kaiwen Zha, Lijun Yu, Alireza Fathi, David A. Ross, Cordelia Schmid, Dina Katabi, and Xiuye Gu. Language-guided image tokenization for generation, 2024. URL `https://arxiv.org/abs/2412.05796`.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL `https://arxiv.org/abs/1505.04597`.

[40] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, December 2022.

[41] Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. *arXiv preprint arXiv:2405.17430*, 2024.

[42] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Josh Susskind, and Navdeep Jaitly. Matryoshka diffusion models, 2024. URL `https://arxiv.org/abs/2310.15111`.

[43] Charlie Nash, João Carreira, Jacob Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models, 2022. URL `https://arxiv.org/abs/2203.09494`.

[44] Junhong Shen, Liam Li, Lucio M. Dery, Corey Staten, Mikhail Khodak, Graham Neubig, and Ameet Talwalkar. Cross-modal fine-tuning: align then refine. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[45] Junhong Shen, Tanya Marwah, and Ameet Talwalkar. Ups: Towards foundation models for pde solving via cross-modal adaptation. *arXiv preprint arXiv:2403.07187*, 2024.

[46] Nicholas Roberts, Samuel Guo, Cong Xu, Ameet Talwalkar, David Lander, Lvfang Tao, Linhang Cai, Shuaicheng Niu, Jianyu Heng, Hongyang Qin, Minwen Deng, Johannes Hog, Alexander Pfefferle, Sushil Ammanaghatta Shivakumar, Arjun Krishnakumar, Yubo Wang, Rhea Sanjay Sukthanker, Frank Hutter, Euxhen Hasanaj, Tien-Dung Le, Mikhail Khodak, Yuriy Nevmyvaka, Kashif Rasul, Frederic Sala, Anderson Schneider, Junhong Shen, and Evan R. Sparks. Automl decathlon: Diverse tasks, modern methods, and efficiency at scale. In *Neural Information Processing Systems*, 2021. URL `https://api.semanticscholar.org/CorpusID:265536645`.

[47] Luyao Yuan, Zipeng Fu, Jingyue Shen, Lu Xu, Junhong Shen, and Song-Chun Zhu. Emergence of pragmatics from referential game between theory of mind agents, 2021. URL https://arxiv.org/abs/2001.07752.

[48] Junhong Shen, Neil Tenenholtz, James Brian Hall, David Alvarez-Melis, and Nicolo Fusi. Tag-llm: Repurposing general-purpose llms for specialized domains, 2024.

[49] Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models, 2024.

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[51] Stability AI. sd-vae-ft-mse. https://huggingface.co/stabilityai/sd-vae-ft-mse, 2022.

[52] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

[53] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[54] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. URL https://arxiv.org/abs/2305.06500.

[55] An Yang, ..., and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

[56] Abhimanyu Dubey, ..., and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[57] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

[59] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. URL https://arxiv.org/abs/1911.05722.

[60] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

[61] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. doi: 10.1109/ICPR.2010.579.

[62] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11305–11315, 2022. URL https://api.semanticscholar.org/CorpusID:246680316.

[63] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, Matthew Yu, Abhishek Kadian, Filip Radenovic, Dhruv Mahajan, Kunpeng Li, Yue Zhao, Vladan Petrovic, Mitesh Kumar Singh, Simran Motwani, Yi Wen, Yiwen Song, Roshan Sumbaly, Vignesh Ramanathan, Zijian He, Peter Vajda, and Devi Parikh. Emu: Enhancing image generation models using photogenic needles in a haystack, 2023. URL https://arxiv.org/abs/2309.15807.

[64] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. doi: 10.1109/CVPR.2014.461.

[65] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *Int. J. Comput. Vision*, 119(1): 3–22, August 2016. ISSN 0920-5691. doi: 10.1007/s11263-014-0748-y. URL `https://doi.org/10.1007/s11263-014-0748-y`.

[66] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL `https://arxiv.org/abs/1706.08500`.

[67] Xin Ding, Yongwei Wang, Zuheng Xu, William J. Welch, and Z. Jane Wang. Continuous conditional generative adversarial networks: Novel empirical losses and label input mechanisms, 2023. URL `https://arxiv.org/abs/2011.07466`.

[68] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. URL `https://arxiv.org/abs/1606.03498`.

[69] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019. URL `https://arxiv.org/abs/1904.06991`.

[70] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL `https://arxiv.org/abs/2207.12598`.

[71] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. URL `https://arxiv.org/abs/1812.04948`.

[72] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128 (7):1956–1981, March 2020. ISSN 1573-1405. doi: 10.1007/s11263-020-01316-z. URL `http://dx.doi.org/10.1007/s11263-020-01316-z`.

[73] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[74] Zongzhe Xu, Ritvik Gupta, Wenduo Cheng, Alexander Shen, Junhong Shen, Ameet Talwalkar, and Mikhail Khodak. Specialized foundation models struggle to beat supervised baselines, 2024. URL `https://arxiv.org/abs/2411.02796`.

[75] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *ArXiv*, abs/2208.05592, 2022.

# A Broader Impact

This work introduces CAT, a content-adaptive tokenizer designed to improve the efficiency and flexibility of image modeling. By enabling dynamic compression based on image content, CAT has the potential to reduce computational and energy costs in large-scale generative and classification systems, contributing to more sustainable AI practices. However, as with any generative model, misuse of this technology for creating misleading or harmful content remains a concern. We encourage responsible deployment, including proper safeguards and usage guidelines, particularly in sensitive domains such as media generation, surveillance, or automated decision-making.

# B Prompt for LLM Scorer

Our caption complexity pipeline works as follows:

Step 1: Use a VLM to generate image description, with the following prompts:

- What's in the image? $\rightarrow$ Caption
- Are there text or numbers in the image? $\rightarrow$ Yes/No.
- Are there faces in the image? $\rightarrow$ Yes/No.

Step 2: Use the same VLM or a separate LLM to generate the complexity score with the prompt:

Given the description of a 512px image, determine its complexity based on the following factors:
1. Number of distinct objects
2. Color variance
3. Texture complexity
4. Foreground and background
5. Symmetry and repetition
6. Human perception factors, like the presence of human faces or text
You will be given the caption, whether there are text or numbers, and whether there are faces in the image. Assign a complexity score such that a higher number means the image is more complex. Note that text and facial details are intrinsically complex because they are crucial to human perception. Here are some examples for scoring:
- Score 1: A plane in a sky
- Score 2: A t-shirt with a emoji on it
- Score 3: A dog lying on the grass
- Score 4: A woman skiing in the snow
- Score 5: Two kids walking on the beach
- Score 6: A dinning table full of food
- Score 7: A close-up shot of a old man
- Score 8: Many people gathering in the stadium
- Score 9: Newspapers or graphs with text and numbers
Now determine the complexity for the caption:
[*Insert caption here*]
[*Insert one of the following based on the Yes/No questions:*
*- There are text visible in the image. There are also facial details.*
*- There are text visible in the image, but there is no human face.*
*- There is no obvious text in the image, but there are facial details.*
*- There is no text or human face in the image.* ]
Respond with "Score: ? out of 9", where "?" is a number between 1 and 9. Then provide explanations.

Note these two steps can be combined into a single inference call.

# C Other Complexity Metrics

Table 9: We also tried various other metrics. However, they are less effective (e.g., caption length) compared to our LLM-based score.

| Metric | Pearson $r$ |
|---|---|
| JPEG | 0.31 |
| MSE | 0.36 |
| LPIPS | 0.23 |
| Caption Length | 0.33 |
| CAT (Ours) | **0.55** |

# D Compression Ratio Distributions with Different LLMs

Table 10: To study whether our caption score is robust to LLM choice, we test multiple captioners and LLMs for scoring. The fact that these combinations generate similar score and compression ratio distributions show that our scoring method is robust.

| | Scoring Model | 8x | 16x | 32x | Avg Rate |
|---|---|---|---|---|---|
| ImageNet | InstructBLIP + Llama3.1-8B-Instruct | 6% | 49% | 45% | 17.43 |
| | InstructBLIP + Qwen2.5-7B-Instruct | 2% | 70% | 28% | 17.35 |
| | LLaVA1.5 7B | 2% | 65% | 33% | 17.75 |
| | LLaVA1.5 7B (Longer Caption) | 3% | 62% | 35% | 17.58 |
| CelebA | InstructBLIP + Llama3.1-8B-Instruct | 17% | 83% | 0% | 13.02 |
| | InstructBLIP + Qwen2.5-7B-Instruct | 15% | 70% | 15% | 13.83 |
| | LLaVA1.5 7B | 5% | 95% | 0% | 14.92 |
| | LLaVA1.5 7B (Longer Caption) | 6% | 94% | 0% | 14.72 |
| DTD | InstructBLIP + Llama3.1-8B-Instruct | 0% | 40% | 60% | 21.57 |
| | InstructBLIP + Qwen2.5-7B-Instruct | 0% | 38% | 62% | 21.87 |
| | LLaVA1.5 7B | 0% | 41% | 59% | 21.42 |
| | LLaVA1.5 7B (Longer Caption) | 2% | 44% | 54% | 19.76 |
| EuroSAT | InstructBLIP + Llama3.1-8B-Instruct | 3% | 30% | 67% | 20.87 |
| | InstructBLIP + Qwen2.5-7B-Instruct | 3% | 25% | 72% | 21.57 |
| | LLaVA1.5 7B | 2% | 22% | 76% | 22.85 |
| | LLaVA1.5 7B (Longer Caption) | 4% | 23% | 72% | 21.19 |
| SUN397 | InstructBLIP + Llama3.1-8B-Instruct | 6% | 73% | 21% | 15.82 |
| | InstructBLIP + Qwen2.5-7B-Instruct | 2% | 78% | 20% | 16.77 |
| | LLaVA1.5 7B | 3% | 80% | 17% | 16.30 |
| | LLaVA1.5 7B (Longer Caption) | 5% | 82% | 13% | 15.59 |

# E Reconstruction Experiments

## E.1 Architecture

We implement the nested VAE similar to the `AutoencoderKL` implementation in the `diffusers` library. The network configuration is:

- `sample_size`: 512
- `in_channels`: 3
- `out_channels`: 3
- `down_block_types`: [`DownEncoderBlock2D`] $\times$ 6
- `up_block_types`: [`UpDecoderBlock2D`] $\times$ 6
- `block_out_channels`: [64, 128, 256, 256, 512, 512]
- `layers_per_block`: 2
- `act_fn`: `silu`
- `latent_channels`: 4/8/16
- `norm_num_groups`: 32
- `mid_block_attention_head_dim`: 1
- `num_layers`: 8

The model sizes for different latent channels are shown below. For the discriminator, we use the pretrained StyleGAN [71].

| Nested VAE | $c = 4$ | $c = 8$ | $c = 16$ |
|---|---|---|---|
| # Params (M) | 187.45 | 187.50 | 187.61 |

### E.2 Training

We use *obj* $\in \{$*human faces*, *text*$\}$ for our perception-focused queries. We use the following training configuration:

- GPU: 64 NVIDIA A100
- Per-GPU batch size: 8
- Global batch size: 512
- Training steps: 1,000,000
- Optimizer: AdamW
  - `lr`: 0.0001
  - `beta1`: 0.9
  - `beta2`: 0.95
  - `weight_decay`: 0.1
  - `epsilon`: 1e-8
  - `gradient_clip`: 5.0
- Scheduler: constant with 10,000 warmup steps
- Loss:
  - `recon_loss_weight`: 1.0
  - `kl_loss_weight`: 1e-6
  - `perceptual_loss_weight`: 1.0
  - `moco_loss_weight`: 0.2
  - `gan_loss_weight`: 0.5
  - `gan_loss_starting_step`: 50,000

The discriminator is trained with the standard GAN loss.

### E.3 Baselines

We train fixed compression baselines using the same data, training configuration, and VAE backbone. For smaller compression ratios, e.g., fixed 8x, the last two downsampling blocks and first two upsampling blocks are not used.

For the adaptive JPEG baseline, we use torchvision.io.encode_jpeg to transform the images into JPEG file and compute the number of bytes as the complexity metric. Smaller files correspond to larger complexity. To provide a better understanding of this metric, we show in Figure 6 the distribution of JPEG sizes on the COCO 2014 test set, with relevant statistics included in the caption. Then, based on the JPEG sizes of all images in the Shutterstock training dataset, we set the thresholds $(a, b)$ to $(38761, 65837)$ to categorize the file sizes into three compression ratios. This set of thresholds ensure that the JPEG baseline has the same training compression ratio distribution as CAT.

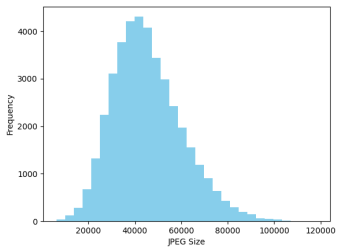



Figure 6: On COCO 2014 test set, the minimum JPEG size is 6128; maximum is 118428; mean is 45474.29; standard deviation is 15037.07.

For LDM VAEs, we follow the instructions in their original repository to use the model checkpoints. Note that LDM VAEs are trained on OpenImages dataset [72], which is different from our training data, so it is hard to fairly compare the reconstruction performance. Nonetheless, we present their rFIDs on the evaluation datasets in Table 11.

Table 11: rFIDs for CAT and LDM VAEs.

| | COCO | ImageNet | CelebA | ChartQA |
|---|---|---|---|---|
| CAT | 0.65 | 0.46 | 1.97 | 5.27 |
| LDM 8x | 0.51 | 0.33 | 2.83 | 8.32 |
| LDM 16x | 0.53 | 0.37 | 3.07 | 8.49 |
| LDM 32x | 0.90 | 0.62 | 5.54 | 10.35 |

### E.4 More Reconstruction Visualization

See Figure 7 in the next page.

### E.5 Fixing Token Count for CAT

We also evaluate the reconstruction performance under fixed compression ratio (token count) for different datasets. Table 12 compares the reconstruction FID for CAT with caption-guided compression ratio vs. fixed 16x compression ratio. We see that adaptive compression based on image complexity outperforms uniform compression using the same architecture in most cases, possibly because error reduction on complex images outweighs the slight error increase on simpler ones.

Table 12: Equalizing test-time token counts.

| CAT rFID | Adaptive | Equalized 16x |
|---|---|---|
| COCO | **0.65** | 0.67 |
| ImageNet | 0.46 | **0.40** |
| CelebA | **1.97** | 2.47 |
| ChartQA | **5.27** | 7.27 |

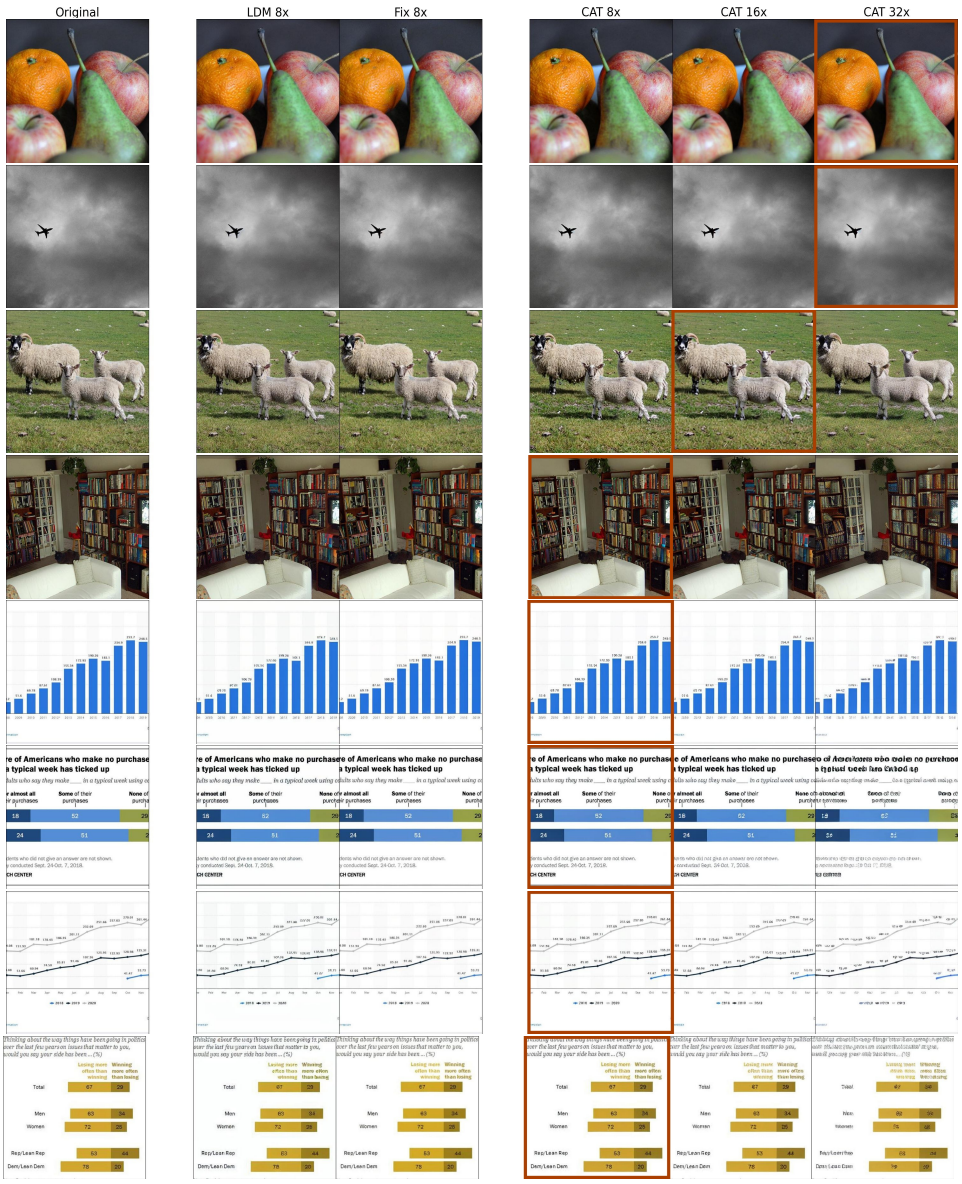



Figure 7: **More CAT reconstruction examples. We highlight the compression ratio selected by our caption complexity in red.**

### E.6 Full Results for Table 4 and Table 5

To complement Table 4, we include the training distribution for different scoring thresholds in Table 13. To complement Table 5, we include the results of fixed-ratio baseline in Table 14.

Table 13: **Compression ratio distribution affects learning outcomes.** Both settings have an average compression of $\sim$16x, but $(4, 7)$ leads to better distribution diversity and empirical results.

| $(a, b)$ | Training Distribution | | | | Reconstruction FID ↓ | | | |
|---|---|---|---|---|---|---|---|---|
| | 8x | 16x | 32x | Average | COCO | ImageNet | CelebA | ChartQA |
| $(4, 7)$ | 10% | 48% | 42% | 16.0x | **0.65** | 0.46 | **1.97** | **5.27** |
| $(2, 8)$ | 0.5% | 89.5% | 10% | 16.5x | 0.67 | **0.43** | 2.58 | 7.70 |

Table 14: **Larger latent channel $c$ generally improves rFID.**

| rFID↓ | $c$ | COCO | ImageNet | CelebA | ChartQA |
|---|---|---|---|---|---|
| Fixed 16x | 4 | 1.25 | 1.32 | 5.89 | 9.45 |
| | 8 | 1.10 | 0.61 | 4.99 | **8.19** |
| | 16 | **0.66** | **0.38** | **2.25** | 8.67 |
| CAT | 4 | 1.66 | 1.10 | 5.83 | 9.13 |
| | 8 | 1.03 | 0.60 | 4.54 | 7.95 |
| | 16 | **0.65** | **0.46** | **1.97** | **5.27** |

# F Classification Experiments

We selected the diverse datasets because they represent out-of-domain distributions where zero-shot models perform poorly [73–75]. Similar to OpenClip [19], we initialize a simple linear classification head that maps from the tokenizer's maximum latent dimension (under 8x compression) to the number of labels for each task. When the image is compressed with 16x or 32x ratio, we zero-pad the latent features to make the length match with the classification head. We evaluate two settings: (1) linear probing, where we keep the image encoder frozen and only fine-tune the classification head; (2) full fine-tuning, where we update both the encoder and the classification head. We train both settings for 20 epochs with a batch size of 64, learning rate 1e-4 and a cosine annealing learning rate schedule with 2 warm-up epochs. We use the AdamW optimizer with weight decay 0.1.

# G Generation Experiments

## G.1 Architecture

We use DiT-XL architecture with a patchify downsampler and patch size of 2. The model size depends on the latent channel, but is generally around 431M parameters. The model TFLOPs is 22.0. All baselines reported in Table 7 use $c = 16$.

## G.2 Training & Inference

We use LLaVA1.5 7B to generate the complexity score for ImageNet training images. For 10 % of the time, we remove the image class label from the input and train unconditional image generation. The training configuration for DiT is:

- GPU: 16 NVIDIA H100
- Per-GPU token batch size: $4096 \times 4$ (equivalent to 64 images for 16x compression ratio)
- Global token batch size: $4096 \times 64$
- Training steps: 400,000
- Optimizer: AdamW
  - `lr`: 0.0001
  - `beta1`: 0.9
  - `beta2`: 0.95
  - `weight_decay`: 0.1
  - `epsilon`: 1e-8
  - `gradient_clip`: 1.0
- Scheduler: Cosine
  - `warmup`: 4000
  - `cosine_theta`: 1.0
  - `cycle_length`: 1.0
  - `lr_min_ratio`: 0.05

At test time, we use "*this is an image of [label]*" as a standardized prompt and manually provide answers to the queries to enable automated evaluation. Then, we use Llama3.1 to obtain the complexity score. DDPM scheduler (`diffusers` implementation) configuration is:

- `num_train_timesteps`: 1000
- `beta_start`: 0.0001
- `beta_end`: 0.02
- `beta_schedule`: `squaredcos_cap_v2`
- `prediction_type`: `epsilon`
- `timestep_spacing`: `leading`
- `num_inference_steps`: 250

All FID-50K and images generated in this paper are using cfg=1.5.

### G.3 Baselines

To ensure we train the baseline with the same compute FLOPs, we fix the token batch size and number of training steps for all settings. For pretrained LDM VAE, we use the scaling factor specified in the model configuration to ensure the input scale and noise scale are similar. For CAT, we use a scaling factor of 1.

### G.4 More Visualization

See Figure 8 in the end.

### G.5 Full Results for Table 8

To complement Table 8, we include the results of fixed-ratio baseline in Table 15.

Table 15: **Larger channel $c$ is not always better for generation.** Contrary to Table 5, we find that increasing channel dimension does not always result in generation gains.

| | $c$ | FID↓ | sFID↓ | IS↑ | Precision↑ | Recall↑ |
|---|---|---|---|---|---|---|
| Fixed 16x | 4 | 5.11 | 10.84 | 158.80 | 0.75 | 0.49 |
| | 8 | 4.96 | **10.39** | **221.85** | **0.76** | **0.51** |
| | 16 | **4.78** | 11.81 | 187.47 | 0.72 | 0.49 |
| CAT | 4 | 5.12 | 11.12 | 152.39 | 0.72 | 0.48 |
| | 8 | **4.38** | **10.31** | 181.03 | **0.76** | 0.48 |
| | 16 | 4.56 | 10.55 | **191.09** | 0.75 | **0.49** |

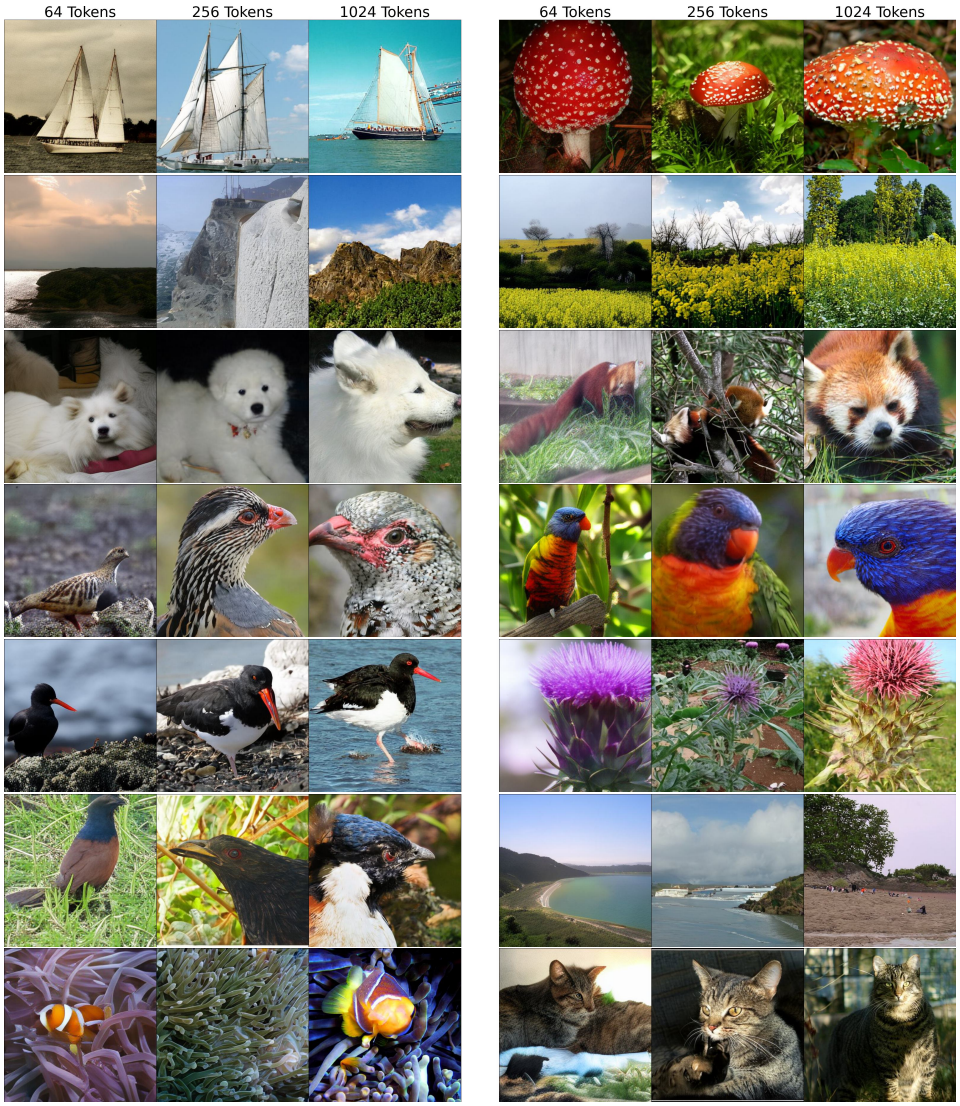



Figure 8: **More DiT-CAT generation examples. Increasing token count (left→right) generally leads to better image quality and higher complexity.**

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contribution is to propose a dynamic tokenizer based on image complexity. These are reflected in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the Conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There is no theoretical result in this paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all training and inference configurations, hyperprameters, LLM prompts in the main text and appendix. We also include the code in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case

of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use open-source benchmarks and release our code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, see the appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report numbers for one trail due to the cost associated with each experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we specify the GPUs we used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper follows the NeurIPS Code of Ethics.

   Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Broader Impact section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not use data or train models with high risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The datasets are cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We discussed how we prompt LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.