3 Method

3.1 Preliminary: autoregressive modeling via next-token prediction

Formulation. Consider a sequence of discrete tokens $x = (x_1, x_2, \dots, x_T)$, where $x_t \in [V]$ is an integer from a vocabulary of size V . The next-token autoregressive posits the probability of observing the current token x_t depends only on its prefix $(x_1, x_2, \dots, x_{t-1})$. This **unidirectional token dependency assumption** allows for the factorization of the sequence x 's likelihood:

$$p(x_1, x_2, \dots, x_T) = \prod_{t=1}^{T} p(x_t | x_1, x_2, \dots, x_{t-1}). \qquad (1)$$

Training an autoregressive model p_θ involves optimizing $p_\theta(x_t | x_1, x_2, \dots, x_{t-1})$ over a dataset. This is known as the ·next-token prediction·, and the trained p_θ can generate new sequences.

Tokenization. Images are inherently 2D continuous signals. To apply autoregressive modeling to images via next-token prediction, we must: 1) tokenize an image into several *discrete* tokens, and 2) define a 1D *order* of tokens for unidirectional modeling. For 1), a quantized autoencoder such as [30] is often used to convert the image feature map $f \in \mathbb{R}^{h \times w \times C}$ to discrete tokens $q \in [V]^{h \times w}$:

$$f = \mathcal{E}(im), \quad q = \mathcal{Q}(f), \qquad (2)$$

where im denotes the raw image, $\mathcal{E}(\cdot)$ a encoder, and $\mathcal{Q}(\cdot)$ a quantizer. The quantizer typically includes a learnable codebook $Z \in \mathbb{R}^{V \times C}$ containing V vectors. The quantization process $q = \mathcal{Q}(f)$ will map each feature vector $f^{(i,j)}$ to the code index $q^{(i,j)}$ of its nearest code in the Euclidean sense:

$$q^{(i,j)} = \left(\arg \min_{v \in [V]} \| \text{lookup}(Z, v) - f^{(i,j)} \|_2 \right) \in [V], \qquad (3)$$

where lookup(Z, v) means taking the v -th vector in codebook Z . To train the quantized autoencoder, Z is looked up by every $q^{(i,j)}$ to get $\hat{f}$, the approximation of original f . Then a new image $\hat{im}$ is reconstructed using the decoder $\mathcal{D}(\cdot)$ given $\hat{f}$, and a compound loss $\mathcal{L}$ is minimized:

$$\hat{f} = \text{lookup}(Z, q), \qquad \hat{im} = \mathcal{D}(\hat{f}), \qquad (4)$$

$$\mathcal{L} = \|im - \hat{im}\|_2 + \|f - \hat{f}\|_2 + \lambda_P \mathcal{L}_P(\hat{im}) + \lambda_G \mathcal{L}_G(\hat{im}), \qquad (5)$$

where $\mathcal{L}_P(\cdot)$ is a perceptual loss such as LPIPS [97], $\mathcal{L}_G(\cdot)$ a discriminative loss like StyleGAN's discriminator loss [46], and λ_P, λ_G are loss weights. Once the autoencoder $\{\mathcal{E}, \mathcal{Q}, \mathcal{D}\}$ is fully trained, it will be used to tokenize images for subsequent training of a unidirectional autoregressive model.

The image tokens in $q \in [V]^{h \times w}$ are arranged in a 2D grid. Unlike natural language sentences with an inherent left-to-right ordering, the order of image tokens must be explicitly defined for unidirectional autoregressive learning. Previous AR methods [30, 92, 50] flatten the 2D grid of q into a 1D sequence $x = (x_1, \dots, x_{h \times w})$ using some strategy such as row-major raster scan, spiral, or z-curve order. Once flattened, they can extract a set of sequences x from the dataset, and then train an autoregressive model to maximize the likelihood in (1) via next-token prediction.

Discussion on the weakness of vanilla autoregressive models. The above approach of tokenizing and flattening enable next-token autoregressive learning on images, but introduces several issues:

1) **Mathematical premise violation.** In quantized autoencoders (VQVAEs), the encoder typically produces an image feature map f with inter-dependent feature vectors $f^{(i,j)}$ for all i, j . So after quantization and flattening, the token sequence $(x_1, x_2, \dots, x_{h \times w})$ retains *bidirectional* correlations. This contradicts the *unidirectional* dependency assumption of autoregressive models, which dictates that each token x_t should only depend on its prefix $(x_1, x_2, \dots, x_{t-1})$.

2) **Inability to perform some zero-shot generalization.** Similar to issue 1), The unidirectional nature of image autoregressive modeling restricts their generalizability in tasks requiring bidirectional reasoning. E.g., it cannot predict the top part of an image given the bottom part.

3) **Structural degradation.** The flattening disrupts the spatial locality inherent in image feature maps. For example, the token $q^{(i,j)}$ and its 4 immediate neighbors $q^{(i \pm 1,j)}$, $q^{(i,j \pm 1)}$ are closely correlated due to their proximity. This spatial relationship is compromised in the linear sequence x , where *unidirectional* constraints diminish these correlations.