NEXT-DENSE-STRIDE PREDICTION FOR MULTIMODAL AUTOREGRESSIVE VISUAL MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce DenseAR, a new generative paradigm that reformulates autoregressive image generation as coarse-to-fine next-dense-stride prediction using a compact single-scale tokenizer. Our key insight is that the quality gain of coarse-to-fine generation needs no multi-resolution tokenizer—the prediction order on one compact grid can provide it. This addresses two limitations of existing autoregressive models at once: the slow inference of raster-order autoregression, which DenseAR avoids by predicting multiple tokens in parallel, and the heavy cost of multi-scale approaches, which need long, multi-resolution token sequences to achieve coarse-to-fine prediction. Building on our efficient framework and the flexibility of autoregressive modeling, we further extend DenseAR to a unified model that handles multiple modalities and imaging tasks within a single backbone. We validate DenseAR on both medical and natural images. On multi-contrast brain MRI, a single DenseAR model unifies cross-modal translation, modality-conditioned generation, and tumor segmentation, while remaining competitive with task-specific methods. On ImageNet, DenseAR improves class-conditional generation quality (FID and IS) over both a single-grid baseline without stride ordering and a multi-scale tokenizer-based baseline.

1 INTRODUCTION

Visual generative modeling has advanced rapidly, driven in large part by diffusion models, which produce detailed, coherent images through iterative denoising (Ho et al., 2020; Song et al., 2021; Park et al., 2025; 2026). Following the success of Large Language Models (LLMs), an alternative paradigm treats images as discrete sequences: a visual tokenizer compresses an image into vector-quantized (VQ) tokens, and an autoregressive (AR) model predicts these tokens one by one, like words in a sentence (Sun et al., 2024; Tian et al., 2024; Yu et al., 2025). Within this paradigm, how visual tokens are ordered for generation has emerged as a central design axis that trades quality against efficiency: two orders dominate visual AR—raster order, which generates tokens row by row, and next-scale order, used by visual autoregressive (VAR) modeling (Tian et al., 2024), which generates a sequence of token maps from low to high resolution. By generating a low-resolution token map before the finer ones, this coarse-to-fine order improves generation quality over raster-order AR (Tian et al., 2024).

Each order has a complementary strength and weakness. Raster order (Sun et al., 2024; Esser et al., 2021) operates on a single, compact latent grid, but generates tokens strictly one after another; this sequential decoding makes inference slow. Next-scale order (Tian et al., 2024; Han et al., 2024) instead achieves coarse-to-fine generation, but at a cost: it represents each image as a stack of token maps at several resolutions. The model processes all multi-scale tokens at once, so the sequence becomes far longer than a single grid, sharply raising memory and compute. This worsens in conditional generation like image translation and editing, where each source image adds its own multi-scale stack on top of the target (Liu et al., 2025; Mao et al., 2025). AR is therefore caught between the slow generation of raster order and the heavy cost of multi-scale sequences.

To resolve this trade-off, we introduce **DenseAR**, an autoregressive paradigm based on *next-dense-stride prediction* that keeps the compact single grid of raster order while ordering its tokens coarse to fine, without the sequential slowness of raster order or the lengthy multi-scale sequence of next-scale order. Our key insight is that a coarse-to-fine schedule can come from the order in which a single grid is visited rather than from the tokenizer (Figure 4 in Appendix A.1). DenseAR first commits a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

few widely spaced tokens, then halves the stride until the grid is full, so every position is predicted with committed context nearby; the tokens of each stride are predicted in parallel, so generation takes only a few steps. Keeping every modality on one short grid—rather than a multi-scale stack—is also what makes unification practical: several modalities fit in a single sequence without the blow-up that burdens multi-scale backbones, so one model can serve many tasks. This is especially valuable in medical imaging, where multi-contrast synthesis, cross-modal translation, and segmentation are usually handled by separate models and remain far less explored by AR than natural images. We validate DenseAR on ImageNet generation and multi-contrast brain MRI, where a single model handles all three tasks at once.

In summary, our contributions are:

- A new generative paradigm, *next-dense-stride prediction*, for coarse-to-fine generation on a single-scale latent grid, with neither raster-order slowness nor the sequence inflation of next-scale models.
- A unified autoregressive model for medical imaging that handles multi-contrast synthesis, cross-modal translation, and segmentation—to our knowledge, the first AR model to span all three.
- A *task-conditional* classifier-free guidance scheme that controls how strongly each prediction follows its source modalities, tunable per task with no extra components.

2 Background

2.1 Visual Tokenization

Most autoregressive (AR) visual models operate not on raw pixels but on discrete tokens. The standard choice is a vector-quantized tokenizer such as the vector-quantized variational autoencoder (VQ-VAE) (Oord et al., 2017). For example, VQ-VAE's encoder maps an input image of size $H \times W \times 3$ to a lower-resolution feature map of size $h \times w \times d$ (with $h = H/f$, $w = W/f$ for a downsampling factor f), and each d-dimensional feature vector is replaced by its nearest entry in a learned codebook, yielding an $h \times w$ grid of token indices. This grid is what most visual AR models predict over.

Visual autoregressive modeling (VAR) (Tian et al., 2024) introduces a multi-scale tokenizer to enable a coarse-to-fine representation. Instead of a single-scale grid per image, it encodes each image into K grids $R = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K)$, where each $\boldsymbol{r}_k$ is of size $h_k \times w_k$ and the resolutions grow from coarse to fine, with the last grid $\boldsymbol{r}_K$ matching the full $h \times w$. Reconstruction sums all K grids before a single decode, $\texttt{Decode}\left(\sum_{k=1}^{K} \texttt{Expand}_k(\boldsymbol{r}_k)\right)$, where $\texttt{Expand}_k(\cdot)$ takes the index grid $\boldsymbol{r}_k \in \{1, \ldots, V\}^{h_k \times w_k}$, looks up its codebook vectors to obtain a feature grid of size $h_k \times w_k \times d$, upsamples it to the shared resolution $h \times w \times d$, and applies a per-scale convolution; $\texttt{Decode}(\cdot)$ then maps the summed feature grid back to an image. The reconstruction therefore emerges only from the entire stack, not any single grid. Because each image is now represented by K stacked grids rather than one, any model predicting over this representation must process their *cumulative* length $\sum_{k=1}^{K} h_k \times w_k$—far more tokens than the single $h \times w$ grid of a single-scale tokenizer.

2.2 Autoregressive Generation and Token Ordering

Given a token grid, an AR model factorizes the joint distribution over its tokens and predicts them one at a time, $p(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T) = \prod_{t=1}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{t-1})$. The order in which tokens are visited is a design choice, and it is where visual AR methods differ most.

Single-scale order. Single-scale methods predict over one $h \times w$ grid. The standard choice is raster order, as in LlamaGen (Sun et al., 2024), which flattens the grid into hw tokens and predicts them one at a time with causal self-attention—hundreds of sequential steps per image, hence slow inference. A line of work keeps the single grid but predicts tokens in parallel to cut this step count: the masked generative image transformer (MaskGIT) (Chang et al., 2022) decodes subsets of positions simultaneously via masked prediction. The parallelized autoregressive model (PAR) (Wang et al., 2024) groups spatially distant, weakly dependent tokens for simultaneous prediction to accelerate inference. The autoregressive model with randomized parallel generation (ARPG) (Li et al., 2026)

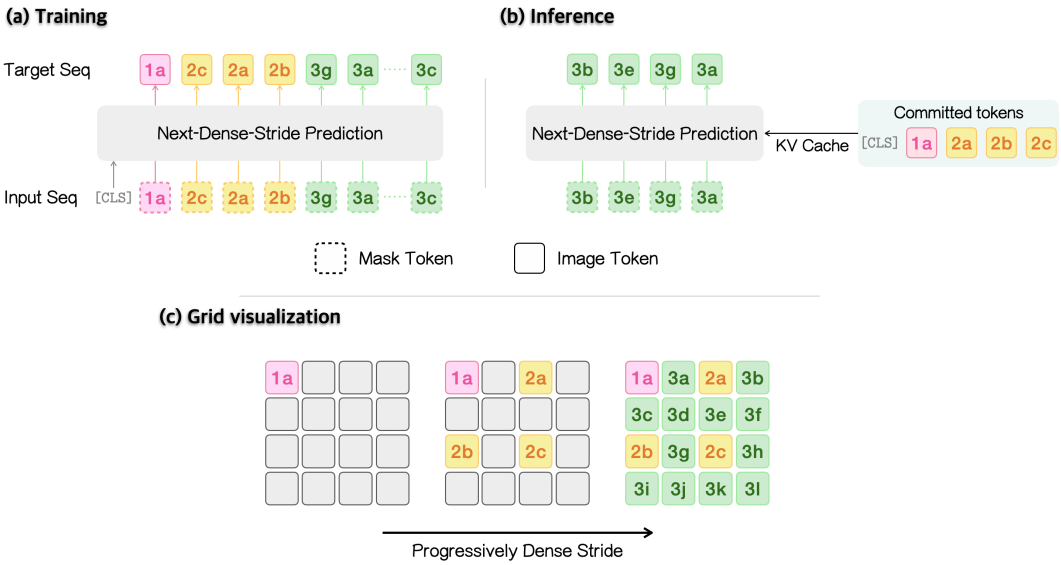


Figure 1: Visual illustration of DenseAR's next-dense-stride prediction. **(a) Training:** after the fixed `[CLS]` prefix, each target token is predicted from a position-encoded *mask-token query* (dashed) at that position. Queries are ordered by stride—stage 1 (pink), stage 2 (yellow), stage 3 (green). Stages run coarse-to-fine, but tokens *within* a stage are randomly ordered (e.g. `2c, 2a, 2b`). **(b) Inference:** from `[CLS]`, each step predicts the next group of tokens in parallel, conditioned on the committed context (class condition and predicted tokens). **(c) Grid visualization:** the stride densifies left to right—coarse positions fill first, then denser stages complete the grid. The 4×4 grid and three stages are illustrative; DenseAR uses a 16×16 token grid in all experiments.

trains causally on randomized token orders and, at inference, uses position-encoded mask query tokens to predict tokens in any order in parallel, caching each committed group so it is encoded once and reused as context rather than recomputed at every step.

Multi-scale order. To combine parallel decoding with a coarse-to-fine order, VAR (Tian et al., 2024) changes the prediction unit from a single token to one entire token map per scale. Using the K grids from Section 2.1, it factorizes generation across scales, $p(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_K) = \prod_{k=1}^{K} p(\boldsymbol{r}_k \mid \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{k-1})$, predicting all tokens within a grid $\boldsymbol{r}_k$ in parallel and cutting the step count to the number of scales K. This coarse-to-fine prior is not free: the model processes the full stack of grids rather than the final $h \times w$ grid alone, so the larger token budget translates into higher memory and attention compute, making these models expensive to scale, especially in multitask settings.

The two families trade off differently: single-scale orders keep one compact grid but predict in a flat order, without the coarse-to-fine structure that next-scale models use to improve quality (Tian et al., 2024); multi-scale orders provide that structure but inflate the token budget. DenseAR introduces a third option—a coarse-to-fine *order* on the single grid, obtained by varying the prediction stride, retaining parallel decoding without the multi-scale inflation.

2.3 Unified Generative Models for Multimodal and Multi-task Vision

Beyond single-task generation, a growing body of work builds *unified* models spanning multiple modalities and tasks within one generative model. We review how diffusion- and autoregressive-based approaches address multimodal and multi-task problems across natural and medical images.

Diffusion-based unification. Diffusion-based unification generally follows two paradigms. The first incorporates separate adapters or conditioning modules for each distinct condition type (Zhang et al., 2023; Mou et al., 2023; Huang et al., 2023). The second relies on a single shared backbone across tasks (Fu et al., 2025), though often requiring task- or modality-specific encoders to apply each condition. Similar architectural strategies have emerged in medical imaging. For instance, medical multi-modal generation (MedM2G) (Zhan et al., 2024) trains a separate diffusion model per modality



Figure 2: **Unified multi-task inference. (a) Multi-task inference.** A task is one token sequence: a task marker ([TASK]), zero or more source modalities each with a class marker ([SRC]), and a target after [TGT]. **(b) Token sequence layout.** Each square is one token, ordered left to right. The prefix—task token (gray), class tokens (orange), and source-image tokens (blue)—is fixed context; only the target tokens (right) are generated, coarse to fine (pink, then yellow, green). Source modalities are drawn as images for readability; in the sequence, they are discrete tokens.

and couples them via a central text latent space and coupled through shareable cross-attention mechanisms. Similarly, Adaptive LDM (Kim & Park, 2024) integrates switchable, target-specific normalization blocks to translate a single source modality into various targets. Both approaches integrate modality-specific modules directly into the network architecture. In contrast, Any2all (Gan et al., 2026) instead stacks modalities as input channels, avoiding per-modality components but fixing the set of modalities in advance, so adding one requires retraining the whole model.

Autoregressive-based unification. Autoregressive models extend to new tasks by adding conditioning to the token sequence rather than new per-task modules. EditAR (Mu et al., 2025) casts editing and translation as next-token prediction, but depends on an external text encoder and a foundation-model distillation module. VAREdit (Mao et al., 2025) casts editing as next-scale prediction on a VAR backbone. To cut the costly multi-scale source, it conditions on only the finest grid $\boldsymbol{r}_K$, discarding the $K-1$ coarser scales—yet the target stays a full cumulative multi-scale stack, so the cost persists. Medical efforts remain single-task: MedVAR (He et al., 2026) adapts VAR to medical synthesis, and AR-Seg (Chen et al., 2025) applies next-scale prediction to segmentation, both inheriting VAR's heavy sequence. To our knowledge, no prior AR backbone handles multi-contrast synthesis, cross-modal translation, and segmentation in one inference-efficient model. DenseAR fills this gap: one backbone, no text encoder, both continuous images and discrete segmentation maps, with coarse-to-fine generation from stride scheduling on a single grid.

Outside of diffusion and autoregressive pipelines, medical vision generalist (MVG) (Ren et al., 2024) unifies image-to-image generation tasks (such as cross-modal translation, segmentation, and inpainting) using in-context learning. MVG specifies each task by a labeled example pair stitched with the test image onto one canvas, which a masked vision transformer (Dosovitskiy et al., 2020) completes. Two drawbacks follow: every prediction needs a fully labeled pair at inference, and the single canvas holds one source image, so MVG cannot condition on several inputs (Appendix D).

3 Method

DenseAR performs coarse-to-fine autoregression on a *single* latent grid through *next-dense-stride prediction*. It generates an image stride by stride: a few widely spaced tokens are committed first, then progressively denser strides fill in the positions between them. At inference, the stride order lets the model predict many tokens in parallel at each step, so a full image takes only a few steps rather than token-by-token generation. This gives a coarse-to-fine schedule on one compact grid, without the long multi-scale token sequences of next-scale models. We define the stride order (Section 3.1) and then extend DenseAR into unified multimodal, multi-task modeling (Section 3.2).

3.1 Next-dense-stride Order

A tokenizer maps an image to a grid $\boldsymbol{x} \in \{1, \ldots, V\}^{G \times G}$ of G^2 discrete tokens. DenseAR visits these tokens from coarse to fine along a *stride pyramid*. At stride k, we keep every k-th token along

each axis, giving a sub-grid D_k of $(G/k)^2$ evenly spaced tokens—for example, on a 16×16 grid, stride $k = 8$ keeps a 2×2 sub-grid of 4 tokens, while stride $k = 1$ keeps all 256. The strides form a decreasing sequence $K,\ K/2,\ \ldots,\ 1$, where K is the coarsest stride. Halving the stride nests the sub-grids, $D_K \subset D_{K/2} \subset \cdots \subset D_1$, with D_1 the full grid. Generation walks down this sequence: it commits all of D_K, then repeatedly halves the stride, each time adding only the positions not already covered by a coarser stride, until the grid is complete (Figure 1). Choosing $K = G$ starts from a single token; a smaller K starts from a denser sub-grid.

Training. DenseAR uses the same maximum-likelihood objective as a standard autoregressive model, only over a sampled stride order π rather than a fixed raster one. For each image, we draw an order π: we pick the coarsest stride k_0 from a fixed set, then shuffle the tokens within each stage. Following the order π, the model predicts each token conditioned on those preceding it,

$$\mathcal{L} = -\,\mathbb{E}_{\boldsymbol{x},\,\pi}\left[\sum_{i=1}^{G^2} \log p_\theta\big(\boldsymbol{x}_{\pi_i} \mid \boldsymbol{x}_{\pi_{<i}}\big)\right]. \tag{1}$$

We shuffle within each stage so the model learns to predict any position in the next stage given any coarser context. Training, therefore, covers a distribution of stride orders rather than a single fixed path, and this is exactly what lets inference decode an entire stage in parallel: whatever grouping we choose at test time, the model has been trained to predict it from the committed coarser tokens. Despite the multi-stage coarse-to-fine design, every stage stays on the same $G \times G$ grid, so the conditioning sequence never exceeds G^2—unlike next-scale order.

Inference. At inference, DenseAR follows the same coarse-to-fine order, but decodes many tokens per step rather than one at a time. Denoting $\boldsymbol{x}_{<t}$ for the tokens committed in earlier steps and P_t for the group predicted at step t, the group is predicted in parallel, $p_\theta\big(\boldsymbol{x}_{P_t} \mid \boldsymbol{x}_{<t}\big) = \prod_{j \in P_t} p_\theta\big(\boldsymbol{x}_j \mid \boldsymbol{x}_{<t}\big)$, where every token in the step is drawn from the same committed context $\boldsymbol{x}_{<t}$, not from the other tokens in P_t. Training is fully token-causal, with each token conditioning on all earlier tokens in the order; inference relaxes only this, predicting a whole group P_t in parallel so its tokens do not condition on one another. The random within-stage order seen during training is what makes this parallel grouping safe. Committed tokens are stored in a key-value (KV) cache and reused across steps, so context is encoded once rather than recomputed as in Figure 1, and the number of tokens in P_t follows a schedule chosen at inference (detail is in Section 4).

3.2 Unified Multimodal, Multi-task Model

We extend DenseAR to a unified model that handles multiple modalities and tasks within one backbone. Prior unified models add structure for each task: diffusion approaches attach a new encoder or conditioning module per task (Kim & Park, 2024; Zhan et al., 2024), and autoregressive approaches inherit the long multi-scale sequence of next-scale tokenizers and stack further conditioning and refinement modules on top of it (Qu et al., 2025; Rajagopalan et al., 2025). DenseAR needs none of these. Its coarse-to-fine order comes from the prediction stride on a single grid, so it avoids the multi-scale sequence growth that is especially costly when several modalities share one sequence. Because every modality shares the same token domain, a task is simply a choice of which grids are given and which are predicted, expressed entirely in the token sequence with a small modality marker prepended to each grid. Concretely, each modality is a grid of discrete vector-quantized (VQ) tokens $\boldsymbol{x}^m \in \{1, \ldots, V\}^{G \times G}$, where m denotes the modality (e.g. $m = \texttt{seg_mask}$ for a segmentation mask). A task predicts one target grid from a set of source grids S, with each grid preceded by a modality marker. A task is written as one token sequence with three parts,

$$\boldsymbol{z} = \big[\ \underbrace{\texttt{[TASK]}}_{\text{task marker}}\ \underbrace{\texttt{[SRC 1]}\ \boldsymbol{x}^1\ \texttt{[SRC 2]}\ \boldsymbol{x}^2 \ldots}_{\text{sources (raster)}}\ \underbrace{\texttt{[TGT]}\ \boldsymbol{x}^\star}_{\text{target (stride order } \pi)}\ \big]. \tag{2}$$

where each modality grid is preceded by a marker: a task marker, a `[SRC]` marker per source modality (laid out in raster order, since sources are fixed context rather than generated), and a `[TGT]` marker for the target grid $\boldsymbol{x}^\star$, the only portion generated, in stride order π. Tasks differ only in the task marker `[TASK]` and the source set S, drawn from the same sequence (Figure 2). With `[TASK] = generate`, $S = \varnothing$, and `[TGT] = FLAIR`, the model generates a `FLAIR` image; with `[TASK] = translate` and S holding the available source contrasts, it predicts the same `FLAIR` target conditioned on those sources.

Task-conditional classifier-free guidance. The same design supports classifier-free guidance (CFG) (Ho & Salimans, 2021) with no extra components. Standard CFG is widely used in class-conditional image generation, where it trades off sample fidelity against diversity by guiding toward a class label. We extend it to a task-conditional form controlling how strongly the target follows its source modalities, per task. This is useful because the right amount of source adherence differs across tasks: translation should stay close to the source contrast, whereas modality-conditioned generation should follow the task prior more loosely. During training, we replace the source blocks with dummy blocks of the same length with probability p_{drop}, while keeping the task marker, so the model also learns a source-free prior $p_\theta(\boldsymbol{x}^\star \mid \texttt{[TASK]})$ that is still specific to the task. At inference, we combine the source-conditioned and source-free predictions for each token, with guidance strength w,

$$\begin{aligned}\log \tilde{p}_\theta \;&= (1+w)\,\log p_\theta\big(\cdot \mid \texttt{[TASK]}, S\big) \\ &\quad -\; w\,\log p_\theta\big(\cdot \mid \texttt{[TASK]}\big),\end{aligned} \tag{3}$$

so guidance steers toward the sources while staying within the task. We train all tasks jointly under the objective above, drawing each task from its own data and mixing tasks across steps; the task mixture, guidance strength w, source-dropout rate, and data splits are given in Section 4.

4 Numerical Evaluations

We evaluate DenseAR's two contributions separately: (1) the next-dense-stride order, which delivers coarse-to-fine generation on a single grid, and (2) the unified backbone, which handles many modalities and tasks in one model. To test (1), we evaluate class-conditional image generation on ImageNet (Deng et al., 2009), the standard benchmark for visual autoregressive models across raster, masked, and next-scale orders (Sun et al., 2024; Li et al., 2026; Chang et al., 2022). To test (2), we consider multi-contrast brain MRI, BraTS-2023 (Menze et al., 2015). We describe datasets and training details in Section 4.1, baselines in Section 4.2, and results in the following sections.

4.1 Experimental Setup

Datasets. For medical imaging, we use BraTS-2023 (Menze et al., 2015; Bakas et al., 2018), which provides four co-registered MRI contrasts per subject—native T1-weighted (T1), contrast-enhanced T1-weighted (T1ce), T2-weighted (T2), and T2 fluid-attenuated inversion recovery (FLAIR)—with an expert tumor segmentation. We take axial slices and keep the central 20–80% of each volume, discarding near-empty end slices so every retained slice contains substantial brain anatomy. Each slice is center-cropped from 240×240 to 228×228 (verified to remove no brain tissue across all slices), resized to 256×256, and encoded to a 16×16 token grid, matching the natural-image setting. Because the official test set has no released segmentation masks, we re-split the 1,251 subjects into 1,000 for training ($93{,}000 \times 4$ slices) and 240 for testing ($22{,}320 \times 4$ slices), with 11 held out for validation ($1{,}023 \times 4$ slices). For the segmentation task, we exclude slices with an empty tumor mask.

For natural images, we use ImageNet (Deng et al., 2009) at 256×256, covering 1,000 classes over 1.28 M training images; following common practice, we train class-conditional generation on an encoded image, which is in a 16×16 grid of VQ tokens (Oord et al., 2017).

Implementation. For natural images we tokenize with the pretrained VQ-VAE of LlamaGen (Sun et al., 2024). For the medical setting, we train two VQ-VAEs of identical architecture: an image-only tokenizer encoding the four MRI contrasts, used by all AR baselines, and one that additionally encodes the tumor mask into the same shared codebook, used by DenseAR. We thus treat the segmentation mask as another image, so segmentation needs no dedicated tokenizer or architecture. For natural images, we train DenseAR at two model sizes (L and XL); for the medical setting, we train a single unified model spanning translation, generation, and segmentation. All model scales, mixture weights, dropout and guidance strengths, step budget, and optimization settings are listed in the appendix.

4.2 Baselines

Medical baselines. On BraTS, we compare DenseAR per task against task-specific specialists from each dominant paradigm (GAN, diffusion, autoregressive), and against the closest unified models. For multi-contrast generation, we use the autoregressive model LlamaGen (Sun et al., 2024) and the

Table 1: Unified evaluation across translation, generation, and segmentation on BraTS. The left block (Capability) shows which tasks each method is designed for; the Efficiency block reports inference cost; the right block reports quantitative metrics. A single DenseAR leads on translation, ranks second on generation, and is competitive on segmentation, while being the only model that addresses all three tasks. Best and second-best results are color-coded per task among the generalist models.

Method	Capability			Efficiency	Translation				Generation			Segmentation	
	Trans.	Gen.	Seg.	Mem (GB)	PSNR ↑	SSIM ↑	LPIPS ↓	L1 ↓	FID ↓	RadFID ↓	KID ↓	Dice ↑	IoU ↑
Specialists													
Pix2Pix (Isola et al., 2017)	✓	✗	✗	0.56	21.44	0.829	0.100	0.036	–	–	–	–	–
CycleGAN (Zhu et al., 2017)	✓	✗	✗	0.92	20.29	0.819	0.105	0.048	–	–	–	–	–
MSG-LDM (Lin et al., 2026)	✓	✗	✗	8.16	21.10	0.814	0.112	0.039	–	–	–	–	–
LlamaGen (Sun et al., 2024)	✗	✓	✗	6.43	–	–	–	–	10.92	0.011	0.005	–	–
DiT (Peebles & Xie, 2023)	✗	✓	✗	9.11	–	–	–	–	8.03	0.005	0.004	–	–
nnU-Net (Isensee et al., 2021)	✗	✗	✓	1.30	–	–	–	–	–	–	–	0.898	0.828
MedSegDiff-V2 (Wu et al., 2023)	✗	✗	✓	15.07	–	–	–	–	–	–	–	0.851	0.765
Generalists													
EditAR (Mu et al., 2025)	✓	✓	✗	24.61	20.78	0.811	0.112	0.041	15.88	0.029	0.010	–	–
MVG (Ren et al., 2024)	✓	✗	✓	12.51	20.64	0.825	0.113	0.040	–	–	–	0.846	0.763
Ours	✓	✓	✓	10.72	21.89	0.839	0.097	0.034	8.50	0.012	0.004	0.851	0.756

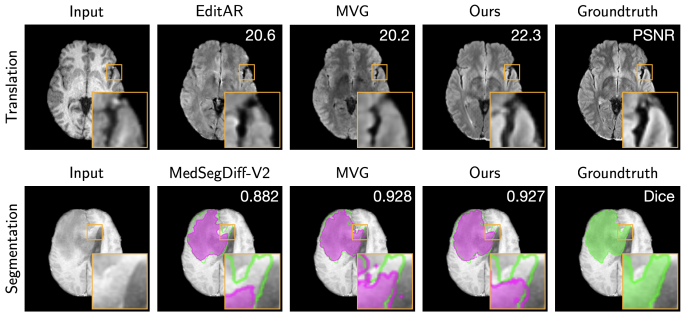


Figure 3: Visual comparison of translation and segmentation for representative methods from Table 1. In segmentation, pink mask is the prediction and green mask is the ground truth.

diffusion model DiT (Peebles & Xie, 2023). For cross-modal translation, we use the GAN-based translators Pix2Pix (Isola et al., 2017) and CycleGAN (Zhu et al., 2017), and MSG-LDM (Lin et al., 2026), a latent-diffusion model performing any-to-any multi-contrast translation—covering the GAN and diffusion families. For segmentation, we use nnU-Net (Isensee et al., 2021), the standard end-to-end medical segmentation model, and MedSegDiff-V2 (Wu et al., 2023), a diffusion segmentation model. We additionally compare to two unified models. EditAR (Mu et al., 2025), the closest unified autoregressive model, addresses translation and generation but has no native segmentation head. MVG (Ren et al., 2024) is an in-context unified model addressing translation and segmentation; since it cannot condition on multiple source images, we include it only in the single-source experiments, not the multi-source translation task. All medical baselines are trained on BraTS following each method's standard setup, and Appendix C.2 lists each one's parameter count, epochs, batch size, learning rate, and schedule. Table 5 (Appendix B.1) summarizes which tasks each model supports and the components it requires.

Natural-image baselines. On ImageNet, we compare DenseAR against token-based visual generators covering the dominant decoding strategies—raster-order autoregression (LlamaGen (Sun et al., 2024)), masked iterative decoding (MaskGIT (Chang et al., 2022), MAR (Li et al., 2024)), alternative autoregressive orders (RAR (Yu et al., 2025), RandAR (Pang et al., 2024), PAR (Wang et al., 2024), NAR (He et al., 2025), ARPG (Li et al., 2026)), and next-scale autoregression (VAR (Tian et al., 2024))—together with the latent diffusion model DiT (Peebles & Xie, 2023). Two of these comparisons are the most informative. The first quantifies what next-dense-stride contributes on its own. By evaluating it under the same capacity and compute as a random-order baseline (ARPG (Li et al., 2026)), we attribute the resulting gains directly to coarse-to-fine stride ordering. The second concerns coarse-to-fine modeling itself: VAR is the closest competitor, but obtains its coarse-to-fine

Table 2: Quantitative comparison of different methods on ImageNet benchmarks. Up (↑) and down (↓) arrows indicate that higher and lower values are better, respectively.

Type	Model	Param.	Steps	Mem. (GB)↓	FID↓	IS↑	Pre.↑	Rec.↑
Diffusion	DiT-L/2 (Peebles & Xie, 2023)	458 M	250	1.62	5.02	167.2	0.75	0.57
	DiT-XL/2 (Peebles & Xie, 2023)	675 M	250	2.14	2.27	278.2	0.83	0.57
Mask	MaskGIT (Chang et al., 2022)	227 M	8	1.71	6.18	182.1	0.80	0.51
	MAR-B (Li et al., 2024)	208 M	100	1.47	2.31	281.7	0.82	0.57
	MAR-L (Li et al., 2024)	479 M	100	2.32	1.78	296.0	0.81	0.60
AR	RAR-B (Yu et al., 2025)	261 M	256	4.65	1.95	290.5	0.82	0.58
	RAR-L (Yu et al., 2025)	461 M	256	6.37	1.70	299.5	0.81	0.60
	PAR-L (Wang et al., 2024)	343 M	147	10.25	3.76	218.9	0.81	0.60
	PAR-XL (Wang et al., 2024)	775 M	147	17.13	2.61	259.2	0.80	0.62
	NAR-L (He et al., 2025)	372 M	31	10.25	3.06	263.9	0.81	0.53
	NAR-XL (He et al., 2025)	816 M	31	17.13	2.70	277.5	0.81	0.58
	RandAR-L (Pang et al., 2024)	343 M	88	7.32	2.55	288.8	0.81	0.58
	RandAR-XL (Pang et al., 2024)	775 M	88	13.52	2.25	317.8	0.80	0.60
	ARPG-L (Li et al., 2026)	320 M	64	2.64	2.37	293.7	0.82	0.55
	ARPG-XL (Li et al., 2026)	719 M	64	4.57	1.99	340.6	0.80	0.61
Raster AR	LlamaGen-L (Sun et al., 2024)	343 M	576	10.23	3.07	256.1	0.83	0.52
	LlamaGen-XL (Sun et al., 2024)	775 M	576	17.11	2.62	244.1	0.80	0.57
Multi-scale AR	VAR-d16 (Tian et al., 2024)	310 M	10	10.85	3.30	274.4	0.84	0.51
	VAR-d20 (Tian et al., 2024)	600 M	10	15.97	2.57	302.6	0.83	0.56
	VAR-d24 (Tian et al., 2024)	1.0 B	10	22.29	2.09	312.9	0.82	0.59
DenseAR	DenseAR-L	320 M	64	2.64	2.23	305.6	0.81	0.58
	DenseAR-XL	719 M	64	4.57	1.90	341.1	0.79	0.61

prior by stacking token maps across resolutions, whereas DenseAR obtains its schedule from the prediction order—so we compare not only generation quality but also efficiency.

Metrics. We report standard metrics per task: FID, IS and precision/recall for natural-image generation, plus decoding steps and peak memory for efficiency, measured with a batch size of 64 and bfloat16 precision; PSNR, SSIM, L1 and LPIPS for translation; FID, the radiology-feature variant RadFID and the kernel inception distance (KID) for medical generation, which better reflect perceptual quality on MRI; and the Dice coefficient and intersection over union (IoU) for segmentation.

4.3 Unified Multi-task Evaluation

Table 1 evaluates the unified medical model on BraTS against task-specific specialists and two unified competitors, EditAR and MVG; Figure 3 shows representative translations and segmentation masks, with the full qualitative comparisons in Appendix B.2. The capability block tells the headline at a glance: DenseAR is the only model that addresses translation, generation, and segmentation at once. Every specialist covers a single task, EditAR has no segmentation path, and MVG cannot support class-conditional generation from a modality label alone. DenseAR does so with the leanest set of required components (Table 5, Appendix B.1): a single VQ-VAE, no text encoder, no pretrained backbone, and no in-context example—whereas EditAR relies on a text encoder and a pretrained backbone, and MVG on a pretrained backbone and a labeled in-context example. DenseAR uses the task-conditional classifier-free guidance of Section 3, with a guidance weight w set separately per task; Appendix A.3 sweeps w from 0.5 to 15 for all three tasks and shows the optima differ markedly.

Translation. For the single direction comparable across all baselines (T1 $\rightarrow$ FLAIR, Table 1), a single DenseAR backbone achieves the best cross-modal translation, improving every metric over the GAN and diffusion specialists as well as EditAR and MVG. DenseAR also generalizes across all twelve source–target contrast pairs, where it generally outperforms EditAR and MVG (Table 6, Appendix B.2), and generally improves when more than one contrast are supplied (Appendix A.4).

Generation. For each method, we generate 12,500 samples per class and compute perceptual metrics over the 50,000 samples against 12,500 real samples per class. In FID, DiT leads at 8.03, DenseAR follows at 8.50, and LlamaGen is last at 10.92, with comparable KID—and unlike LlamaGen and DiT, which only perform class-conditional generation, DenseAR also handles translation and segmentation. The next-dense-stride schedule is what drives this quality: removing it—predicting the same grid without the stride stages—degrades every generation metric (Table 3, Appendix A.2), confirming the gain comes from the stride ordering.

Segmentation. For each method, we evaluate on slices excluding near-empty slices where the metric is unstable. DenseAR reaches 0.851 Dice and 0.756 IoU, on par with the diffusion-based MedSegDiff-V2 (0.851 Dice, 0.765 IoU) and behind the specialist nnU-Net (0.898 Dice). DenseAR recovers much of nnU-Net's accuracy through the same token interface it uses for generation and translation, with no segmentation-specific head.

Across the three tasks, one DenseAR model leads on translation, ranks second on generation, and is competitive on segmentation. The breadth comes with little accuracy cost: where a specialist leads, the margin is modest, and DenseAR delivers all three from one backbone trained under one objective.

4.4 Natural-image generation

Table 2 reports class-conditional ImageNet generation from a DenseAR model trained separately from the medical model of Section 4.3. Baseline numbers, memory included, are quoted from Li et al. (2026) rather than re-measured; DenseAR uses the same protocol; Appendix A.5 compares the two model scales and Appendix B.3 shows samples against LlamaGen and VAR.

Against raster order. Compared with raster-order autoregression, the parallel stride decoding cuts the step count sharply: DenseAR generates in 64 steps against LlamaGen's 576, while improving FID (2.23 vs. 3.07 at L). DenseAR reaches this quality with both few decoding steps and low memory, which matters when many sequences are packed together in the multi-task setting of Section 4.3.

Coarse-to-fine without multi-scale inflation. VAR obtains its coarse-to-fine prior by stacking token maps across resolutions, which inflates the sequence and the memory it occupies. DenseAR reaches comparable or better quality on a single grid at a fraction of that memory: DenseAR-XL attains 1.90 FID using 4.57 GB, against VAR-d24's 2.09 FID at 22.29 GB—better quality with roughly 5× less memory and fewer parameters (719 M vs. 1.0 B). This is the central efficiency claim of the single-grid design made concrete: a coarse-to-fine schedule costs almost nothing in memory when it comes from the prediction order, not a multi-scale tokenizer.

Effect of the stride order. Because DenseAR and ARPG share the same backbone, parameter count, step budget, and memory, the only difference between the two rows is the decoding order, so this pair isolates the effect of next-dense-stride prediction. At both scales, DenseAR improves ARPG's FID—2.23 vs. 2.37 at L and 1.90 vs. 1.99 at XL—and IS at L (305.6 vs. 293.7), though the XL pair is within half a point; precision is 0.01 lower in both and recall higher at L. The stride order therefore adds a coarse-to-fine schedule that ARPG lacks.

RAR and MAR report lower FID than DenseAR, but at a higher decoding cost, and MAR relies on a continuous tokenizer and sampling framework; Appendix B.3 compares them in detail.

5 Conclusion

We introduce DenseAR, an autoregressive paradigm that reformulates image generation as coarse-to-fine next-dense-stride prediction on a single latent grid. We obtain a coarse-to-fine schedule through the order in which tokens are predicted—a few widely spaced tokens first, then progressively denser strides—keeping the compact grid of raster order while avoiding its slow inference and the lengthy sequences of next-scale models. On class-conditional generation, DenseAR is far more efficient than raster-order models at higher quality, and as strong as next-scale models using a fraction of their memory. We further develop DenseAR into a unified backbone: one DenseAR model performs cross-modal translation, modality-conditioned generation, and tumor segmentation on multi-contrast brain MRI, the first medical autoregressive model to unify all three. The key design is a single token sequence in which small markers at the front specify the task and each modality, so the model handles multiple modalities without any dedicated module. Because a task is simply a choice of which grids are given and which are predicted, extending DenseAR to new modalities or tasks requires no architectural change, only a new arrangement of the same sequence. We hope this view of generation, translation, and segmentation as one prediction problem over a shared grid offers a practical path toward general-purpose autoregressive models for medical imaging.

AI USE STATEMENT

In preparing this work, we used large language models (LLMs) in two roles. As a coding assistant, we used them to help implement and debug portions of our training and evaluation code. All such code was reviewed and tested by the authors, and every number reported in this paper comes from runs we conducted and verified ourselves. As a writing assistant, we used them to check grammar and edit for readability. LLMs were not used for research ideation or method design: the next-dense-stride formulation, the unified task interface, and the experimental design are the authors' own. We have reviewed all AI-assisted work and take responsibility for the final content of this paper.

ETHICS STATEMENT

This work uses BraTS-2023 (Menze et al., 2015; Bakas et al., 2018), a publicly released, de-identified brain MRI dataset distributed for research use; no new human-subjects data was collected and no IRB approval was required. Generative models for medical imaging can produce anatomically plausible but incorrect structure, so the synthesis and segmentation results reported here are intended for methodological evaluation and not for clinical use or diagnosis.

REPRODUCIBILITY STATEMENT

An anonymous code package reproducing training and sampling for both the ImageNet and BraTS settings is included in the supplementary material. Section 4.1 describes the datasets, preprocessing, and the subject-level re-split used for BraTS-2023, together with the tokenizers and model scales. Appendix C gives the method illustrations and the training configuration for DenseAR and every baseline, including parameter counts, epochs, batch sizes, learning rates, schedules, and the per-task guidance weights. Appendix A reports the ablations supporting each design choice, each holding the tokenizer, backbone, and step budget fixed so that the reported difference is attributable only to the ablated choice.

REFERENCES

Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. *arXiv preprint arXiv:1811.02629*, 2018. doi: 10.48550/arxiv.1811.02629.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022.

Tao Chen, Chenhui Wang, Zhihao Chen, and Hongming Shan. Autoregressive medical image segmentation via next-scale mask prediction. In *MICCAI*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12868–12878, 2021. doi: 10.1109/cvpr46437.2021.01268.

Tsu-Jui Fu, Yusu Qian, Chen Chen, Wenze Hu, Zhe Gan, and Yinfei Yang. UniVG: A generalist diffusion model for unified image generation and editing. In *ICCV*, 2025.

Weijie Gan, Xucheng Wang, Tongyao Wang, Wenshang Wang, Chunwei Ying, Yuyang Hu, Yasheng Chen, Hongyu An, and Ulugbek S. Kamilov. A unified framework for multimodal image reconstruction and synthesis using denoising diffusion models. *arXiv preprint arXiv:2602.08249*, 2026.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2412.04431.

Yefei He, Yuanyu He, Shaoxuan He, Feng Chen, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Neighboring autoregressive modeling for efficient visual generation. *arXiv preprint arXiv:2503.10696*, 2025.

Zhicheng He, Yunpeng Zhao, Junde Wu, Ziwei Niu, Zijun Li, Bohan Li, Lanfen Lin, and Yueming Jin. Medvar: Towards scalable and efficient medical image generation via next-scale autoregressive prediction, 2026. URL https://arxiv.org/abs/2602.14512.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Lianghua Huang, Di Chen, Yifei Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023.

Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021. doi: 10.1038/s41592-020-01008-z.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

Jonghun Kim and Hyunjin Park. Adaptive latent diffusion model for 3D medical image to image translation: Multi-modal magnetic resonance imaging study. In *WACV*, 2024.

Haopeng Li, Jinyue Yang, Guoqi Li, and Huan Wang. Autoregressive image generation with randomized parallel decoding. In *The Fourteenth International Conference on Learning Representations*, 2026.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *NeurIPS*, 2024.

Jianqiang Lin, Zhiqiang Shen, Peng Cao, Jinzhu Yang, Osmar R. Zaiane, and Xiaoli Liu. Multiscale structure-guided latent diffusion for multimodal mri translation. *arXiv preprint arXiv:2603.12581*, 2026.

Yi Liu, Shengqian Li, Zuzeng Lin, Feng Wang, and Si Liu. Cyclevar: Repurposing autoregressive model for unsupervised one-step image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15194–15204, 2025.

Qingyang Mao, Qi Cai, Yehao Li, Yingwei Pan, Mingyue Cheng, Ting Yao, Qi Liu, and Tao Mei. Visual autoregressive modeling for instruction-guided image editing. *arXiv preprint arXiv:2508.15772*, 2025.

Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34:1993–2024, 2015. doi: 10.1109/tmi.2014.2377694.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

Jiteng Mu, Nuno Vasconcelos, and Xiaolong Wang. Editar: Unified conditional generation with autoregressive models. *arXiv preprint arXiv:2501.04699*, 2025.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. doi: 10.48550/arxiv.1711.00937.

Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T. Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. *arXiv preprint arXiv:2412.01827*, 2024.

Chicago Y. Park, Michael T. McCann, Cristina Garcia-Cardona, Brendt Wohlberg, and Ulugbek S. Kamilov. Random walks with Tweedie: A unified view of score-based diffusion models [in the spotlight]. *IEEE Signal Processing Magazine*, 42(3):40–51, 2025. doi: 10.1109/MSP.2025.3590608. URL https://doi.org/10.1109/MSP.2025.3590608.

Chicago Y. Park, Michael T. McCann, Cristina Garcia-Cardona, Brendt Wohlberg, and Ulugbek S. Kamilov. Deep parameter interpolation for scalar conditioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3884–3892, 2026.

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

Yunpeng Qu, Kun Yuan, Jinhua Hao, Kai Zhao, Qizhi Xie, Ming Sun, and Chao Zhou. Visual autoregressive modeling for image super-resolution. *arXiv preprint arXiv:2501.18993*, 2025.

Sudarshan Rajagopalan, Kartik Narayan, and Vishal M. Patel. Restorevar: Visual autoregressive generation for all-in-one image restoration, 2025. URL https://arxiv.org/abs/2505.18047.

Sucheng Ren, Xiaoke Huang, Xianhang Li, Junfei Xiao, Jieru Mei, Zeyu Wang, Alan Yuille, and Yuyin Zhou. Medical vision generalist: Unifying medical imaging tasks in context. *arXiv preprint arXiv:2406.05565*, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024.

Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation. *arXiv preprint arXiv:2412.15119*, 2024.

Junde Wu, Wei Ji, Huazhu Fu, Min Xu, Yueming Jin, and Yanwu Xu. Medsegdiff-v2: Diffusion based medical image segmentation with transformer. *arXiv preprint arXiv:2301.11798*, 2023.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *ICCV*, 2025.

Chenlu Zhan, Yu Lin, Gaoang Wang, Hongwei Wang, and Jian Wu. MedM2G: Unifying medical multi-modal generation via cross-guided diffusion with visual invariant. In *CVPR*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.