Adaptive Inference Routing for Autoregressive-Diffusion Generative Models

Gia Ancone [1] Cayleigh Haag [1]
[1]Department of Computer Science, Stanford University

Motivation & Background

Both diffusion models and autoregressive (AR) models are at the forefront of generative modeling. AR models tend to be fast at inference and effective at capturing global structure. However, errors made early have enduring impact on later token prediction. Diffusion models are more robust in this sense, as they iteratively refine samples toward the known data manifold, but pose a greater computational strain [1, 2]. Prior work shows generation difficulty varies across seeds [2, 4], and hybrid AR-diffusion strategies can balance cost and quality [3]. Hence, our objectives are:

1. To explore and compare the strengths of two distinct hybrid generative architectures.
2. To test whether early diagnostic signals from one pipeline can inform adaptive routing decisions that reduce inference cost without sacrificing output quality.

Data

- **CIFAR-10** [6] "dog" class: 5,000 training images, 32×32 RGB, normalized to $[-1, 1]$
- Model B patches each image into **16 non-overlapping** 8×8 patches (192-dim vectors)
- **600 seeds** labeled by human annotators comparing Model A vs. B side-by-side to train model R.

Label	Count	%
Model A needed	403	67.2%
`cheap_ok` (B is fine)	197	32.8%

Features

Model A takes full 32×32 RGB images; **Model B** flattens 8×8 patches to 192-dim vectors.

Router Feature Selection via Ablation

We initially extracted **24 candidate features** from Model B's first 12 AR steps using forward hooks. We performed **leave-one-out ablation**: removing each feature individually and measuring the impact on router precision. Features whose removal *improved* precision were discarded as noise. This reduced the set from 24 $\rightarrow$ **18 features**, improving precision from 38% $\rightarrow$ **58%**. **Final 18 router features:**

- **AR Dynamics (10):** MSE stats, spatial variance, autocorrelation, attention entropy/trend/focus, frequency energy
- **Image-Level (4):** pixel std, dynamic range, edge energy, symmetry error
- **Self-Consistency (2):** teacher forcing discrepancy
- **Per Layer (1):** Layer 3 max attention weight

Model A: Blockwise Diffusion (800 passes)

- Generates images **block-by-block** (4 blocks of 16×16)
- Conditional **U-Net** with self-attention
- Full DDPM reverse process [1]
- Each block conditioned on prior blocks via **binary context mask**
- Trained 500 epochs, Adam (lr 3×10^{-4}).

$$\mathcal{L}_A = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{m})\|^2\right]$$

Total: 128 base channels, 200 steps per block $\times$ 4 blocks = **800 forward passes**

Model B: AR + Shallow Diffusion (46 passes)

Stage 1: Autoregressive Patch Generation

- Transformer decoder (6 layers, 512 dim, 8 heads) with **causal self-attention**
- Generates **16 patches** (8×8 px) sequentially from a learned start token
- Teacher forcing during training; temperature $T = 0.25$ noise at inference. 250 epochs, Adam (lr 3×10^{-4})

$$\mathcal{L}_{AR} = \sum_{i=1}^{16} \|\hat{\mathbf{p}}_i - \mathbf{p}_i\|^2, \quad \hat{\mathbf{p}}_i = f_\theta(\mathbf{p}_{<i})$$

Stage 2: Shallow Diffusion Corrector

- Small U-Net (64 base channels, 100 total timesteps)
- Adds noise to AR output, then runs **30 denoising steps** to remove artifacts
- 150 epochs, Adam (lr 3×10^{-4})

Total: $16 + 30 =$ **46 forward passes** $\rightarrow$ **17**$\times$ cheaper than Model A

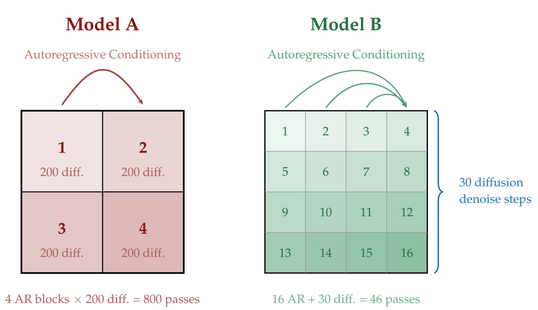


Figure 1. Left: Model A generates four 16×16 blocks via diffusion, each conditioned AR on prior blocks. Right: Model B generates sixteen 8×8 patches autoregressively, then refines with 30 diffusion steps.

Router (Model R): Gradient Boosting

- **Gradient Boosting** classifier [5] (2000 trees, depth 4, lr 0.05)
- Z-score normalized features; routes to B if $P(\texttt{cheap\_ok}) \geq 0.55$, else A
- Best of 30 random 80/20 splits saved
- Probe overhead: only **0.0023s/image**

Key Design Choices

- **GB over neural networks:** With only 480 training samples & 18 features, tree-based methods are more data-efficient and less likely to overfit. GB's sequential residual correction with shallow trees (depth 4) regularizes naturally
- **Threshold tuning:** Hyperparameter tuned from 0.40 to 0.70. Threshold **0.55** balances cost savings (23%) with precision (up to 58%): higher thresholds are too conservative, lower ones degrade quality

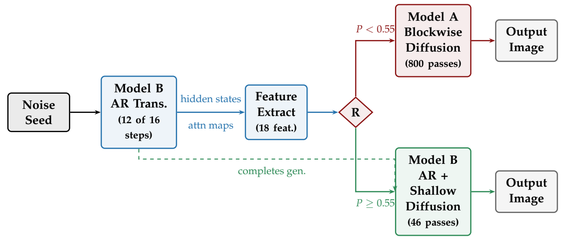


Figure 2. Routing pipeline. Seed enters Model B's AR stage; features are extracted and passed to Router R, which routes to Model A (expensive) or lets Model B complete (cheap).

Results

Router Classification

Metric	Train (n=480)	10-fold CV (test)	Best of 30 (test)
Accuracy	100.0%	65.3% $\pm$ 3.8%	65.8%
Precision	100.0%	**44.4%** $\pm$ 11.3%	**57.9%**
Recall	100.0%	28.1% $\pm$ 8.2%	25.0%

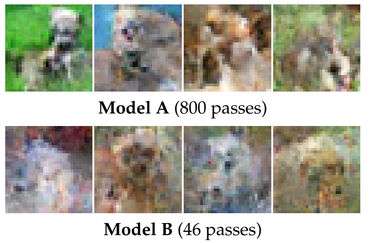
80/20 split (**480** train, **120** test). Train overfitting is expected with high-capacity GB, but this configuration maximized **test precision**, which is our primary objective. Random baseline precision is **32.8%** (class prior), not 50%.

Inference Cost (T4 GPU, $N = 50$)

Strategy	Cost/img	% to A	Savings
Always Model A	0.493s	100%	-
Always Model B	0.009s	0%	98.2%
Human Router	0.334s	67%	32.3%
Learned Router (Model R)	**0.378s**	**76%**	**23.2%**

Model A is **56.6**$\times$ **slower** than Model B on GPU.

Qualitative Comparison



Model A (800 passes)

Model B (46 passes)

Figure 3. Top row: Model A outputs. Bottom row: Model B outputs. Quality gap motivates adaptive routing.

Discussion

- Router precision (**44-58%**) substantially beats random baseline (**32.8%**), early AR dynamics carry **real signal** about generation quality
- Gap between router savings (**23.2%**) and ground truth (**32.3%**) validates that **per sample adaptive routing is feasible**
- Conservative threshold (0.55) prioritizes **quality**: better to waste compute than degrade output
- Framework is **modular**, better generators yield larger savings with the same routing pipeline

Future Work

- Scale labeled dataset beyond 600 seeds for more reliable estimates
- **Continuous compute allocation**: adapt diffusion steps per sample instead of binary routing
- Extend to **multiple CIFAR-10 classes** or higher resolution
- Replace human labels with **automated quality metrics** (per seed FID)

Key Takeaways

- Early diagnostic signals from an AR generation process can predict per sample difficulty and enable **adaptive compute allocation**, achieving **23.2% cost savings** with a lightweight routing overhead of just **0.0023s/image**.
- Beyond these preliminary results, this work highlights the potential of adaptive routing in hybrid AR-diffusion pipelines, where per sample compute allocation could yield greater gains with higher-capacity models, supporting AR-diffusion hybrids as a promising direction for efficient generative modeling.

References

[1] Ho et al., NeurIPS 2020. [2] Dhariwal & Nichol, NeurIPS 2021. [3] Salimans & Ho, ICLR 2022. [4] Karras et al., NeurIPS 2022. [5] Friedman, Ann. Stat. 2001. [6] Krizhevsky, 2009.

giaancone@stanford.edu cayleigh@stanford.edu