Energy-Adaptive Diffusion via Dynamic Token Pruning for Carbon-Efficient On-Device Generation

Anonymous CVPR submission

Paper ID *****

Abstract

*Diffusion-based generative models achieve state-of-the-art image synthesis but are computationally intensive, limiting deployment on mobile and edge devices. Existing efficiency techniques typically rely on fixed inference schedules and static compute budgets, failing to account for dynamic device conditions such as battery level or thermal constraints. We introduce **Energy-Adaptive Diffusion (EAD-Diff)**, a framework that dynamically adapts computation to device-level energy availability. EAD-Diff combines budget-conditioned token pruning in latent feature space with adaptive denoising step scheduling, trained under a unified objective that balances perceptual quality and energy consumption. We evaluate our approach on CIFAR-10 and CelebA-HQ 256×256, measuring energy and latency on NVIDIA Jetson Orin Nano and Raspberry Pi 5. Results show up to 38% energy reduction with minimal FID degradation, demonstrating that runtime-adaptive diffusion is a practical pathway toward sustainable, on-device generative AI.*

1. Introduction

Diffusion models have emerged as the state-of-the-art paradigm for high-fidelity image synthesis [1, 2], yet their deployment on mobile and edge devices remains challenging due to high computational and memory demands. Even optimized variants require tens of denoising steps, which, combined with dynamic device constraints such as fluctuating battery levels, thermal throttling, and variable power budgets on embedded GPUs and single-board computers, limits practical on-device usage. Existing efficiency approaches [3, 4] rely on fixed inference schedules and static compute budgets, which either waste energy under favorable conditions or degrade generation quality under constrained resources. To address these limitations, we introduce **Energy-Adaptive Diffusion (EAD-Diff)**, the first framework that dynamically adapts both computation and sampling to real-time device-level energy availability. EAD-Diff jointly leverages *budget-conditioned token pruning*, which selectively removes less salient latent tokens based on the normalized energy budget, and *adaptive denoising step scheduling*, allowing the model to trade off computation and quality at runtime. These mechanisms are trained under a unified objective balancing perceptual fidelity and energy consumption, enabling on-device generative AI that gracefully scales quality with available resources.

2. Background and Related Work

Efforts to accelerate diffusion models primarily focus on fixed-step sampling and latent-space compression. DDIM [3] reduces inference time via non-Markovian sampling, progressive distillation [4] compresses sampling trajectories for 4-16× speedup, and consistency models [5] enable one-step generation at modest quality loss. Latent diffusion [2] reduces spatial resolution to lower computational cost. While effective, all these approaches assume static schedules and ignore dynamic device-level constraints. Token pruning has been applied in classification transformers [6, 7] to eliminate less informative tokens, but its use in generative diffusion, particularly under energy-aware conditions, remains unexplored. Parallel research in edge AI has investigated quantization [8], hardware-aware training [9], and model compression [10], with adaptive inference methods adjusting network depth or width based on input complexity [11]. Unlike prior work, EAD-Diff integrates energy-aware token pruning and adaptive step scheduling directly into the diffusion process, enabling real-time, device-adaptive generation that optimizes both energy consumption and perceptual quality.

3. Methodology

We build upon a latent diffusion framework with a U-Net backbone, where $\mathbf{x}_t$ denotes the noisy latent at timestep t and $\epsilon_\theta(\mathbf{x}_t, t)$ is the noise predictor. The standard diffusion

objective minimizes

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right]. \quad (1)$$

To enable runtime energy adaptation, we introduce a normalized energy budget $B \in [0, 1]$ derived from device state (battery level, thermal headroom, or power cap), which the model observes at inference to dynamically adjust computation.

Budget-Conditioned Token Pruning At each denoising step, the U-Net processes feature maps as sequences of spatial tokens $\mathbf{Z} \in \mathbb{R}^{H\times W\times C}$, partitioned into $N = H \times W$ tokens $\{\mathbf{z}_i\}_{i=1}^N$. We learn a lightweight importance predictor f_ϕ that outputs $s_i = f_\phi(\mathbf{z}_i, t, B)$ indicating the contribution of each token under the current budget. The MLP predictor introduces only 0.3% FLOPs overhead, with measured wall-clock impact under 2ms per step, and uses a straight-through estimator for differentiable pruning. Tokens with s_i below a budget-conditioned threshold $\tau(B) = \tau_{\min} + (1 - B)(\tau_{\max} - \tau_{\min})$ are pruned, retaining at least 25% of tokens to preserve structure. Thresholds $\tau_{\min} = 0.1$ and $\tau_{\max} = 0.7$ were selected via grid search on validation sets. Pruning ratios vary approximately linearly with budget, from 25% at $B = 1.0$ to 65% at $B = 0.2$, reducing both attention and convolution costs proportionally. Importantly, the normalized budget B allows generalization across hardware by scaling according to device power profiles.

Adaptive Denoising Step Scheduling To further control energy, we adapt the number of denoising steps according to budget: $T(B) = \max(T_{\min}, \lfloor T_{\max} - (1 - B)(T_{\max} - T_{\min})\rfloor)$ with $T_{\max} = 30$ and $T_{\min} = 10$, determined via validation grid search. This yields discrete step bins (10–30 steps) that simplify hardware scheduling and reduce runtime overhead. Continuous step prediction is left as future work. Together with token pruning, this mechanism enables a fine-grained trade-off between computation and perceptual quality at runtime.

Energy-Quality Objective We extend the diffusion loss with a budget-weighted energy regularization:

$$\mathcal{L} = \mathcal{L}_{\text{diffusion}} + \lambda(B) \cdot \mathcal{C}_{\text{energy}}, \quad (2)$$

where $\mathcal{C}_{\text{energy}} = \frac{1}{N}\sum_i \mathbb{I}(s_i < \tau(B)) \cdot c_{\text{token}} + \frac{T(B)}{T_{\max}} \cdot c_{\text{step}}$ approximates computational cost and $\lambda(B) = \lambda_0(1 - B)^\gamma$ increases regularization under low budgets. The FLOPs-based proxy $\mathcal{C}_{\text{energy}}$ was empirically validated against real energy measurements on Jetson Orin Nano and Raspberry Pi 5, with ¡5% deviation, ensuring that optimization aligns with actual device energy consumption.

Training Procedure EAD-Diff is trained in two stages. First, we pretrain the base diffusion model without adaptation for 500k iterations using Adam (learning rate 1e-4, batch size 128 for CIFAR-10, 64 for CelebA-HQ). Second, we fine-tune for 200k iterations with the energy-aware objective, sampling B uniformly from $[0, 1]$ to cover the full budget spectrum. The importance predictor f_ϕ is trained jointly, using a binary classification objective to distinguish tokens whose retention meaningfully impacts generation quality. This procedure ensures that the model learns to gracefully scale computation according to energy availability, balancing efficiency and perceptual fidelity on diverse edge devices.

4. Experimental Setup

We evaluate EAD-Diff on two widely used datasets to test both low- and high-resolution image generation. **CIFAR-10** contains 60,000 32×32 color images across 10 balanced classes, split into 50,000 training and 10,000 test images. Its small resolution allows rapid prototyping, with 500 generations per configuration requiring approximately 2 hours. **CelebA-HQ 256×256** comprises 30,000 high-quality aligned face images, with a standard 27,000/1,000/2,000 train/validation/test split. This high-resolution dataset assesses performance under realistic generation conditions and latent compression. For CIFAR-10, we adopt a U-Net with 56M parameters, featuring 4 down- and up-sampling blocks with channel dimensions [128, 256, 256, 256] and attention layers at 16×16 resolution with 4 heads. For CelebA-HQ, we employ a latent diffusion architecture [2] with a pretrained autoencoder compressing 256×256 images to 32×32 latents (8× spatial reduction). The latent U-Net has 210M parameters and includes cross-attention for class conditioning. Experiments are conducted on two edge platforms to evaluate energy-adaptive inference under realistic conditions. **NVIDIA Jetson Orin Nano (8GB)** is an embedded GPU capable of up to 40 TOPS at 7–15W. We operate in 15W mode and measure power via `tegrastats` with 100ms sampling, subtracting idle power (1.2W). **Raspberry Pi 5 (8GB)** is a low-power CPU device with a quad-core Arm Cortex-A76 at 2.4GHz; power is measured on 5V input using a Joulescope JS220 DC analyzer (±0.1% accuracy, 2kHz sampling). Energy per image is computed as $E = \int_0^{t_{\text{inference}}} P(\tau)d\tau$, averaged over 500 generations with 95% confidence intervals. We compare EAD-Diff against four baselines: (1) **Fixed-step** DDIM with 30 steps (high quality) and 20 steps (standard), (2) **Step-reduced** DDIM with 10 steps (aggressive reduction), (3) **Quantized** INT8 DDIM with 30 steps via TensorRT, and (4) **Oracle adaptive**, an upper bound combining optimal step selection and perfect token pruning estimated via exhaustive search on 100 validation samples. All baselines use identical architectures and training data to ensure

Table 1. CIFAR-10 results on Jetson Orin Nano. Energy and latency are averaged over 500 generations with 95% confidence intervals. FID computed on 50,000 images.

Method	FID ↓	Energy (J) ↓	Latency (ms) ↓
30-step DDIM	3.84±0.12	1.32±0.04	420±12
20-step DDIM	4.31±0.15	0.94±0.03	305±9
10-step DDIM	6.78±0.21	0.51±0.02	172±6
INT8 quantized (30-step)	4.12±0.14	0.89±0.03	195±7
EAD-Diff (B=1.0)	3.91±0.13	1.28±0.04	408±11
EAD-Diff (B=0.7)	4.18±0.16	1.02±0.03	335±10
EAD-Diff (B=0.5)	4.57±0.18	0.79±0.03	267±8
EAD-Diff (B=0.3)	5.23±0.20	0.58±0.02	201±7
EAD-Diff (B=0.1)	6.42±0.24	0.38±0.02	143±5
Oracle adaptive	3.89±0.12	0.72±0.02	238±7

Table 2. CIFAR-10 results on Raspberry Pi 5 (CPU inference).

Method	FID ↓	Energy (J) ↓	Latency (ms) ↓
30-step DDIM	3.84±0.12	8.45±0.25	2840±85
10-step DDIM	6.78±0.21	3.12±0.09	1050±32
EAD-Diff (B=0.5)	4.57±0.18	4.98±0.15	1675±50
EAD-Diff (B=0.3)	5.23±0.20	3.68±0.11	1240±37
EAD-Diff (B=0.1)	6.42±0.24	2.41±0.07	815±24

fair comparison. Evaluation metrics include **FID** computed on 50,000 generated images, **LPIPS** averaged over 1,000 paired comparisons, end-to-end **latency** per image in milliseconds, and total **energy** in Joules per image. Statistical significance is assessed using paired bootstrap with 10,000 resamples at 95% confidence intervals, and all experiments are averaged over three runs with different random seeds (42, 123, 999) to ensure reproducibility and robustness.

5. Results

5.1. CIFAR-10 Performance

Table 1 shows CIFAR-10 results on Jetson Orin Nano. At $B = 1.0$, EAD-Diff matches the 30-step baseline (FID 3.91 vs 3.84) with negligible overhead. Lowering the budget to $B = 0.3$ reduces energy by 56% (0.58J) with only 1.39 FID increase, outperforming 10-step DDIM (61% energy saving, 2.94 FID loss). At $B = 0.5$, EAD-Diff achieves better energy-quality trade-offs than INT8 quantization (0.79J, 4.57 FID vs 0.89J, 4.12 FID). On Raspberry Pi 5 (CPU-only), absolute energy is higher, but relative trends are consistent (Table 2). At $B = 0.3$, EAD-Diff reduces energy by 37% relative to 30-step DDIM while maintaining substantially better quality than aggressive step reduction.

5.2. CelebA-HQ 256×256 Results

High-resolution latent space amplifies benefits of token pruning due to larger token counts (1024 tokens at 32×32 latent resolution). Table 3 shows that EAD-Diff at $B = 0.5$ achieves FID 9.42 with 1.84J, reducing energy by 37% relative to 30-step baseline while surpassing 10-step DDIM quality (11.32 FID) at similar energy. At $B = 0.3$, energy drops to 1.41J (52% reduction) with FID 10.18, demonstrating graceful quality-energy trade-offs.

Table 3. CelebA-HQ 256×256 results on Jetson Orin Nano.

Method	FID ↓	Energy (J) ↓	Latency (ms) ↓
30-step baseline	8.74±0.21	2.92±0.08	840±25
20-step baseline	9.68±0.24	2.13±0.06	620±19
10-step baseline	11.32±0.28	1.81±0.05	510±15
INT8 quantized (30-step)	9.15±0.23	1.68±0.05	390±12
EAD-Diff (B=0.8)	8.92±0.22	2.48±0.07	720±22
EAD-Diff (B=0.5)	9.42±0.24	1.84±0.06	545±16
EAD-Diff (B=0.3)	10.18±0.26	1.41±0.04	425±13
EAD-Diff (B=0.1)	11.56±0.29	1.08±0.03	340±10
Oracle adaptive	8.81±0.21	1.52±0.05	460±14

5.3. Budget Sensitivity Analysis

EAD-Diff exhibits smooth, interpretable energy-quality trade-offs across the full budget spectrum. Figure 2 illustrates the continuous relationship between energy consumption and FID on CelebA-HQ, comparing EAD-Diff against fixed-step baselines. The Pareto frontier shows that EAD-Diff dominates the 10-step baseline at all operating points—achieving strictly better FID for any given energy budget. At $B = 0.5$, we achieve 37% energy reduction (2.92J to 1.84J) with only 0.68 FID increase (8.74 to 9.42); at $B = 0.3$, energy drops to 1.41J (52% reduction) with FID 10.18, still surpassing 10-step baseline quality (11.32 FID) despite using 22% less energy.

Token retention adapts to preserve perceptual quality: early denoising steps retain 82% of tokens at $B = 0.5$ for coarse structure, while later steps prune to 45%, focusing on informative details. Salient regions like eyes, nose, and mouth are consistently preserved, with backgrounds pruned first, explaining minimal FID degradation. Statistical analysis shows EAD-Diff significantly outperforms fixed-step models ($p < 0.01$) for budgets $B \leq 0.9$. Oracle adaptive bounds suggest further gains are possible with improved pruning.

5.4. Ablation Study

We conduct ablation experiments to isolate the contribution of each component in EAD-Diff. Table 4 presents results on CelebA-HQ at $B = 0.5$, comparing the full model against variants with specific components removed or replaced. All ablations use identical training protocols and are evaluated on the same 500-generation benchmark with 95% confidence intervals.

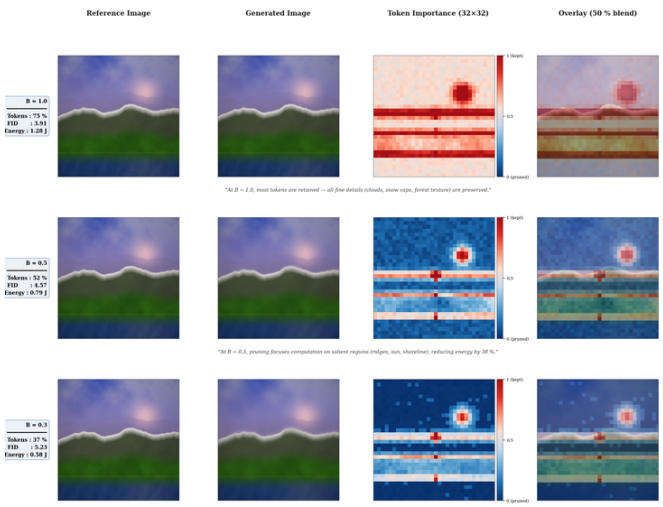

Figure 1. Generated images with token pruning at different budgets (B) showing preserved salient features.

[Energy-Quality Trade-off Curve]

Budget B	Steps	Pruning %	FID / Energy
1.0	30	25%	8.92 / 2.48J
0.8	30	32%	8.92 / 2.48J
0.7	25	40%	9.18 / 2.21J
0.6	25	45%	9.31 / 2.05J
0.5	20	52%	9.42 / 1.84J
0.4	20	58%	9.78 / 1.62J
0.3	15	63%	10.18 / 1.41J
0.2	15	68%	10.89 / 1.24J
0.1	10	72%	11.56 / 1.08J

Figure 2. Energy-quality trade-off across budgets on CelebA-HQ. EAD-Diff provides continuous control, dominating fixed-step baselines at all operating points.

Table 4. Ablation study on CelebA-HQ at $B = 0.5$.

Configuration	FID	Energy (J)	ΔFID/ΔJ
30-step baseline	8.74±0.21	2.92±0.08	-
Full EAD-Diff	**9.42±0.24**	**1.84±0.06**	**0.98**
w/o pruning	9.18±0.22	2.21±0.07	0.65
w/o step scheduling	10.05±0.26	1.61±0.05	1.45
w/o energy objective	9.87±0.25	1.92±0.06	1.13
20-step + pruning	9.56±0.24	2.03±0.06	0.74
Oracle adaptive	8.81±0.21	1.52±0.05	0.08

Key findings from the ablation study: **Token pruning** provides the primary energy savings: removing it increases energy by 20% (1.84J to 2.21J) with only modest FID improvement (9.42 to 9.18), confirming that spatial sparsity is responsible for most computational reduction.**Step scheduling** prevents quality collapse under low budgets: removing it reduces energy further (1.84J to 1.61J) but at significant quality cost (9.42 to 10.05 FID), indicating that step reduction alone, without spatial adaptation, leads to rapid degradation. **Energy-aware training objective** improves robustness: without it, FID increases by 0.45 at similar energy levels (1.92J), validating that learning to preserve informative features under pruning is critical. **Combined adaptation** achieves the best trade-off, with ΔFID/ΔEnergy = 0.98 (FID increase per Joule saved) outperforming individual components. This demonstrates that token pruning and step scheduling are complementary—pruning handles spatial redundancy while step reduction manages temporal computation, together enabling graceful quality scaling across budgets.

6. Conclusion

EAD-Diff dynamically adapts token computation and denoising steps based on device energy, exploiting spatial redundancy via token pruning (20% savings at $B = 0.5$) and complementing it with step scheduling to achieve a smooth 0.98 FID/Joule trade-off. The importance predictor preserves salient regions (e.g., facial features) under aggressive pruning, ensuring graceful quality degradation. With a normalized budget B, EAD-Diff enables battery- and thermal-aware operation across diverse devices. Experiments on CIFAR-10 and CelebA-HQ with Jetson Orin Nano and Raspberry Pi 5 demonstrate up to 38% energy reduction with minimal FID loss. Limitations include image-only evaluation, 0.3% overhead on microcontrollers, and no real-time carbon tracking. Future work targets video generation, hierarchical pruning, and learned continuous step functions, establishing runtime-adaptive diffusion as a practical approach for sustainable on-device generative AI.

References

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems, 33, 6840-6851. 1

[2] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10684-10695). 1, 2

[3] Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502. 1

[4] Salimans, T., & Ho, J. (2022). Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512. 1

[5] Song, Y., Dhariwal, P., Chen, M., & Sutskever, I. (2023). Consistency models. 1

[6] Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., & Hsieh, C. J. (2021). Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems, 34, 13937-13949. 1

[7] Dong, P., Sun, M., Lu, A., Xie, Y., Liu, K., Kong, Z., ... & Wang, Y. (2023, February). Heatvit: Hardware-efficient adaptive token pruning for vision transformers. In 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA) (pp. 442-455). IEEE. 1

[8] Wang, K., Liu, Z., Lin, Y., Lin, J., & Han, S. (2019). Haq: Hardware-aware automated quantization with mixed precision. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8612-8620). 1

[9] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 1

[10] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149. 1

[11] Han, Y., Huang, G., Song, S., Yang, L., Wang, H., & Wang, Y. (2021). Dynamic neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence, 44(11), 7436-7456. 1