EdgeDiffuse: Optimization of Image Generation Models for Edge

Students: Ethan Li, Sean He, Yijie Huang, Pin-Yu Lin, Yichen Jin, Cecilia Hu

Motivation

Stable Diffusion has transformed creative AI — but remains **cloud-dependent**. Edge inference enables **privacy-preserving, fully offline generation** critical for IoT and embedded environments, yet standard SD models require **4+ GB VRAM** — far beyond what low-cost hardware like the **Orange Pi RK3588** can provide.

Stable Diffusion 850M+ parameters 4GB+ VRAM required → **Deployment Gap** Resource Constraints → **Orange Pi** Limited RAM Fully offline

Requirements

⇨ Offline text-to-image generation

≥20%	≥25%	On-device	Accept.
Quantization compression	Pruning compression	Orange Pi RK3588 (ARM + NPU)	Perceptual quality* retained

*Quality threshold: LPIPS <0.05 excellent · <0.10 good · <0.20 acceptable | CLIP score typical range 0.20–0.35

Implementation

Stable Diffusion v1.5/Turbo 2.1 → 01 PRUNING **Taylor-Importance** 4 rounds x ~ 7% → 02 QUANTIZATION **INT8 PTQ W8A16** Conv + Linear → 03 DEPLOYMENT **ONNX → RKNN NPU inference** Unet on NPU

Pruning + Distillation

Strategy	Iterative soft-pruning
Importance metric	Taylor-Importance
Sensitivity metric	Latent Divergence
Rounds · per-round	4 rounds · ~7% each
Total compression	25.3% (865M → 647M)
Post-distillation teacher	DreamShaper v8 · 50K steps
Distillation loss	Feature MSE + Noise MSE

SD 1.5 (20 steps) — Timestep-weighted sensitivity; final step comparison only. More steps provide robustness, making per-step weighting tractable.

SD Turbo 2.1 (4 steps) — Every intermediate step compared to teacher. Large timestep gaps make weighting impractical; each step is critical.

Quantization

Method	Weight-only PTQ
Scheme	Symmetric linear, per-channel
Precision	W8A16 (INT8 weights, FP16 activations)
Target layers	Conv + Linear in U-Net
Calibration	Histogram (beats MinMax by 2.6× scale accuracy)

INT4 rejected — weight error 312× larger than INT8; no native PyTorch INT4 type (stored as INT8, zero VRAM benefit); LPIPS 0.689 vs 0.023, structural collapse observed.

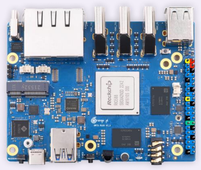
Implementation (cont.)

Edge Deployment

Hardware	Orange Pi RK3588 16 GB × 2
Conversion pipeline	PyTorch → ONNX → RKNN
U-Net inference	NPU (RK3588)
CLIP + VAE inference	CPU
● **Pruned model on NPU**	⇨ **Validated**
● **Full pipeline on NPU**	⇨ **Pending toolkit support:** Quantized model RKNN conversion not feasible with current toolkit

Fake INT8 on CPU is counterproductive — compute unit remains FP32; dequant overhead causes 1.87× slowdown vs FP32. Real speedup requires NPU with native INT8 compute units.

Experimental Setup



Target Edge Hardware

Device	Orange Pi RK3588
CPU	ARM Cortex-A76/A55
NPU	6 TOPS
RAM	16 GB
Mode	Fully offline

Reference Platform

Instance	Amazon EC2 G5
GPU	NVIDIA A10G
Role	Cloud baseline
Used for	Compression eval

Dataset & Evaluation Protocol

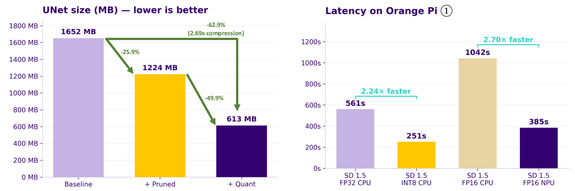
Eval set	Fixed prompt set, 10 categories
Resolution	512 × 512
Seed	Fixed (same seed across all runs)
Eval sample size	n = 200–500 per experiment
Calibration set	Sampled prompts for INT8/INT4

Experimental Design

#	Model	Hardware	Pruning	Quant	Precision	Metrics
1	Stable Diffusion Turbo 2.1	**Cloud CUDA** A10G	—	—	FP32 (Baseline)	• Size • Latency • LPIPS • CLIP score
2			✓	—	FP16 (+Pruned)	
3			✓	✓	W8A16 (+Quant)	
4	Stable Diffusion 1.5	**Orange Pi** CPU	—	—	FP32 (Baseline)	Latency • UNet • Total
5			✓	—	FP16 (+Pruned)	
6			✓	✓	INT8 (+Quant)	
7		**Orange Pi** CPU + NPU	✓	—	FP16	

Results

"cherry blossoms in full bloom, kyoto temple, spring" — generated at 512×512, upscaled to 1060×1060 for print.

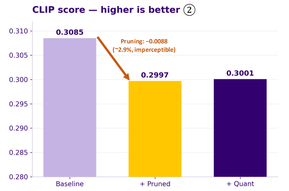


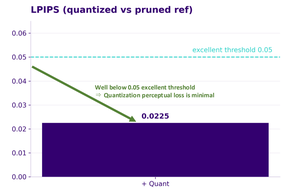
2.69× Total compression vs baseline

2.70× NPU speedup vs CPU baseline

99.2% CLIP score retained NPU vs CPU

0.023 LPIPS (excellent <0.05 threshold)





① SD 1.5, 20 steps. SD Turbo 2.1 (4 steps) projects ~5x speedup, but preliminary results showed unexpected deviations pending further analysis — only SD 1.5 shown here.

② LPIPS measures perceptual similarity, not absolute quality. Since pruning shifts the model's output distribution, LPIPS is reported only for the quantized model relative to the pruned reference, isolating the perceptual cost of quantization alone.

SD Turbo 2.1 · Compression Pipeline (Cloud CUDA · n=500, 4-step)

Metrics	Baseline	+Pruned	+Quant (W8A16)
UNet size (MB)	1651.6	1224.1 (1.35×)	613.0 (2.69×)
LPIPS (vs pruned ref)	—	0.000	0.023
CLIP score	0.309	0.300	0.300
CLIP δ vs baseline	0.000	−0.009	−0.009

Conclusion

- ✓ **Compression validated:** 2.69× size reduction with excellent quality (LPIPS 0.023, CLIP virtually unchanged), latency reduced.
- ✓ **NPU deployment succeeds:** UNet runs on the Orange Pi NPU via RKNN.
- ✗ **INT8 optimal:** 2× compression over pruned with negligible quality loss; INT4 rejected (312× error, structural collapse)
- ○ **NPU acceleration blocked:** toolkit cannot convert the quantized UNet; NPU shows no speedup over CPU yet

⇒ Pruning and quantization are validated and on-device NPU inference is feasible — unlocking real acceleration awaits more mature RKNN toolkit support.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

Acknowledgements

We gratefully acknowledge Arman Kazemi, Jin Wang, and Sankalp Dayal (Amazon) for industry mentorship, Prof. Radha Poovendran for faculty guidance, and Qifan Lu and Steven Sun for their additional support. We also thank the UW Research Computing Club for providing computational resource access and the Student Technology Fee for financial support.