Evaluating Slang for View Synthesis

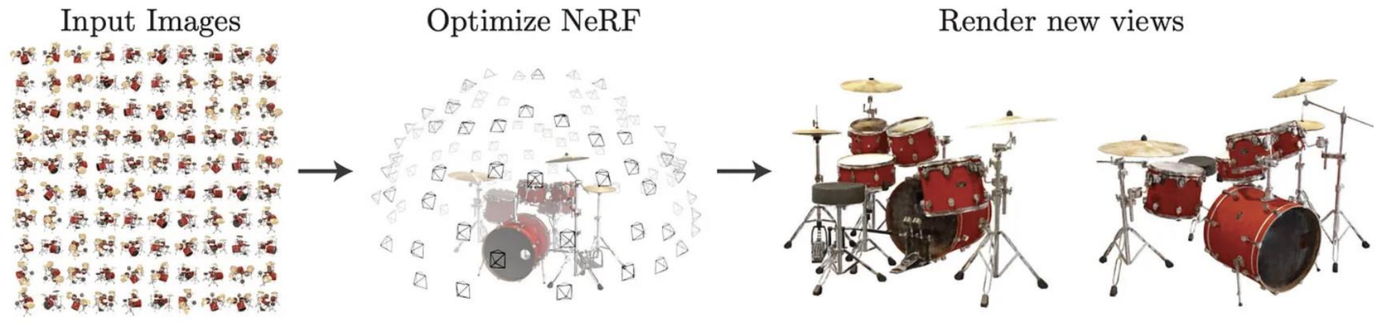
David Choi, Taiki Yoshino, Rick Rodness, Hayden Kwok
Advised By: Ravi Ramamoorthi, Tzu Mao Li

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Computer Science and Engineering

ERSP
UC San Diego
Early Research Scholars Program

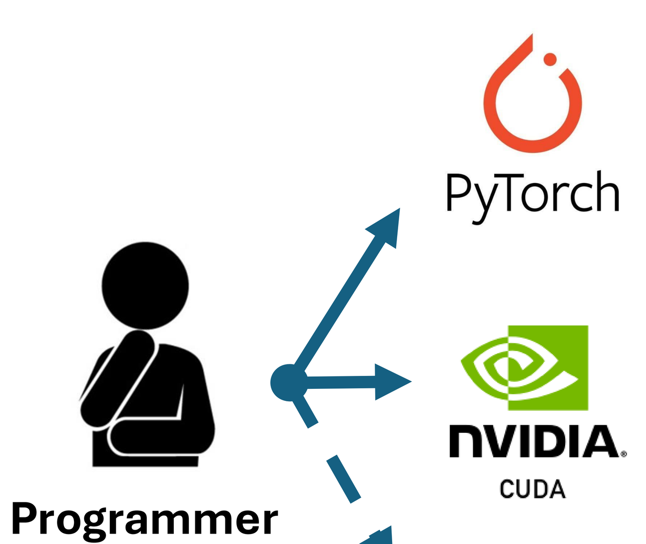
Background

View synthesis (aka 3D reconstruction) is a field that aims to generate 3D models from 2D images. Many approaches (NeRF, Gaussian Splatting, etc.) leverage insights from machine learning, specifically optimization through differentiation.



(from [1]):
Take 100 pictures of a drum from different views of a hemisphere. Using the images, optimize an MLP to represent the 3D scene of the object and render new images from arbitrary camera views.

These models are usually implemented in either PyTorch or custom CUDA kernels. Each option carries tradeoffs.



PyTorch Modules:
- Auto-diff. (easier dev.)
- Slow data access. (higher perf.)

Self-defined CUDA Kernel:
- No auto-diff. (harder dev.)
- Fast memory access (better perf.)

New Option

SLANG:
- auto-diff. (easier dev?)
- Fast memory access (better perf?)

Slang is a new programming language boasting both auto-differentiation and fast memory access.

Motivation

The Slang.D paper claims:

"SLANG.D demonstrates that [...] we can achieve a substantial advancement in expressiveness, performance and usability of an AD [Auto Differentiation] system" [2]

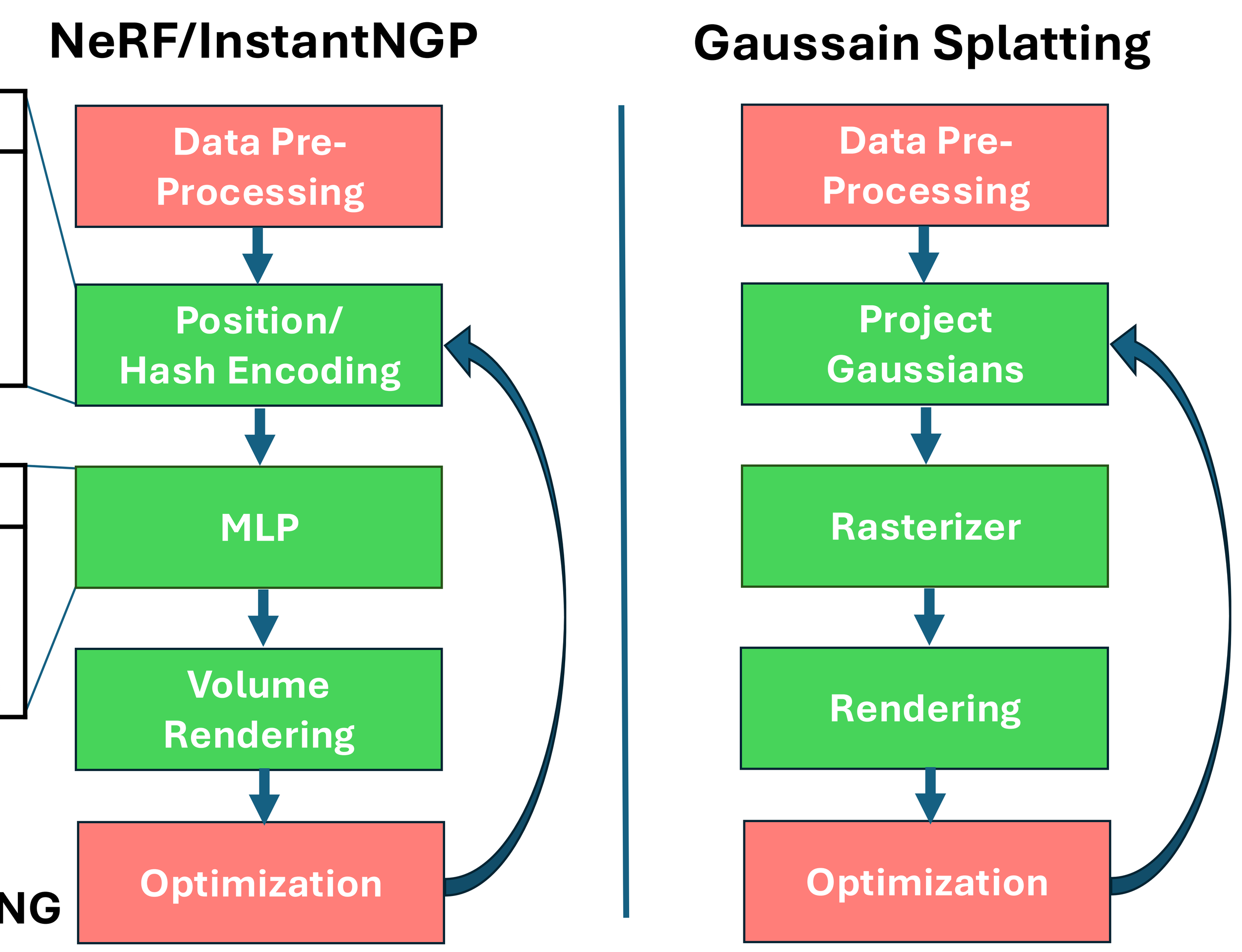
The purpose of our research is to affirm or reject the claims of the original Slang.D paper and the viability of Slang in differentiable rendering applications.

Methodology

To verify the performance of Slang, we implemented three view synthesis algorithms: NeRF, InstantNGP, and 3D Gaussian Splatting. We replaced the computationally heavy processes with Slang. We measured the average forward pass, forward + backward pass over 1000 iterations, and the final PSNR.

Encoding Parameters	
Multi-Res Level	16
Number of feature dimensions per entry	2
Position Encoding Level	5

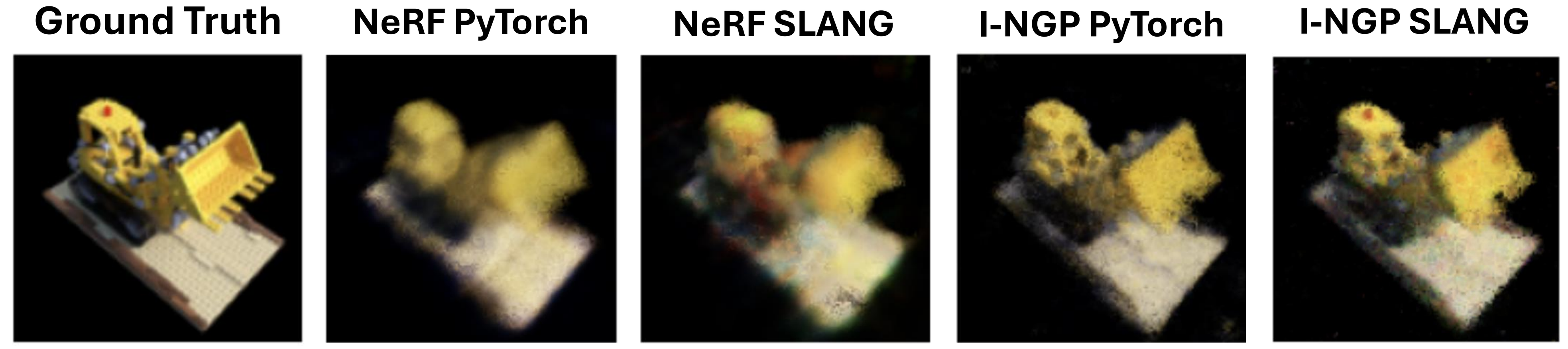
MLP Parameters	
Number of Neurons	32
Number of Layers	3
Picture size	128 x 128



Results

	NeRF		InstantNGP		3DGS	
	PyTorch	SLANG	PyTorch	SLANG	PyTorch	SLANG*
Fwd (ms)	2.60	0.657	9.91	8.42	8.03	7.45
Fwd + Bwd (ms)	4.61	19.3	43.1	54.6	12.82	12.50
PSNR	21.4	20.1	22.7	22.8	22.4	22.3

Ground Truth | NeRF PyTorch | NeRF SLANG | I-NGP PyTorch | I-NGP SLANG

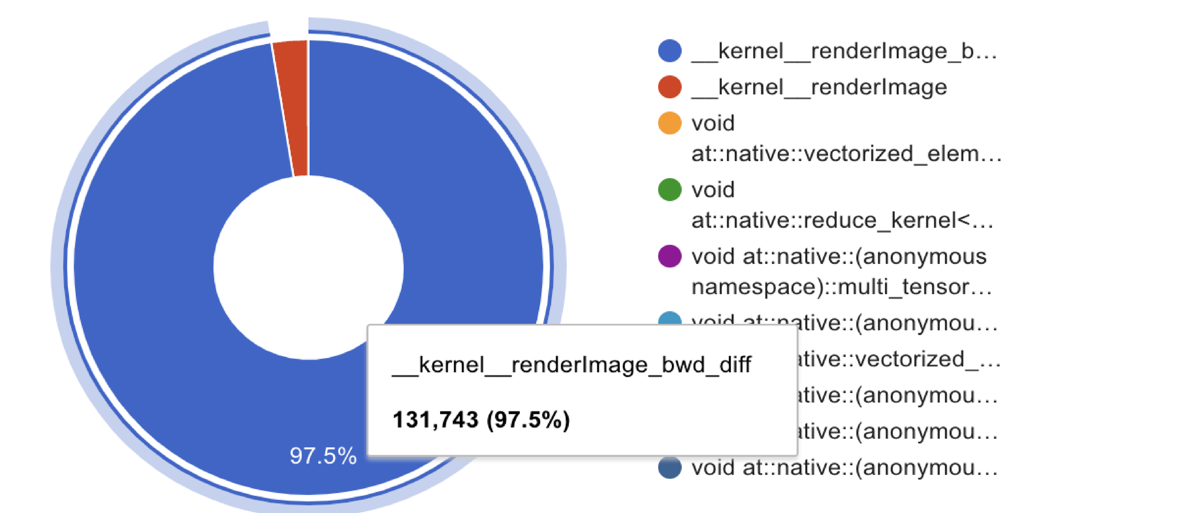


Ground Truth | 3DGS PyTorch | 3DGS Slang*

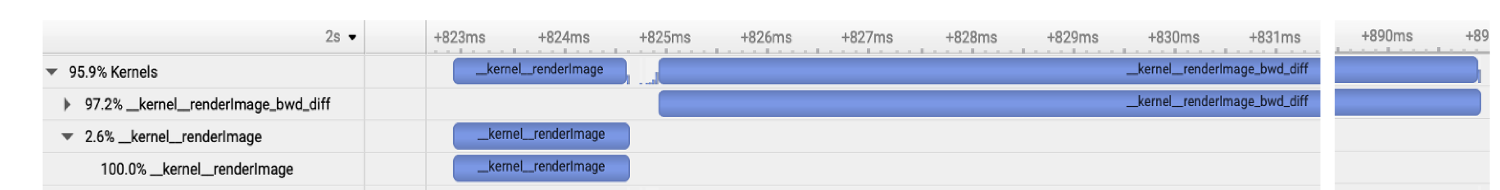
*We were only able to replace the projection of gaussians and the rasterizer for 3DGS in Slang so much of our discussion and conclusions will focus on NeRF and Instant-NGP

Discussion

For the performance of NeRF, we found rendering (fwd) in Slang was 4x faster than PyTorch. However, training (fwd + bwd) in Slang was 4x slower than in PyTorch.



PyTorch Profiler with TensorBoard



NVIDIA Nsight Systems

We found +97% of training time was from the backward pass generated by Slang's auto diff. We identified 2 plausible causes:

1. Our computation graph became too complicated in our forward implementation
2. Slang's auto diff is not well optimized.

Conclusion

In our research we could not affirm the claims of the Slang.D paper. While there were promising performance gains in the forward pass, code generated with Slang's auto diff was inferior to PyTorch. further analysis is required.

Acknowledgments

Thank you to our advisors Tzu-Mao Li and Ravi Ramamoorthi for their guidance. Additional thanks to Javahir Abbasova, Christine Alvarado, and Lisa Huang.

References

[1] Ben Mildenhall, et al. Nerf: Representing scenes as neural radiance fields for view synthesis. In ECCV, 2020.
[2] Bangaru, S. P., et al. (2023). SLANG.D: Fast, Modular and Differentiable Shader Programming. In ACM Transactions on Graphics (Vol. 42, Issue 6, pp. 1–28). Association for Computing Machinery (ACM).