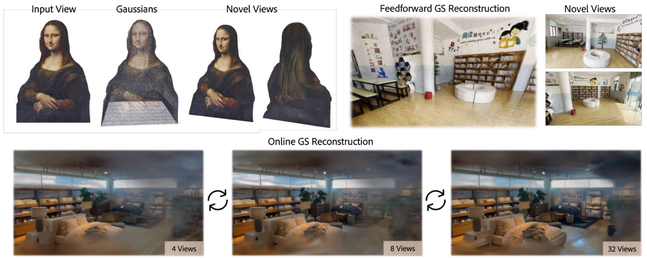
tttLRM: Test-Time Training for Long Context and Autoregressive 3D Reconstruction

Chen Wang, Hao Tan, Wang Yifan, Zhiqin Chen, Yuheng Liu, Kalyan Sunkavalli, Sai Bi, Lingjie Liu†, Yiwei Hu†

University of Pennsylvania, Adobe Research

CVPR JUNE 3-7, 2026 DENVER COLORADO

TL,DR: We design a large reconstruction models (LRM) with Test-time training that supports high-resolution, long sequence and feedforward or online 3D reconstruction with linear complexity.



Introduction and Motivation

Goal: reconstruct explicit 3D representations from long sequence (and streaming) input images

Problems with Existing 3D Reconstruction Models:

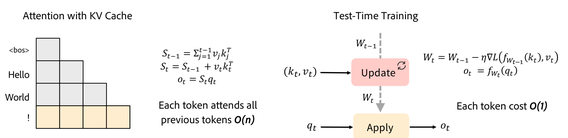
1). Existing LRM has attention layers, which limits scalability

2). Latent 3D models like LVSM has slow rendering, lack controllability and interpretability

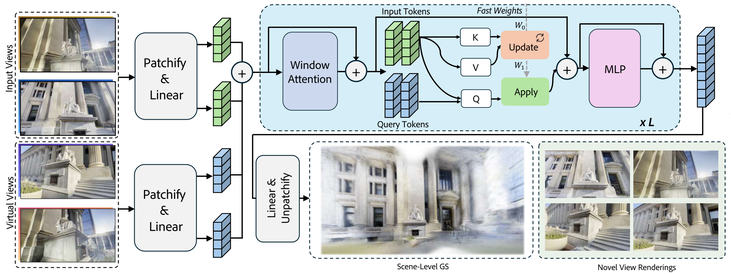
Key Idea: Compress inputs into the fast weights of TTT layers and decode to explicit 3D representations when needed

Our Contributions:

1) First TTT-based model for feedforward and online explicit 3D reconstruction with linear complexity
2) A scalable and unified framework that interprets TTT fast weights for controllable explicit 3D representations
3) State-of-the-art GS quality on both object and scene-level datasets with high-efficiency



Method



Overview

Given a set of input images, we project them into tokens and use them to update fast weights, we have another set of tokens from virtual views to query the fast weights and decode into 3D representations

Model Architecture

Tokenization: patchify and project images to tokens:

$$\{\mathbf{T}_{i,j}\}_{i=1}^{N}{}_{j=1}^{HW/p^2} = \text{Tokenize}\big(\text{Patchify}([\{\mathbf{I}_i\}_{i=1}^{N}, \{\mathbf{R}\}_{i=1}^{N}])\big)$$

Window Attn: attention for each view for local relationships:

$$\mathbf{T}_i = \mathbf{T}_i + \text{WinAttn}(\mathbf{T}_i),$$

Fast Weight Update: Use input tokens to update fast weights with Muon:

$$W = \text{Update}(\{\mathbf{T}_i\}_{i=1}^{N}),$$
$$\mathbf{T}_i = \text{Apply}(W, \mathbf{T}_i)$$

Query and Decode: Use virtual tokens to query fast weights:

$$\mathbf{T}_i^{\text{v}} = \text{Apply}(W, \mathbf{T}_i^{\text{v}})$$

Depending on 3D representations, virtual tokens can either be image, triplanes, or others.

Streaming 3DGS Reconstruction

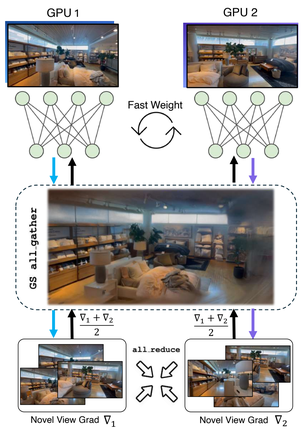
Algorithm 1 Streaming 3DGS Reconstruction

Input: Reconstructor $\mathcal{F}$ with initial fast weights W_0; input/query view batches $\{(\mathcal{I}_{(b)}, \mathcal{I}^v_{(b)})\}_{b=1}^{B}$

Output: Reconstructed GS G

```
W ← W_0
for b = 1 to B do
    _, W ← F(W, I_(b))
    G_(b), _ ← F(W, I^v_(b))
end for
return G_(B)
```

Distributed Training Across Multiple GPUs



Input tokens are shared, each GPU predicts partial GS and then to be merged and back-propagated.

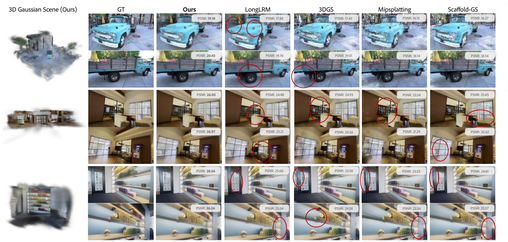
Training

$$\mathcal{L} = \mathcal{L}_{\text{RGB}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{opacity}}\mathcal{L}_{\text{opacity}}$$

Use monocular depth estimation regularize GS position and prune opaque GS.

Results and Discussion

Scene-Level 3DGS Reconstruction (960x540, 16-64 views)

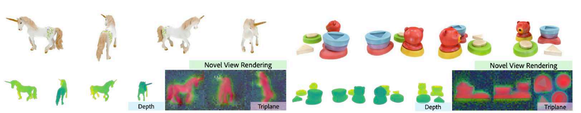


Views	Method	Time↓	DL3DV-140 PSNR↑	DL3DV-140 SSIM↑	DL3DV-140 LPIPS↓	Tanks&Temples PSNR↑	Tanks&Temples SSIM↑	Tanks&Temples LPIPS↓
16	3D GS$_{30k}$	13m	21.20	0.708	0.264	16.76	0.598	0.334
	Mip-Splatting$_{30k}$	13m	20.88	0.712	0.274	16.82	0.616	0.332
	Scaffold-GS$_{30k}$	16m	**22.13**	**0.738**	**0.250**	**17.02**	**0.634**	**0.321**
	Long-LRM (16v model)	**0.4s**	22.66	0.740	0.292	17.51	0.555	0.408
	Ours (single model)	3.6s	**23.60**	**0.784**	**0.255**	**18.15**	**0.613**	**0.360**
32	3D GS$_{30k}$	13m	23.60	0.779	0.213	18.10	0.688	0.269
	Mip-Splatting$_{30k}$	13m	23.32	0.784	0.217	18.39	**0.700**	**0.262**
	Scaffold-GS$_{30k}$	16m	**24.77**	**0.805**	**0.205**	**18.41**	0.691	0.290
	Long-LRM (32v model)	**1s**	24.10	0.783	0.254	18.38	0.601	0.363
	Long-LRM (32v model w/ optim)	12s	24.99	0.809	0.243	18.69	0.623	0.360
	Ours (single model, AR)	7.5s	24.31	0.803	0.237	18.96	0.653	0.322
	Ours (single model)	7.2s	**25.07**	**0.822**	**0.215**	**19.22**	**0.662**	**0.305**
64	3D GS$_{30k}$	13m	26.55	0.852	**0.164**	20.78	**0.778**	**0.205**
	Mip-Splatting$_{30k}$	13m	26.29	0.850	0.166	20.08	0.759	0.220
	Scaffold-GS$_{30k}$	16m	**27.07**	**0.857**	0.173	**20.96**	0.768	0.240
	Long-LRM (64v model)	**3.7s**	24.63	0.799	0.243	19.11	0.627	0.346
	Ours (single model, AR)	15.2s	24.81	0.814	0.225	19.80	0.675	0.308
	Ours (single model)	14.8s	**25.95**	**0.844**	**0.195**	**20.31**	**0.700**	**0.274**

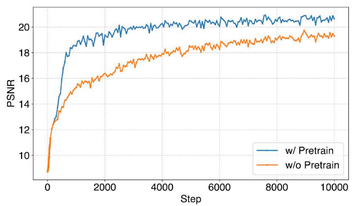
Object-Level 3DGS Reconstruction (1024x1024)

Method	Resolution	Views	Time (s) ↓	PSNR↑	SSIM↑	LPIPS↓
GS-LRM [68]	256 × 256	8	0.1	31.55	0.964	0.028
Ours		8	**0.1**	**33.14**	**0.972**	**0.024**
GS-LRM [68]		8	0.7	32.83	0.969	0.029
Ours		8	**0.3**	**34.02**	**0.974**	**0.025**
GS-LRM [68]	512 × 512	16	2.5	33.55	0.976	0.023
Ours		16 (10 V.)	**0.8**	**34.67**	**0.978**	**0.022**
GS-LRM [68]		24	5.5	33.26	0.976	**0.022**
Ours		24 (10 V.)	**1.1**	**34.80**	**0.979**	**0.022**

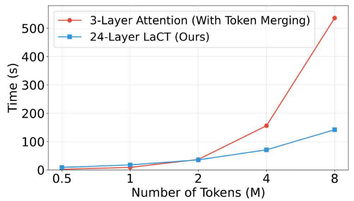
Decode into Triplanes



Pretraining from TTT-LVSM



Comparison with Attention Layers



Project Page, Video and Code