NEURAL INFORMATION PROCESSING SYSTEMS

Direct Conditional Parameterization for N-Dimensional Splatting

Zhongpai Gao*, Benjamin Planche, Van Nguyen Nguyen, Meng Zheng, Anwesa Choudhuri, Arun Innanje, Terrence Chen, Ziyan Wu
United Imaging Intelligence, Boston MA, USA

UNITED IMAGING Intelligence

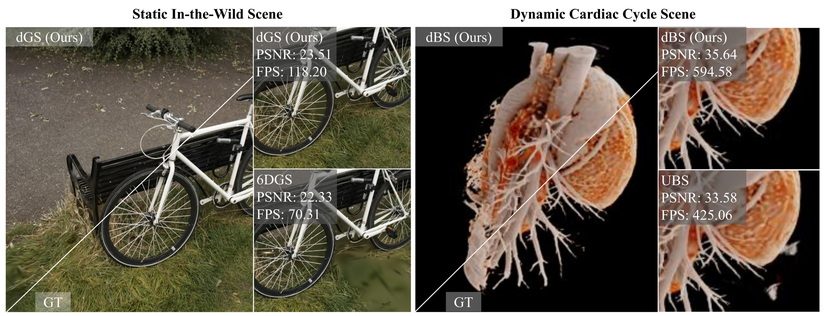
Project page | Job openings

Motivation

N-dimensional splatting lifts each primitive into a joint Gaussian over position and view/time, then conditionally slices it at every query.

Slicing is the bottleneck. N-DGS and UBS invert the query block, build a regression matrix and correct the covariance for every primitive, every frame.

Entangled coordinates. Opacity, position and covariance share one joint covariance, so they cannot be controlled separately.



Contributions

Direct conditional parameterization. A Cholesky precision factor **L** drives opacity; a factorized regression with learnable coupling **Λ** drives position. The underlying coordinates form a globally bijective chart of joint Gaussian covariances.

Kernel-agnostic. dGS (Gaussian) and dBS (Beta), each tested against its covariance-derived twin: N-DGS and UBS.

Fused CUDA kernels. 6.9–7.7× faster production slicing and 1.25–1.91× faster complete renders.

6.9–7.7×
faster slicing (A100, 1M prims)

+1.26 dB
best MCMC gain (dBS, 7DGS-PBR)

2.66×
CUDA FPS, Mip-NeRF 360 (dGS)

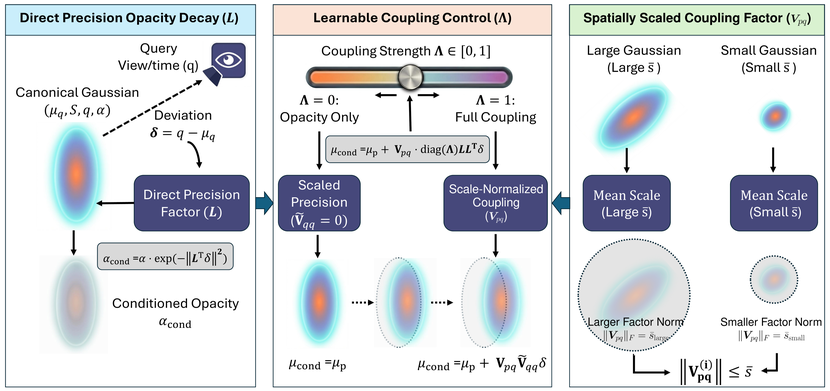
1.25–1.91×
complete-render speedup

Ablation: Coupling and Spatial Scale

	NeRF Synthetic		6DGS-PBR	
Config	**PSNR↑**	**LPIPS↓**	**PSNR↑**	**LPIPS↓**
Λ = 0 (dGS-O)	33.84	0.030	36.54	0.059
Λ = 1 (fixed)	33.78	**0.029**	37.31	0.053
dGS w/o spatial scale	31.26	0.046	34.78	0.075
Λ learned (dGS)	**33.91**	**0.029**	**37.75**	**0.052**

Standard 3DGS densification. Removing the spatial scale costs about 3 dB; learned Λ beats both fixed couplings.

Method: Read Out What Rendering Needs



Implicit (N-DGS / UBS): derive everything from a joint covariance

$$\boldsymbol{\mu}_{\text{cond}} = \boldsymbol{\mu}_p + \Sigma_{pq}\,\Sigma_{qq}^{-1}(\mathbf{q} - \boldsymbol{\mu}_q)$$
$$\Sigma_{\text{cond}} = \Sigma_{pp} - \Sigma_{pq}\,\Sigma_{qq}^{-1}\,\Sigma_{pq}^{\top}$$
$$\alpha_{\text{cond}} = \alpha \cdot \mathcal{K}\big(\mathbf{q} - \boldsymbol{\mu}_q;\, \Sigma_{qq}\big)$$

Direct (dGS, ours): parameterize what rendering reads

$$\alpha_{\text{cond}} = \alpha \cdot \exp\big(-\lambda_o \|\mathbf{L}^\top \boldsymbol{\delta}\|_2^2\big), \qquad \text{no inversion}$$
$$\boldsymbol{\mu}_{\text{cond}} = \boldsymbol{\mu}_p + \mathbf{V}_{pq}\,\text{diag}(\boldsymbol{\Lambda})\,\mathbf{V}_{qq}\,\boldsymbol{\delta}, \qquad \text{no Schur complement}$$
$$\mathbf{V}_{qq} = \mathbf{L}\mathbf{L}^\top, \quad \mathbf{V}_{pq} = \bar{s}\,\boldsymbol{\Theta}_{pq}, \quad \|\boldsymbol{\Theta}_{pq}\|_F = 1, \quad \boldsymbol{\delta} = \mathbf{q} - \boldsymbol{\mu}_q$$

No Inversion, No Schur Complement

Per primitive	N-DGS	UBS	dGS (ours)	dBS (ours)
Invert Σ_{qq}	Σ_{qq}^{-1}	Σ_{qq}^{-1}	—	—
Regression P	$\Sigma_{pq}\Sigma_{qq}^{-1}$	$\Sigma_{pq}\Sigma_{qq}^{-1}\Lambda_\beta$	—	—
Cov. correction	$P\Sigma_{pq}^{\top}$	$P\Sigma_{pq}^{\top}$	—	—
Slicing, C=3 (ms)	0.71	0.67	**0.10** (7.1×)	**0.10** (6.9×)
Slicing, C=4 (ms)	0.96	1.34	**0.12** (7.7×)	**0.13** (7.2×)

Production CUDA slicing at 1M primitives on an A100, excluding autograd and rasterization.

Same Recipe, Beta Kernel: dBS

Only the slicing step changes, so UBS's Beta decay drops in on $\mathbf{z} = \mathbf{L}^\top\boldsymbol{\delta}$; position reuses the bandwidth as coupling.

$$\alpha_{\text{cond}} = \alpha \prod_{i=1}^{C} \big(1 - \tanh(z_i^2)\big)^{\beta_i}, \quad \mathbf{z} = \mathbf{L}^\top\boldsymbol{\delta}, \quad \boldsymbol{\Lambda}_\beta = \min(\boldsymbol{\beta}/4, \mathbf{1})$$

Takeaways

- Parameterize the conditional quantities rendering reads, not a joint covariance.
- Same quality or better on 5 benchmarks, 2 kernels; up to +1.26 dB.
- 6.9–7.7× faster slicing; learned Λ beats both fixed couplings.

Results with MCMC Densification

Data	Method	#K	Time↓	FPS_{CUDA}↑	PSNR↑	SSIM↑	LPIPS↓
NeRF Synthetic	N-DGS	300	10.43	348.95	34.04	0.973	0.027
	dGS (ours)	300	**10.26**	**593.94**	**34.34**	**0.973**	**0.026**
	UBS	300	6.80	306.33	34.48	0.975	0.026
	dBS (ours)	300	**5.48**	**415.66**	**34.53**	**0.975**	**0.025**
Mip-NeRF 360	N-DGS	2,852	51.23	75.31	26.88	0.775	0.263
	dGS (ours)	2,574	**49.85**	**200.56**	**27.67**	**0.794**	**0.227**
	UBS	3,057	28.59	76.18	28.50	0.841	**0.183**
	dBS (ours)	3,057	**24.42**	**157.45**	**28.70**	**0.842**	0.185
6DGS-PBR	N-DGS	163	54.80	476.79	40.06	0.976	0.039
	dGS (ours)	163	**47.54**	**551.02**	**40.34**	**0.976**	**0.038**
	UBS	163	8.70	348.63	39.53	0.975	0.042
	dBS (ours)	163	**8.47**	**401.93**	**39.72**	**0.976**	**0.042**
D-NeRF	N-DGS	150	**10.31**	501.52	34.63	0.976	0.033
	dGS (ours)	150	10.61	**746.65**	**34.64**	**0.976**	**0.030**
	UBS	150	**8.37**	371.81	32.91	0.967	0.040
	dBS (ours)	150	9.34	**463.09**	**33.55**	**0.972**	**0.033**
7DGS-PBR	N-DGS	175	**28.44**	511.12	29.20	0.944	0.063
	dGS (ours)	175	31.12	**577.38**	**29.90**	**0.948**	**0.061**
	UBS	175	9.90	396.33	30.61	0.951	0.057
	dBS (ours)	175	**8.79**	**539.57**	**31.87**	**0.955**	**0.053**

1 run per scene; pairs share primitive caps (#K = thousands). Time in minutes; FPS_{CUDA} uses fused CUDA slicing.

Qualitative Comparison

