Variational Multi-scale Representation for Estimating Uncertainty in 3D Gaussian Splatting

Ruiqi Li, Yiu-ming Cheung

Department of Computer Science, Hong Kong Baptist University

{csrqli, ymc}@comp.hkbu.edu.hk

Background

- Uncertainty information in 3D representation is useful
  - Remove the noisy components in the model
  - Provide confidence maps to assess the quality of synthesized views and depth
  - Important in autonomous driving simulation and robotics navigation

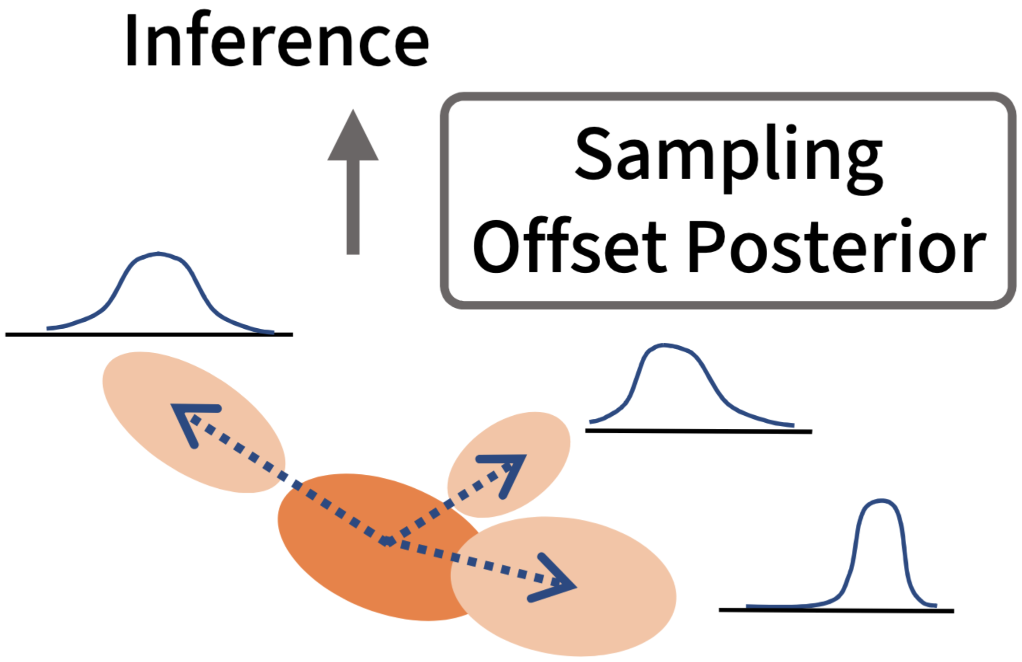
- Naïve 3DGS cannot provide uncertainty information
  - Uncertainty of model parameters
  - Uncertainty of predictions

Motivation

- In Bayesian learning
  - Computational efficiency
  - Exploration of model parameter space samples

- Multi-scale variational representation
  - Learn variational distribution of 3DGS attributes
  - Build local models of multiple scales to increase model parameter space diversity

Method: Multi-scale Variational Representation



- Offsetting the 3DGS attributes

$$\mathbf{p}^* = \mathbf{p} + \chi_{\mathbf{p}}; \qquad S^* = S + \chi_S,$$

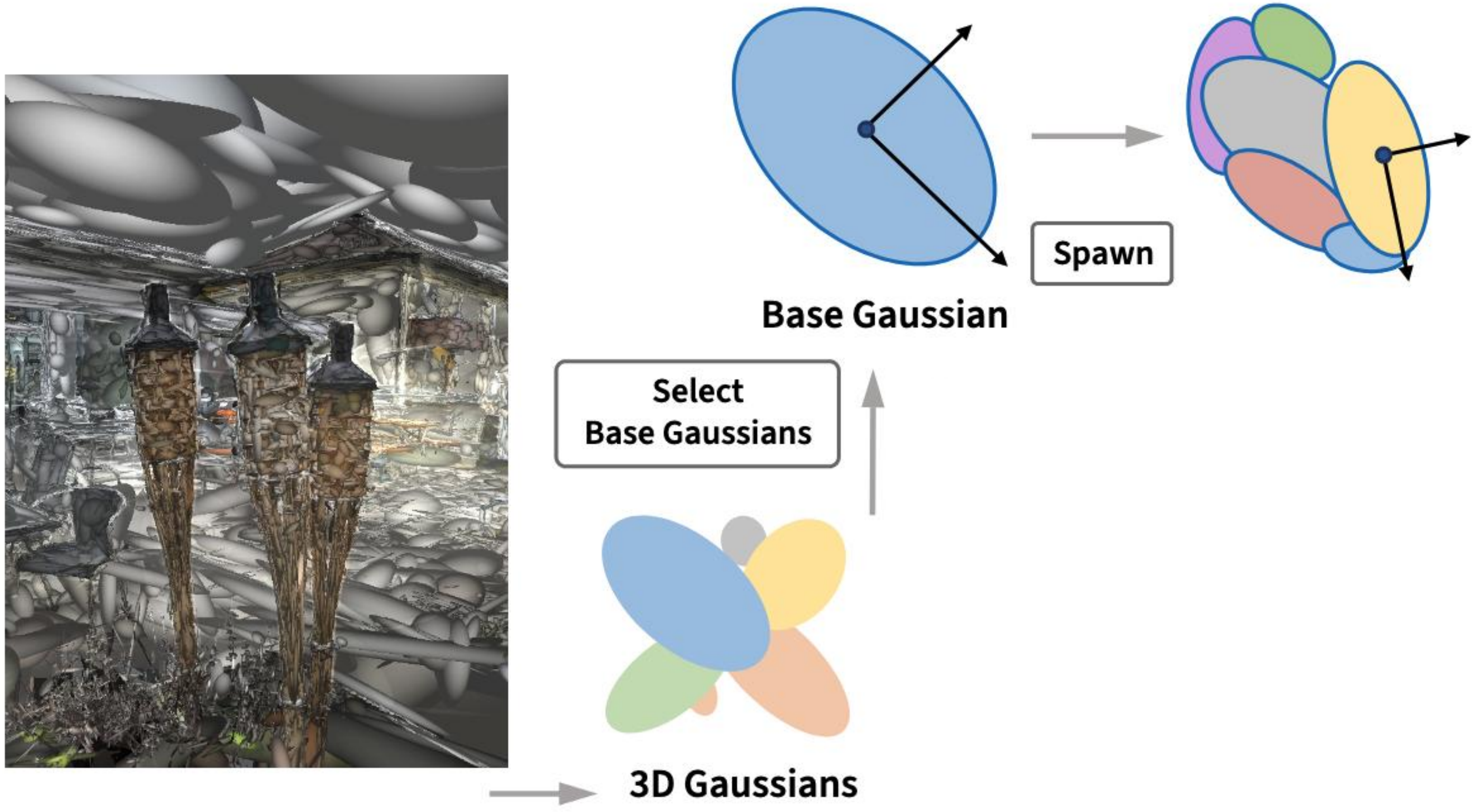
- Variational inference with multi-scale priors

$$q(\chi_S) \sim U(-S_{base} + (1 - 1/K)S_{base}, 0); \quad q(\chi_{\mathbf{p}}) \sim \mathcal{N}(0, \delta^2),$$

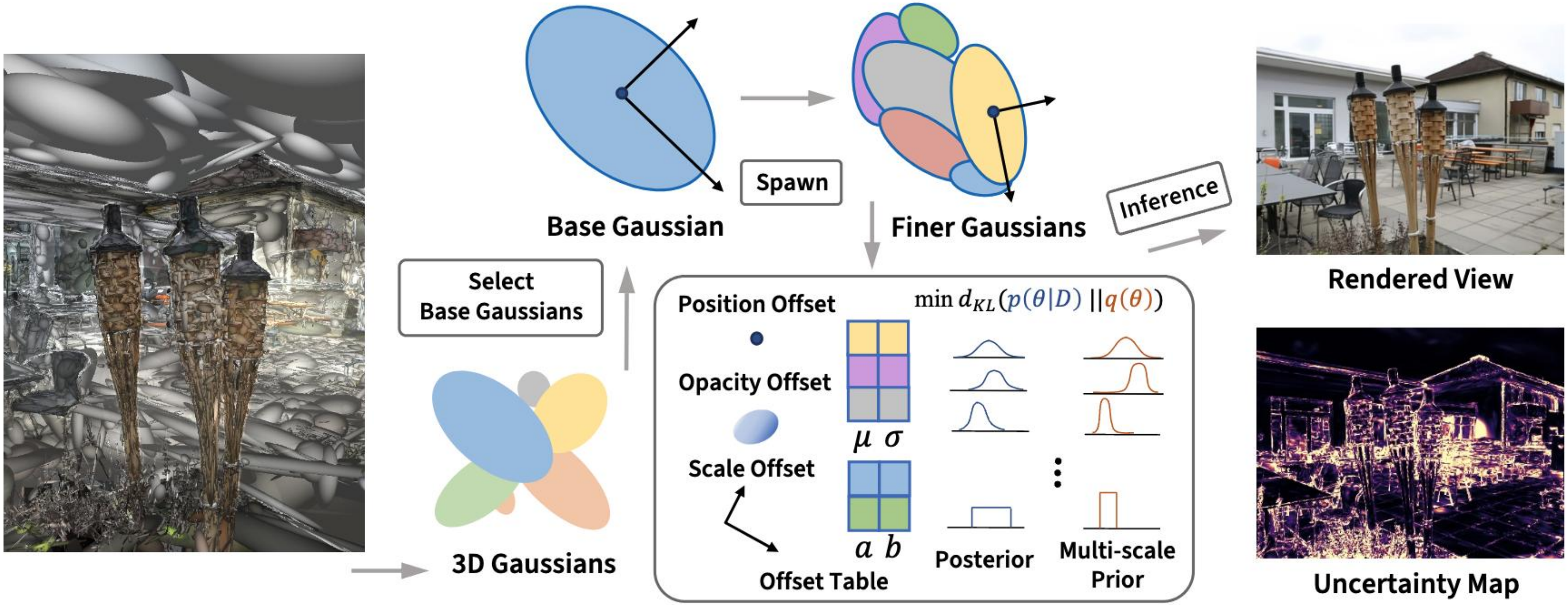
- Inference with learned posterior

$$p(\mathbf{c} \mid x, \mathcal{D}) = \mathop{\mathbb{E}}_{\chi \sim p(\chi|\mathcal{D})}[p(\mathbf{c} \mid x, \chi)] = \int p(\mathbf{c} \mid x, \chi) p(\chi \mid \mathcal{D}) \mathrm{d}\chi,$$

Method: Pipeline



Method: Pipeline



Algorithm: Pipeline

Algorithm 1 The pseudo-code of the training process of our uncertainty-aware 3DGS.

Input: Images and corresponding camera poses
Parameter: Maximum training step T; Spawn interval t; Threshold τ
Output: Trained scene representation with parameter $\boldsymbol{\theta}$; offset table ϕ

while step $< T$ **do**
  if step % t == 0 **then**
    Select $\mathcal{G}_{base} = \{\mathcal{G}_n | \sum ||\nabla\boldsymbol{\theta}|| > \tau_{\boldsymbol{\theta}}, ||S_n|| > \tau_S, \alpha > \tau_\alpha\}$
    Spawn $\mathcal{G}_{base}$, create offset table $\phi = \{\phi_S,\ \phi_{\mathbf{p}},\ \phi_\alpha\}$
    Assign prior $p(\chi)$ for offsets
  end if
  Sample offset χ, render image **c** and compute image loss $\mathcal{L}_1, \mathcal{L}_{SSIM}$
  Compute KL divergence $\mathcal{L}_{KL} = d_{KL}(p(\chi|\mathcal{D})||q(\chi))$
  Optimize $\boldsymbol{\theta}, \phi$ with total loss $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_{SSIM} + \mathcal{L}_{KL}$
end while

Experiments: Depth Uncertainty

Table 1: The depth uncertainty estimation performance on the LF dataset, quantified by the AUSE with MAE error.

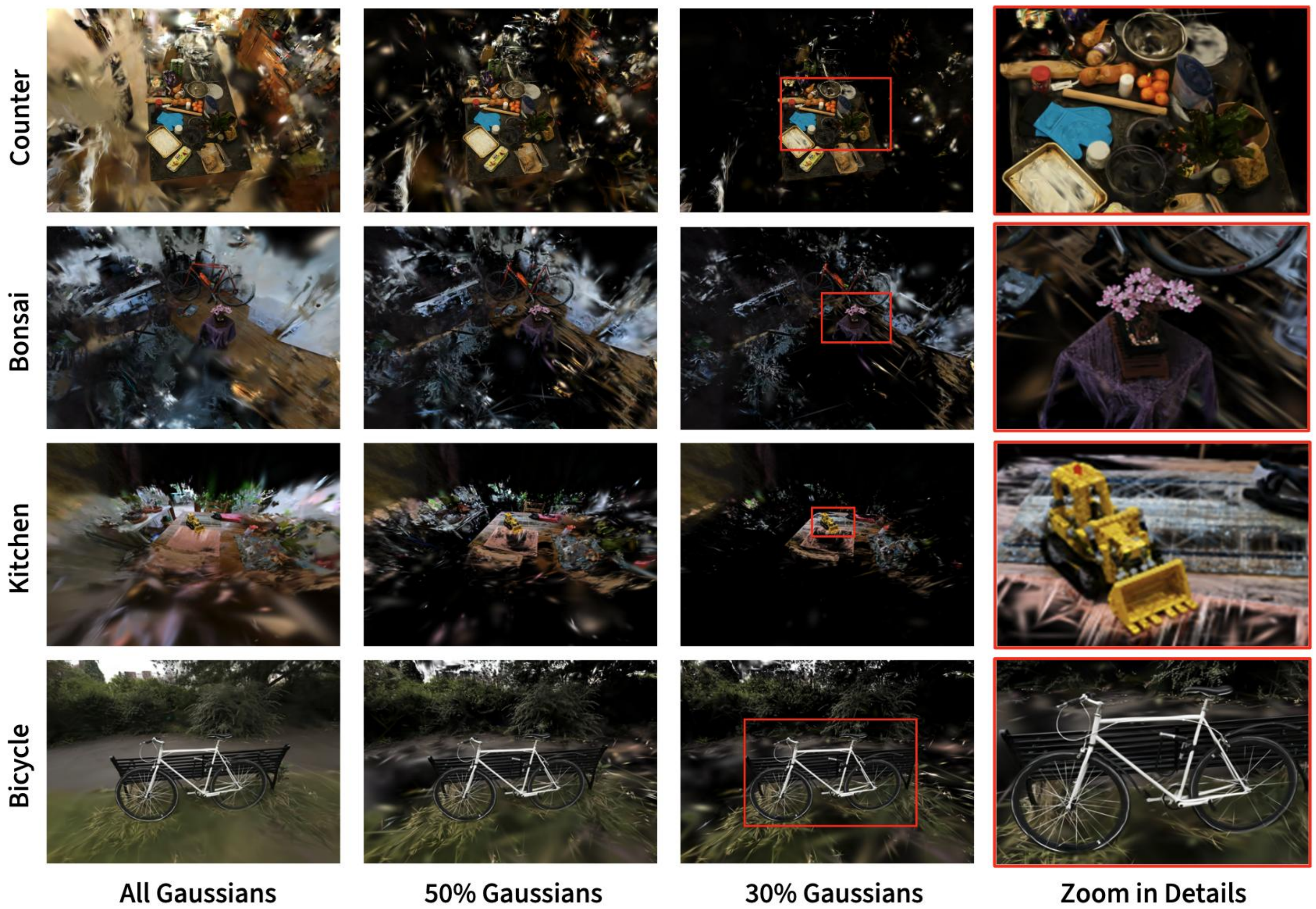
LF Dataset	africa	basket	statue	torch	**Average**
CF-NeRF	0.35	0.31	0.46	0.97	0.52
S-NeRF	0.66	0.38	0.67	0.74	0.61
Bayes' Ray	0.27	0.28	0.17	0.22	0.23
Ensemble GS ($\times$10)	0.16	0.22	0.17	0.26	0.20
Ours	0.19	0.13	0.21	0.23	0.19

Experiments: Novel View Quality and Uncertainty

Table 2: The performance of novel view rendering and uncertainty estimation on rendered images within the LF and LLFF dataset.

		Synthesized View Quality			Uncertainty Quality	
	Method	**PSNR** $\uparrow$	**SSIM** $\uparrow$	**LPIPS** $\downarrow$	**AUSE** $\downarrow$	**NLL** $\downarrow$
LF Dataset	CF-NeRF	24.32	0.835	0.202	0.49	-0.37
	S-NeRF	20.21	0.761	0.248	0.62	1.32
	Ensemble GS ($\times$10)	27.64	0.902	0.088	0.29	-0.34
	Ours	27.39	0.914	0.101	0.26	-0.30
LLFF Dataset	CF-NeRF	21.74	0.782	0.190	0.48	0.58
	S-NeRF	20.10	0.744	0.221	0.59	0.91
	Ensemble GS ($\times$10)	24.54	0.810	0.157	0.30	0.26
	Ours	23.97	0.806	0.172	0.32	0.23

Experiment: Noisy Gaussian Removal



Thank you for listening!

Code: Github Repo

Inquiries to Ruiqi Li: WeChat QR Code
or Mailto: csrqli@comp.hkbu.edu.hk