Robustness and Uncertainty in Neural Radiance Fields

Andrea Tagliasacchi (@taiyasaki)

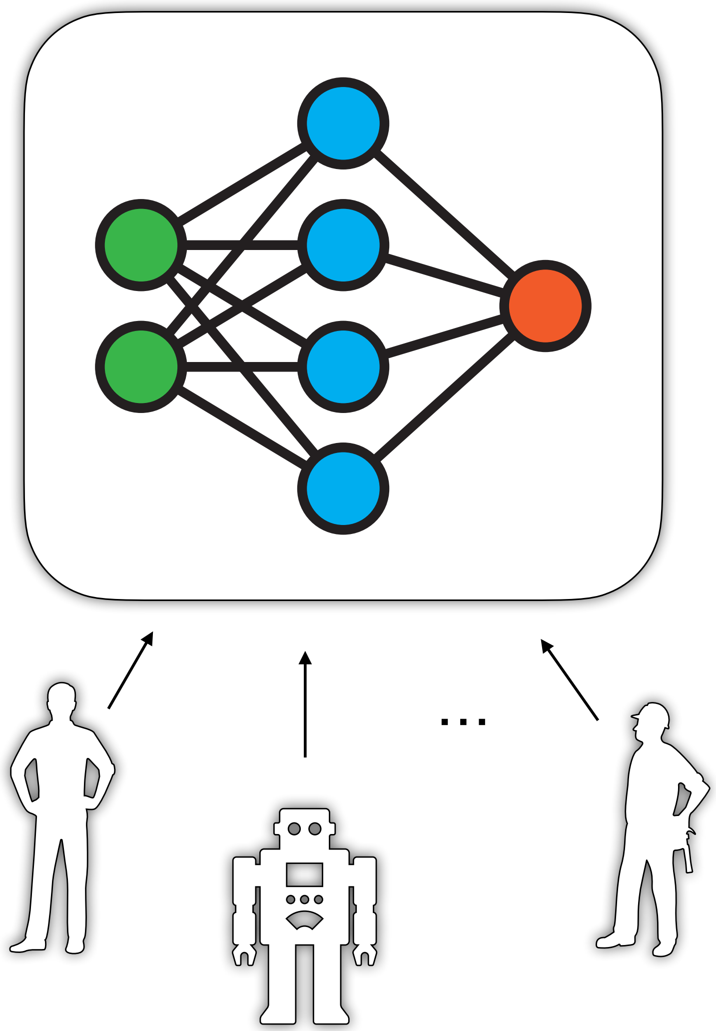
Associate Professor – Simon Fraser University
Associate Professor (Status Only) – University of Toronto
Staff Research Scientist – Google DeepMind

Talk at SZU – May 21, 2024

Objective: Neural World Models

$$\mathbf{x} \in \mathbb{R}^3$$

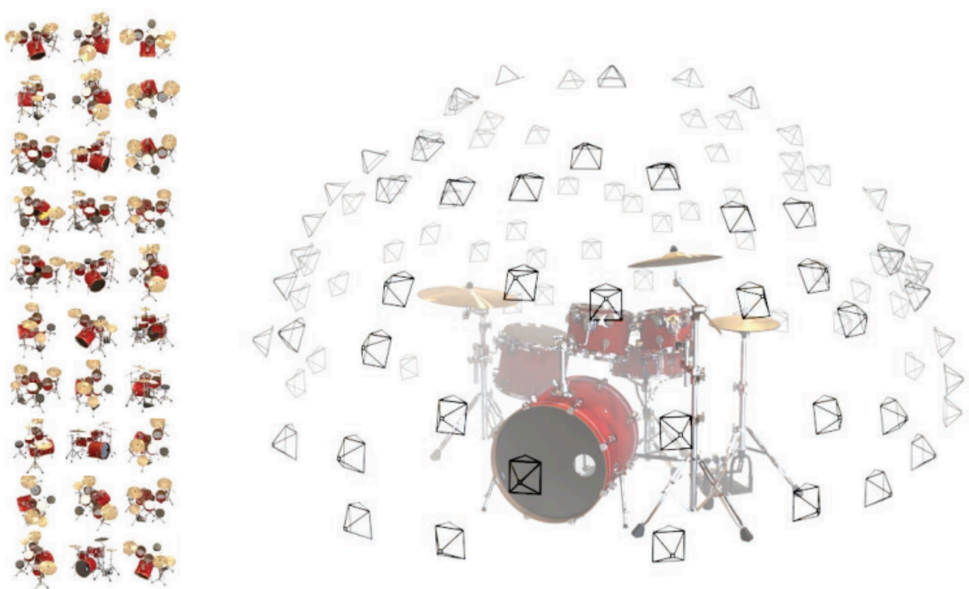
$$\mathbf{t} \in \mathbb{R}$$



$$\mathbf{f} \in \mathbb{R}^F$$

Neural Radiance Fields (NeRF)

- Use collections of calibrated images (with known camera parameters)
- Optimize 3D model so that rendered images match the training images

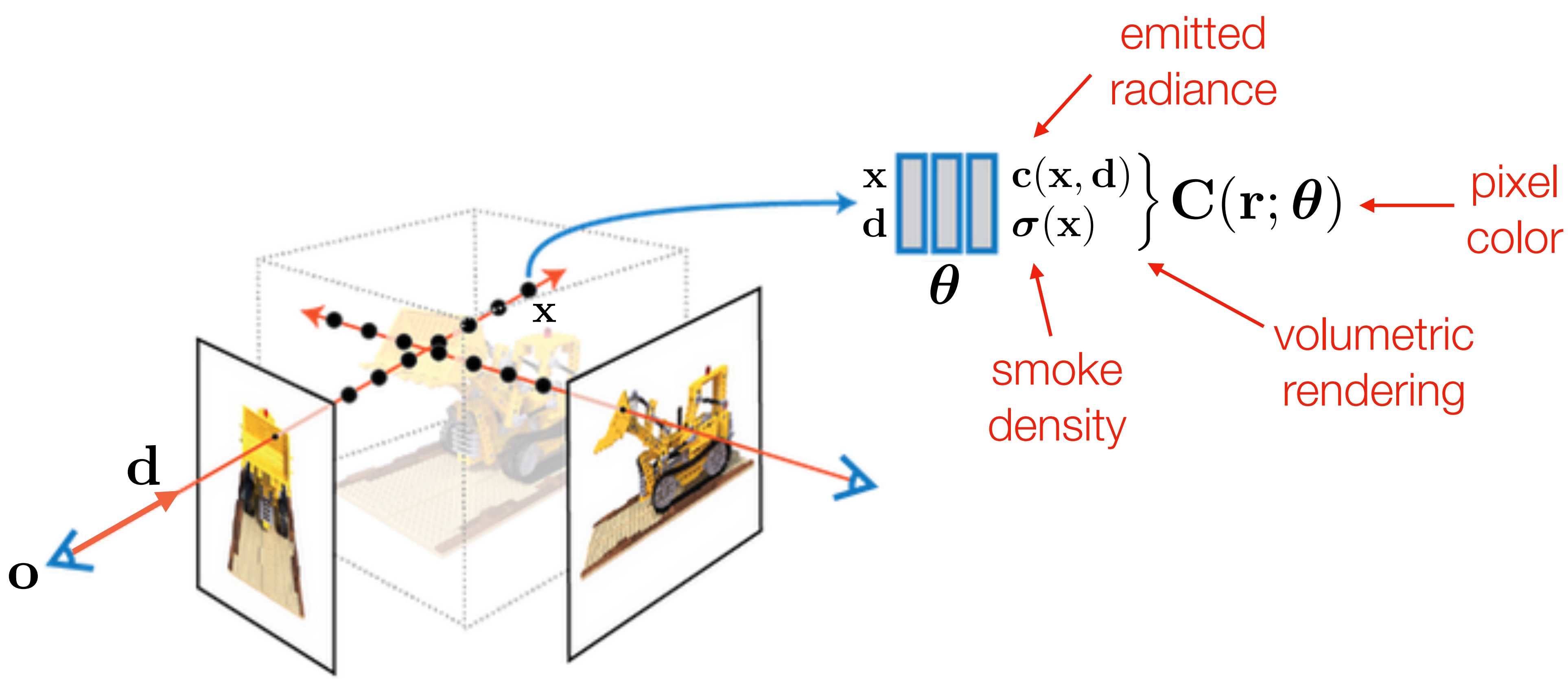


set of posed images

novel view synthesis

Basics of NeRF

- The 3D representation is a light emitting cloud of smoke
- Images from the 3D representation obtained via volumetric rendering



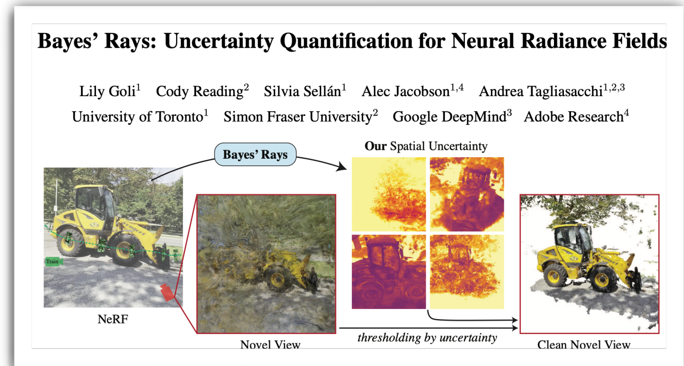
Tagliasacchi and Mildenhall "Volume Rendering Digest (for NeRF)" arXiv

courtesy of Ben Mildenhall

Types of Uncertainty

- Epistemic uncertainty
  - lack of knowledge in the model
  - i.e. uncertainty in the output

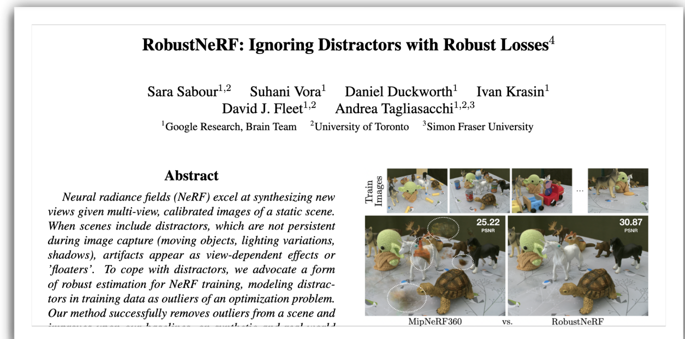
- Aleatoric uncertainty
  - randomness of data (e.g. sensor noise)
  - i.e. uncertainty in the input



Bayes' Rays: Uncertainty Quantification for Neural Radiance Fields

Lily Goli[1] Cody Reading[2] Silvia Sellán[1] Alec Jacobson[1,4] Andrea Tagliasacchi[1,2,3]
University of Toronto[1] Simon Fraser University[2] Google DeepMind[3] Adobe Research[4]

spotlight @ CVPR'2024



RobustNeRF: Ignoring Distractors with Robust Losses[4]

Sara Sabour[1,2] Suhani Vora[1] Daniel Duckworth[1] Ivan Krasin[1]
David J. Fleet[1,2] Andrea Tagliasacchi[1,2,3]
[1]Google Research, Brain Team [2]University of Toronto [3]Simon Fraser University

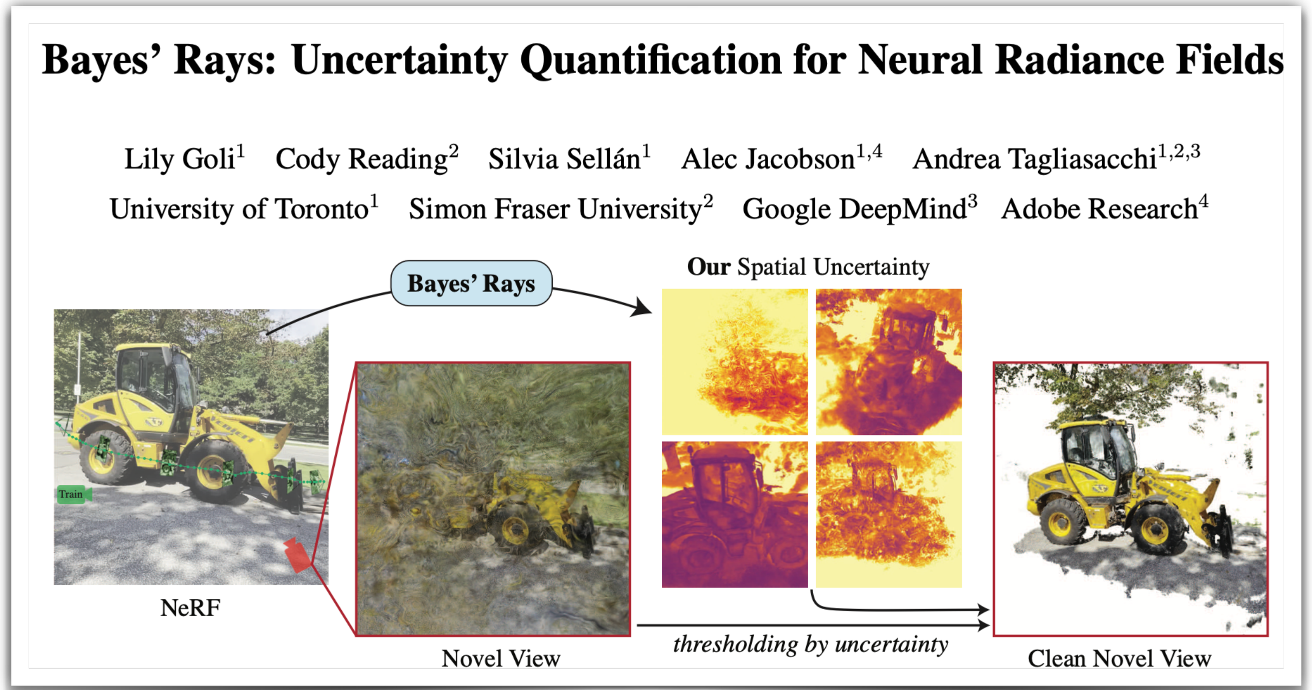
Abstract

Neural radiance fields (NeRF) excel at synthesizing new views given multi-view, calibrated images of a static scene. When scenes include distractors, which are not persistent during image capture (moving objects, lighting variations, shadows), artifacts appear as view-dependent effects or 'floaters'. To cope with distractors, we advocate a form of robust estimation for NeRF training, modeling distractors in training data as outliers of an optimization problem. Our method successfully removes outliers from a scene and

highlight @ CVPR'2023

Talk Agenda

- Epistemic uncertainty (uncertainty in the output)
- Aleatoric uncertainty (uncertainty in the input)



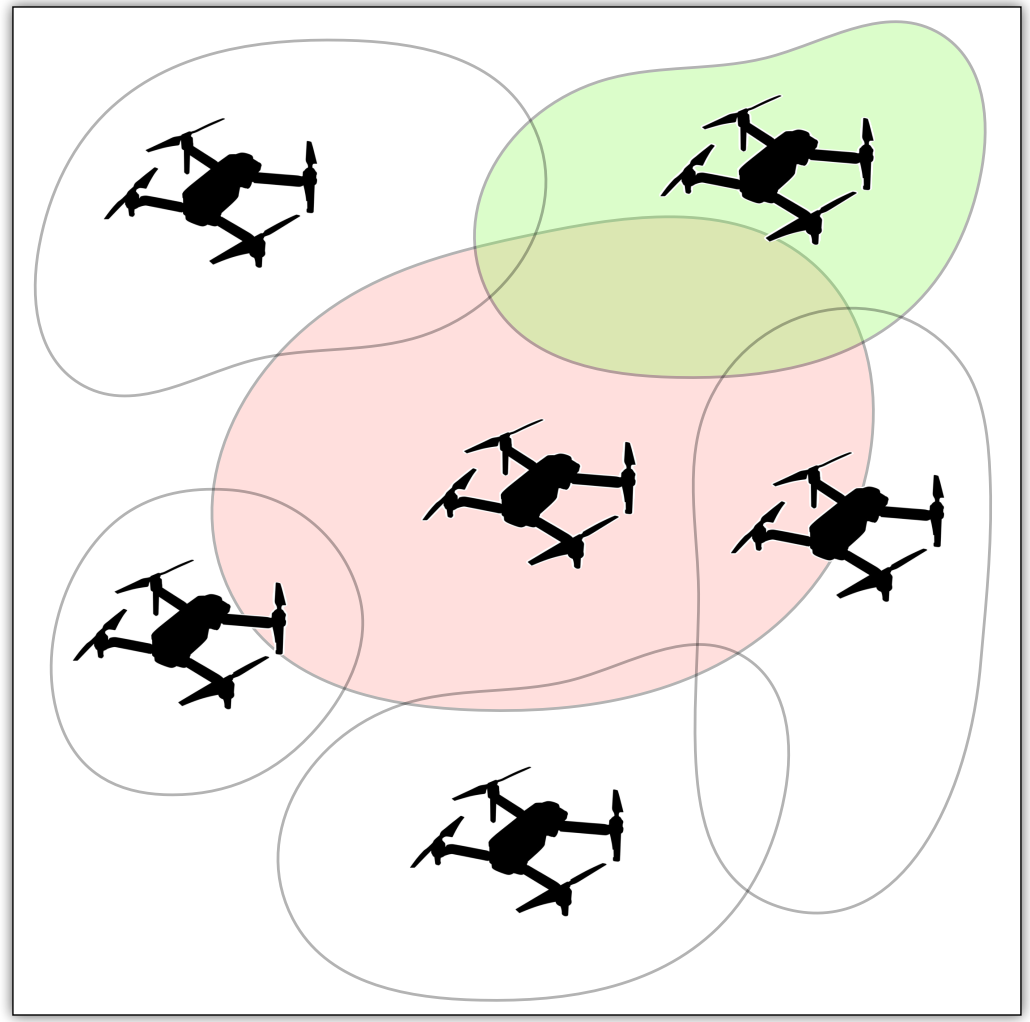
Bayes' Rays: Uncertainty Quantification for Neural Radiance Fields

Lily Goli[1] Cody Reading[2] Silvia Sellán[1] Alec Jacobson[1,4] Andrea Tagliasacchi[1,2,3]
University of Toronto[1] Simon Fraser University[2] Google DeepMind[3] Adobe Research[4]

Lily Goli

Why is this important?

Safety in Robotics



Multi-Agent Coordination

Uncertainty

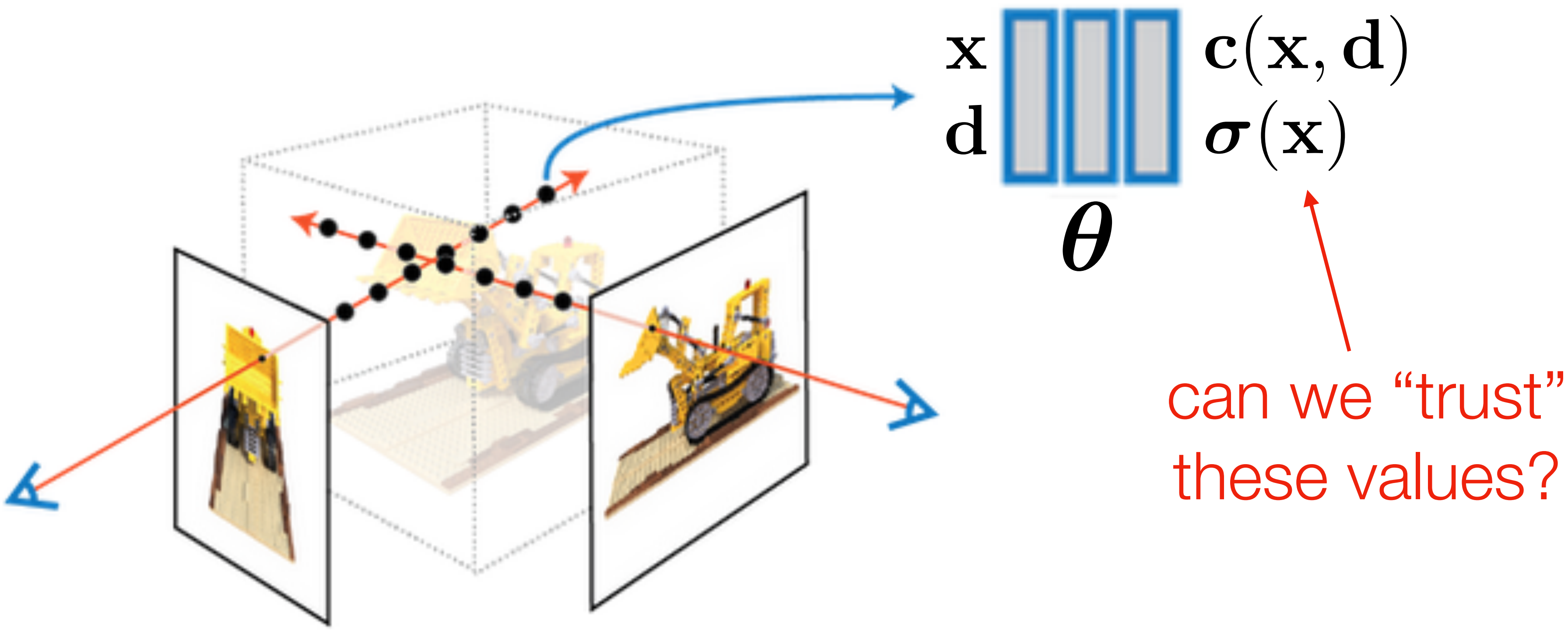
- Epistemic uncertainty
  - i.e. uncertainty in the model output

training trajectory
(training images)

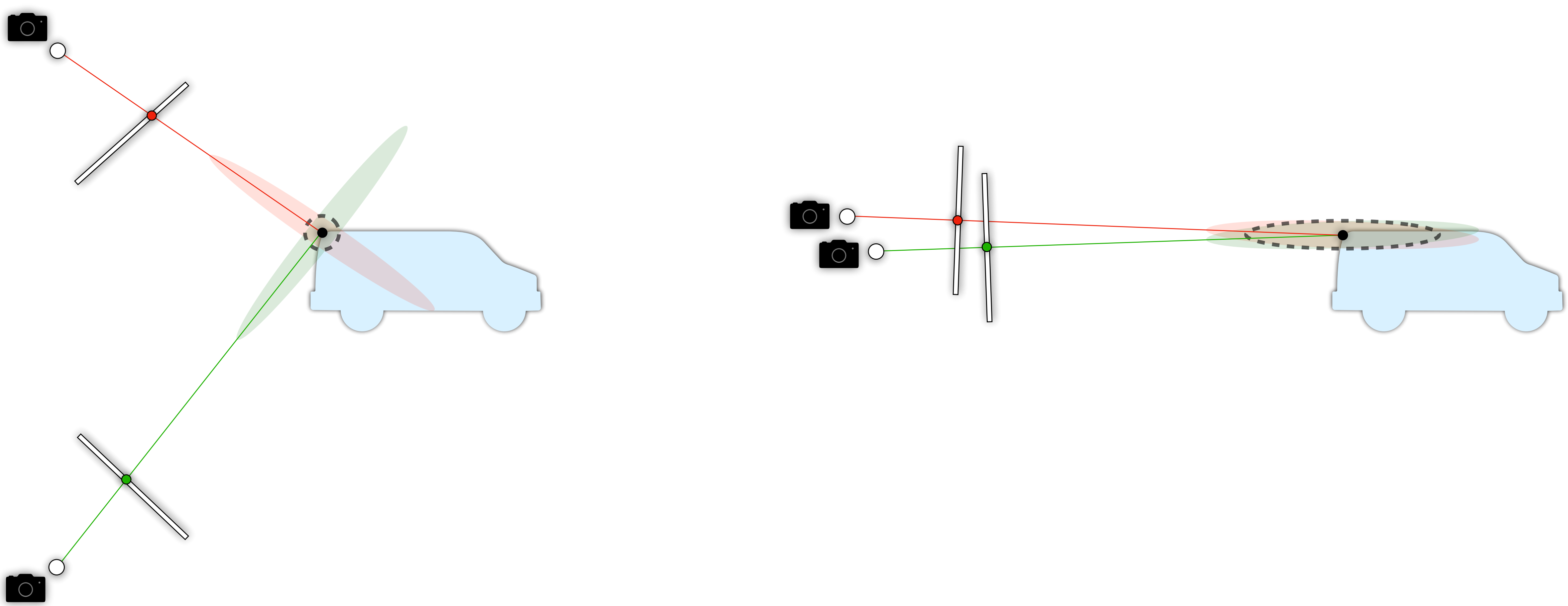
test trajectory
(test cameras)

test trajectory
(novel view synthesis)

Uncertainty



Geometric Uncertainty



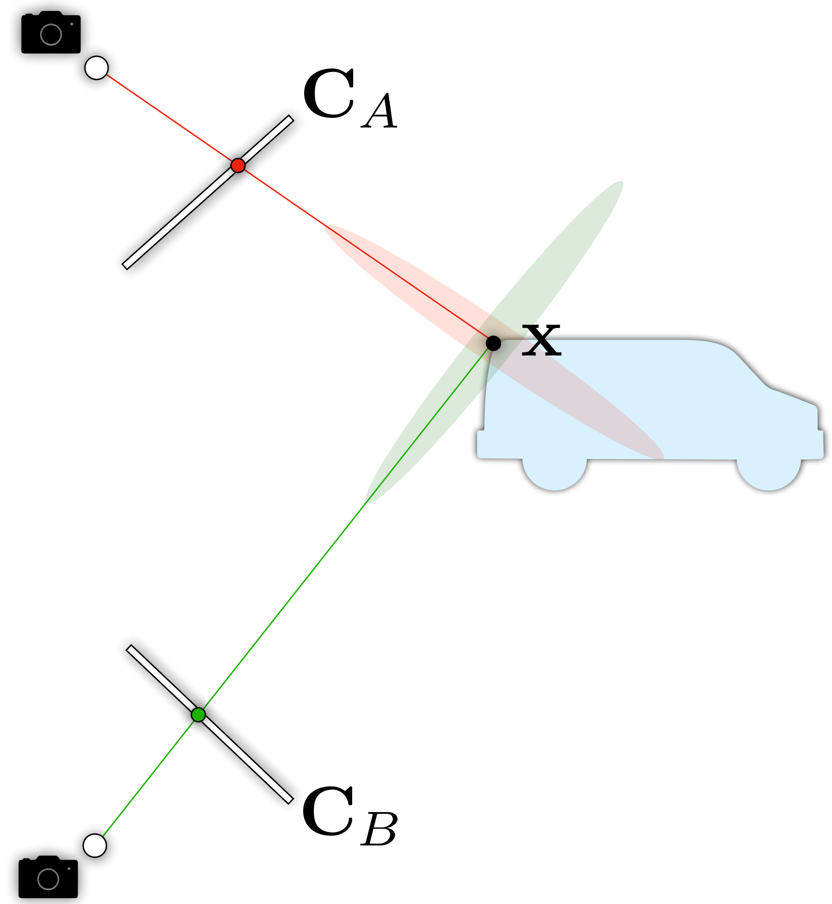
Geometric Uncertainty

- Assume we know the color/position $(\mathbf{x}, \mathbf{c})$
- Can we wiggle its position w/o effect?

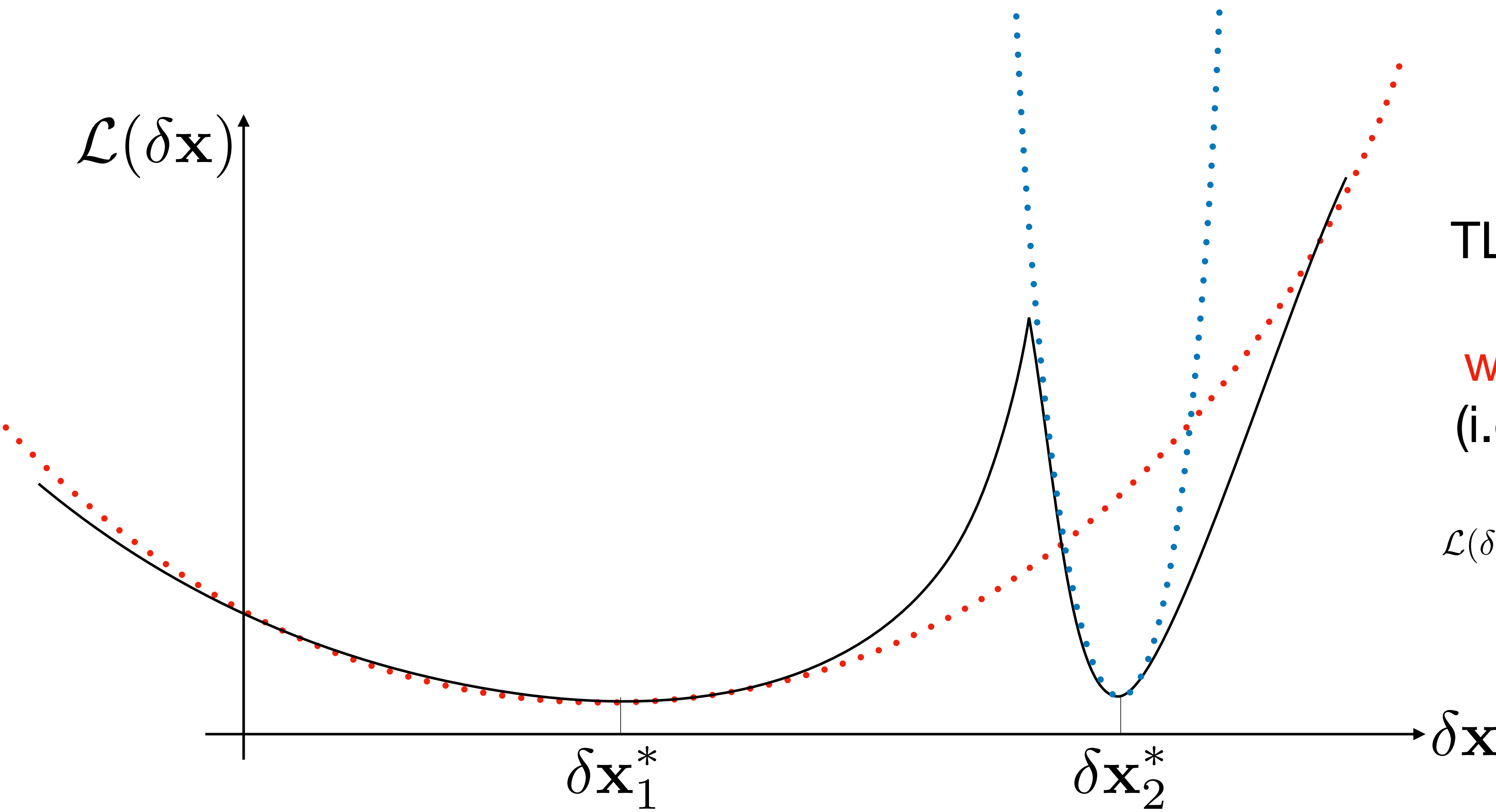
can I wiggle

$$\mathcal{L}(\delta\mathbf{x}) = \sum_{c \in A,B} \|\mathbf{C}_c[\Pi_c(\mathbf{x} + \delta\mathbf{x})] - \mathbf{c}\|_2^2$$

without affecting
the outcome



Laplace approximation

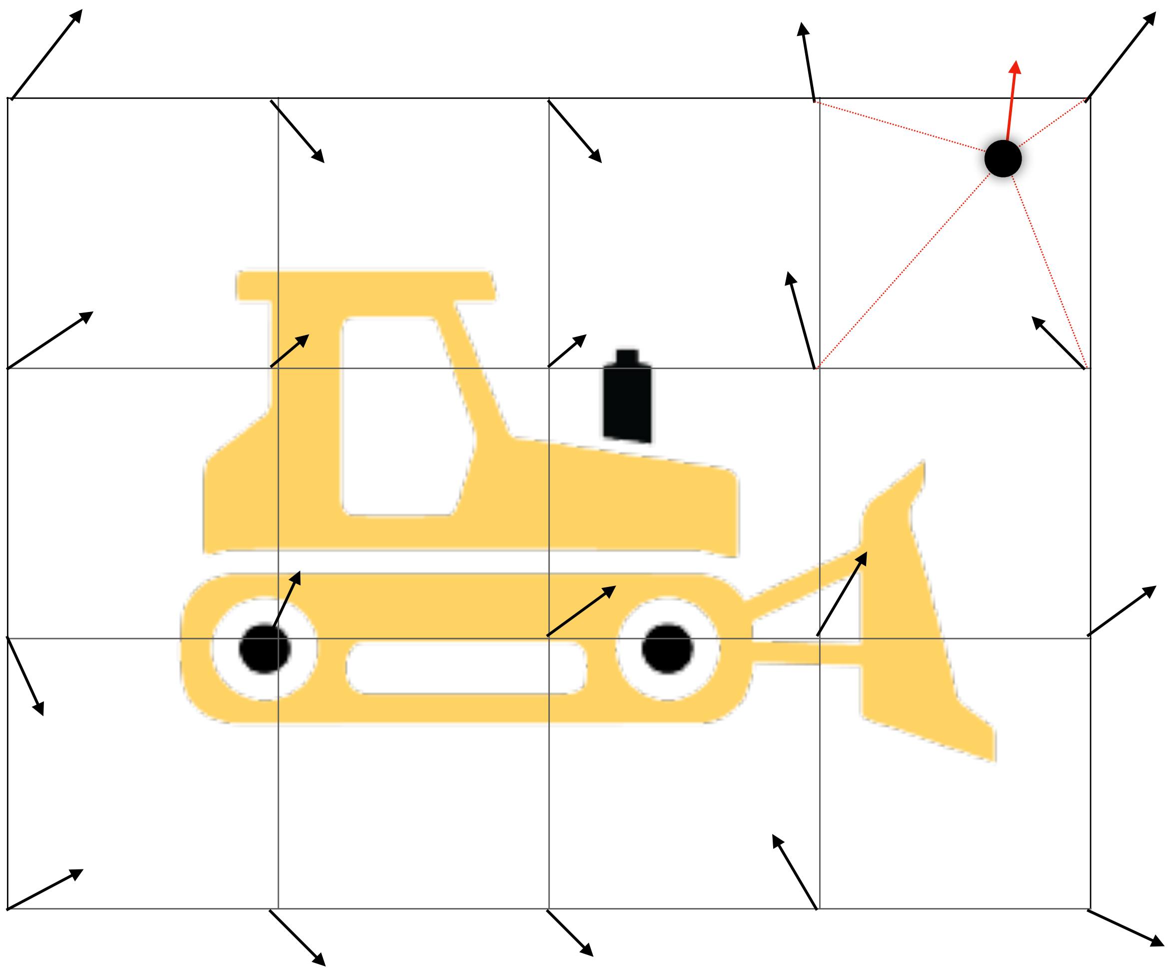


TL;DR: Jacobians of the energy at the optima can be used to determine whether your solution is a null-space (i.e. tell you how flat the landscape is)

$$\mathcal{L}(\delta\mathbf{x}) \propto (\delta\mathbf{x}-\delta\mathbf{x}^*)^\top \left(\frac{\partial\mathcal{L}(\delta\mathbf{x}^*)}{\partial\delta\mathbf{x}}\right)^\top \left(\frac{\partial\mathcal{L}(\delta\mathbf{x}^*)}{\partial\delta\mathbf{x}}\right) (\delta\mathbf{x}-\delta\mathbf{x}^*)$$

Wikipedia: Laplace's approximation provides an analytical expression for a posterior probability distribution by fitting a Gaussian distribution with a mean equal to the MAP solution and precision equal to the observed Fisher information. The approximation is justified by the Bernstein–von Mises theorem, which states that under regularity conditions the posterior converges to a Gaussian in large samples.

"Dip" it in a displacement field



$$\Theta := \text{grid vector field}$$

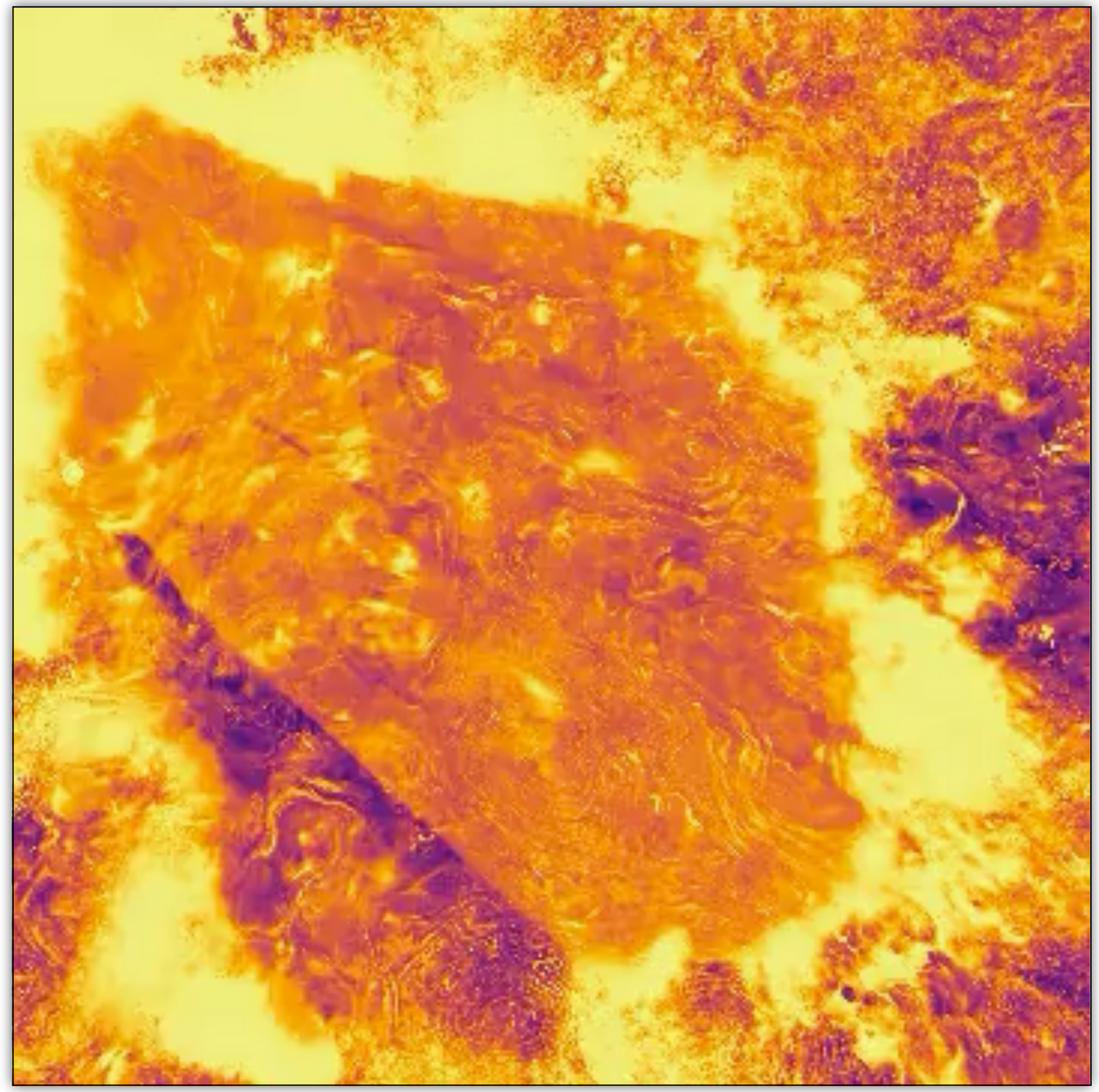
$$-\mathbf{H} \approx \mathrm{diag}\left(\frac{1}{R}\sum_{\mathbf{r}} \mathbf{J}_{\Theta}(\mathbf{r})^{\top}\mathbf{J}_{\Theta}(\mathbf{r})\right) + \lambda\mathbf{I}$$

$$\mathcal{U}(\mathbf{x}) = \mathrm{Trilinear}(\mathbf{x}, -\mathbf{H}^{-1})$$

uncertainty field

Example of Uncertainty


novel-view synthesis (test)


uncertainty

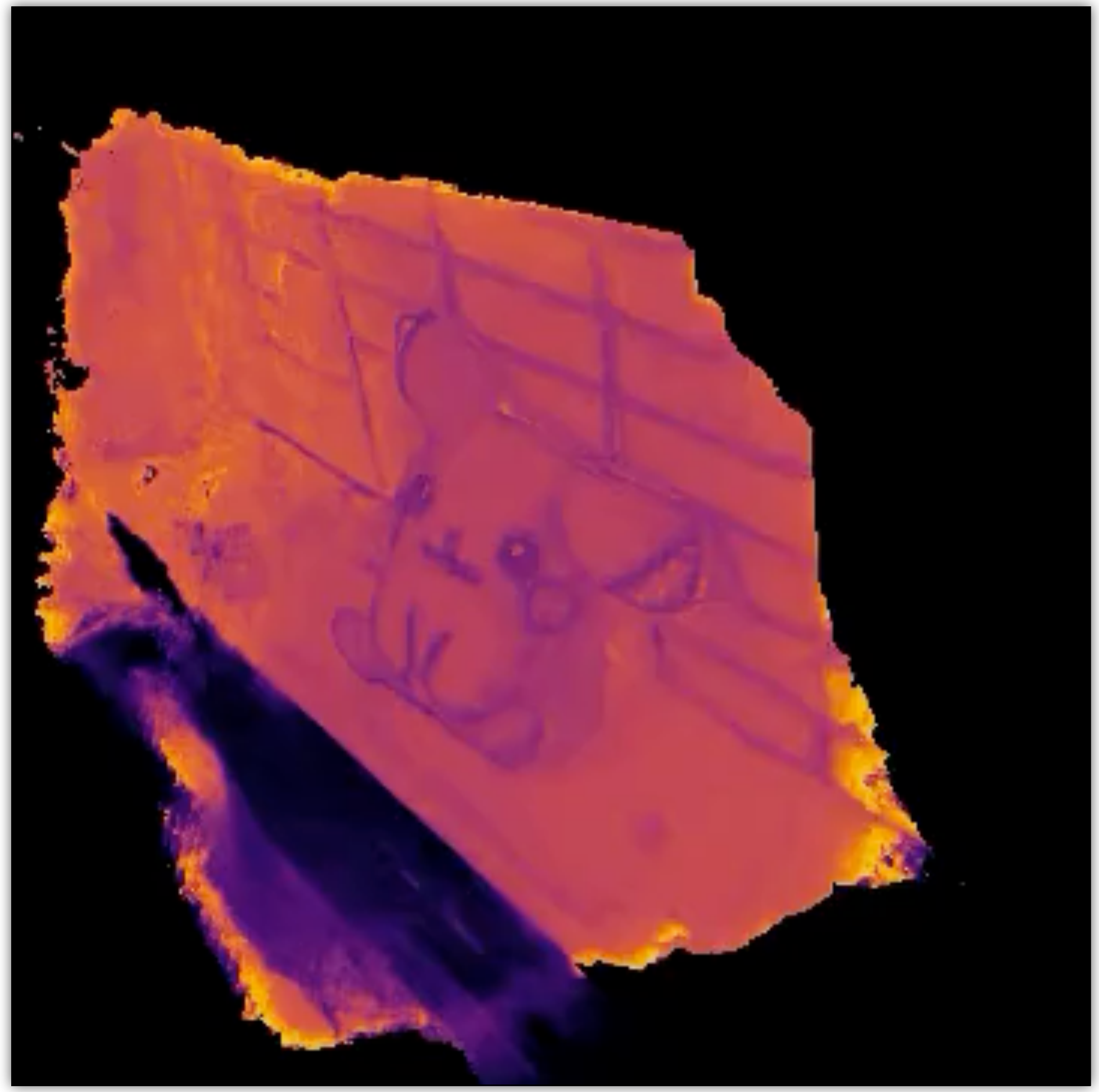
Floater removal (a-posteriori)

pre-filtering

post-filtering

Uncertainty Beyond Floaters


post-filtering

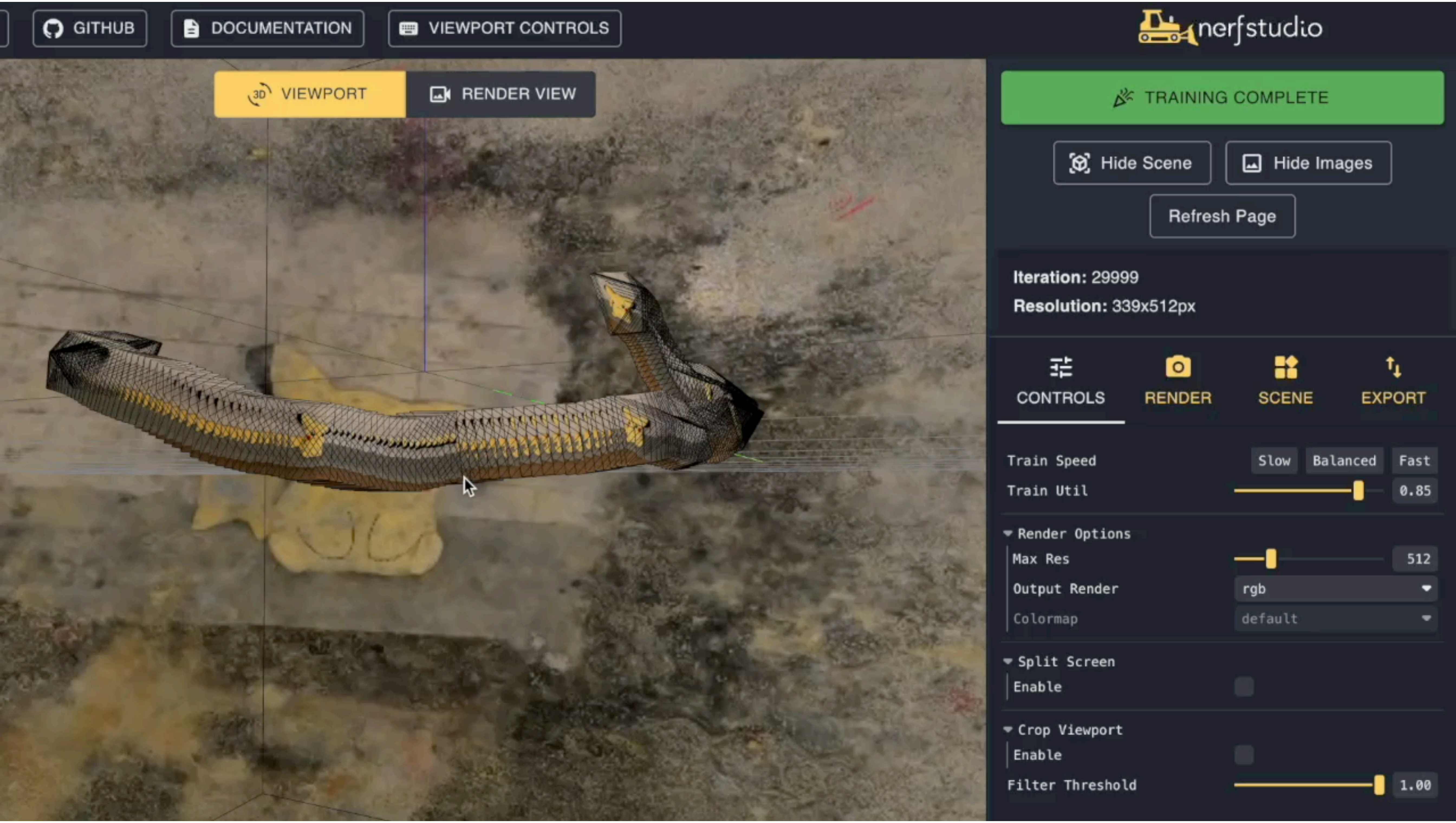

post-filtering (uncertainty)

Comparisons vs. SOTA

- Significantly faster than (co-developed) Berkeley's NerfBuster
  - NerfBuster requires lengthy (10m+) test-time optimization
  - BayesRays is fully interactive
  - …Berkeley's team now uses BayesRays (🥳)

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Coverage ↑
Nerfacto (base)	16.83	0.52	0.39	0.89
Nerfbusters	17.99	**0.60**	**0.25**	0.63
NerfU-0.9	17.66	0.56	0.34	**0.83**
NerfU-0.4	17.78	0.57	0.31	0.78
NerfU-best	**18.27**	**0.60**	0.27	0.70

Warburg et al. "Nerfbusters: Removing Ghostly Artifacts from Casually Captured NeRFs" ICCV 2023


GITHUB
DOCUMENTATION
VIEWPORT CONTROLS
nerfstudio
VIEWPORT
RENDER VIEW
TRAINING COMPLETE
Hide Scene
Hide Images
Refresh Page
Iteration: 29999
Resolution: 339x512px
CONTROLS
RENDER
SCENE
EXPORT
Train Speed
Slow
Balanced
Fast
Train Util
0.85
Render Options
Max Res
512
Output Render
rgb
Colormap
default
Split Screen
Enable
Crop Viewport
Enable
Filter Threshold
1.00

Recap – BayesRays

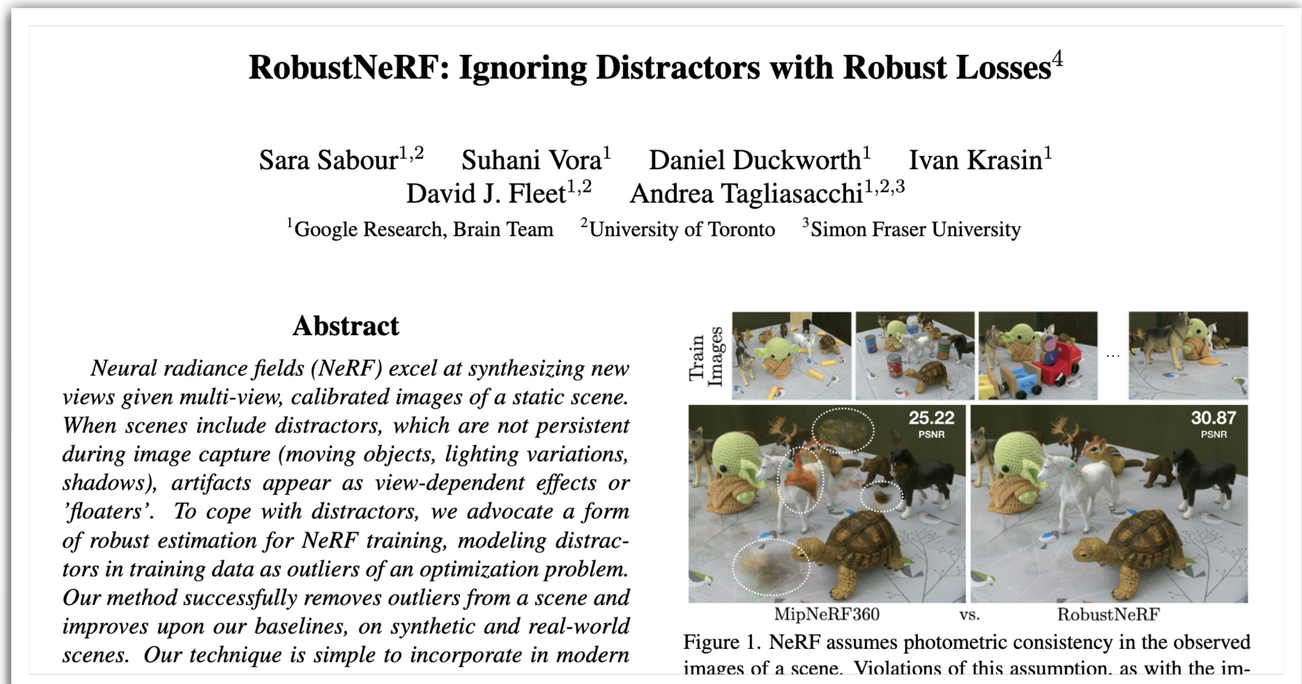
Quantify geometric uncertainty in novel-view synthesis workloads

https://bayesrays.github.io

- Limitations
  - How model uncertainty in view-dependent effects? (to boost generality)
  - Hierarchical solve for uncertainty? (to boost efficiency)
  - Autonomous exploration via uncertainty minimization?
  - ~~Can it be quickly adapted to 3D Gaussian Splatting?~~

Talk Agenda

- Epistemic uncertainty (uncertainty in the output)
- Aleatoric uncertainty (uncertainty in the input)



RobustNeRF: Ignoring Distractors with Robust Losses[4]

Sara Sabour[1,2] Suhani Vora[1] Daniel Duckworth[1] Ivan Krasin[1]
David J. Fleet[1,2] Andrea Tagliasacchi[1,2,3]
[1]Google Research, Brain Team [2]University of Toronto [3]Simon Fraser University

Abstract

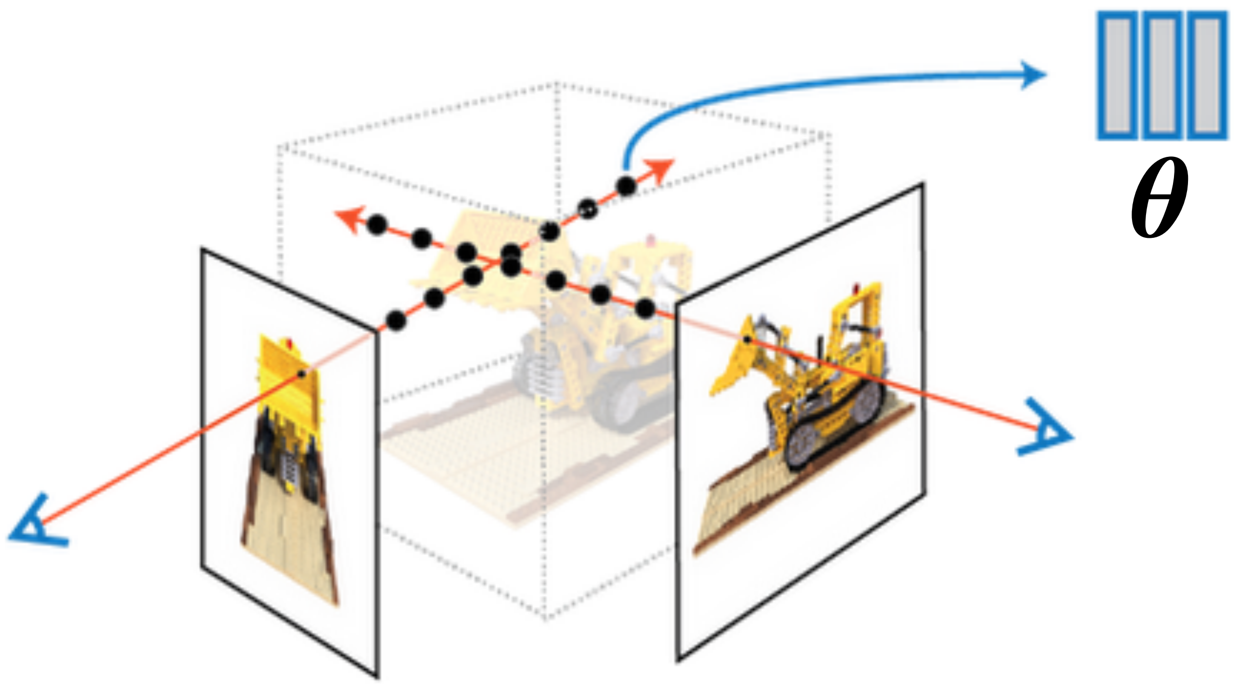
Neural radiance fields (NeRF) excel at synthesizing new views given multi-view, calibrated images of a static scene. When scenes include distractors, which are not persistent during image capture (moving objects, lighting variations, shadows), artifacts appear as view-dependent effects or 'floaters'. To cope with distractors, we advocate a form of robust estimation for NeRF training, modeling distractors in training data as outliers of an optimization problem. Our method successfully removes outliers from a scene and improves upon our baselines, on synthetic and real-world scenes. Our technique is simple to incorporate in modern

Figure 1. NeRF assumes photometric consistency in the observed images of a scene. Violations of this assumption, as with the im-

Sara Sabour

Photometric Consistency

- NeRF: optimizes the model parameters $\boldsymbol{\theta}$ by:
  - assuming all pictures taken are of exactly the same scene



$$\mathcal{L}_{\text{rgb}}(\boldsymbol{\theta}) = \sum_i \mathbb{E}_{\mathbf{r}\sim\mathbf{C}_i}\left[\mathcal{L}^{\mathbf{r},i}_{\text{rgb}}(\boldsymbol{\theta})\right]$$

$$\mathcal{L}^{\mathbf{r},i}_{\text{rgb}}(\boldsymbol{\theta}) = \boxed{||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2^2}$$

photometric consistency

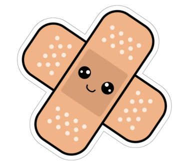
Photometric Consistency

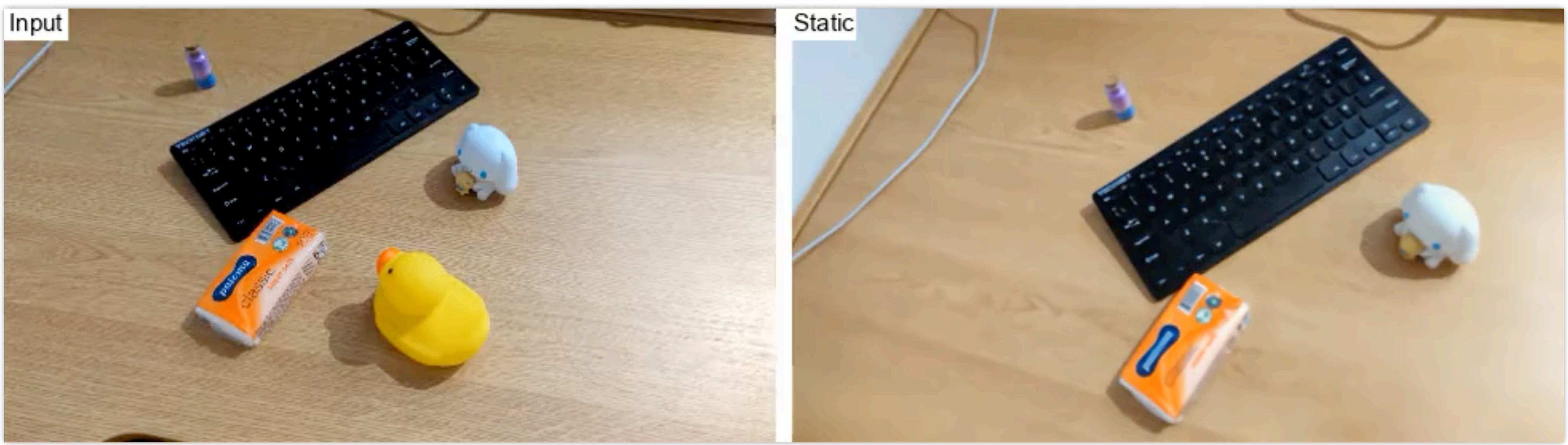
- Assume all pictures taken are of the same scene
  - Is this a resonable assumption?

...

$$\mathcal{L}_{\mathrm{rgb}}^{\mathbf{r},i}(\boldsymbol{\theta}) = ||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2^2$$

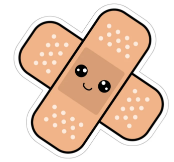
Photometric Consistency

- How has the problem been addressed in the literature? 
    - Factorize into of static + dynamic components



Wu et al. "D2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video" NeurIPS 2022

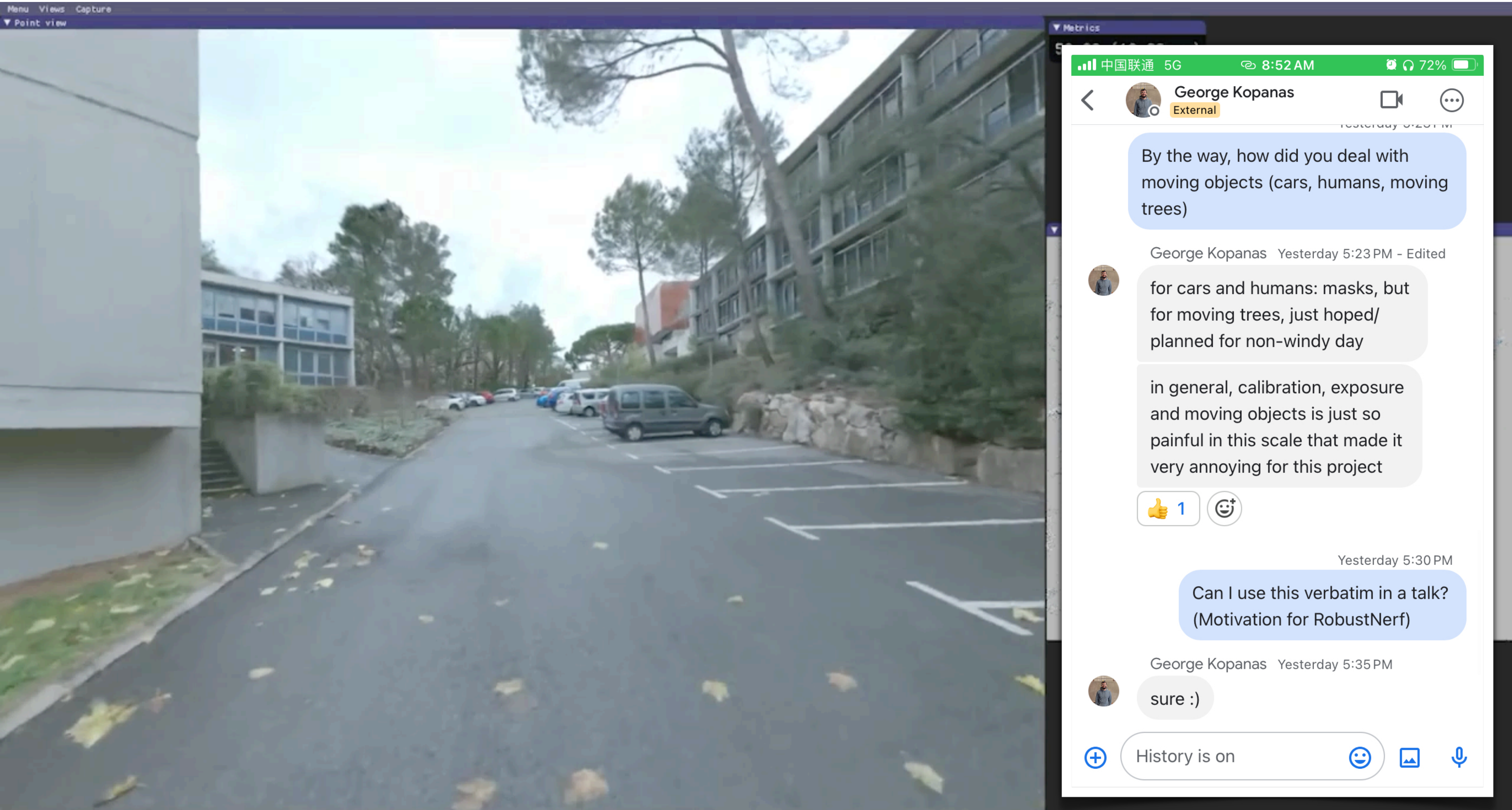
Photometric Consistency

- How has the problem been addressed in the literature? 
  - Exclude pixels of known classes (people, cars, …)

Brandeburg gate – NeRF in the Wild

$$\mathcal{L}^{\mathbf{r},i}_{\text{oracle}}(\boldsymbol{\theta}) = \boxed{\mathbf{S}_i(\mathbf{r})} \cdot ||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2^2$$

semantic segmentation

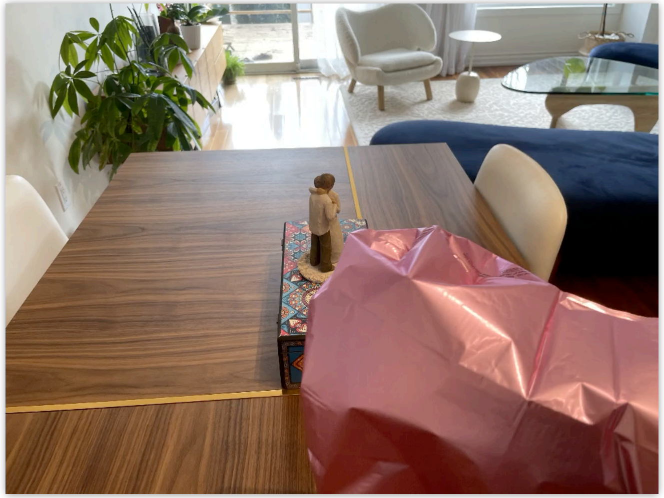


Kerbl et al. “A Hierarchical 3D Gaussian Representation for Real-Time Rendering of Very Large Datasets” SIGGRAPH 2024

Photometric Consistency

- But are distractors always easily segmentable?

...



$$\mathcal{L}^{\mathbf{r},i}_{\text{oracle}}(\boldsymbol{\theta}) = \boxed{\mathbf{S}_i(\mathbf{r})} \cdot ||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2^2$$

???


training images

Photometric Consistency

- Distractors degrades the quality of the 3D model in two ways

spurious density
"floaters"

incorrect view
dependent effects

Photometric Consistency

- Assume all pictures taken are of the same scene
  - What violates this assumpion in our optimization?

...

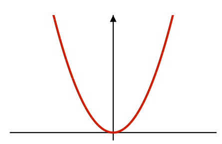
$$\mathcal{L}_{\text{rgb}}^{\mathbf{r},i}(\boldsymbol{\theta}) = ||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2^2$$

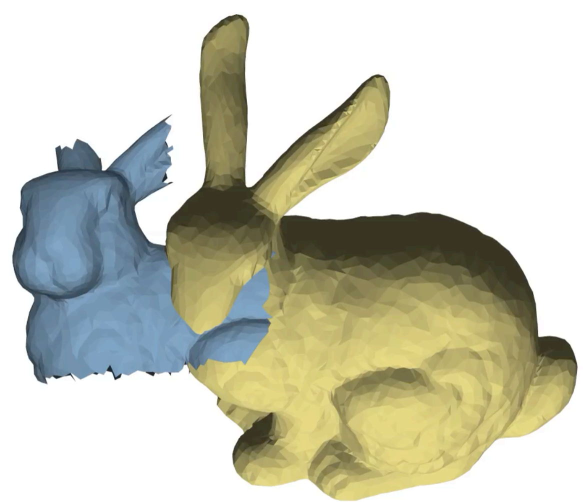
“Steal” ideas from SparseICP

$$\underset{\mathbf{R},\mathbf{t}}{\arg\min} \sum_{n=1}^{N} \kappa(\|...\|_2)$$

Classical ICP

$$\kappa(x) = x^2$$

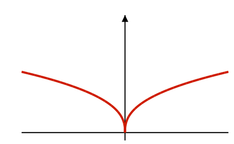


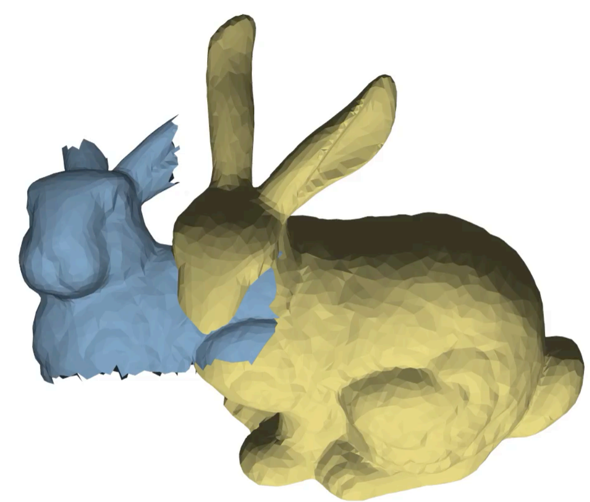


vs.

Sparse ICP

$$\kappa(x) = x^p$$





Bouaziz, Tagliasacchi, Pauly “Sparse Iterative Closest Point” Symposium on Geomety Processing 2013

Robust Optimization w/ IRLS

Iteratively
Re-Weighted
Least
Squares

$$\mathcal{L}^{\mathbf{r},i}_{\text{robust}}(\boldsymbol{\theta}) = \kappa(||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2)$$

identical
gradients

$$\mathcal{L}^{\mathbf{r},i}_{\text{robust}}(\boldsymbol{\theta}^{(t)}) = \omega(\boldsymbol{\epsilon}^{(t-1)}(\mathbf{r})) \cdot ||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}^{(t)}) - \mathbf{C}_i(\mathbf{r})||_2^2$$

$$\boldsymbol{\epsilon}^{(t-1)}(\mathbf{r}) = ||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}^{(t-1)}) - \mathbf{C}_i(\mathbf{r})||_2$$

IRLS vs. Oracle

- Losses look similar? …steal the oracle's properties!
  - weights are categorical (due to UNet's softmax)
  - weights are spatially coherent (due to UNet's bottleneck)

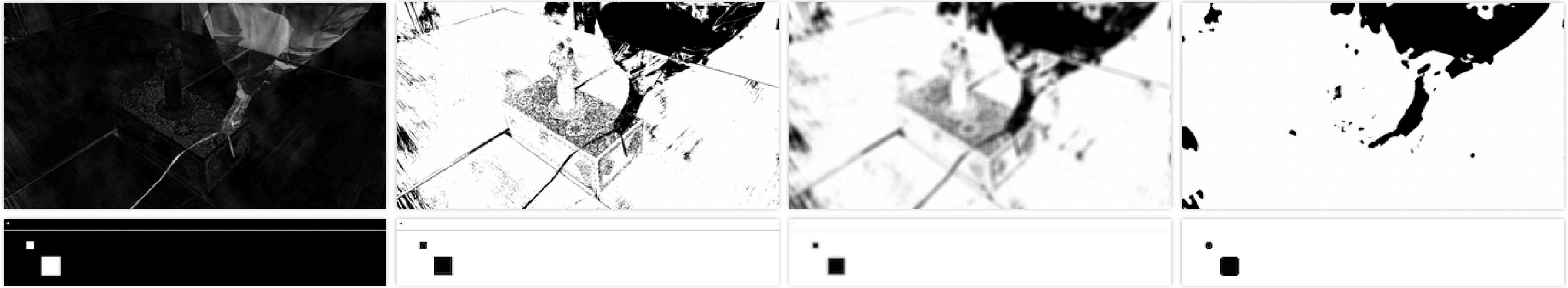
$$\mathcal{L}^{\mathbf{r},i}_{\text{oracle}}(\boldsymbol{\theta}) = \mathbf{S}_i(\mathbf{r}) \cdot ||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2^2$$

v.s.

$$\mathcal{L}^{\mathbf{r},i}_{\text{robust}}(\boldsymbol{\theta}) = \omega(\boldsymbol{\epsilon}^{(t-1)}(\mathbf{r})) \cdot ||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2^2$$

Trimmed+Diffused Least Squares

- Losses look similar? …steal the oracle's properties!
  - weights are categorical (due to UNet's softmax)
  - weights are spatially coherent (due to UNet's bottleneck)



residuals – $\epsilon(\mathbf{r})$ inliers – $\tilde{\omega}(\mathbf{r})$ diffusion – $\tilde{\omega}(\mathbf{r}) \circledast \mathcal{B}_{3\times3}$ (IRLS) weights – $\mathcal{W}(\mathbf{r})$


training images


training images

Quantitative Analysis

	Statue			Android			Crab			BabyYoda		
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
mip-NeRF 360 (L_2)	0.36	0.66	19.09	0.40	0.65	19.35	0.27	0.77	25.73	0.31	0.75	22.97
mip-NeRF 360 (L_1)	0.30	0.72	19.55	0.40	0.66	19.38	0.22	0.79	26.69	0.22	0.80	26.15
mip-NeRF 360 (Ch.)	0.30	0.73	19.64	0.40	0.66	19.53	0.21	0.80	27.72	0.23	0.80	25.22
D^{2}NeRF	0.48	0.49	19.09	0.43	0.57	20.61	0.42	0.68	21.18	0.44	0.65	17.32
RobustNeRF	**0.28**	**0.75**	**20.89**	**0.31**	**0.65**	**21.72**	**0.21**	**0.81**	**30.75**	**0.20**	**0.83**	**30.87**
mip-NeRF 360 (clean)	0.19	0.80	23.57	0.31	0.71	23.10	0.16	0.84	32.55	0.16	0.84	32.63

MipNeRF360 (L2)

MipNeRF360 (L1)

MipNeRF360 (Ch)

D^{2}NeRF

RobustNeRF

Recap – RobustNerf

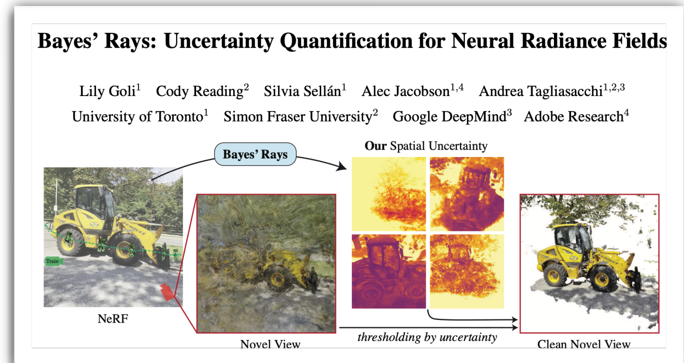
robustness to any capture outliers via a simple loss change

https://robustnerf.github.io

- Limitations
  - can this be useful for Lambertian decomposition?
  - can we be more statistically efficient?
  - ~~can it be adapted to 3D gaussian splatting?~~
  - ~~can we move away from patch-based training?~~
  - ~~can we make the model theoretically gounded?~~

Conclusions

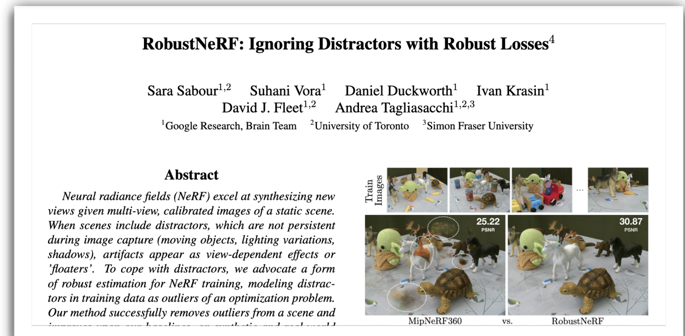
- **Epistemic uncertainty: BayesRays**
  - safe understanding of 3D space
  - enables multi-agent sensor fusion


Bayes' Rays: Uncertainty Quantification for Neural Radiance Fields

Lily Goli[1] Cody Reading[2] Silvia Sellán[1] Alec Jacobson[1,4] Andrea Tagliasacchi[1,2,3]
University of Toronto[1] Simon Fraser University[2] Google DeepMind[3] Adobe Research[4]

spotlight @ CVPR'2024

- **Aleatoric uncertainty: RobustNerf**
  - ignores what is difficult to model
  - simplifies capture prep requirements


RobustNeRF: Ignoring Distractors with Robust Losses[4]

Sara Sabour[1,2] Suhani Vora[1] Daniel Duckworth[1] Ivan Krasin[1]
David J. Fleet[1,2] Andrea Tagliasacchi[1,2,3]
[1]Google Research, Brain Team [2]University of Toronto [3]Simon Fraser University

Abstract

Neural radiance fields (NeRF) excel at synthesizing new views given multi-view, calibrated images of a static scene. When scenes include distractors, which are not persistent during image capture (moving objects, lighting variations, shadows), artifacts appear as view-dependent effects or 'floaters'. To cope with distractors, we advocate a form of robust estimation for NeRF training, modeling distractors in training data as outliers of an optimization problem. Our method successfully removes outliers from a scene and

highlight @ CVPR'2023