Poster ID: THU-PM-002

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

RobustNerf: Ignoring Distractors with Robust Losses

Sara Sabour, Suhani Vora, Daniel Duckworth,
Ivan Krasin, David Fleet, Andrea Tagliasacchi

Google Brain
UNIVERSITY OF TORONTO
VECTOR INSTITUTE
SFU SIMON FRASER UNIVERSITY

NeRF is amazing at **3D multi view reconstruction from 2D images**

Inputs photos

NeRF is amazing at **3D rendering from 2D images**

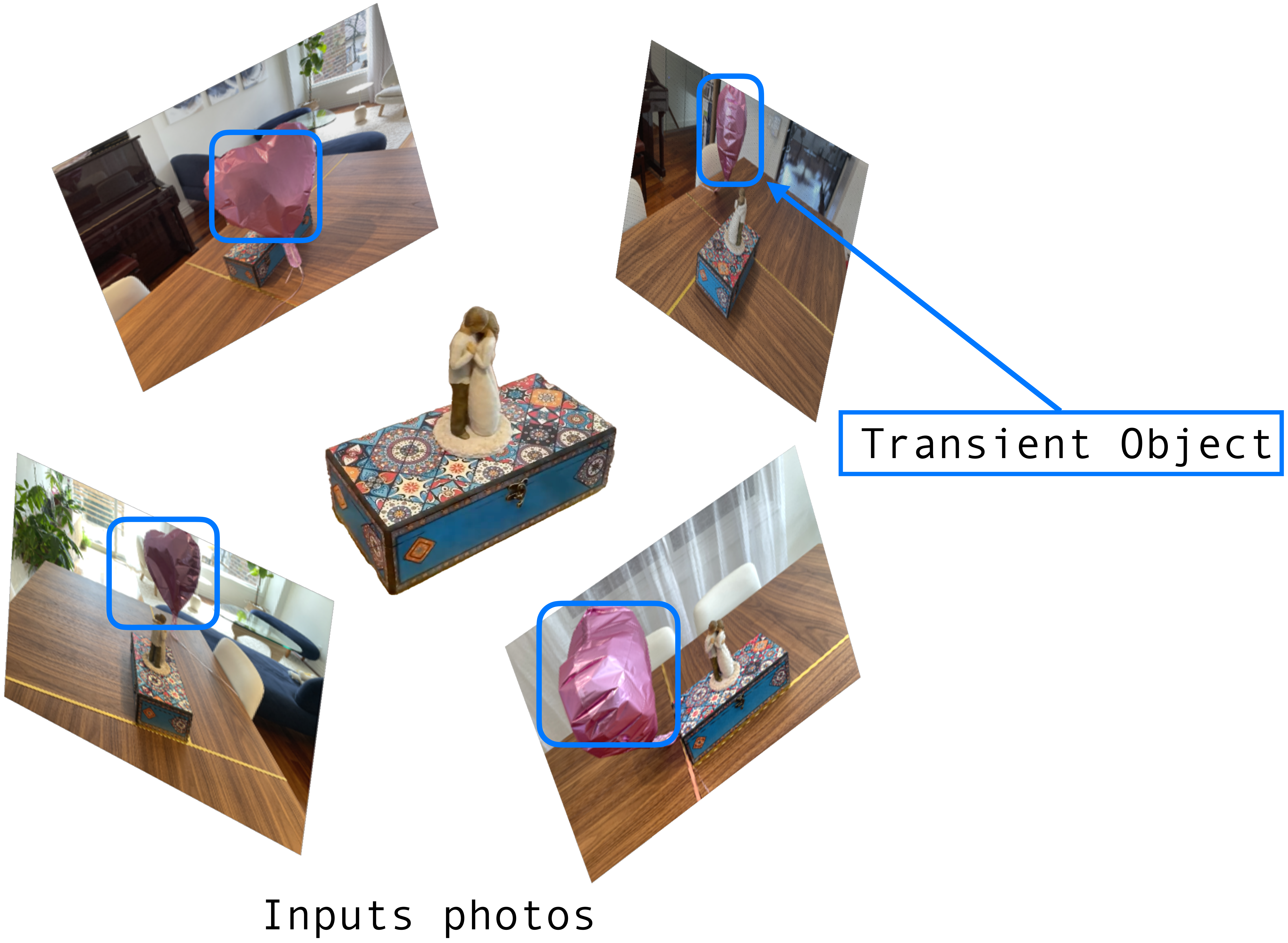
Inputs photos

MipNerf360

NeRF struggles with **photometric inconsistencies**

Inputs photos

NeRF struggles with **photometric inconsistencies**



NeRF struggles with **photometric inconsistencies**

Inputs photos

MipNerf360:Charbonnier Loss

NeRF struggles with **photometric inconsistencies**

Inputs photos

MipNerf360:Charbonnier Loss

Cloudy

NeRF struggles with **photometric inconsistencies**

Inputs photos

MipNerf360:Charbonnier Loss

Cloudy Floater

NeRF struggles with **photometric inconsistencies**

Inputs photos

RobustNeRF

Clean & Clear

RobustNerf successfully removes artifacts!

Photometric inconsistencies are Outliers in a robust optimization task.

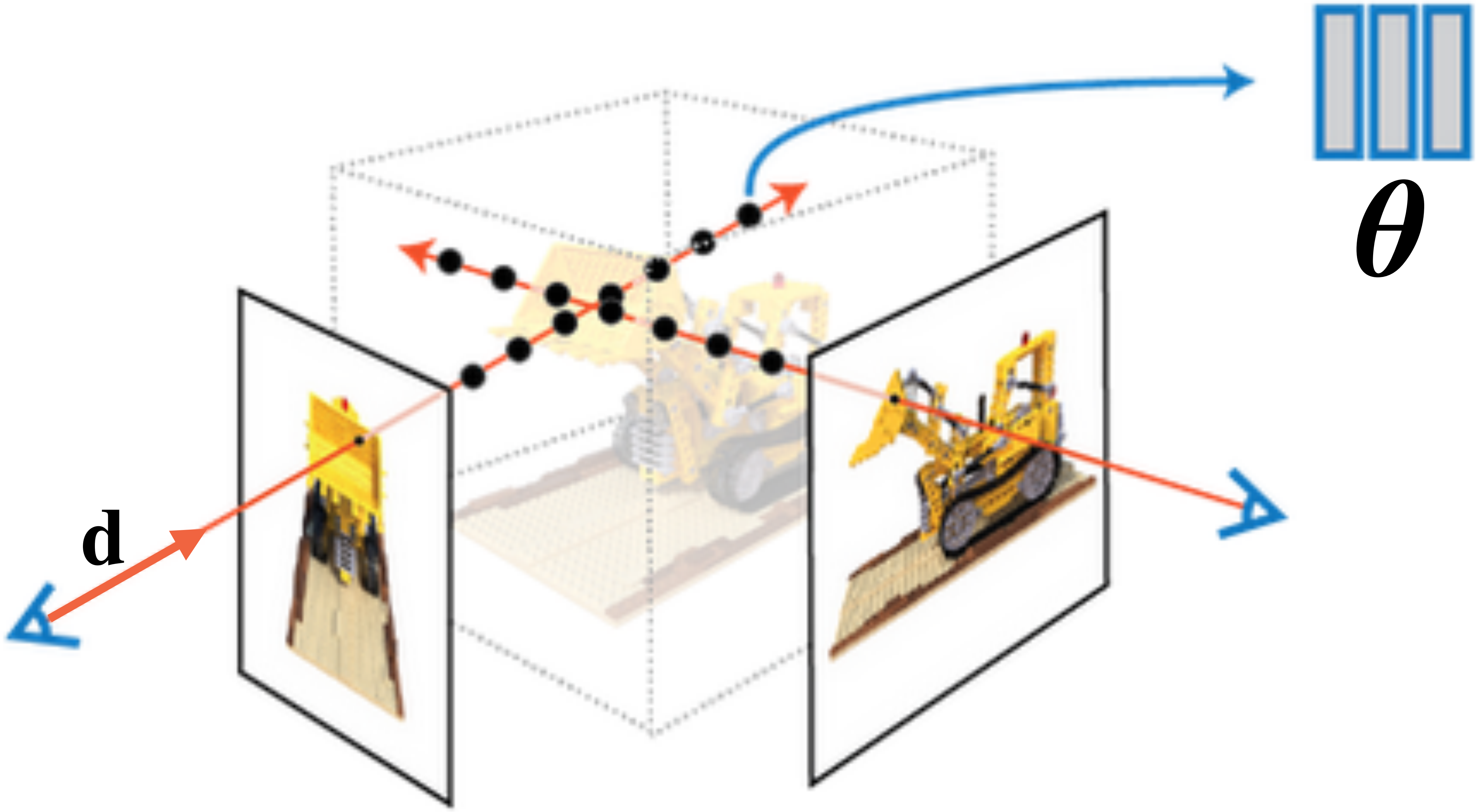
MipNeRF360

RobustNeRF

Poster ID: THU-PM-002 link: https://robustnerf.github.io

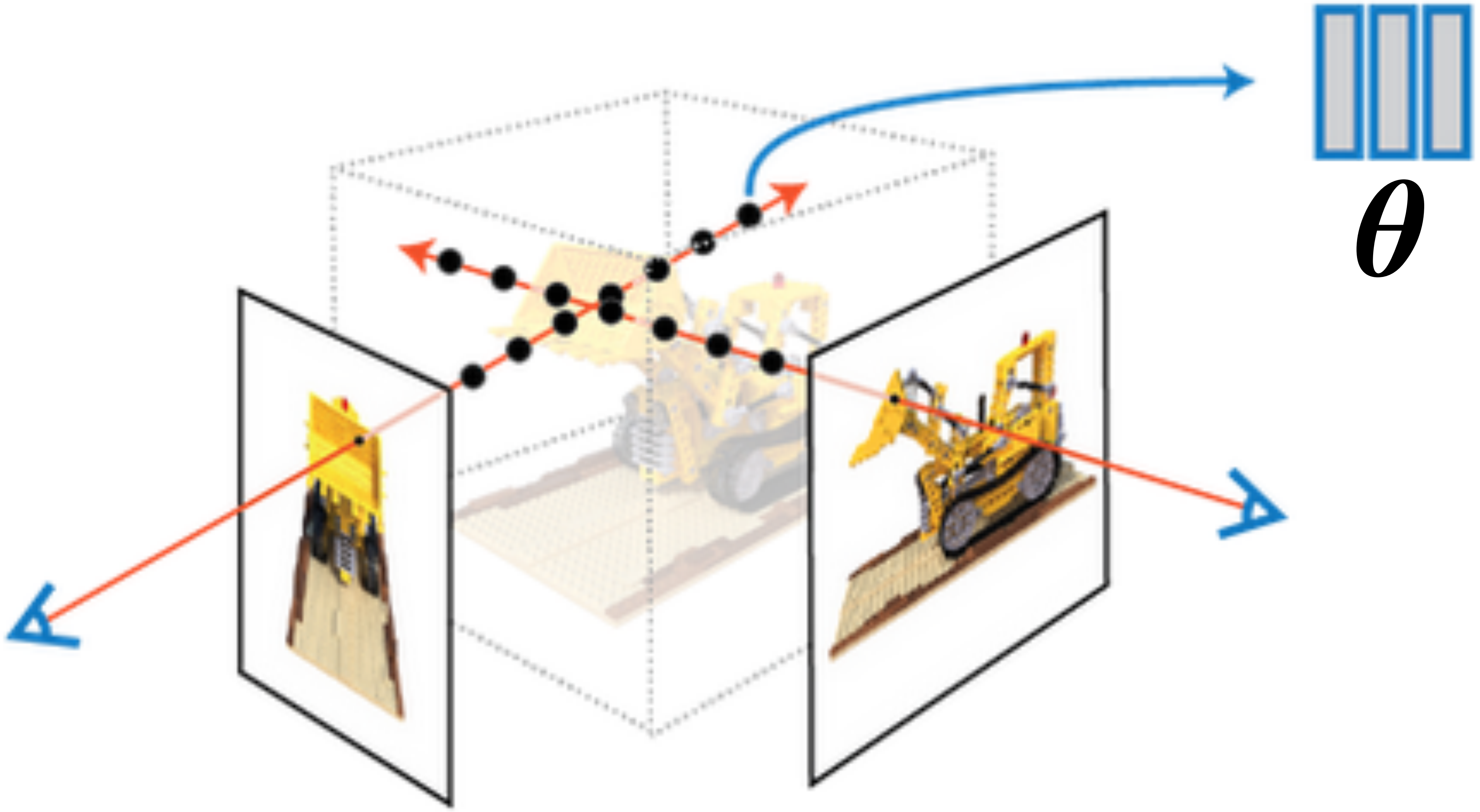
Photometric Consistency

- NeRF: optimizes the model parameters $\boldsymbol{\theta}$ by:
  - Assuming all pictures taken are of exactly the same scene
  - It is ideal for gaussian noise



Photometric Consistency

- NeRF: optimizes the model parameters $\boldsymbol{\theta}$ by:
    - Assuming all pictures taken are of exactly the same scene
    - It is ideal for gaussian noise



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

photometric consistency

Robust Estimation

- automatically identify non-consistent information
- model distractors as optimization outliers

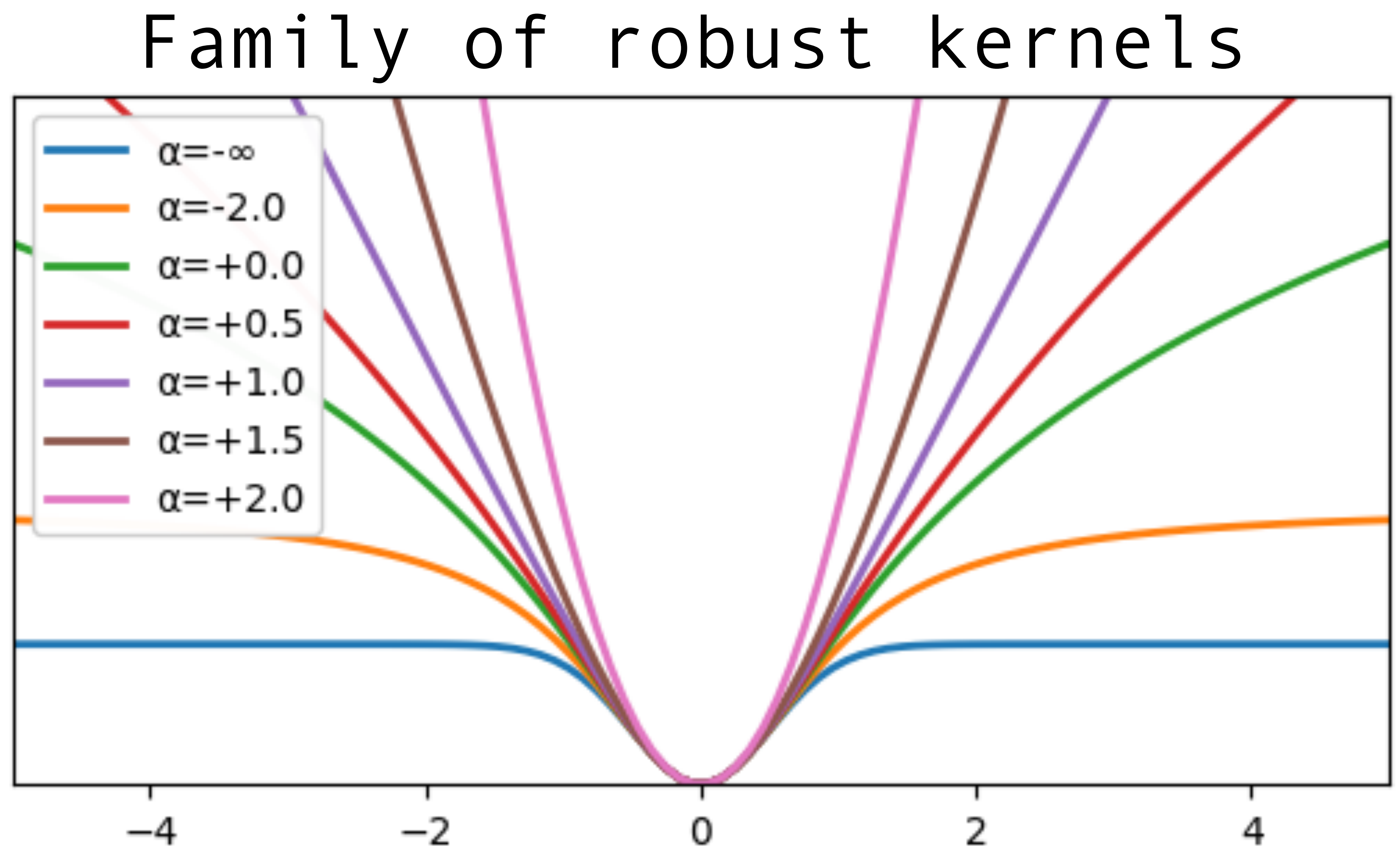
Robust Estimation

- automatically identify non-consistent information
- model distractors as optimization outliers

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

v.s.

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



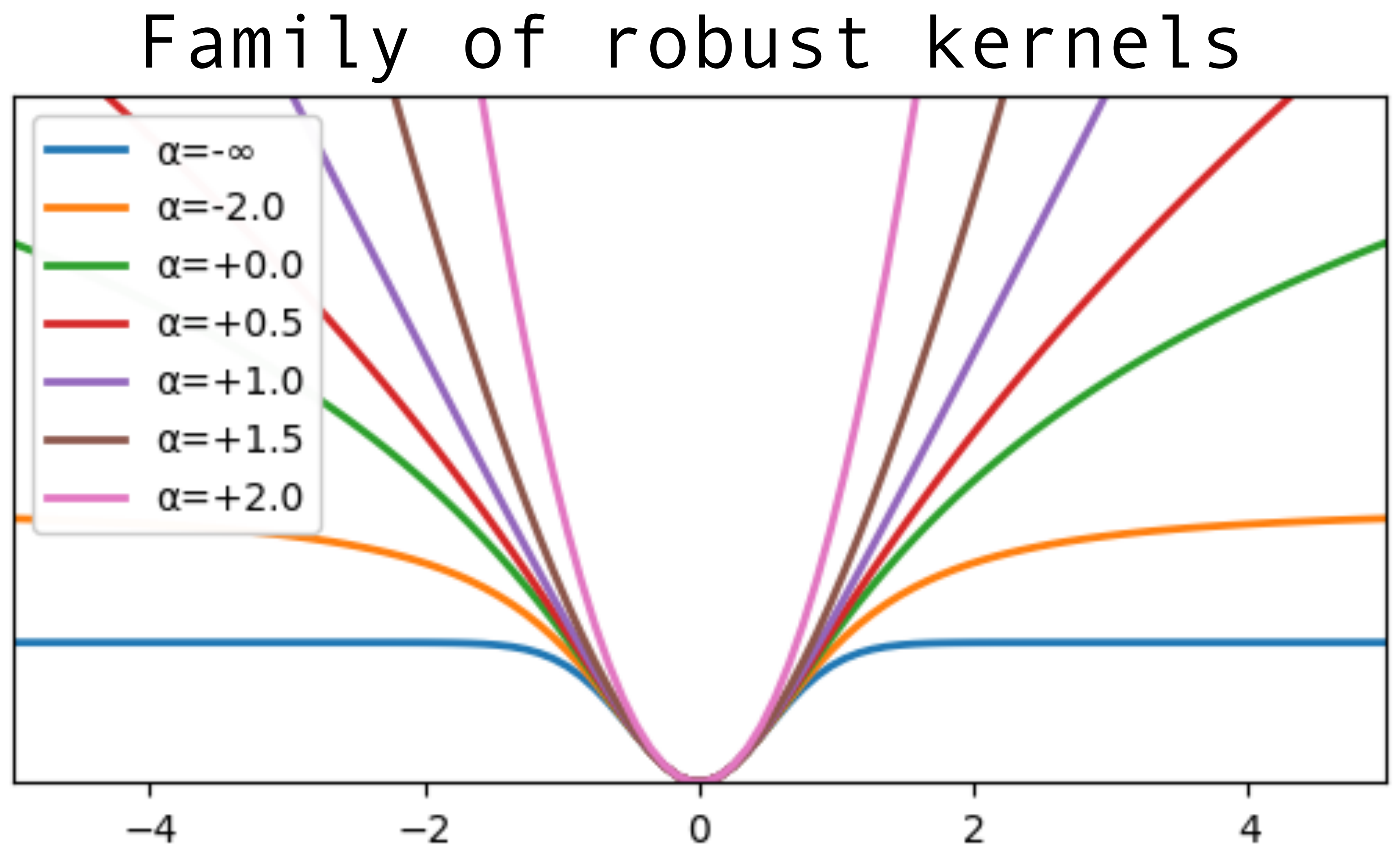
Robust Estimation

- automatically identify non-consistent information
- model distractors as optimization outliers

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

v.s.

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



Robust Estimation

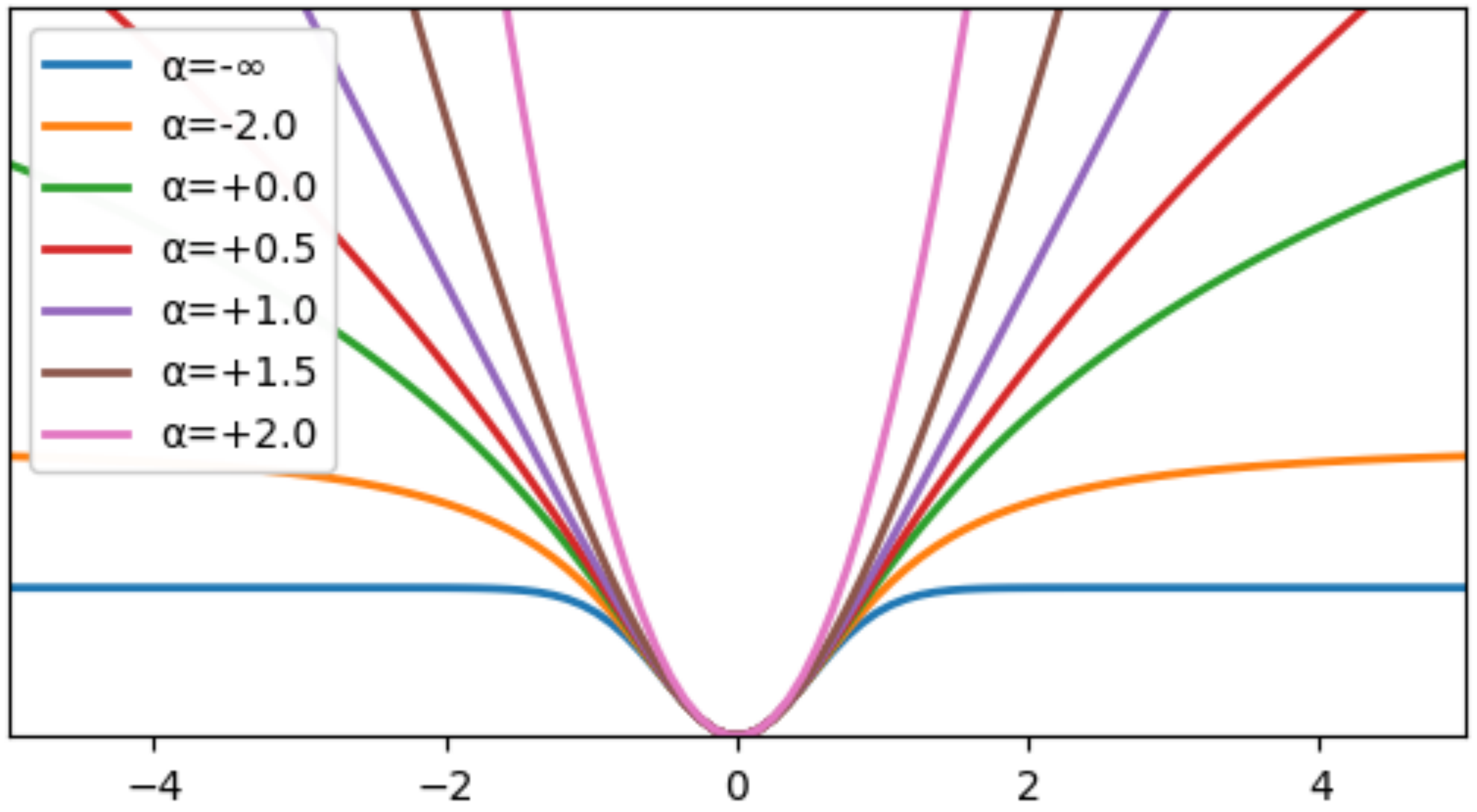
- automatically identify non-consistent information
- model distractors as optimization outliers

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

v.s.

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

Family of robust kernels



Robust Estimation

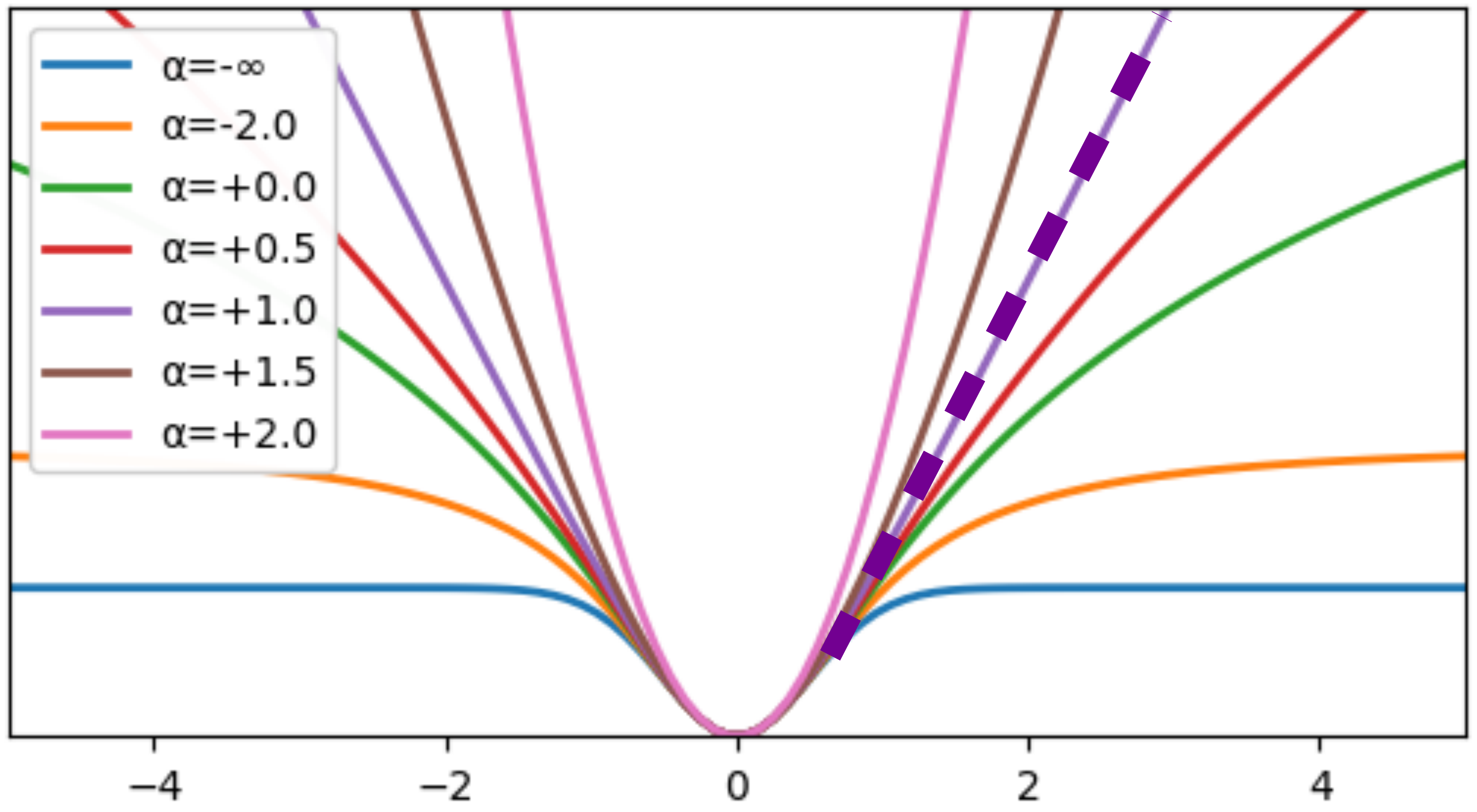
- automatically identify non-consistent information
- model distractors as optimization outliers

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

v.s.

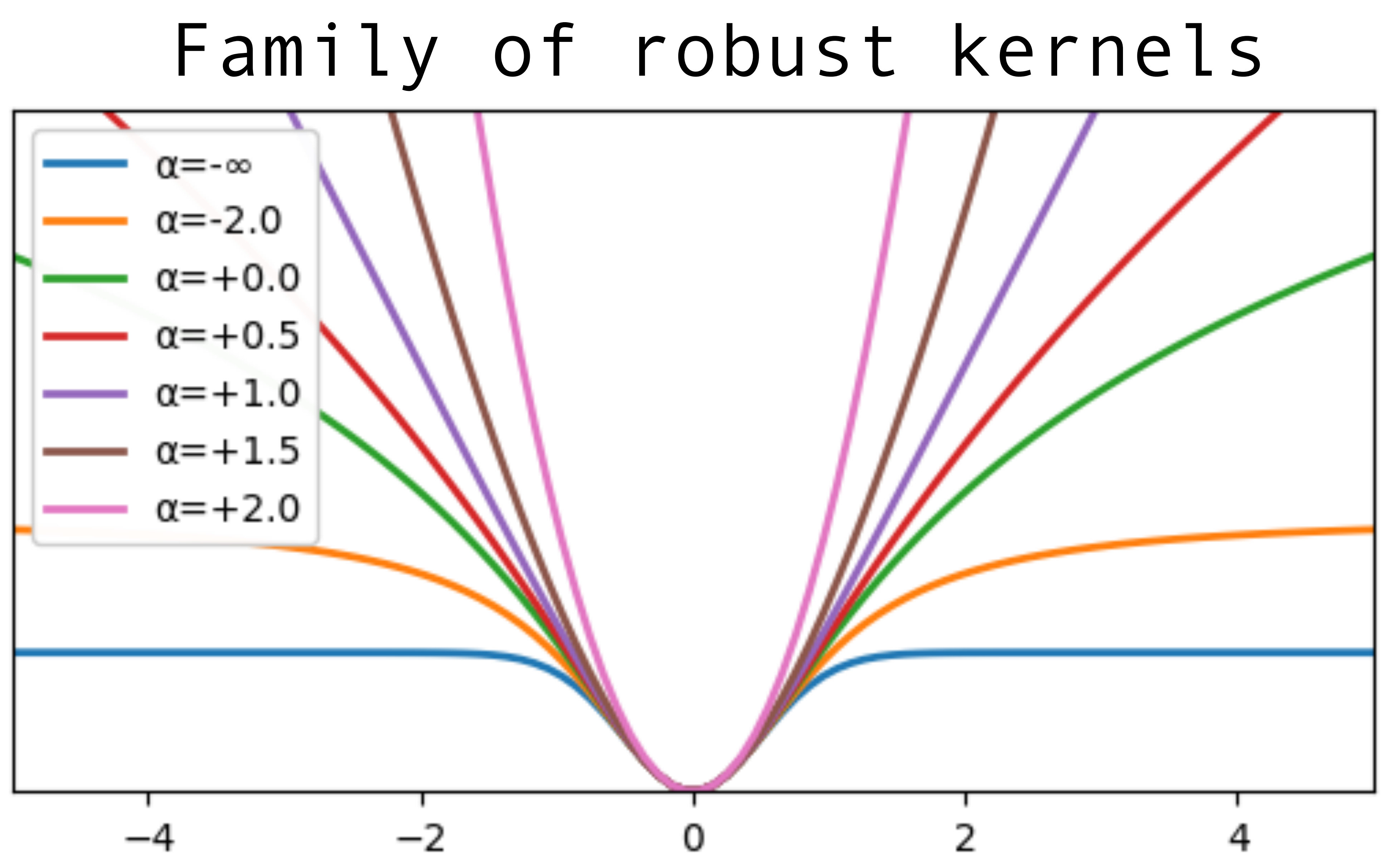
$$\mathcal{L}^{\mathbf{r},i}_{\text{robust}}(\boldsymbol{\theta}) = \kappa(||\mathbf{C}(\mathbf{r};\boldsymbol{\theta}) - \mathbf{C}_i(\mathbf{r})||_2)$$

Family of robust kernels

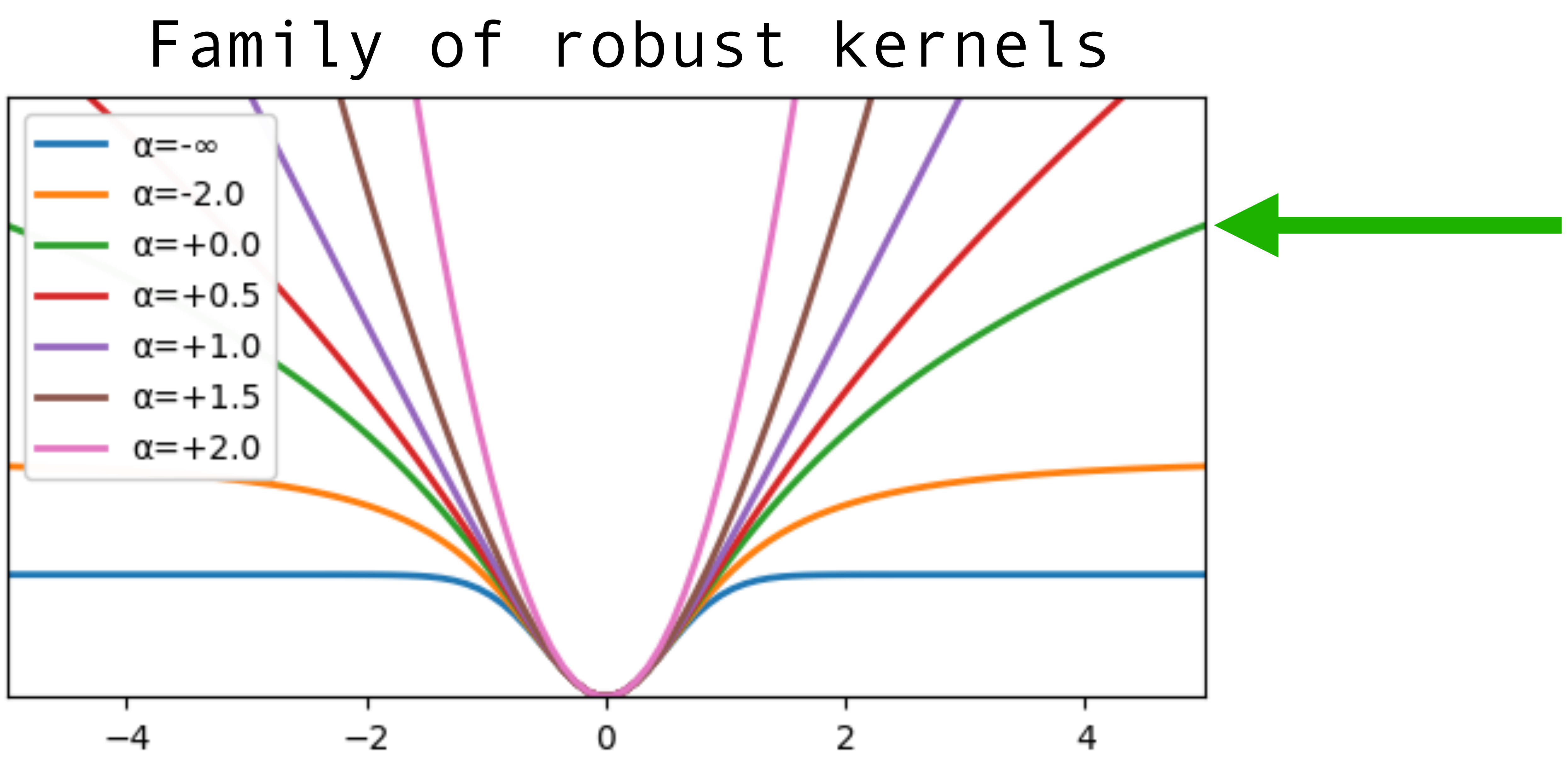


MipNerf360: Charbonnier Loss

Does it work?

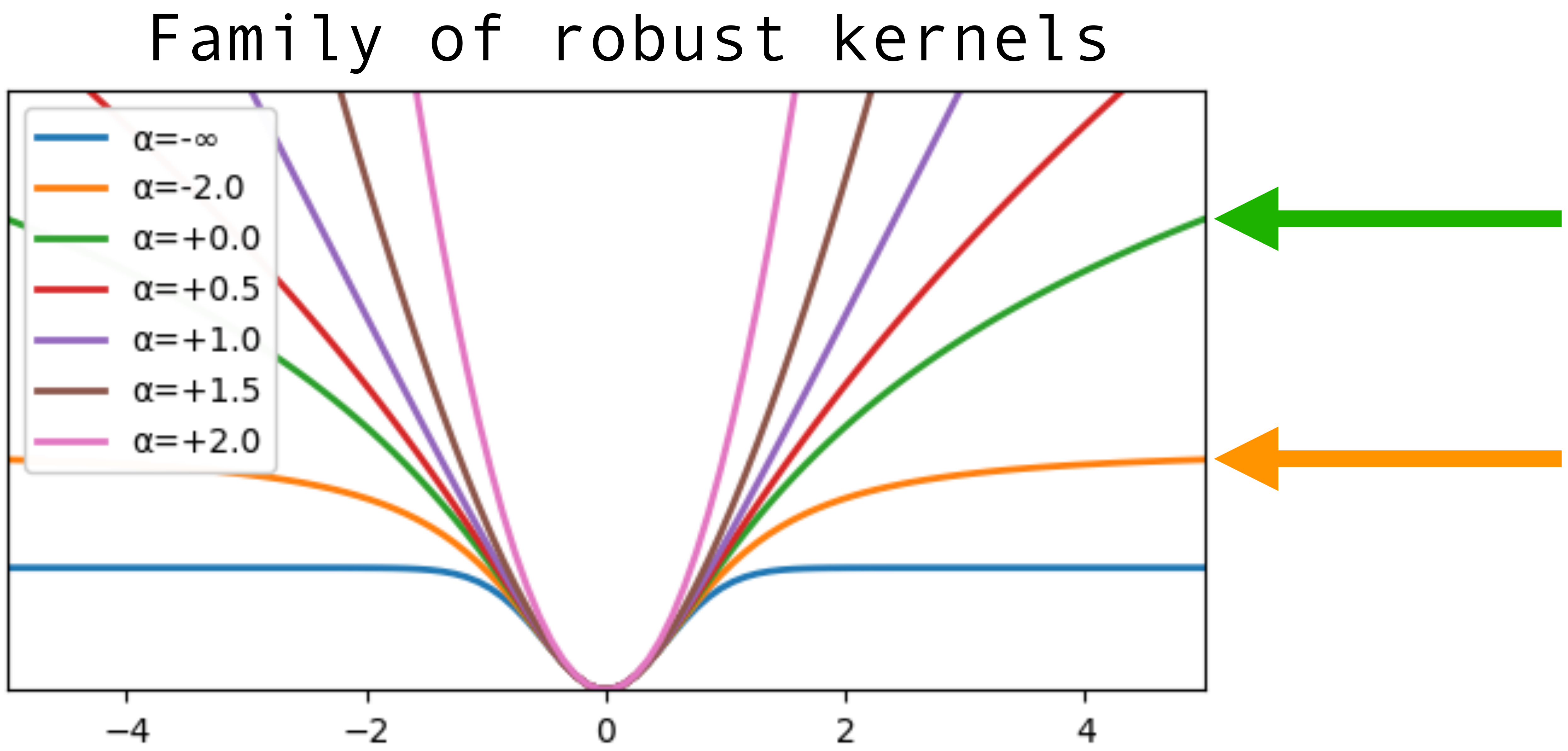


Does it work?



details preserved, but floaters

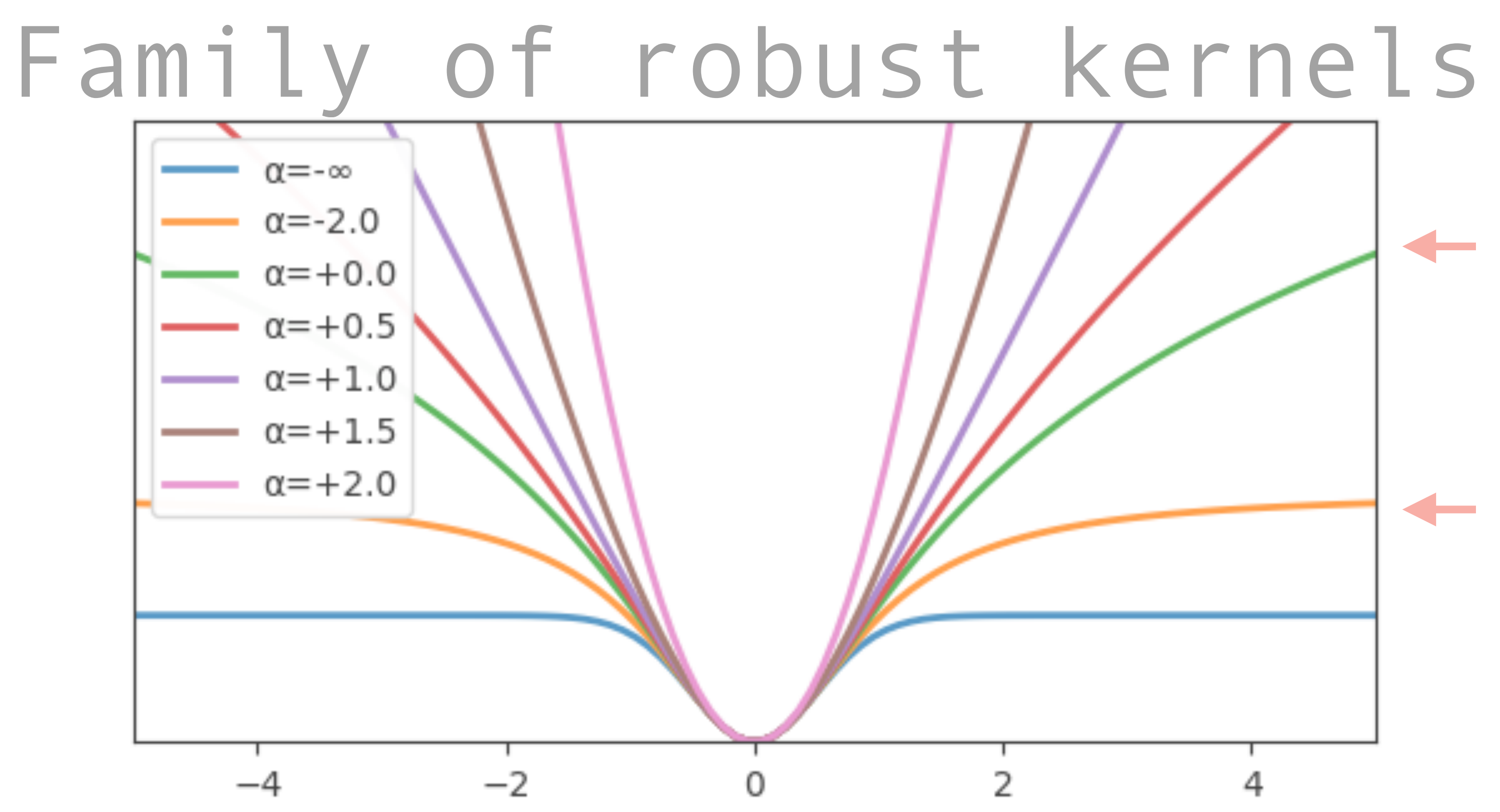
Does it work?

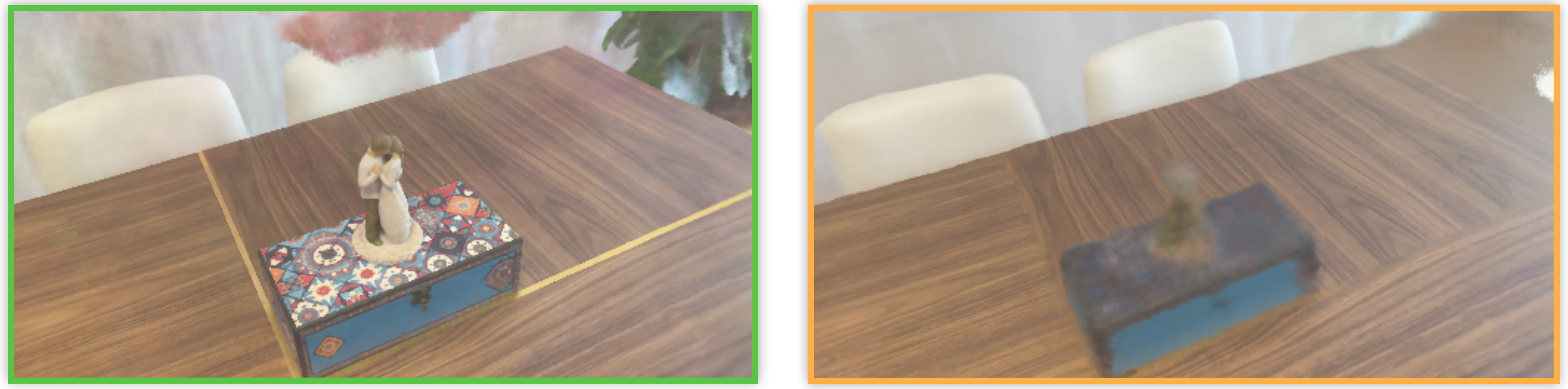


details preserved, but floaters

floaters gone, but details not preserved

Are outlier pixels independant in transient objects?





Good at **independent** noise

VS

Bad at **structured** noise

Robust Optimization w/ IRLS

- IRLS: Iteratively Re-Weighted Least Squares
  - weights are spatially consistent
  - weights are in {0, 1}   (hard weights)

Robust Optimization w/ IRLS

- IRLS: Iteratively Re-Weighted Least Squares
  - weights are spatially consistent
  - weights are in {0, 1}   (hard weights)

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

v.s.

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

Robust Optimization w/ IRLS

- IRLS: Iteratively Re-Weighted Least Squares
  - weights are spatially consistent
  - weights are in {0, 1}   (hard weights)

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

v.s.

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

Residuals at the previous iteration

Trimmed / Diffused Least Squares

$$\tilde{\omega}(\mathbf{r}) = \epsilon(\mathbf{r}) \leq \mathcal{T}_\epsilon \ , \quad \mathcal{T}_\epsilon = \mathrm{Median}_{\mathbf{r}}\{\epsilon(\mathbf{r})\} \ .$$

threshold the residuals

residuals – $\epsilon(\mathbf{r})$

inliers – $\tilde{\omega}(\mathbf{r})$

Trimmed / Diffused Least Squares

$$\tilde{\omega}(\mathbf{r}) = \epsilon(\mathbf{r}) \leq \mathcal{T}_\epsilon \ , \quad \mathcal{T}_\epsilon = \text{Median}_\mathbf{r}\{\epsilon(\mathbf{r})\} \ .$$

threshold the residuals

$$\mathcal{W}(\mathbf{r}) = (\tilde{\omega}(\mathbf{r}) \circledast \mathcal{B}_{3\times3}) \geq \mathcal{T}_\circledast \ , \quad \mathcal{T}_\circledast = 0.5 \ .$$

diffuse the classification

residuals – $\epsilon(\mathbf{r})$

inliers – $\tilde{\omega}(\mathbf{r})$

diffusion – $\tilde{\omega}(\mathbf{r}) \circledast \mathcal{B}_{3\times3}$

Trimmed / Diffused Least Squares

$$\tilde{\omega}(\mathbf{r}) = \epsilon(\mathbf{r}) \leq \mathcal{T}_\epsilon \;, \quad \mathcal{T}_\epsilon = \text{Median}_\mathbf{r}\{\epsilon(\mathbf{r})\} \;.$$

threshold the residuals

$$\mathcal{W}(\mathbf{r}) = (\tilde{\omega}(\mathbf{r}) \circledast \mathcal{B}_{3\times 3}) \geq \mathcal{T}_\circledast \;, \quad \mathcal{T}_\circledast = 0.5 \;.$$

diffuse the classification

$$\omega(\mathcal{R}_8(\mathbf{r})) = \mathbb{E}_{\mathbf{s}\sim\mathcal{R}_{16}(\mathbf{r})}\left[\mathcal{W}(\mathbf{s})\right] \geq \mathcal{T}_\mathcal{R} \,, \quad \mathcal{T}_\mathcal{R} = 0.6 \,.$$

threshold again

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

Trimmed / Diffused Least Squares

$$\tilde{\omega}(\mathbf{r}) = \epsilon(\mathbf{r}) \leq \mathcal{T}_\epsilon \ , \quad \mathcal{T}_\epsilon = \mathrm{Median}_{\mathbf{r}}\{\epsilon(\mathbf{r})\} \ .$$

threshold the residuals

$$\mathcal{W}(\mathbf{r}) = (\tilde{\omega}(\mathbf{r}) \circledast \mathcal{B}_{3\times3}) \geq \mathcal{T}_\circledast \ , \quad \mathcal{T}_\circledast = 0.5 \ .$$

diffuse the classification

$$\omega(\mathcal{R}_8(\mathbf{r})) = \mathbb{E}_{\mathbf{s}\sim\mathcal{R}_{16}(\mathbf{r})}\left[\mathcal{W}(\mathbf{s})\right] \geq \mathcal{T}_\mathcal{R} \, , \quad \mathcal{T}_\mathcal{R} = 0.6 \, .$$

threshold again

residuals – $\epsilon(\mathbf{r})$

inliers – $\tilde{\omega}(\mathbf{r})$

diffusion – $\tilde{\omega}(\mathbf{r}) \circledast \mathcal{B}_{3\times3}$

(IRLS) weights – $\mathcal{W}(\mathbf{r})$

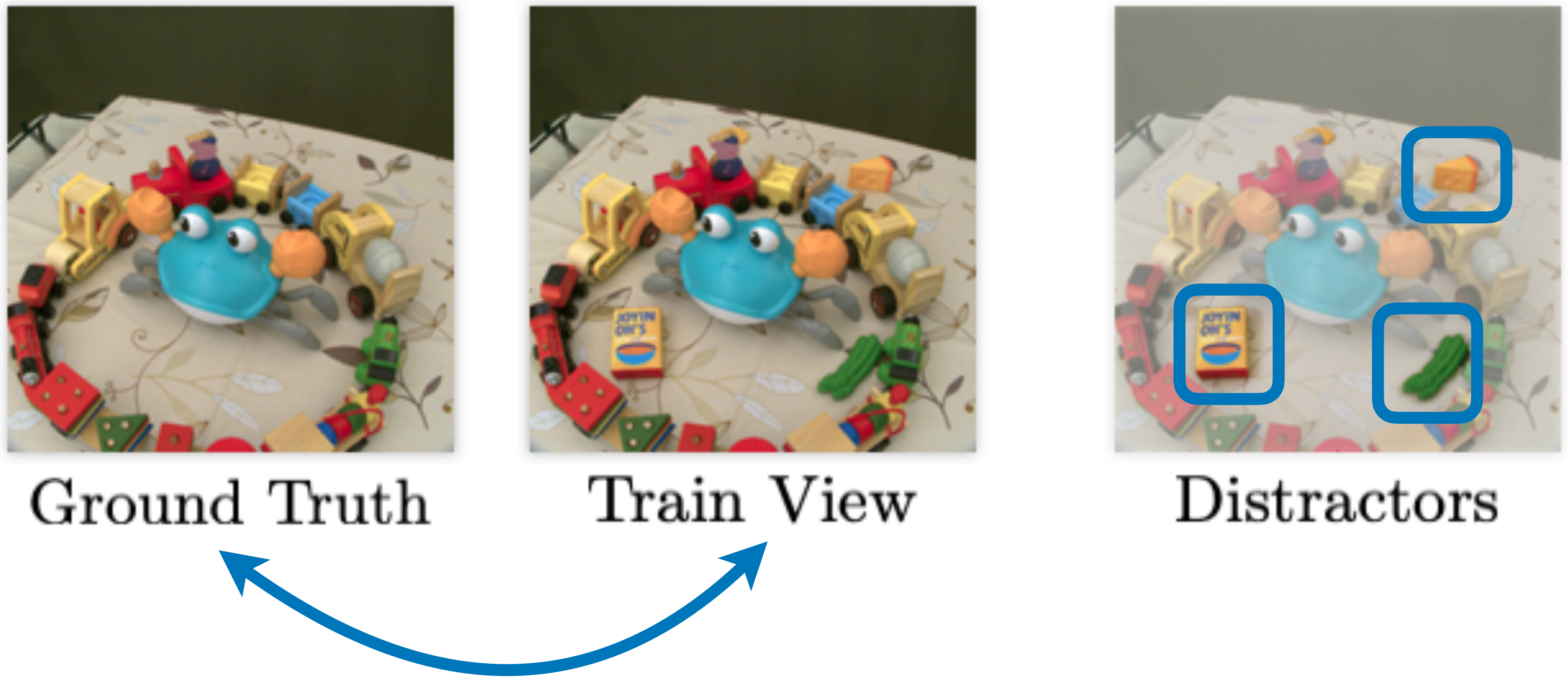
Training Progress w/ IRLS

Ground Truth

Train View

Distractors

Training Progress w/ IRLS



Training Progress w/ IRLS

Distractors

.5% 2% 12% 100% (t/T)

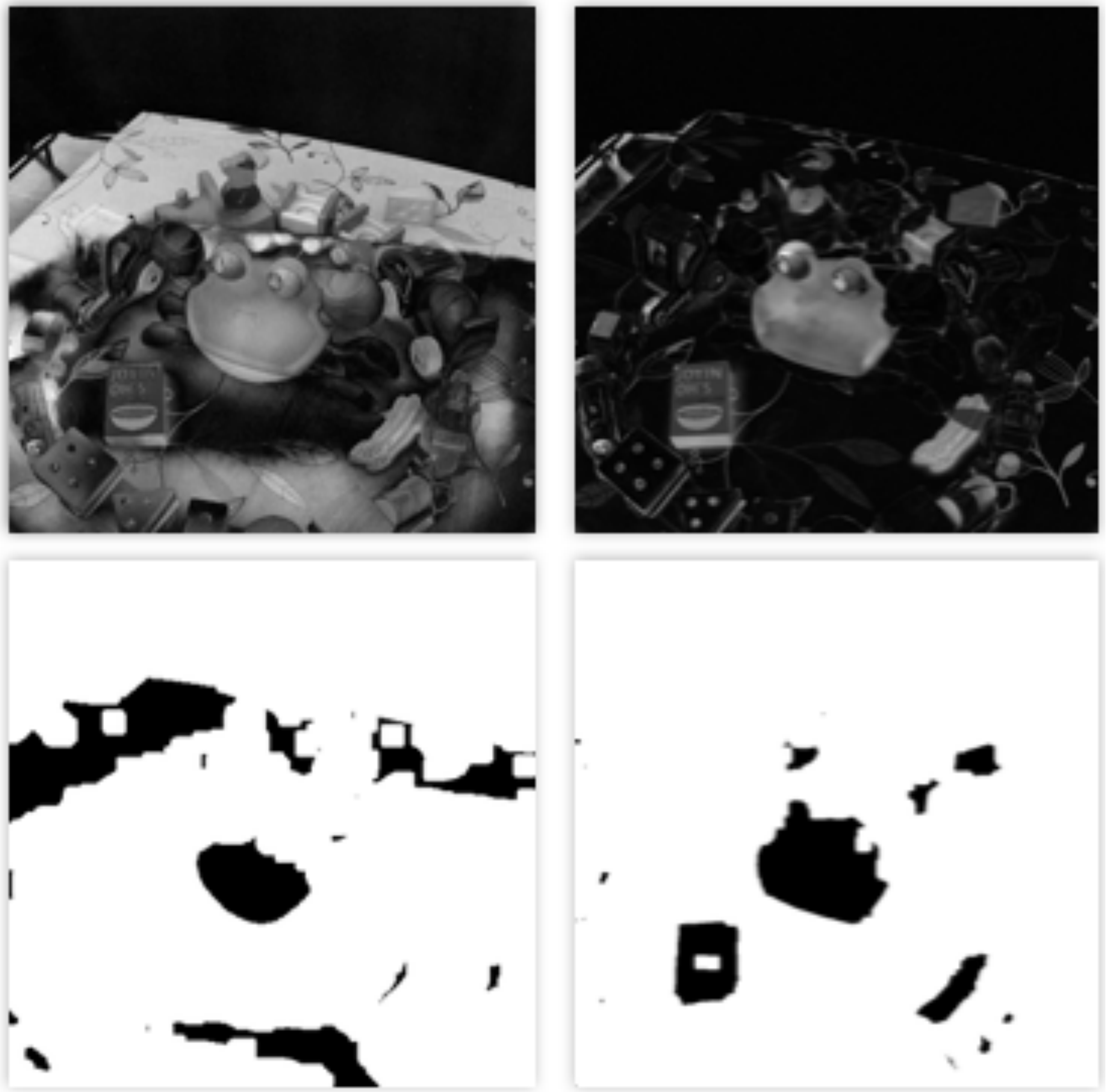
residuals
$\epsilon(\mathbf{r})$

weights
$\mathcal{W}(\mathbf{r})$

Training Progress w/ IRLS

Distractors

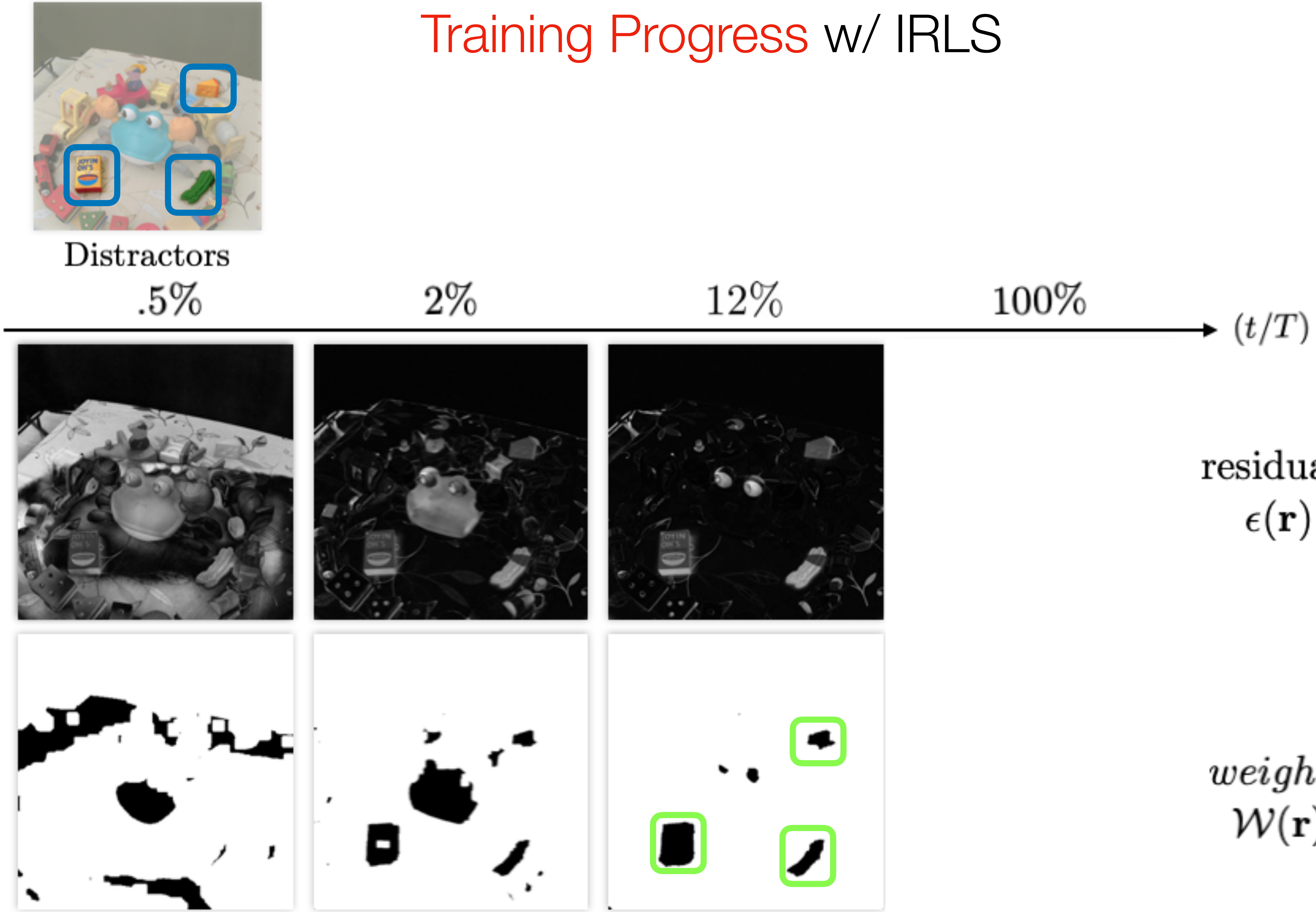
.5% 2% 12% 100% → (t/T)



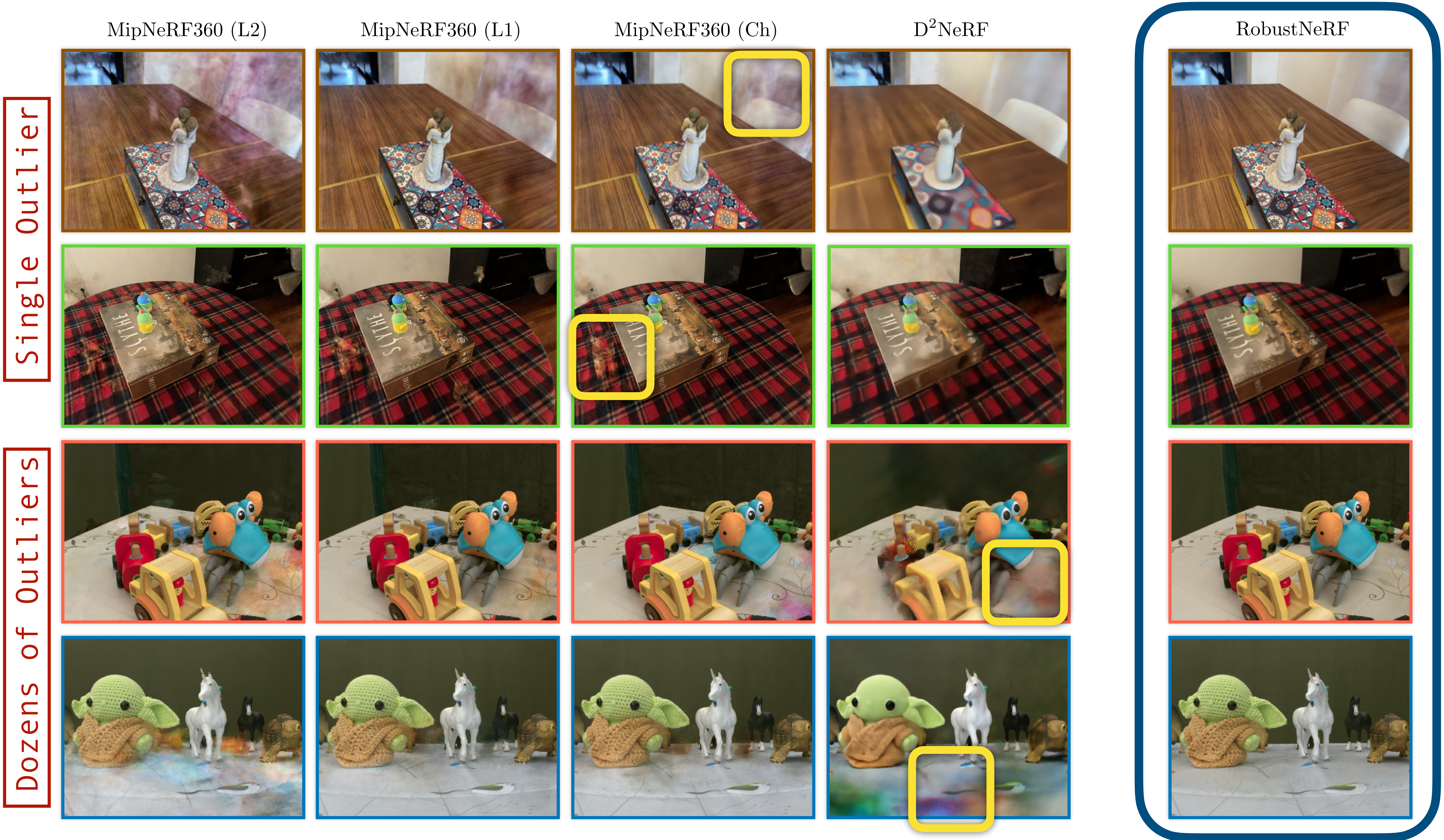
residuals $\epsilon(\mathbf{r})$

weights $\mathcal{W}(\mathbf{r})$

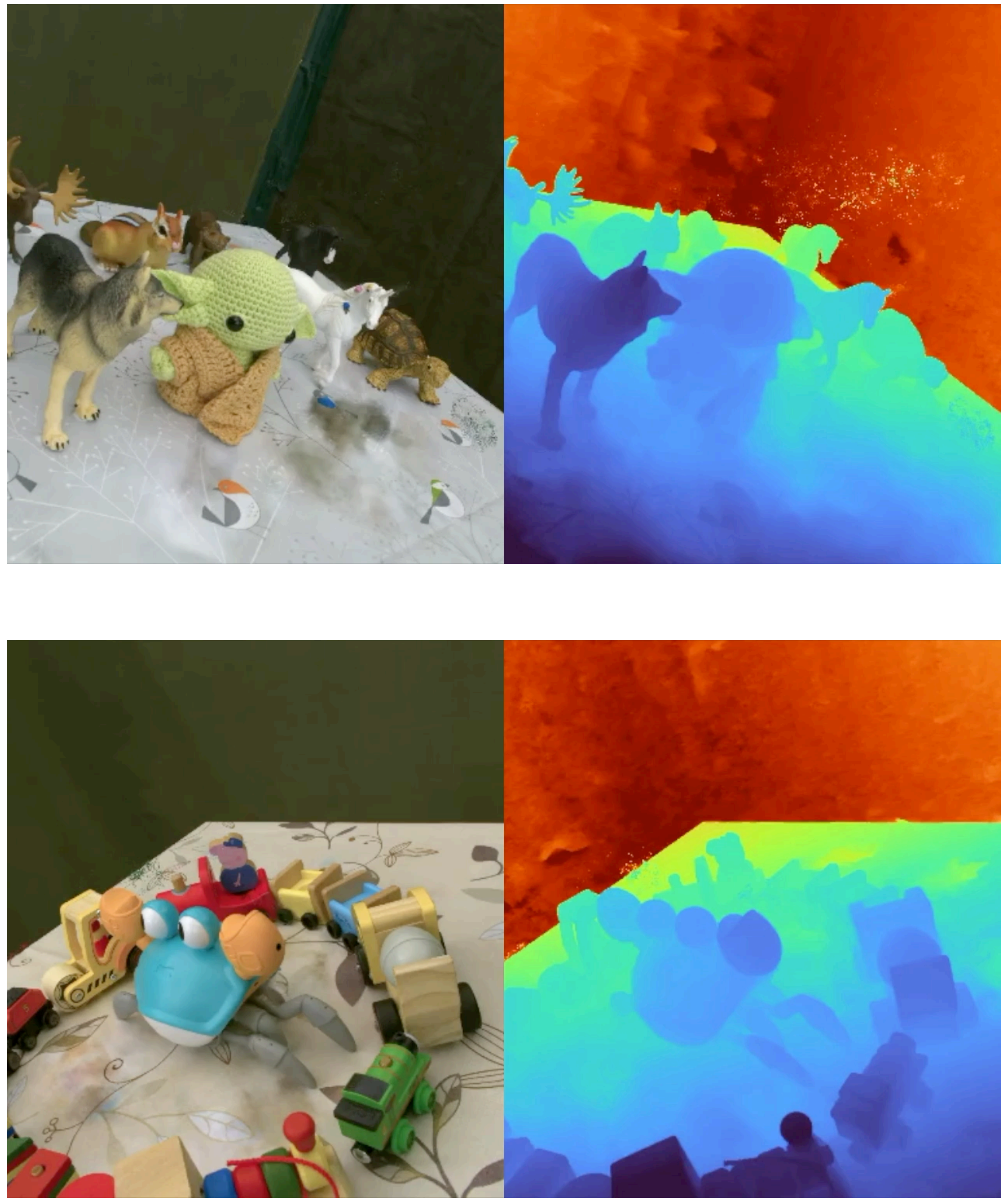
Training Progress w/ IRLS



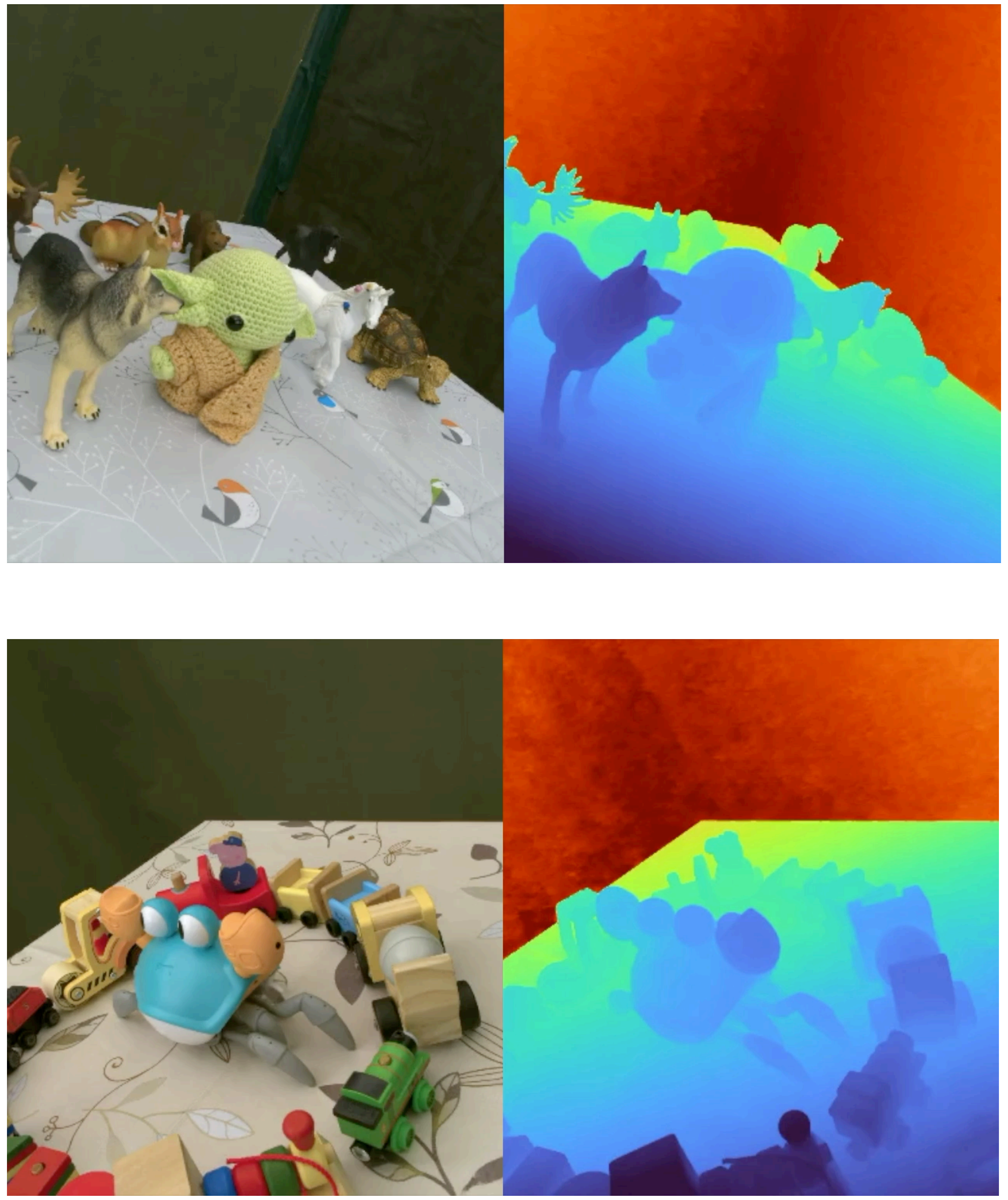

MipNeRF360 (L2)
MipNeRF360 (L1)
MipNeRF360 (Ch)
D2NeRF
RobustNeRF
Single Outlier
Dozens of Outliers


MipNeRF360 (L2)
MipNeRF360 (L1)
MipNeRF360 (Ch)
D^{2}NeRF
RobustNeRF
Single Outlier
Dozens of Outliers

Comparison with MipNeRF360



MipNeRF360



RobustNeRF

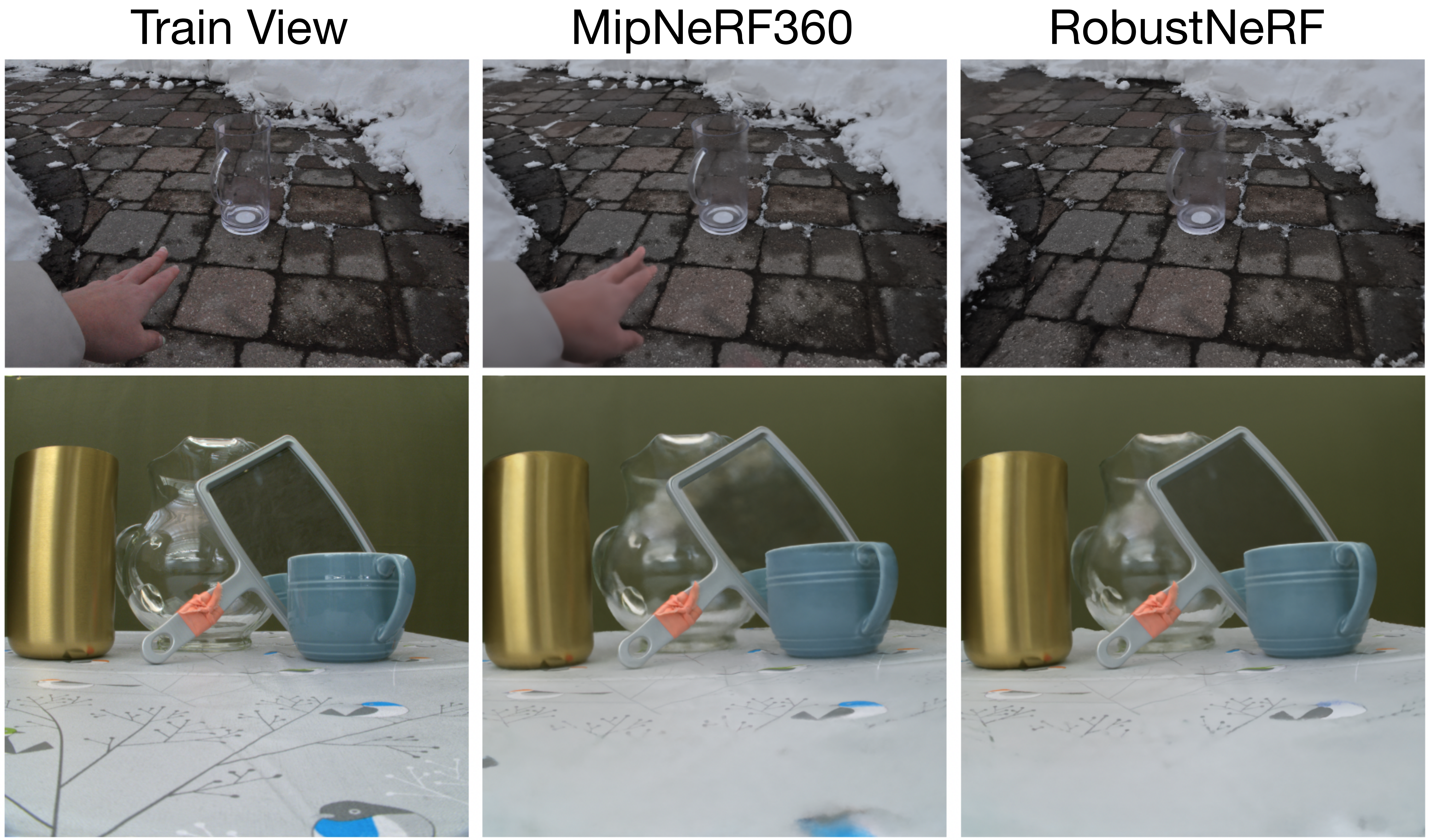
Viewdependent Scene Challenge

Train View

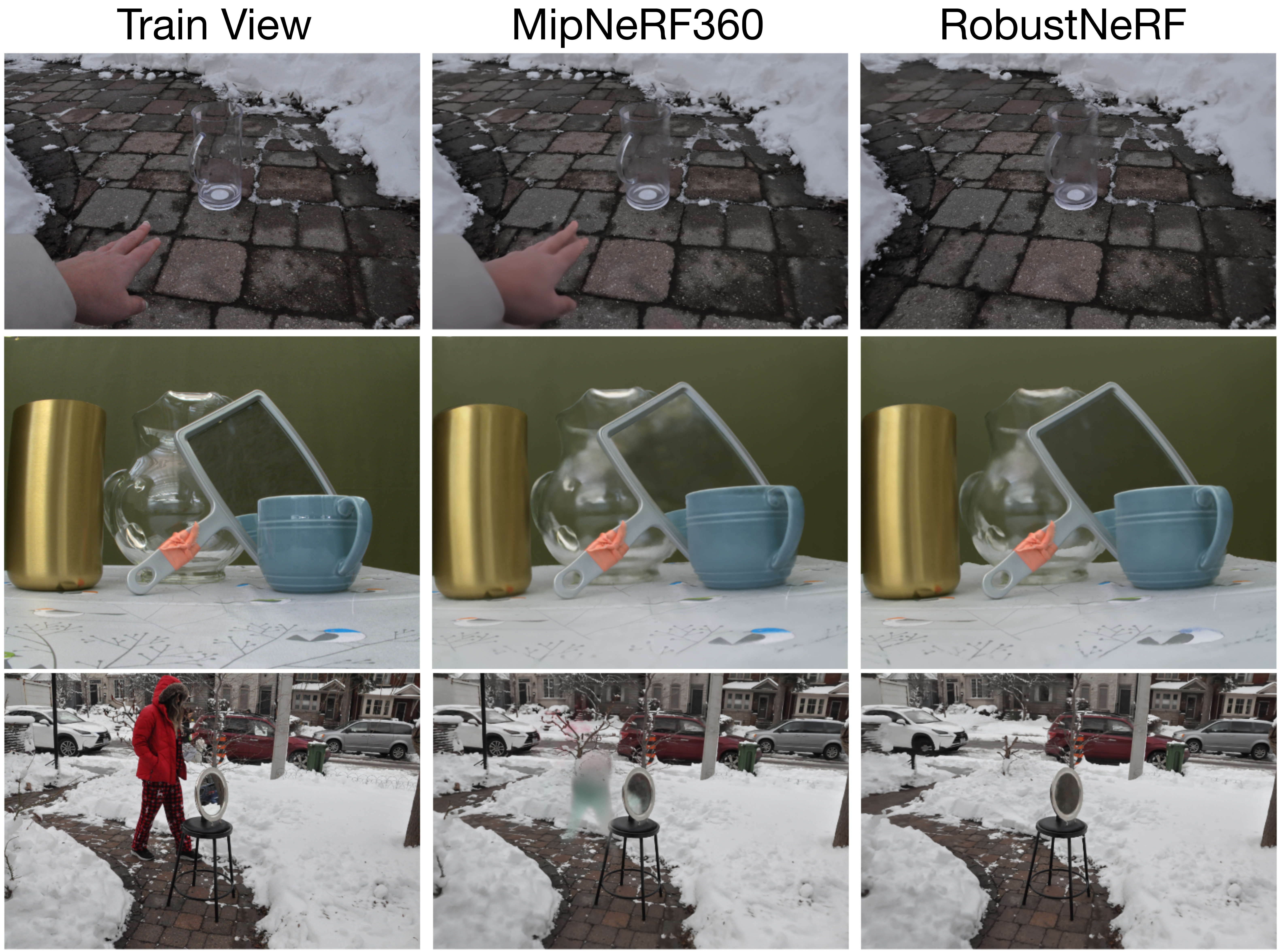
MipNeRF360

RobustNeRF

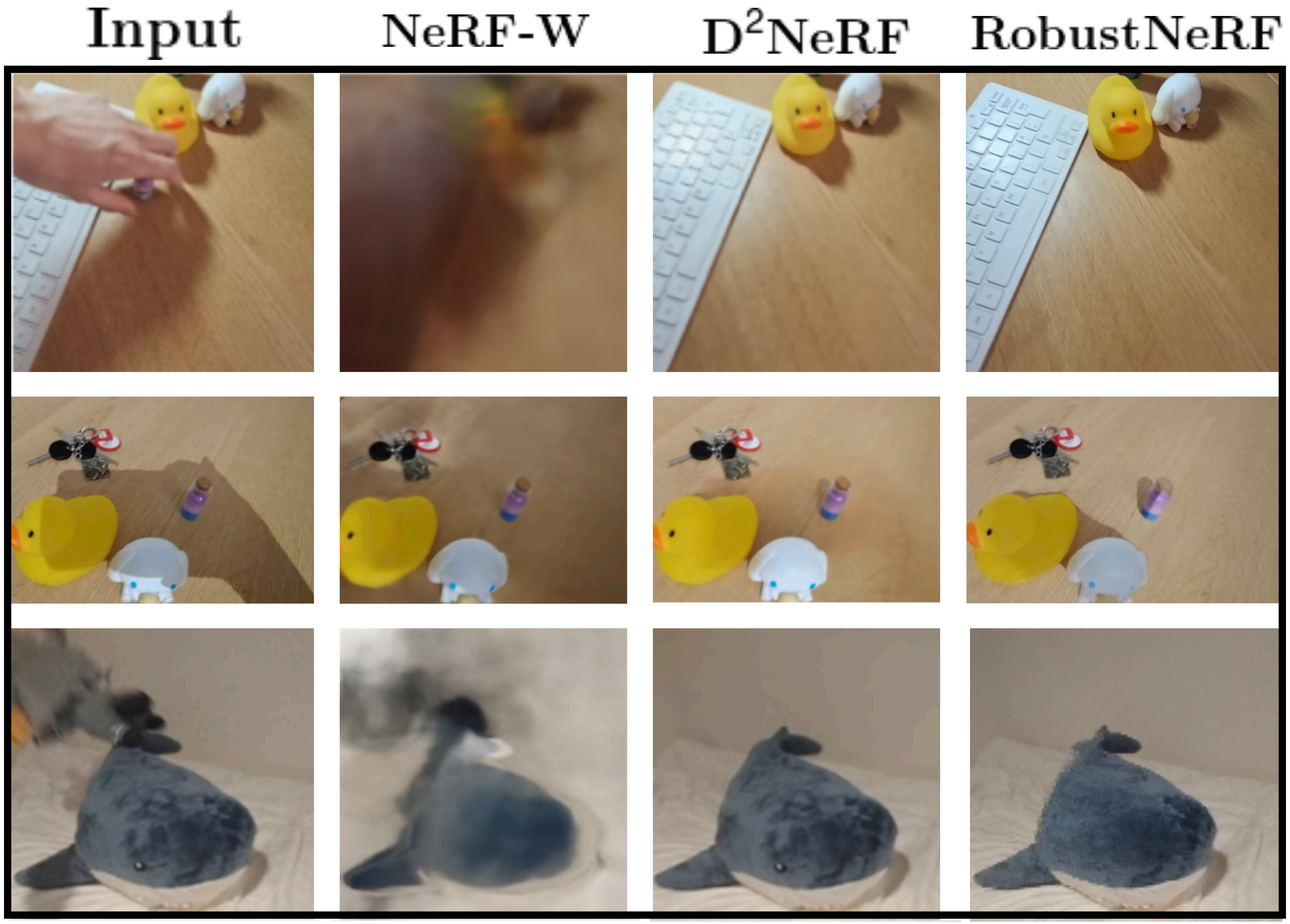
Viewdependent Scene Challenge



Viewdependent Scene Challenge

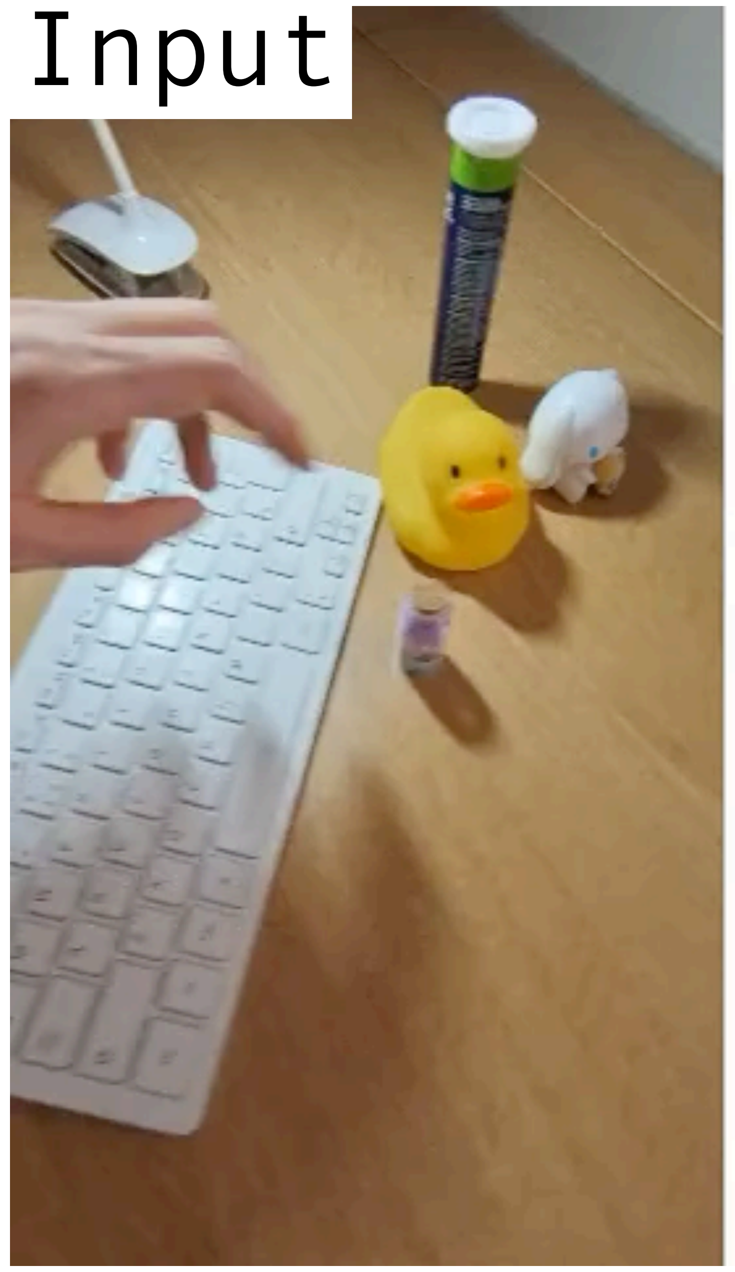


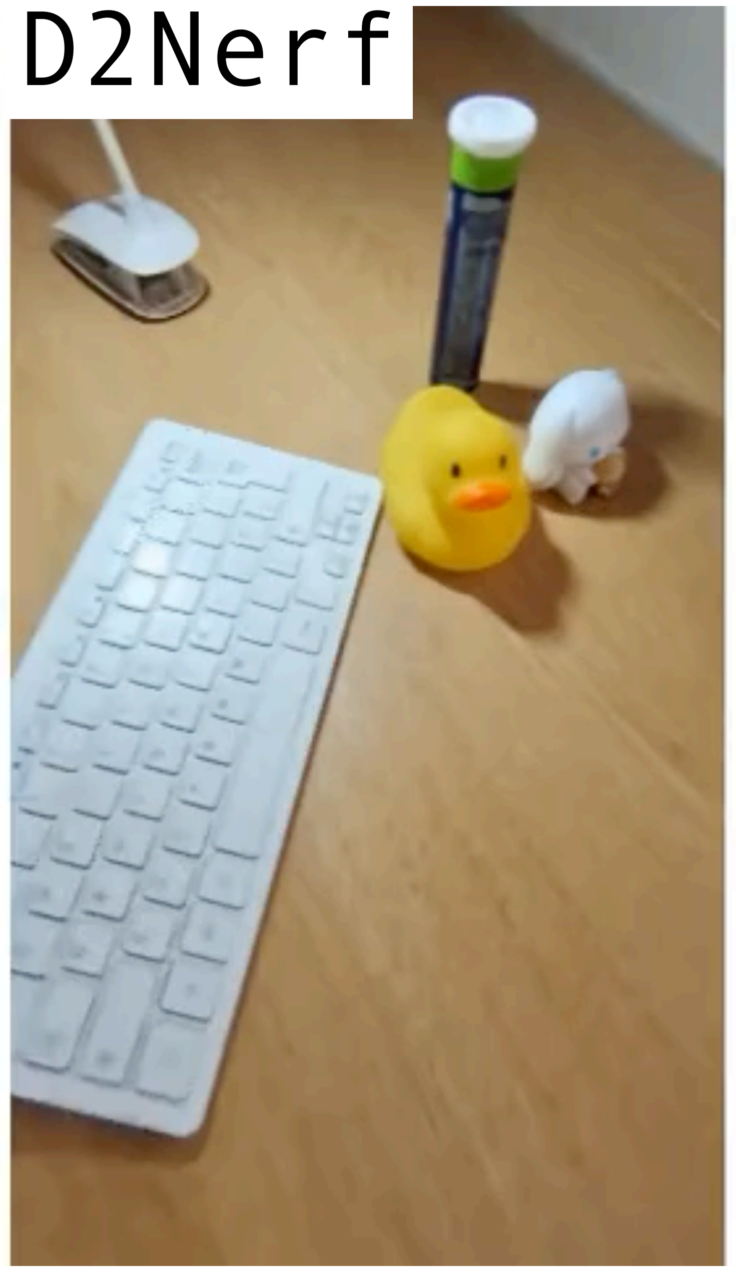
Transient Shadow Challenge

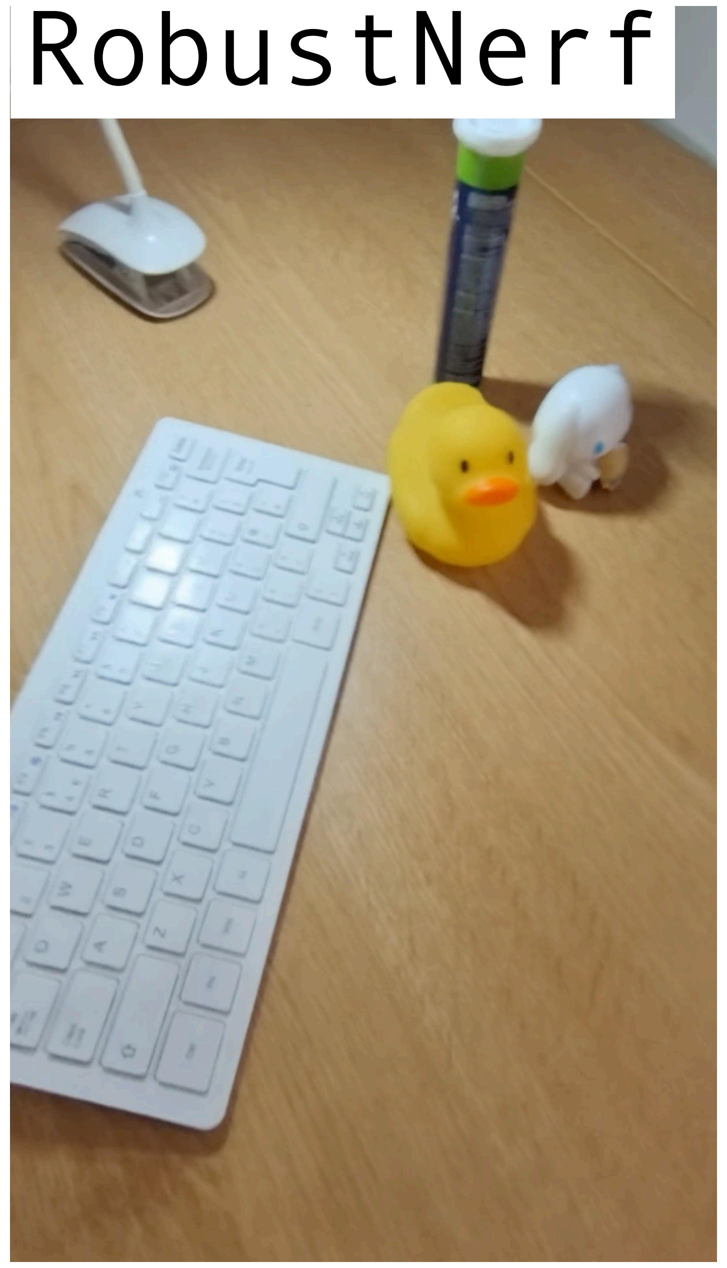


Wu, Tianhao, et al. "D $^ 2$ NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video." *arXiv preprint arXiv:2205.15838* (2022).

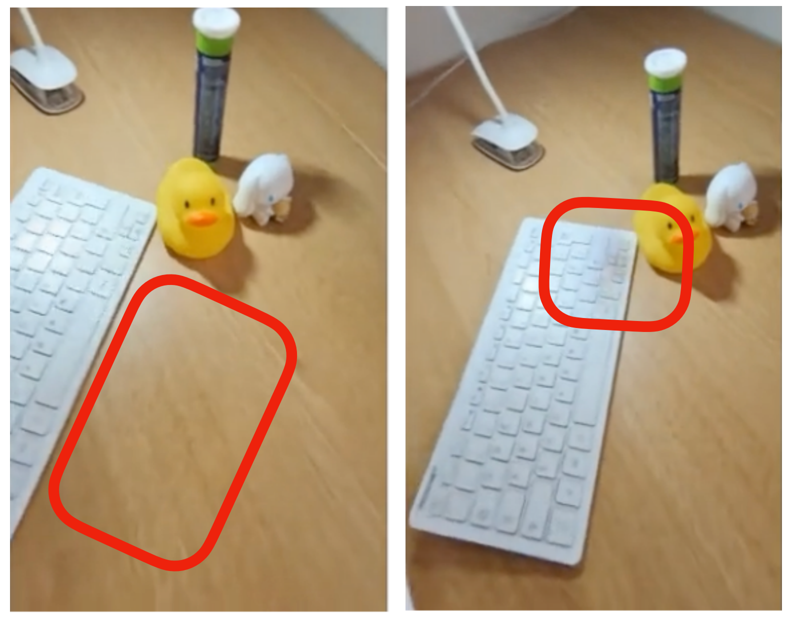
Side by Side Video on ”D2NeRF Pick”







1`` 6``

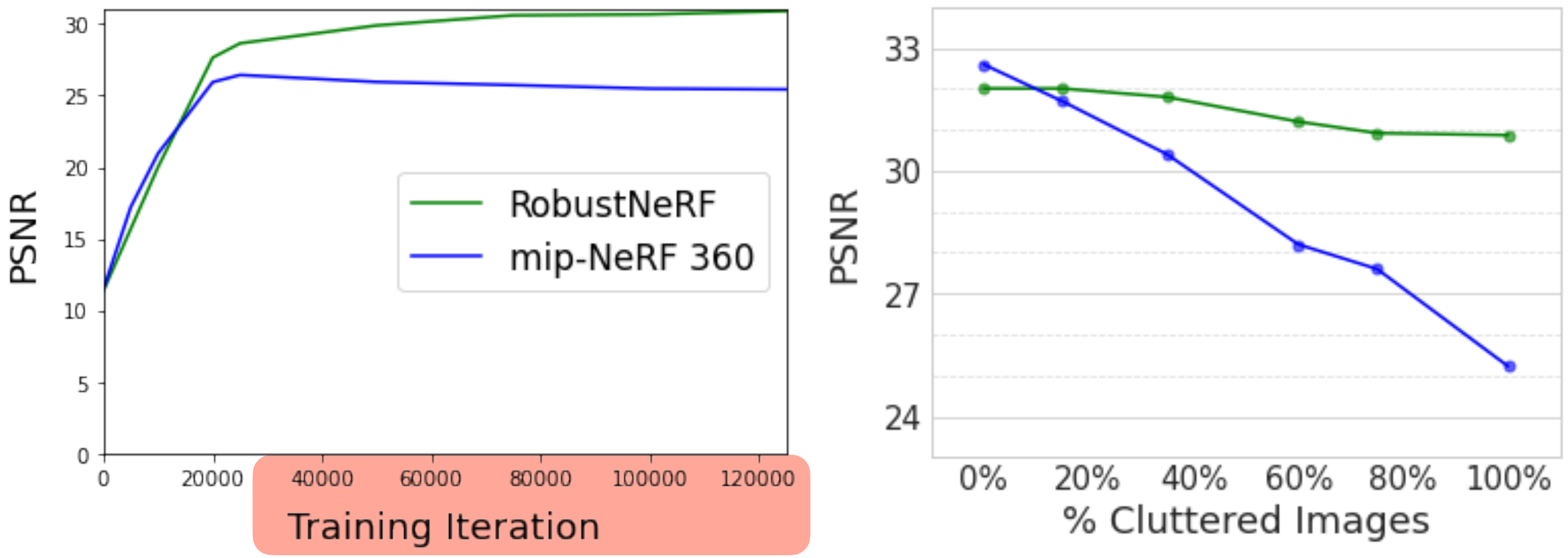


shadow

Floater

Wu, Tianhao, et al. "D $^ 2$ NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video." *arXiv preprint arXiv:2205.15838* (2022).

Limitations

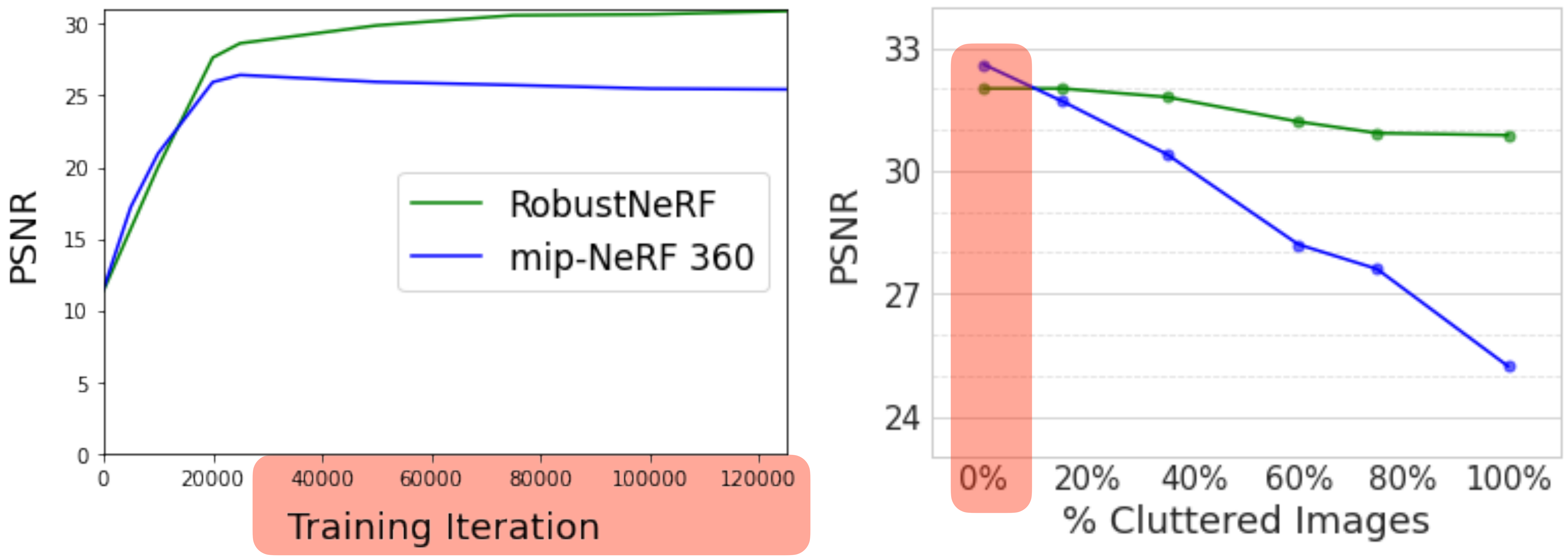


Statistical inefficiency

Scenes take longer to reach peak accuracy. Clean scenes suffer a bit.

Cause: robust loss always considers a (fixed) portion of data as outliers

Limitations



Statistical inefficiency

Scenes take longer to reach peak accuracy. Clean scenes suffer a bit.

Cause: robust loss always considers a (fixed) portion of data as outliers

Limitations

Lots of transients will confuse Colmap.

No Colmap No Camera No NeRF.

Transients in the mirror -> The mirrored surface is replaced with blanks.

RobustNeRF

Poster ID: THU-PM-002

https://robustnerf.github.io