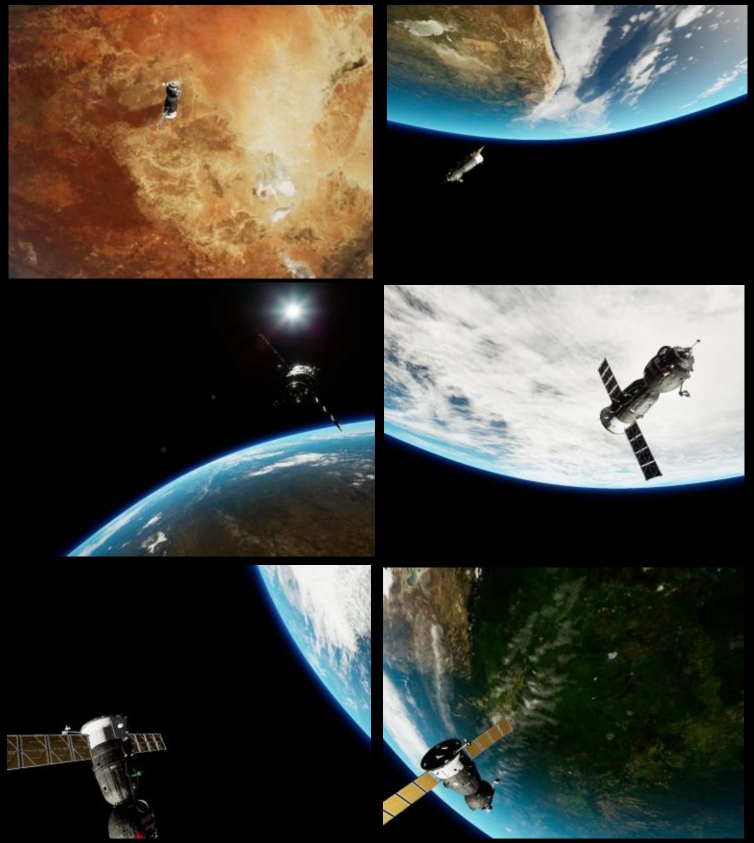
Deep learning and Unreal Engine for Spacecraft Pose Estimation

Pedro F. Proença and Yang Gao

URSO: Unreal Rendered Spacecraft On Orbit

Features

- Built on UE4
- Earth:
  - 21600x10800 land and cloud images
  - Ocean specular mask
  - Atmospheric scattering
- Lighting:
  - Sun emissive body
  - Simulated albedo
  - Bloom scatter convolution + Lens flare
- Spacecrafts:
  - High poly models imported from 3D model repositories
  - UE4 materials
- Interface:
  - UnrealCV via server/client TCP connection



URSO: Unreal Rendered Spacecraft On Orbit

Datasets

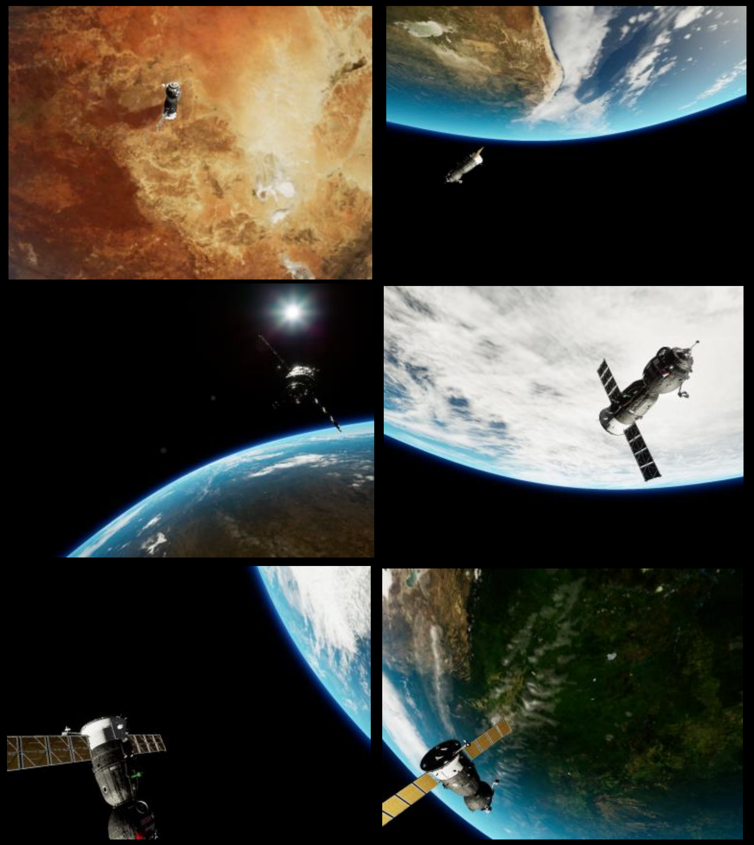
Random Sampling:

- Earth yaw
- Orbit height (LEO)
- Camera and object orientation
- Object location within FOV
- Object distance: [10, 40] m

Camera:

- 1080x960 px
- 90 deg horizontal FOV
- Auto-exposure

Dataset size: 5000 images



URSO: Unreal Rendered Spacecraft On Orbit

Why Unreal Engine 4?

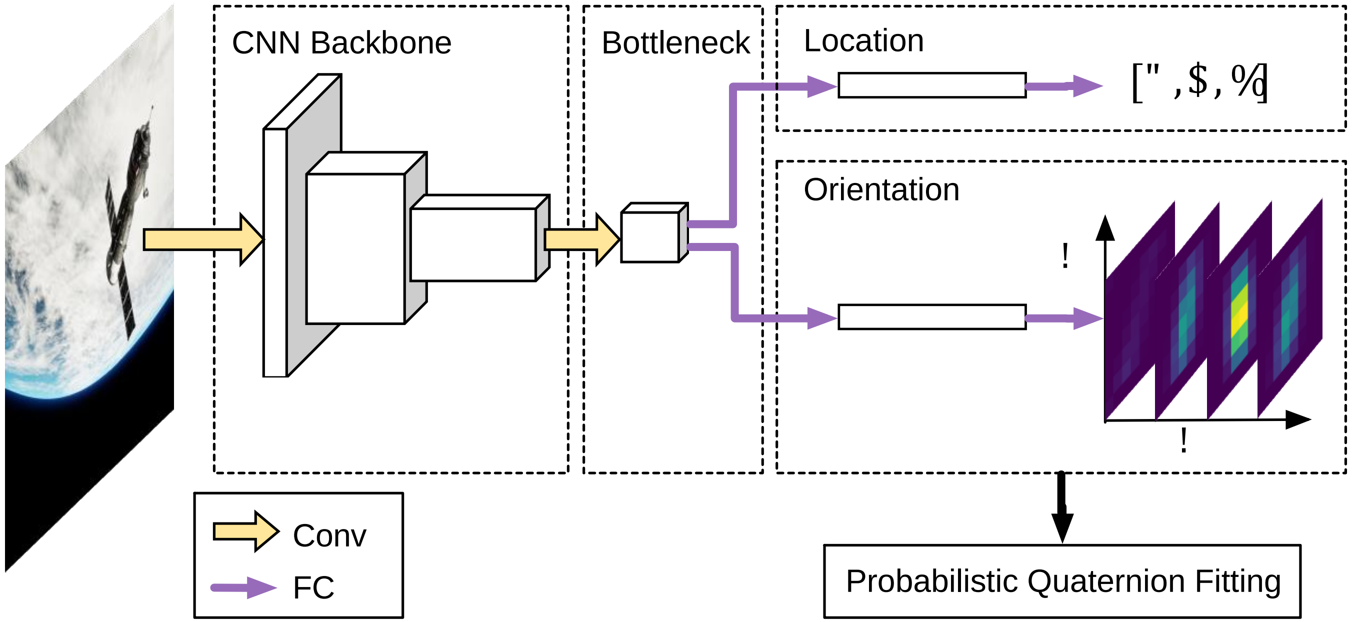
- Other space visual simulators:
    - **PANGU** by University of Dundee supported by ESA
    - **SurRende**r by Airbus
        - Ray tracing
        - ‘Giant textures’

- UE4 pros:
    - State-of-the-art graphics
    - Community
    - Industry (e.g. autonomous driving, robotics)
    - Source-available
    - Physical based shading (materials, lighting)
    - Physical based cameras:
        - Exposure metering mode, F-Stop, ISO, etc.

- UE4 cons:
    - Steep learning curve
    - Limited, sparse documentation
    - Max texture resolution: 8192 x 8192 px
    - World size vs precision error
    - Importing materials from other tools (e.g. Blender)

Spacecraft Pose Estimation

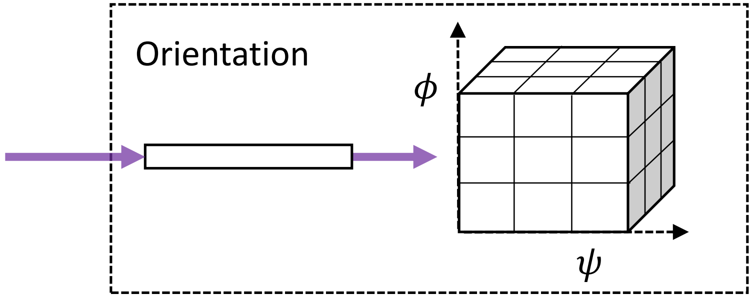
Architecture

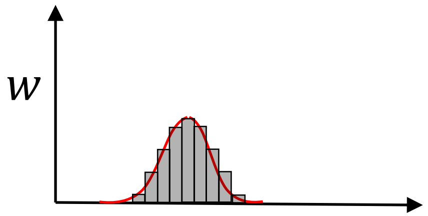


- To regress or to classify that is the question?

Spacecraft Pose Estimation

Probabilistic orientation estimation





Probability mass function

Training — Soft Encoding:

$$w_i = \frac{e^{-\frac{\theta_i^2}{2\sigma^2}}}{\sum_i^N e^{-\frac{\theta_i^2}{2\sigma^2}}} \qquad \theta_i = \frac{2\cos^{-1}\left|b_i^T q_{gt}\right|}{\pi}$$

Testing — Weighted average quaternion

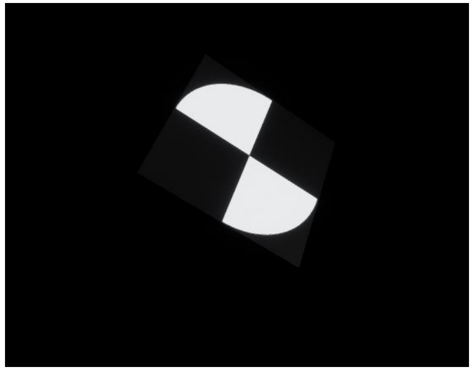
$$\bar{q} = Null\left(\sum_i^N w_i\left(b_i^T b_i\right)\right)$$

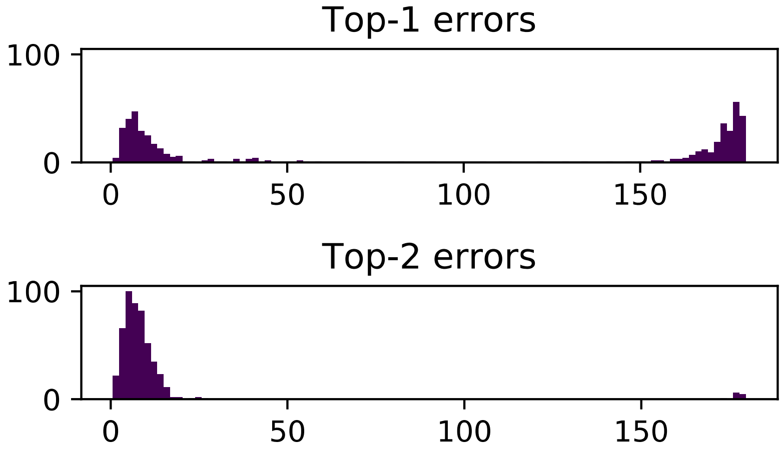
Spacecraft Pose Estimation

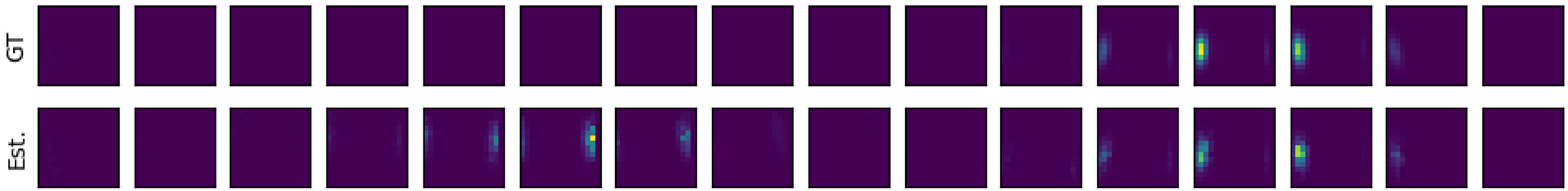
Ambiguous orientation

Multimodal orientation estimation:

- Expectation Maximization to fit N Gaussians
- Use log likelihood to score Mixture Model

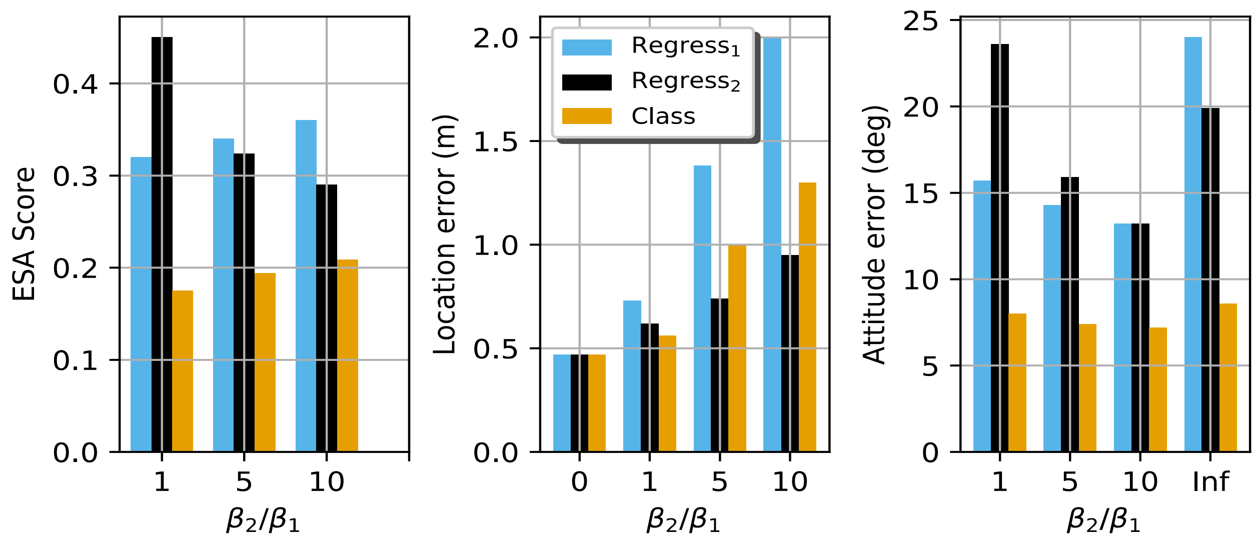






Spacecraft Pose Estimation

Results: Weighting Loss Functions



$$L_{\text{total}} = \sum_{i}^{m} \beta_1 \frac{\left\| \hat{t}^{(i)} - t_{gt}^{(i)} \right\|_2}{\left\| t_{gt}^{(i)} \right\|_2} + \beta_2 L_{\text{orientation}}$$

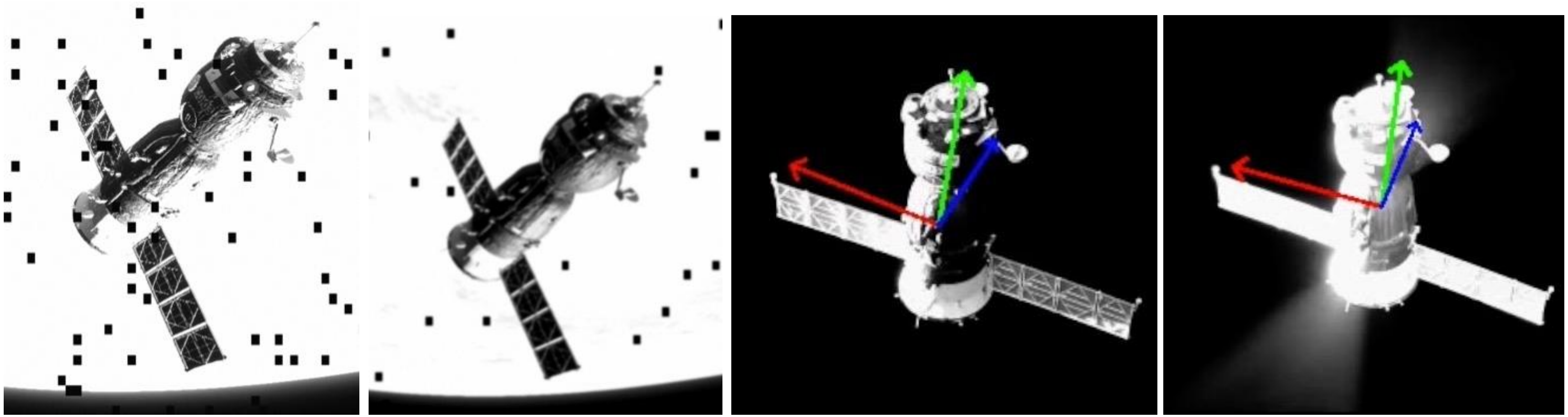
$$L_{\text{orientation}} = \begin{cases} 1 - \left| \hat{q}^{(i)} . q_{gt}^{(i)} \right|, & \text{regress}_1 \\ \cos^{-1} \left| \hat{q}^{(i)} . q_{gt}^{(i)} \right|, & \text{regress}_2 \\ -w_{gt}^{(i)} . \log\left(\widehat{w}^{(i)}\right), & \text{class} \end{cases}$$

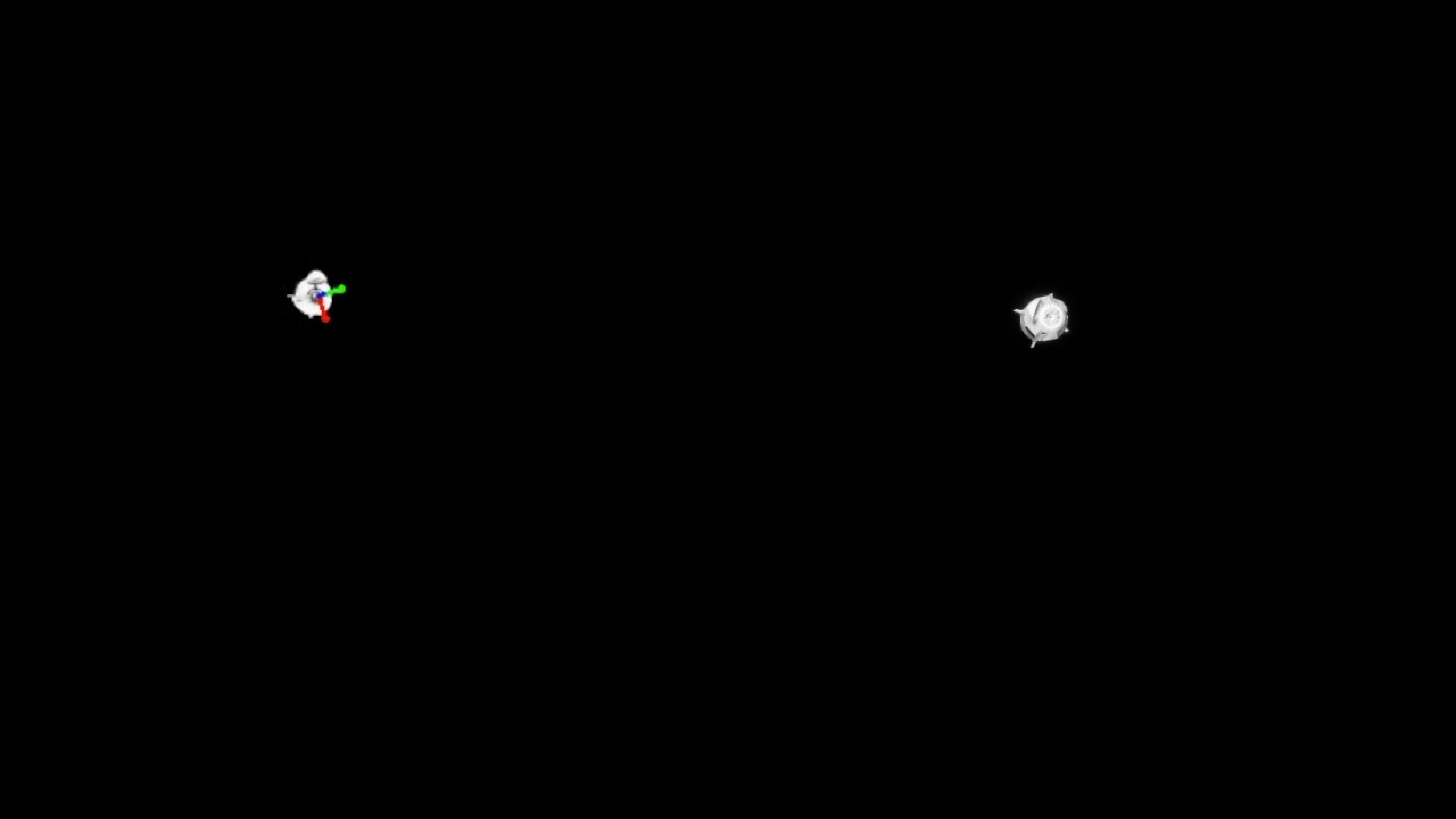
Spacecraft Pose Estimation

Data Augmentation and Sim-to-real

- Image Augmentation during Training:
  - In-plane rotation
  - Small camera orientation perturbations
  - Change image exposure & contrast
  - Convert to grayscale
  - Add AWG noise and blur
  - Drop-out patches

Aug.	Loc error	Ori. error
None	1.06 m	19.5 deg
Rotation	0.56 m	8.0 deg



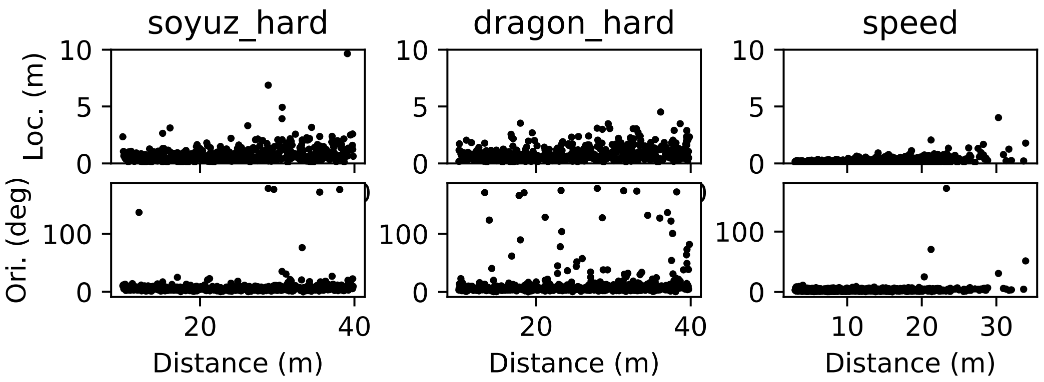


Spacecraft Pose Estimation

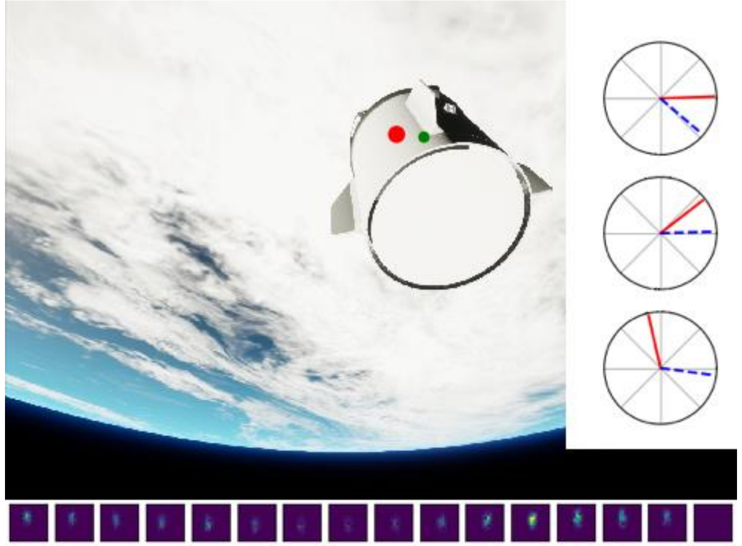
Results

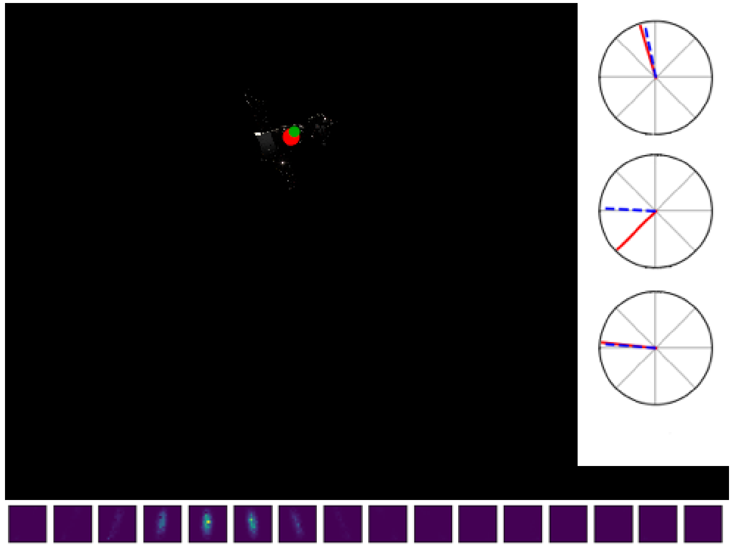
Dataset	Location error	Attitude error
SPEED	0.17 m	4.0 deg
Soyuz hard	0.8 m	7.7 deg
Dragon hard	0.9 m	13.9 deg

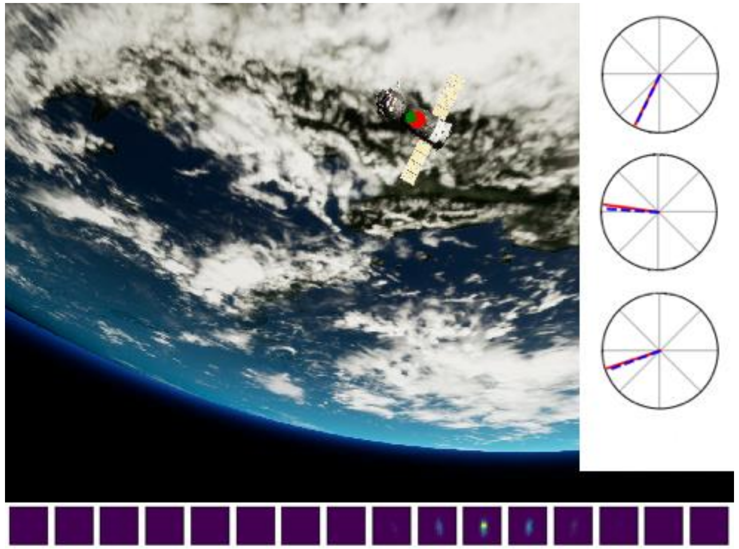
Error distribution



Resnet-50, 128 bottleneck filters, 24 bins per angle ~ 50 M params







Spacecraft Pose Estimation

ESA competition entry

Backbone	# bottleneck filters	# bins per Angle	Real ESA score	Synth. ESA score
Resnet-101	512	32	**0.1443** ⭑	0.0670 ⭑
	800	64	0.1630 ⭑	0.0604 ⭑
	Triple Ensemble		0.1555	**0.0571**

⭑ Public Leaderboard

~500M params

Other Teams	Real ESA score	Synth. ESA score
UniAdelaide	0.3752	**0.0095**
EPFL_cvlab	**0.1140**	0.0215
Top 10 average	1.3848	0.1515

Spacecraft Pose Estimation

Conclusions & Future work

Contribution:

- Effective data augmentation
- Alternative to Keypoint-based approach
- Probabilistic orientation estimation method to model orientation ambiguity
- Ablation study

Future work:

- More efficient architecture
- Improve URSO
- Address tracking

https://pedropro.github.io/project/urso/

P. Proença and Y. Gao, *Deep Learning for Spacecraft Pose Estimation from Photorealistic Rendering*, arXiv:1907.04298