

# MixRI: Mixing Features of Reference Images for Novel Object Pose Estimation

## Supplementary Material

### A. Appendix

In this section, we provide detailed information about our work. In Suppl. A.1, we describe the procedure for generating the ground truth correspondences. In Suppl. A.2, we outline the strategy for selecting reference images in our experiments. In Suppl. A.3, we present the design of MixRI+, the variant of MixRI. Next, we discuss some limitations and failure cases in Suppl. A.4 and detail the evaluation datasets in Suppl. A.5. We also provide qualitative results in Suppl. A.6. In Suppl. A.7 we discuss the balance between time and memory cost in different methods. Finally, we detail the training process in Suppl. A.8 and address ethical considerations in Suppl. A.9.

#### A.1. Ground Truth Correspondences

We train our network entirely with synthetic images, using the ground truth 3D information provided in GSO-Datasets [9]. To sample the 3D object points, we sample

the 2D image points in all reference images and unproject the 2D image points into the 3D space. For each reference image, we randomly select  $M$  image points within the object mask and find their corresponding points along all the other reference images with the query image using the ground truth pose and rendered depth. We also use depth to judge whether the corresponding location is occluded and set  $\tau$  to 4mm. In our default setting,  $M$  is 10, and  $S$  is 24, which produces  $N = 240$  points in total.

Because depth can have a sharp change in edge, we use the close operation in morphology to shrink the mask. We set the kernel size to  $3 \times 3$  and repeat 3 iterations. If the mask is really small and cannot sample enough points, we just repeat the sampled points.

We use the same procedures during the evaluation to produce the correspondences along all the reference images as training procedures, except for the query image.

We also use some augmentations during training, such

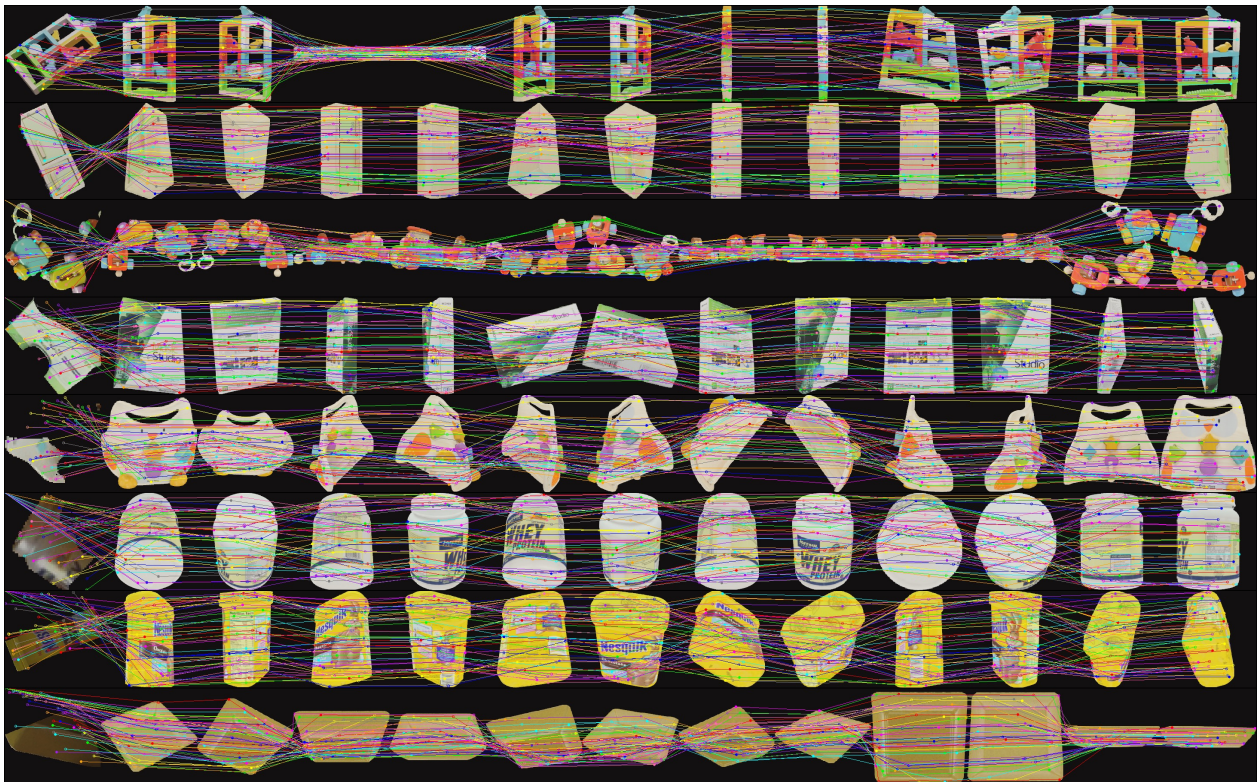


Figure 1. **Training samples.** Each row represents one batch. The left-most image is the query image and the remaining are 12 reference images. We show the correspondences across all reference images with the query image. Solid points represent visible points, while hollow points indicate occluded points. For the query image, some points are projected outside of the image. We set them to the location of  $[-1, -1]$  and regard them as occluded points during training.

as Gaussian blur, contrast, brightness, sharpness, and color change, as done in [10]. We provide some illustrations to show the training samples in Fig. 1.

## A.2. Reference Images Selection Strategy

In the default setting, our method uses only 24 reference images to have a balance of accuracy and performance. To demonstrate the generality of our method, all reference images are randomly selected from a reference image bank for each inference. In detail, to generate the reference image bank, we generate 162 reference images from viewpoints defined on a regular icosphere, which is created by subdividing each triangle of the Blender icosphere primitive into four smaller triangles, just as [10] does. We use the farthest sampling strategy (FPS) during each inference session to sample  $S$  (in most experiments,  $S = 24$ ) reference images. Because in-plane rotations do not provide additional information about the visibility of 3D object points, we only measure the out-of-plane distance between two rotations:

$$d(\hat{\mathbf{R}}_0, \hat{\mathbf{R}}_1) = \arccos\left(\frac{\text{tr}(\hat{\mathbf{R}}_0^\top \hat{\mathbf{R}}_1) - 1}{2}\right) / \pi, \quad (1)$$

where  $\hat{\mathbf{R}}_i$  means rotation removed the in-plane components from the origin rotation matrix  $\mathbf{R}_i$ . It is worth mentioning that for real-world scenarios, we can omit the process of establishing the reference image bank and directly generate only  $S$  reference images, pre-sampling the corresponding relationships between them in advance, as stated in Suppl. A.1.

## A.3. MixRI Variant

MixRI is designed for faster inference, a smaller network cache, and faster preparation of reference images. However, we also consider scenarios where more reference images are available, allowing for further accuracy improvements. To effectively select the most relevant reference images from a large reference image pool, we follow other approaches [10, 11] to identify the most suitable reference images. Here we use the same reference image bank as stated in Suppl. A.2 which has 162 reference images in total. We use the feature extractor from [10] as our feature extractor, as it also mitigates the impact of in-plane rotations, similar to our framework. Following [10], we identify the top-1 candidate with the most similar out-of-plane rotations and then select three additional reference images based on their rotation



Figure 2. **Failure Case.** The bowl is texture-less and nearly occluded, which make the matching really challenge. The top shows the matching results, where the point on the far left image is the predicted matched point, the second image from the left shows the ground truth matched point, and the remaining images are reference images, totaling 12. So if the lines between the first and second images on the left are parallel and of equal length, it means the match is correct. For ease of viewing, we randomly sampled 10 points predicted to be visible. The lower part of each result set displays the estimated results. The far-left image is the RGB image, the middle image shows the projection of the ground truth pose (in green) and the estimated pose (in red). The image on the far right displays the error heatmap calculated between the ground truth pose and the predicted pose which darker red indicates higher error with respect to the ground truth pose (legend: 0 cm to 5 cm).



alignment with the retrieved reference image. To effectively gather the closest viewpoint information, we train a MixRI variant which takes only four relevant reference images but keeps the model structure, model size the same. This variant is designed for situations where more reference images are available and the reference images have closer relationships to each other.

#### A.4. Limitations and Failure Cases

The limitations of our method stem from classic challenges in matching, such as matching weak textures and similar areas. Our method may fail when applied to highly texture-less objects. Although we try to handle such situations using attention mechanisms, our method can struggle to match between similar patches, especially when the detection result only captures a part of the whole object or object that is significantly occluded. As shown with the bowl in Figure 2, it is almost completely obscured and with very weak texture.

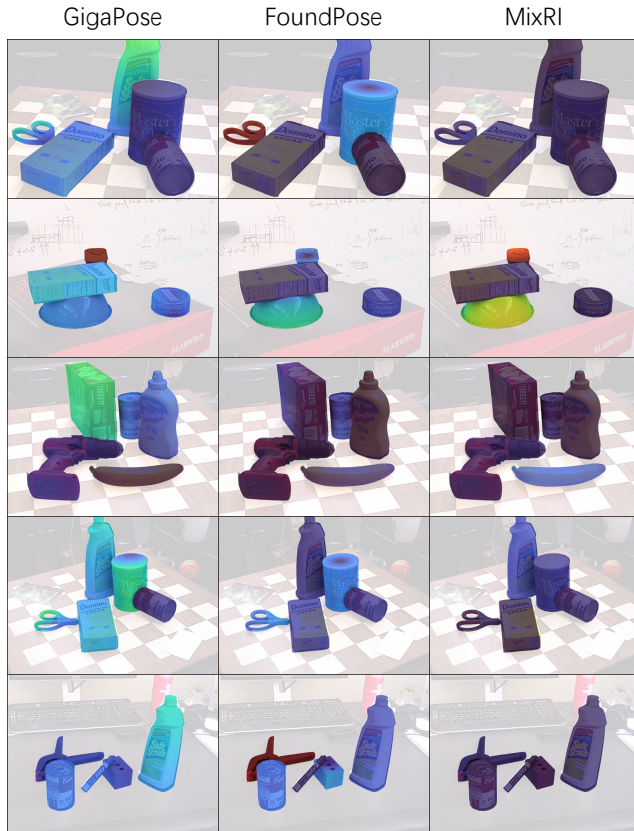



Figure 3. **Qualitative comparison with GigaPose [10] and FoundPose [11].** We compare our method with GigaPose and FoundPose, which are two leading methods in speed and accuracy. All the results are visualized in error heatmap calculated between the ground truth pose and the predicted pose which darker red indicates higher error with respect to the ground truth pose (legend: 0 cm  5 cm).

As the visualization shows, the points considered visible in the query image actually mostly lie in the occluded areas, leading to an incorrect pose estimation in the end.

#### A.5. Evaluation Datasets

We evaluate our method on seven core BOP datasets [7], including LM-O [1], YCB-V [12], T-LESS [5], TUD-L [6], IC-BIN [3], HomebrewedDB(HB) [8] and ITODD [4]. These datasets have 132 objects in total, which are never seen during the training stage. They include the general challenge of the RGB pose estimation, such as illumination change, texture-less object, strong occlusion, and cluttered scenes.

#### A.6. Qualitative Results

In Fig. 3, we present qualitative comparisons with GigaPose [10] and FoundPose [11], which are the two leading methods in speed and accuracy, respectively. Our method can make some mistakes especially in texture-less objects and some really small objects. However, we still achieve comparable results and can have better performance in some cases.

We illustrate more qualitative examples of our method in Fig. 4 for YCB-V [12] and in Fig. 5 for LM-O [1]. All illustrations are composed of the original RGB image, the corresponding matching results, the projection of the object model with ground truth pose and predicted pose, and the error heatmap.

#### A.7. Time & Memory Balance

In Table 1, we compare the feature extraction costs for a single object on an RTX 4090 GPU. Both GigaPose [10] and FoundPose [11] require significant time for feature extraction and pre-processing. On one hand, they rely on a large number of reference images, increasing the time required to extract features from all reference images. On the other hand, they include a view selection stage to identify the closest reference image, introducing additional pre-processing overhead, such as clustering. This time-consuming process follows a space-for-time strategy—pre-processing during the inference stage and caching them, thus leading to increased memory consumption. For memory-limited devices, this constraint reduces the number of objects whose poses can be estimated, as additional memory is required to store reference images, pre-extracted features, and large network parameters. In contrast, MixRIs are highly efficient. Thanks to the fewer reference images, we omit pre-extraction and instead perform feature extraction during inference, significantly reducing memory requirements with no cache needed. With fewer reference images and a lightweight network, MixRIs are more suitable for memory-constrained devices but still keep fast inference speed.

| Method | GigaPose [10] | FoundPose [11] | MixRI(12)    | MixRI(24) |
|--------|---------------|----------------|--------------|-----------|
| Time   | 11.6 s        | 40 s           | <b>21 ms</b> | 42 ms     |
| Memory | 233.5 M       | 523.5 M        | <b>9.2 M</b> | 18.4 M    |

Table 1. **The time and memory cost for processing all the reference images of one object.** MixRIs are much more efficient than GigaPose and FoundPose, making it more suitable for real-world applications.

### A.8. Training Details

We train our network from scratch. During training, we use the GSO-Datasets provided by MegaPose [9], which includes over 1 million images generated by BlenderProc [2]. We train our networks using PyTorch with the AdamW optimizer with  $\beta = (0.9, 0.999)$  and  $5e-4$  weight decay. The learning rate is set to  $1e-4$  and the warm up is 200 iterations. We train our network around 600k iterations, which spend about one week on four 4090 GPUs with batch size 8 on each. We use 12 reference images to train our network and  $N$  is set to 240. All images are cropped to  $224 \times 224$ . We also add some augmentations as done in [10] like: Gaussian blur, contrast, brightness, sharpness, and color change. The feature dimension  $D$  is set to 256. The number of iterations for each module  $n_0, n_1, n_2, n_3$  are set to 4, 2, 2, 2, respectively. Besides, for each module, the weights are shared during the iteration. We set  $\tau_{occ} = 0.8$ .

For the training losses, the predicted 2D coordinates are normalized to the range  $[-1, 1]$  for stable training. The Huber delta is set to 0.0357, corresponding to 4 pixels before normalization.

### A.9. Ethics

This research contributes to the development of object pose estimation, with potential applications in robotics, augmented reality (AR), and machine vision in general. While many of those applications could bring societal benefits (e.g. workload decrease through automation, AR-based teaching or assistance), it could also be used for unethical purposes.

## References

- [1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *Eur. Conf. Comput. Vis.*, pages 536–551, 2014. 3, 6
- [2] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Wendelin Knauer, Klaus H Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 4
- [3] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malasiotis, and Tae-Kyun Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3583–3592, 2016. 3
- [4] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing mvtec itodd-a dataset for 3d object recognition in industry. In *Int. Conf. Comput. Vis. Worksh.*, pages 2200–2208, 2017. 3
- [5] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888, 2017. 3
- [6] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Eur. Conf. Comput. Vis.*, pages 19–34, 2018. 3
- [7] Tomas Hodan, Martin Sundermeyer, Yann Labbe, Van Nguyen Nguyen, Gu Wang, Eric Brachmann, Bertram Drost, Vincent Lepetit, Carsten Rother, and Jiri Matas. Bop challenge 2023 on detection segmentation and pose estimation of seen and unseen rigid objects. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5610–5619, 2024. 3
- [8] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In *Int. Conf. Comput. Vis. Worksh.*, pages 2767–2776, 2019. 3
- [9] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *Conference on Robot Learning (CoRL)*, pages 715–725, 2022. 1, 4
- [10] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9903–9913, 2024. 2, 3, 4
- [11] Evin Pinar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. In *Eur. Conf. Comput. Vis.*, pages 163–182, 2024. 2, 3, 4
- [12] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, 2018. 3, 5



correspondences



results



correspondences




results



Figure 4. **Qualitative results on YCB-V [12].** We presented two sets of results, with the top of each set showing the matching results. In these, the point on the far left image is the predicted matched point, the second image from the left shows the ground truth matched point, and the remaining images are reference images, totaling 12. So if the lines between the first and second images on the left are parallel and of equal length, it means the match is correct. For ease of viewing, we randomly sampled 10 points predicted to be visible. The lower part of each result set displays the estimated results. The far-left image is an RGB image, the middle image shows the projection of the ground truth pose (in green) and the estimated pose (in red). The image on the far right displays the error heatmap calculated between the ground truth pose and the predicted pose which darker red indicates higher error with respect to the ground truth pose (legend: 0 cm 5 cm).



Figure 5. **Qualitative results on LM-O [1].** The top shows the matching results. In these, the point on the far left image is the predicted matched point, the second image from the left shows the ground truth matched point, and the remaining images are reference images, totaling 12. So if the lines between the first and second images on the left are parallel and of equal length, it means the match is correct. For ease of viewing, we randomly sampled 10 points predicted to be visible. The lower part of each result set displays the estimated results. The far-left image is an RGB image, the middle image shows the projection of the ground truth pose (in **green**) and the estimated pose (in **red**). The image on the far right displays the error heatmap calculated between the ground truth pose and the predicted pose which darker red indicates higher error with respect to the ground truth pose (legend: 0 cm  5 cm).