Moon, S., Son, H., Hur, D., & Kim, S. (2025). Co-op: Correspondence-based novel object pose estimation. In CVPR 2025.

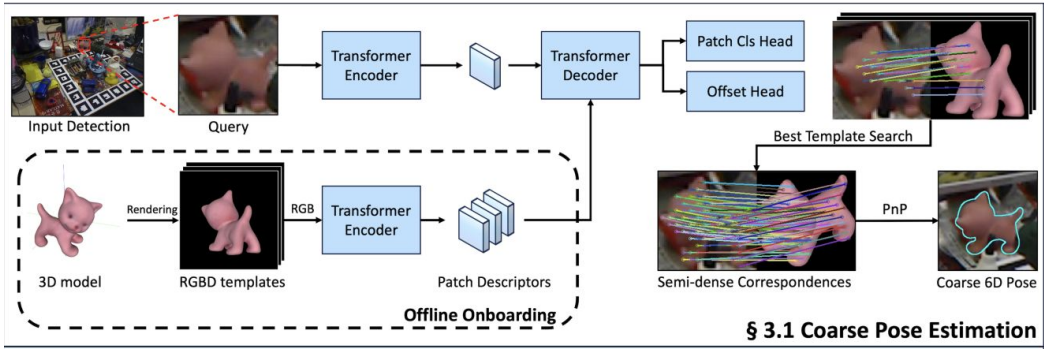
Detailed Methodology Summary - Overall Architecture

The Co-op framework follows a two-stage pipeline designed for 6DoF pose estimation of unseen objects from a single RGB image.

- Coarse Pose Estimation
  - Generates initial 6DoF pose estimates based on a limited number of pre-rendered templates of the desired object.
- Pose Refinement
  - Improves upon initial estimate via a render-and-compare process that iteratively aligns the rendered model with the provided input image.

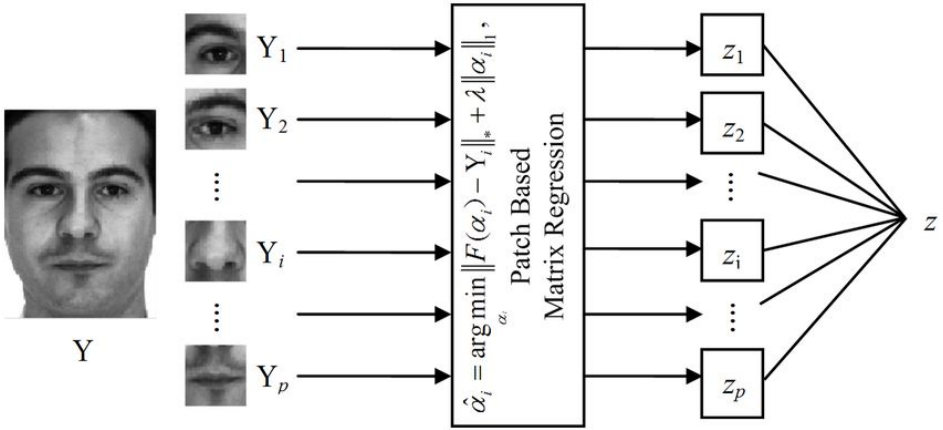
Detailed Methodology Summary - Coarse Pose Estimation

- Aims to obtain a quick, accurate initial pose by estimating semi-dense correspondences between the provided image and a small set of templates.
- Templates are generated from 42 uniformly distributed viewpoints that cover only out-of-plane rotations.
- Each generated template provides RGB and depth information so that the program can use correspondence-based mapping between the 2D image and the 3D templates.



Detailed Methodology Summary - Coarse Pose Estimation

- Main innovation of this stage is the hybrid representation for correspondence estimation.
- Instead of traditional methods, such as direct regression of coordinates, Co-op combines patch-level classification and offset regression to evaluate correspondence between the provided image and the generated templates.

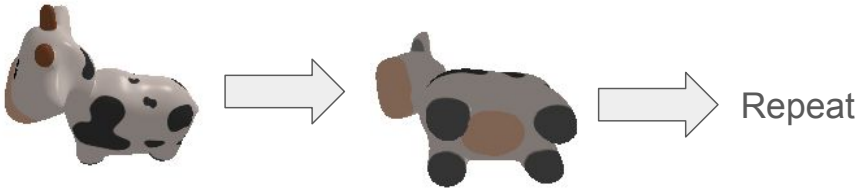


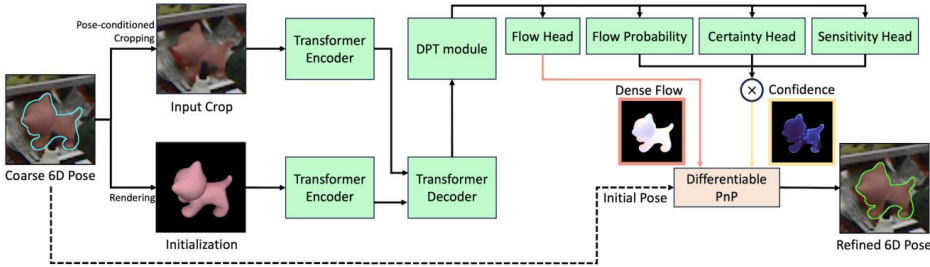
Detailed Methodology Summary - Coarse Pose Estimation

- After finding these correspondences, Co-op applies a Perspective-n-Point (PnP) solver to recover the initial 6DoF pose, using EPnP with RANSAC to ensure resistance to outliers and occlusions.
- Each template is evaluated based on the classification confidence produced earlier, and the template with the highest value is selected for pose estimation.

Detailed Methodology Summary - Pose Refinement

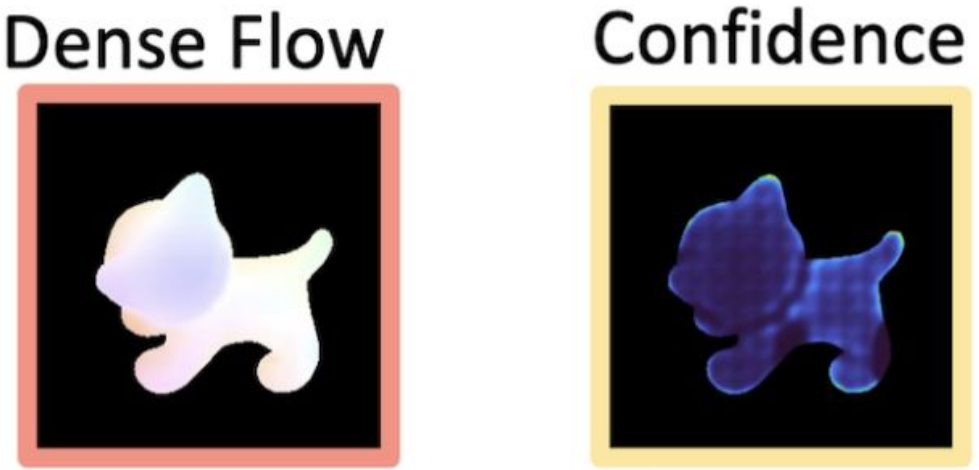
- Use a render-and-compare strategy with probabilistic dense correspondences (article refers to it as ‘flow’).
- Works at a pixel level to correct any misalignments from the coarse stage.
- Renders the object using its CAD model, and then predicts the dense flow field that aligns the rendered image with the query image.





Detailed Methodology Summary - Pose Refinement

- Produces a flow confidence-weight map based on some learning factors, which able to down-weight (ignore) unreliable correspondences, leading to more accurate updates to the pose.

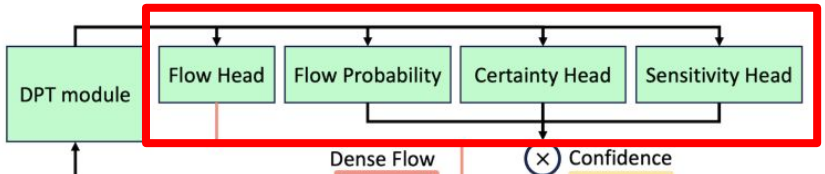


Detailed Methodology Summary - Pose Refinement

- Main innovation of this stage is the introduction of Probabilistic Flow Regression, which models the correspondence field as a Laplace-distributed conditional probability rather than a single, deterministic flow.
- The model predicts the mean flow vector (μ) and an uncertainty scale (b), which represents the flow as a probabilistic distribution.
- These values allow the program to learn where the flow should move pixels, as well as how confident it is for each correspondence it generates.

Detailed Methodology Summary - Pose Refinement

- Co-op computes a flow confidence map (W) by combining three learned factors from this process:
  - Certainty: estimating whether a pixel in the rendered image is visible/occluded;
  - Sensitivity: identifying regions with discriminative visual texture; and
  - Flow Probability (PR): likelihood that the predicted flow lies within a small radius of the true flow.
- The final pose update is done with a differentiable PnP solver (based on the Levenberg-Marquardt algorithm), which uses the confidence map as pixel-wise weights.



Experiments and Results - Setup

- Experiments were performed on the seven core datasets of the BOP challenge.
- Co-op was tested without any retraining, using only the models and test images provided in the challenge.
- Performance was measured using Average Recall (AR), which is the average of three metrics defined in the BOP challenge:
  - Visible Surface Discrepancy (VSD),
  - Maximum Symmetry-Aware Surface Distance (MSSD),
  - and Maximum Symmetry-Aware Projection Distance (MSPD).

Experiments and Results - Quantitative Results

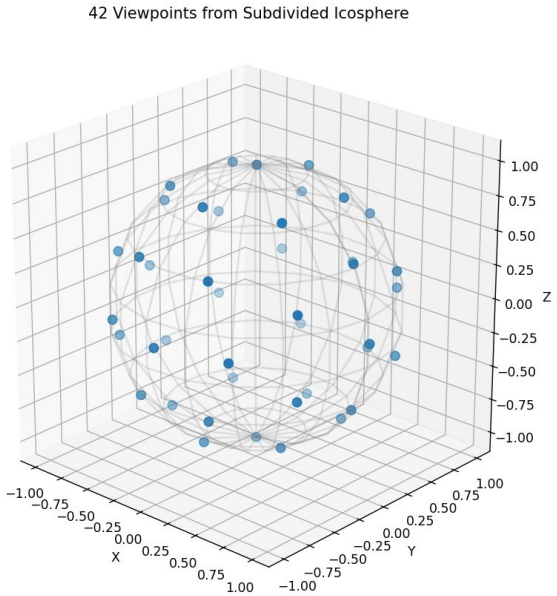
- When evaluating the model using only the coarse estimation stage, Co-op achieved a mean AR of 58.4.
  - Outperformed the next-best method (FoundPose, with a 37.3 AR) by approximately 56.6%.
- When combining the coarse and refinement stages under a single-hypothesis setting, Co-op achieved a mean AR of 64.0.
  - Outperformed all previous single-hypothesis pipelines, including GenFlow and MegaPose.
- Using a five-hypothesis strategy, where multiple coarse poses are refined, and the best result is selected, Co-op further improved to a mean AR of 65.7.
  - This is an improvement of more than 6 points over the previous best method (FoundPose, 59.6 AR).
- Despite the increase in accuracy, Co-op still only required around 4.2 seconds per image on average.

Experiments and Results - Ablation Studies

- When the patch-level classification module was removed and only direct offset regression was used, the model scored much lower, from 58.4 AR to 50.2 AR.
- Replacing Co-op’s probabilistic flow regression with a standard deterministic flow head (like that used in GenFlow) also reduced scores, from 64.0 AR to 63.3 AR.
- Initializing the model from randomized weights rather than the CroCo pre-trained parameters caused the model to score significantly lower, from 58.4 AR to 46.4 AR for coarse estimation and from 64.0 AR to 61.2 AR for refinement.

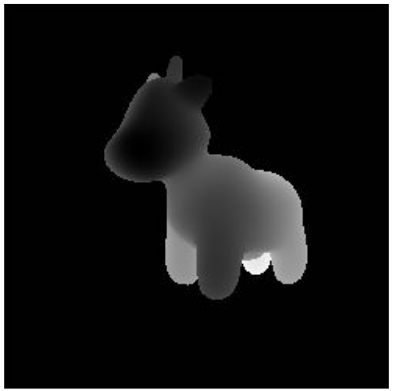
Hands-On Findings & Replication

- Co-op has not published its source code or the models used, so replicating their findings is not possible.
- However, we were able to implement the initial template generation as well as a non-functional skeleton implementation of the pose-refinement.



Hands-On Findings & Replication

- While 42 points may seem like an inadequate number, in our testing, most random viewpoints we chose were quite similar to at least one template.
- This leads us to believe that the number of iterations required by the pose refinement algorithm would generally be small.



End