Foundational feature fusion for conditional flow matching in 6D pose estimation

Amir Hamza[1,2]
ahamza@fbk.eu
Davide Boscaini[1]
dboscaini@fbk.eu
Fabio Poiesi[1]
poiesi@fbk.eu

[1] Fondazione Bruno Kessler
Trento, Italy
[2] University of Trento
Trento, Italy

Abstract

Conditional flow matching has enabled a step forward in object 6D pose estimation, achieving state-of-the-art performance by progressively denoising and registering object representations to observed scenes. Existing methods require training task-specific encoders supervised on object-scene overlap and rely on trivial feature fusion strategies to resolve pose ambiguities. We present FunFlow6D, a novel flow matching-based formulation that leverages features from geometric and appearance foundation models for pose estimation, eliminating the need for task-specific encoder training. We also introduce a cross attention-based fusion mechanism that dynamically combines geometric and appearance features to provide richer conditioning for the flow matching module. Experiments on four datasets from the BOP benchmark show that FunFlow6D outperforms the previous state of the art while reducing supervision requirements and memory overhead. Extensive ablations validate the contribution of each proposed component. Project website: https://tev-fbk.github.io/FunFlow6D/.

1 Introduction

Manipulating and interacting with objects in the physical world requires accurate geometric localization. Estimating an object's 6D pose (its position and orientation in the 3D environment) is key for spatial applications like augmented reality [40], robotic grasping [24], and automated assembly [50]. These tasks demand accuracy under clutter [9, 47] and occlusions [4, 30]. In this work, we focus on *model-based, instance-level 6D pose estimation* from RGBD data [19, 23, 26]. This setting assumes a prior 3D model of the object (CAD or reconstructed mesh) and explicit supervision over a closed set of predefined objects. Traditional approaches [12, 26, 41, 45] cast this as a direct regression problem. Recent work [11], instead, formulates pose estimation as iterative denoising via conditional flow matching (CFM) [18, 42], which needs fewer integration steps than diffusion [25], trains more stably, and follows deterministic trajectories [22]. The flow model learns a displacement field that transports Gaussian noise onto the object's 3D model, making CFM-based pose estimation inherently a registration problem. The flow routes each noisy scene point toward a target point

Figure 1: FunFlow6D estimates object 6D poses from input data (image + 3D model) via flow matching over multimodal foundational features (left). It advances over previous state-of-the-art on two fronts. First, it removes the need to train a task-specific encoder with object-scene overlap supervision (right, pink). Second, instead of relying on predefined fusion schemes such as point-wise summation, it learns how to optimally fuse multimodal features from frozen foundation models during flow matching (right, azure). As a result, FunFlow6D achieves higher accuracy in a single training phase while requiring less supervision.

on the CAD surface. This routing is driven by per-point conditioning features, which enable point-to-point matching and must satisfy three properties: (1) *Geometric distinctiveness*: points on different surface neighborhoods must produce different features, otherwise the flow cannot route them apart. (2) *Local structure preservation*: nearby points must remain distinguishable even in a reduced-dimensional space, for tractability. (3) *Adaptive fusion*: conditioning must account for the fact that geometric and semantic cues are informative at different points.

State-of-the-art methods [11, 42] lack these properties. They condition the flow with an overlap-aware encoder [42, 46] trained to predict per-point visibility between the object's 3D model and the observed scene. However, overlap may fail to encode local 3D structures. We find that correspondences derived from overlap features yield only an average of 3.5 inliers on LM-O [3], indicating that the vast majority of the geometric cues used to condition the flow matching stage are unreliable. The overlap encoder also requires retraining for each new object, tying conditioning quality to per-instance supervision. In contrast, handcrafted [37, 38] and learned [7, 8, 10, 35] descriptors trained at scale are explicitly designed to be discriminative of local 3D structure and to generalize across shapes. These descriptors [10, 35] yield substantially higher inlier ratios than overlap-based features. A 3D descriptor is therefore a more natural fit for the conditioning signal. A separate limitation concerns feature dimensionality: both supervised [11] and unsupervised [5, 6, 32] methods reduce DINOv2 [31] features with PCA [27] to match the dimensionality of the geometric branch. Being linear and unsupervised, PCA preserves the directions of highest global variance across the dataset, but this criterion is not necessarily aligned with what makes two nearby points distinguishable from one another. As a result, directions that carry little variance overall but are locally discriminative, exactly the kind of fine-grained structure per-point conditioning depends on, can be discarded, creating an information bottleneck that limits how well nearby points can be told apart after reduction. Finally, Flose [11] combines appearance and overlap features via point-level summation, which treats semantics and geometry as equally informative at every point. However, the informative modality varies across the object: textured regions rely on semantics, while geometrically distinct regions rely on shape.

We propose FunFlow6D (foundational feature fusion for conditional flow matching in 6D pose estimation), a CFM-based method that meets all three conditioning requirements. For geometric distinctiveness, we condition the flow on frozen dGeDi [10] features that are aware of local 3D structure and generalize well to unseen objects. This raises the average number of

inliers from 3.5 to 7.0 on LM-O (a $\sim 2\times$ increase), without requiring any per-instance training. For local structure preservation, we project frozen DINOv2 features with UMAP [28] instead of PCA. UMAP is non-linear and explicitly preserves local neighborhoods, retaining task-relevant information that PCA may discard. This raises the average number of inliers from 3.2 to 25.6 on LM-O (an $\sim 8\times$ increase), at the same target dimensionality. For adaptive fusion, we combine the two feature modalities with a learnable cross-attention gating module [36]. The module attends across modalities and learns per-point gates that determine how much each modality contributes. This adapts the conditioning to the local structure of each object and enables richer cross-modal interaction than the point-wise summation used by Flose [11]. We evaluate FunFlow6D on four datasets from the BOP benchmark, covering diverse object categories, occlusion levels, textures, and clutter conditions. FunFlow6D outperforms state-of-the-art methods while halving the number of trained parameters and removing the per-instance pre-training requirement.

In summary, our contributions are:

- We condition flow matching for 6D pose estimation on robust correspondence signals extracted from a frozen, zero-shot geometric encoder, providing more discriminative conditioning features without requiring per-instance fine-tuning.
- We leverage UMAP to reduce the dimensionality of semantic features while preserving their local neighborhood structure, retaining task-relevant information that PCA's linear projection would discard.
- We introduce a gated-attention feature fusion module that adapts the flow matching conditioning to local object structure and promotes richer interaction between semantic and geometric cues.

2 Related work

Instance-level 6D pose estimation methods leverage explicit supervision on a closed set of known objects to estimate their 6D poses. *Model-based* approaches split into *indirect* methods, which establish dense correspondences before solving for the pose, and *direct* methods, which regress the transformation end-to-end. *Indirect methods* cast pose estimation as dense coordinate or feature regression [3, 34], increasingly enriched by learned surface representations: SurfEmb [12] learns continuous surface embeddings, ZebraPose [41] introduces a coarse-to-fine surface encoding, and HccePose(BF) [45] extends this encoding to back-facing geometry for ultra-dense 2D–3D correspondences. Complementary refinement [17, 20, 23] and hybrid [19] pipelines improve robustness but remain bound by correspondence quality and degrade under occlusion and texture loss. *Direct methods* regress poses end-to-end on the $SE(3)$ manifold. GDR-Net [44] and GDRNPP [26] couple geometric supervision with differentiable PnP, achieving state-of-the-art results on the BOP benchmark [16]. *Generative formulations*, instead, model the full pose posterior, leveraging advances in denoising diffusion [14] and flow matching [22]. They split into two families: those that learn a flow directly on the $SE(3)$ manifold, and those that transport points in $\mathbb{R}^3$ and recover the rigid transformation in closed form from the resulting correspondences. SE(3)-PoseFlow [18] follows the former one, learning a flow on $SE(3)$ for uncertainty-aware manipulation. Rectified Point Flow [42], GARF [21], and Register Any Point [33] belong to the latter, applying flow matching for registration and assembly in $\mathbb{R}^3$. Closest to our work, Flose [11] casts instance-level 6D pose estimation as conditional flow matching that transports Gaussian

noise onto the object's CAD surface, bypassing correspondence estimation and improving robustness to clutter and occlusion. However, Flose conditions the flow on an overlap-aware encoder that requires retraining for every new object, reduces DINOv2 features with PCA, and fuses geometric and semantic cues by point-wise summation. In contrast, FunFlow6D conditions flow matching on a frozen zero-shot 3D local descriptor to enable unseen-object generalization, introduces the use of UMAP [28] to preserve local neighborhood structure, and combines appearance and geometric cues through a learned cross-attention gating module.

Per-point descriptors for 6D pose estimation. Correspondence-based and generative pose estimation methods recover the rigid transformation by matching or transporting the scene points to model points, respectively. Therefore, per-point descriptors must be discriminative over local surface neighborhoods and generalize to unseen objects without retraining. Foundation models trained at scale provide this signal, and divide into *appearance* and *geometric* encoders. Appearance encoders are trained on large-scale RGB data with self-supervised objectives. DINOv2 [31] outputs semantically-rich features, robust to illumination and viewpoint, and transferable to dense prediction without fine-tuning. It serves as a frozen backbone in both RGB [32] and RGBD [5, 6] pose estimation pipelines. Appearance features disambiguate textured regions but are uninformative about 3D structure: geometrically distinct points with similar texture share near-identical descriptors. Geometric encoders are trained on 3D data to encode local surface structure. GeDi [35] learns generic and distinctive 3D descriptors with a PointNet++ backbone, achieving rotation invariance through local reference frames and generalizing across indoor, outdoor, and object-level scans. dGeDi [10] distills GeDi into a PTV3 [46] backbone that retains its discriminability by achieving higher inlier ratio than GeDi, while making inference faster. The two modalities are complementary. FunFlow6D conditions flow matching jointly on DINOv2 [31] and dGeDi [10] features, and removes the training phase and overlap supervision required by the overlap-aware encoder.

Feature fusion for 6D pose estimation combines individual feature modalities into a unified representation. Existing approaches can be divided into *training-free* and *training-based* methods. *Training-free fusion* combines the two modalities using closed-form operations. FreeZe [5, 6] concatenates semantic and geometric features along the channel dimension for training-free pose estimation. Flose [11] uses point-wise summation to preserve the feature dimensionality required by the pre-trained backbone [42]. Both fusion schemes are non-parameterized and treat the two modalities as equally informative at each point, with no mechanism to suppress unreliable cues or adapt the contribution of each modality independently. *Training-based fusion* relaxes these constraints by learning the optimal fusion strategy from data. Standard cross-attention [43] aggregates information across modalities but lacks an explicit per-point suppression mechanism. Gated attention variants [36] address this limitation by applying a head-specific sigmoid gate to the attention output, introducing input-dependent sparsity that selectively preserves or suppresses features on a per-point basis, with demonstrated gains in expressiveness and training stability. FunFlow6D fuses appearance- and geometry-aware features through a learned cross-attention gating module: cross-attention establishes per-point interactions between the two modalities, while a sigmoid gate determines how much each cue contributes at every point. Unlike concatenation or summation, our fusion strategy adapts to local object structure; unlike plain cross-attention, it can suppress unreliable modalities on a per-point basis.

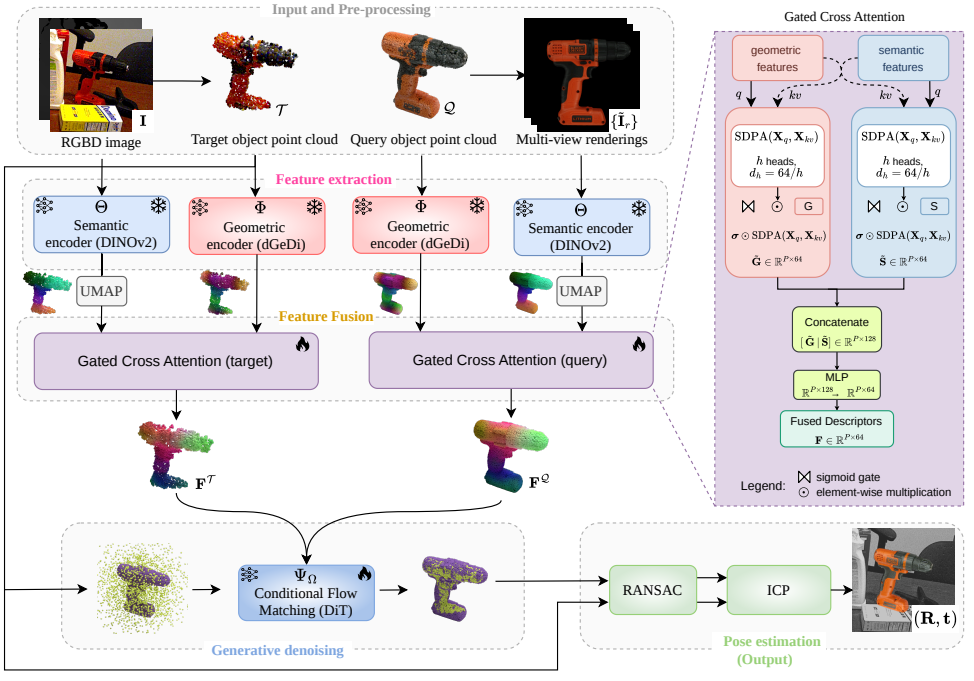


Figure 2: Overview of FunFlow6D. Given the canonical query point cloud $\mathcal{Q}$ and an RGBD image $\mathbf{I}$ as input (top left), FunFlow6D recovers the 6D object pose $(\mathbf{R},\mathbf{t})$ (bottom right) through three stages: feature extraction, generative denoising, and pose estimation. Throughout, $\mathcal{Q}$ acts as a *static anchor* and the flow operates only on the target $\mathcal{T}$, lifted from the masked depth of $\mathbf{I}$. *Feature extraction:* A frozen geometric encoder Φ (dGeDi) and a frozen semantic encoder Θ (DINOv2, reduced via UMAP) produce per-point descriptors that are fused by a bidirectional gated cross-attention module (right inset) into $\mathbf{F}^{\mathcal{Q}},\mathbf{F}^{\mathcal{T}}$. Colors encode feature similarity: corresponding regions across $\mathcal{Q}$ and $\mathcal{T}$ share similar colors. *Generative denoising:* A conditional flow-matching network Ψ_Ω (DiT), conditioned on $\mathbf{F}^{\mathcal{Q}},\mathbf{F}^{\mathcal{T}}$, learns a velocity field that canonicalizes $\mathcal{T}$ into $\hat{\mathcal{T}}$ while the anchor velocities are zeroed. *Pose estimation:* The flow-induced correspondences between $\mathcal{T}$ and $\hat{\mathcal{T}}$ yield the camera-to-canonical transform via a RANSAC-based orthogonal Procrustes solver followed by ICP, and the pose $(\mathbf{R},\mathbf{t})$. ❄ denote frozen modules, while 🔥 indicate trainable ones.

3 Method

3.1 Problem formulation and Overview

Problem formulation. Let $\mathbf{I} \in \mathbb{R}^{H\times W\times 4}$ denote an RGBD observation of a given scene and $\mathcal{Q} \in \mathbb{R}^{N\times 3}$ the dense point cloud of a known *query* object in its canonical reference frame. Our primary objective is to recover the rigid transformation $\mathbf{T} \in \mathrm{SE}(3)$, parameterized by a rotation $\mathbf{R} \in \mathrm{SO}(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$, that maps $\mathcal{Q}$ from its canonical frame into the camera coordinate system. To isolate the object of interest within the cluttered scene, we assume the availability of a binary segmentation mask $\mathbf{M} \in \{0,1\}^{H\times W}$. By applying $\mathbf{M}$ to the depth channel of $\mathbf{I}$ and back projecting the valid pixels into 3D space using the camera intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3\times 3}$, we obtain the *target* point cloud $\mathcal{T} \in \mathbb{R}^{M\times 3}$, expressed in the

camera reference frame. Target point cloud is a partial, occluded, and noisy observation of the query object. Formally, we seek $(\mathbf{R},\mathbf{t})$ such that $\mathbf{R}\mathcal{Q}+\mathbf{t}\approx\mathcal{T}$. To enable supervised alignment in canonical space, we further define $\mathcal{T}^r=\mathbf{R}^\top(\mathcal{T}-\mathbf{t})\in\mathbb{R}^{M\times 3}$ as the target point cloud transported back into the canonical frame of $\mathcal{Q}$ using the ground-truth pose.

Overview. We propose a three-stage pipeline driven by CFM shown in Fig. 2. Here $\mathcal{Q}$ acts as a *static anchor* and is never transformed, the flow operates only on $\mathcal{T}$. First, we process both $\mathcal{Q}$ and $\mathcal{T}$ to extract discriminative point-wise descriptors. Appearance-aware semantic descriptors are obtained from frozen vision foundation model (VFM) and reduced via UMAP to preserve local neighborhood structures, while zero-shot geometric encoders provide complementary structural descriptors. The two modalities are adaptively combined through a gated cross-attention network, providing fused representations $\mathbf{F}^{\mathcal{Q}}$ and $\mathbf{F}^{\mathcal{T}}$. Second, conditioned on $(\mathbf{F}^{\mathcal{Q}},\mathbf{F}^{\mathcal{T}})$, a flow-matching model deforms $\mathcal{T}$ into a canonicalized estimate $\hat{\mathcal{T}}$, predicting where each observed point lies in the canonical frame of $\mathcal{Q}$. Third, the flow induces point-wise correspondence between $\mathcal{T}$ and $\hat{\mathcal{T}}$ from which we recover $\mathbf{T}'=(\mathbf{R}',\mathbf{t}')\in\mathrm{SE}(3)$ with $\mathbf{R}'\mathcal{T}+\mathbf{t}'\approx\hat{\mathcal{T}}$ via orthogonal Procrustes. As $\mathbf{T}'$ is the camera-to-canonical mapping, the desired pose follows by inversion i.e. $\mathbf{T}=\mathbf{T}'^{-1}$.

3.2 Feature extraction

We extract point-wise descriptors by jointly processing geometric features $\mathbf{G}$ and semantic features $\mathbf{S}$ for both the query $\mathcal{Q}$ and the target $\mathcal{T}$, before fusing them.

The geometric encoder Φ is a frozen, distilled multi-scale 3D local descriptor network [10] that, given a point cloud, produces a D-dimensional local geometric descriptor per point. It is distilled to encode information from two complementary spatial extents local neighborhoods covering 30% and 40% of the object's diameter in a single forward pass. Applying Φ to $\mathcal{Q}$ and $\mathcal{T}$ yields multi-scale geometric features $\mathbf{G}^{\mathcal{Q}}\in\mathbb{R}^{N\times D}$ and $\mathbf{G}^{\mathcal{T}}\in\mathbb{R}^{M\times D}$. Operating at two scales provides Φ with both fine-grained surface cues and broader contextual structure, yielding robustness to partiality and local noise in $\mathcal{T}$.

Geometric reasoning alone is insufficient to disambiguate objects exhibiting symmetries or sparse geometric detail. We therefore complement Φ with a semantic encoder Θ that lifts pixel-level features from a frozen VFM onto the 3D points of $\mathcal{Q}$ and $\mathcal{T}$. For the target, Θ is applied to a square crop of $\mathbf{I}$ centered on the barycenter of $\mathcal{T}$, and the resulting pixel features are transferred to the points of $\mathcal{T}$ through the pixel-to-point correspondences induced by the depth channel, producing $\mathbf{S}^{\mathcal{T}}\in\mathbb{R}^{M\times G}$, where G denotes the VFM feature dimensionality. For the query, since no RGB image is available at inference, we synthesize multi-view renderings $\{\widetilde{\mathbf{I}}_r\}_{r=1}^{R}$ of its textured 3D model, apply Θ to each rendering, and back-project the resulting pixel features onto the points of $\mathcal{Q}$ using the renderer's camera intrinsics; whenever multiple pixels project onto the same 3D point, their features are aggregated by mean pooling across views. This rendering pass is performed once per object as an offline pre-processing step, so that at inference time the query features $\mathbf{S}^{\mathcal{Q}}\in\mathbb{R}^{N\times G}$ are loaded from memory.

The dimensionality of $\mathbf{S}^{\mathcal{Q}}$ and $\mathbf{S}^{\mathcal{T}}$ is reduced from G to D via UMAP, which preserves the local manifold structure of the VFM embedding space more faithfully than linear projection based techniques. Crucially, the UMAP embedding is fit only on $\mathbf{S}^{\mathcal{Q}}$ and subsequently applied to project every instance of $\mathbf{S}^{\mathcal{T}}$, ensuring that target and query semantic descriptors live on the same low-dimensional manifold.

3.3 Feature fusion via gated cross-attention

Rather than merging the two streams by point-wise addition, we let them interact through a bidirectional gated cross-attention block that adaptively weights, at every point, how much information each modality contributes. The same module is applied independently to $\mathcal{Q}$ and $\mathcal{T}$; for simplicity we describe it for a generic object with P points ($P{=}N$ for $\mathcal{Q}$, $P{=}M$ for $\mathcal{T}$). Let $\mathbf{G},\mathbf{S} \in \mathbb{R}^{P\times D}$ denote its L_2-normalized geometric and semantic features.

The block contains two symmetric branches that exchange information across modalities. Each branch operates on an ordered pair of streams $(\mathbf{X}_q,\mathbf{X}_{kv})$. The first uses $(\mathbf{X}_q,\mathbf{X}_{kv}) = (\mathbf{G},\mathbf{S})$, while the second swaps the roles to $(\mathbf{S},\mathbf{G})$. Within each branch, queries are linearly projected from $\mathbf{X}_q$, while keys and values are projected from $\mathbf{X}_{kv}$, and multi-head scaled dot-product attention [43] is computed over all P points with h heads of dimension $d_h = D/h$. Following [36], the SDPA output is modulated by a head-specific, element-wise sigmoid gate predicted from the query input:

$$\sigma = \sigma(\mathbf{X}_q\mathbf{W}_g) \in \mathbb{R}^{P\times D}, \quad \mathrm{GAtt}(\mathbf{X}_q,\mathbf{X}_{kv}) = \sigma \odot \mathrm{SDPA}(\mathbf{X}_q,\mathbf{X}_{kv}), \tag{1}$$

where $\mathbf{W}_g \in \mathbb{R}^{D\times D}$ holds the learnable gate parameters, $\sigma(\cdot)$ is the element-wise sigmoid, and $\odot$ the element wise product. The gated output is passed through a linear projection $\mathbf{W}_O$, dropout, and a LayerNorm with a residual from $\mathbf{X}_q$, yielding the refined streams $\tilde{\mathbf{G}},\tilde{\mathbf{S}} \in \mathbb{R}^{P\times D}$ from the two branches respectively. The biases of $\mathbf{W}_g$ are zero-initialized so that σ starts at 0.5 everywhere, letting both modalities contribute on equal footing at the beginning of training; the optimizer is then free to learn per-point, per-channel modality weights.

The two refined streams are concatenated along the channel axis and passed through a two-layer MLP with GELU activation that projects them back to D dimensions, yielding the fused descriptor

$$\mathbf{F} = \mathrm{norm}\big(\mathrm{LN}\big(\mathrm{MLP}([\tilde{\mathbf{G}}\,|\,\tilde{\mathbf{S}}])\big)\big) \in \mathbb{R}^{P\times D}, \tag{2}$$

where $\mathrm{LN}(\cdot)$ denotes LayerNorm, $\mathrm{norm}(\cdot)$ denotes L_2 normalization along the channel axis, and $[\cdot\,|\,\cdot]$ denotes concatenation. Applying the module to $\mathcal{Q}$ and $\mathcal{T}$ yields the per-point fused descriptors $\mathbf{F}^{\mathcal{Q}} \in \mathbb{R}^{N\times D}$ and $\mathbf{F}^{\mathcal{T}} \in \mathbb{R}^{M\times D}$, which condition the flow-matching model.

3.4 Conditional flow matching

Conditional flow matching learns a time dependent velocity field that transports samples from a simple noise distribution to a target data distribution along straight trajectories. In our setup, we sample a noise state $\mathbf{X}(0) \in \mathbb{R}^{2N\times 3}$, where each of the $2N$ points is drawn independently from a standard Gaussian $\mathcal{N}(0,I_3)$, and define the clean state as the concatenation of the anchor with the canonicalized target, $\mathbf{X}(1) = [\,\mathcal{Q};\mathcal{T}^r\,] \in \mathbb{R}^{2N\times 3}$, where $[\,\cdot\,;\,\cdot\,]$ denotes concatenation along the point axis. The two states are linearly interpolated as

$$\mathbf{X}(t) = (1-t)\,\mathbf{X}(0) + t\,\mathbf{X}(1), \qquad t \in [0,1], \tag{3}$$

which yields a constant velocity $\mathbf{X}(1)-\mathbf{X}(0)$ along the trajectory. A flow network Ψ_Ω is trained to regress this velocity conditioned on a signal $\mathbf{C}$,

$$\Psi_\Omega\big(t,\mathbf{X}(t)\,\big|\,\mathbf{C}\big) \;\approx\; \mathbf{X}(1)-\mathbf{X}(0), \tag{4}$$

by minimizing a mean square error function [42], with t sampled uniformly from $[0,1]$ and the pair $(\mathbf{X}(0),\mathbf{X}(1))$ drawn independently at each training step.

Conditioning signal. The flow is conditioned on the fused appearance and geometry descriptors produced by the gated cross-attention module, augmented with a positional encoding,

$$\mathbf{C} = [\mathbf{F} \mid \mathbf{P}] \in \mathbb{R}^{2N\times(D+d_P)}, \tag{5}$$

where $\mathbf{F} = [\mathbf{F}^{\mathcal{Q}}; \mathbf{F}^{\mathcal{T}}]$ stacks the per-cloud fused descriptors along the point axis, $\mathbf{P} \in \mathbb{R}^{2N\times d_P}$ is produced by a positional encoder Ξ, and $[\cdot\,|\,\cdot]$ denotes concatenation along the channel axis. For each point, Ξ uses a 10-dimensional input composed of its 3D coordinates, its surface normal, the coordinates of its counterpart in the noise sample $\mathbf{X}(0)$, and a binary flag distinguishing $\mathcal{Q}$ from $\mathcal{T}$. Sinusoidal embeddings [29] are applied to each geometric attribute independently, concatenated with learned point-wise features, and linearly projected to dimension d_P.

Inference. At test time we recover the predicted position of points by integrating the learned velocity field forward through K uniform Euler steps of size $\Delta t = 1/K$,

$$\hat{\mathbf{X}}(t+\Delta t) = \hat{\mathbf{X}}(t) + \Psi_\Omega\big(t, \hat{\mathbf{X}}(t) \,\big|\, \mathbf{C}\big)\,\Delta t, \tag{6}$$

starting from a fresh Gaussian sample $\hat{\mathbf{X}}(0)$, where each point is again drawn independently from $\mathcal{N}(0, I_3)$. Since $\mathcal{Q}$ acts as a static anchor, the velocity components corresponding to its points are zeroed out at every step, so that only the target portion of the state evolves. The target sub-block of $\hat{\mathbf{X}}(1)$ is the canonicalized estimate $\hat{\mathcal{T}} \approx \mathcal{T}^r$.

3.5 6D pose estimation

The flow induces an explicit one-to-one correspondence between each point of the target $\mathcal{T}$, expressed in the camera frame, and its canonicalized counterpart in $\hat{\mathcal{T}}$. We exploit this correspondence to estimate the camera-to-canonical rigid transform $\mathbf{T}' = (\mathbf{R}', \mathbf{t}') \in \mathrm{SE}(3)$ such that $\mathbf{R}'\mathcal{T} + \mathbf{t}' \approx \hat{\mathcal{T}}$. We obtain a coarse estimate $\mathbf{T}'_c$ by solving orthogonal Procrustes. At each iteration a small subset of correspondences is sampled, a candidate rigid transform is computed in closed form, and the hypothesis is scored by the number of inliers among the remaining pairs. This rejects outliers induced by imperfect flow predictions, particularly in regions affected by heavy occlusion or symmetry ambiguity. The coarse estimate is then refined through ICP aligning the transformed observation $\mathbf{R}'_c\mathcal{T} + \mathbf{t}'_c$ to the anchor $\mathcal{Q}$. The refinement proceeds in a coarse-to-fine manner over multiple decreasing distance thresholds. The final iteration produces the refined transform $\mathbf{T}'$, and the desired object pose follows by inversion $\mathbf{T} = \mathbf{T}'^{-1}$.

4 Results

4.1 Experimental validation

Implementation details. We instantiate Φ with dGeDi [10] and Θ with DINOv2 [31], and set the common descriptor dimensionality to $D = 64$, which matches the native output feature dimensionality of Φ; the DINOv2 features are reduced to the same dimensionality by UMAP. The gated cross-attention block uses $h = 4$ heads, hence $d_h = 16$, with dropout 0.1. We fit UMAP with 30 neighbors, a minimum distance of 0.01, and cosine similarity as metric.

Data. We validate FunFlow6D on four datasets from the BOP benchmark [3, 9, 15, 47], spanning diverse object types (everyday vs. industrial), appearances (textured vs. textureless),

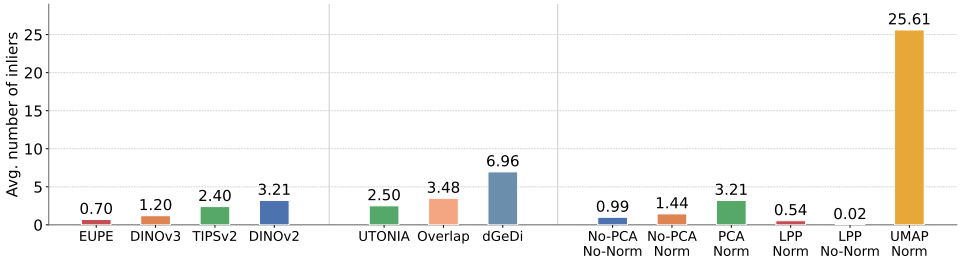


Figure 3: Average number of inliers obtained with different appearance encoders (left), geometric encoders (center), and dimensionality reduction techniques (right).

and symmetries, as well as challenging environmental conditions such as heavy occlusion, dense clutter, and varying illumination.

Metrics. To measure the distinctiveness of the features used to condition the flow matching process, we evaluate the *average number of inliers* among object–scene correspondences. We first establish correspondences between the object 3D model and the scene point cloud via mutual nearest-neighbor matching in the feature space. We then transform the object 3D model using its ground-truth 6D pose to align it with the scene and compute the Euclidean distance between each matched point pair. A correspondence is considered an inlier if this distance is below a threshold of 3% of the object diameter. To measure 6D pose accuracy, we use Average Recall (AR) [16]. AR is the average of the recalls of three metrics: the Visible Surface Discrepancy (VSD), the Maximum Symmetry-Aware Surface Distance (MSSD), and the Maximum Symmetry-Aware Projection Distance (MSPD). VSD measures pose error over the visible object surface, making it inherently invariant to symmetries. MSSD captures the maximum 3D surface deviation under the set of valid symmetry transformations, reflecting suitability for grasping and manipulation. MSPD measures the maximum 2D projection error, relevant for applications such as augmented reality. For each of these three metrics, we compute recall over a range of error thresholds, obtaining $\mathrm{AR}_{\mathrm{VSD}}$, $\mathrm{AR}_{\mathrm{MSSD}}$, and $\mathrm{AR}_{\mathrm{MSPD}}$. AR is then defined as $\mathrm{AR} = (\mathrm{AR}_{\mathrm{VSD}} + \mathrm{AR}_{\mathrm{MSSD}} + \mathrm{AR}_{\mathrm{MSPD}})/3$. To measure the method latency we compute the average inference time per image, measured in milliseconds (ms). Latency is a practical constraint for the closed-loop downstream applications like robotic grasping and manipulation, which require pose updates at the sensor frame rate.

4.2 Quantitative results

Feature discriminativity analysis. During flow matching, FunFlow6D aligns points from a corrupted object representation with those of the observed scene. Since this process is conditioned on point-level features, highly discriminative features are essential for establishing reliable object-scene correspondences. To improve correspondence quality, Flose [11] fine-tunes the overlap-aware encoder introduced in RPF [42] on each BOP dataset [16], a large-scale benchmark for 6D pose estimation. However, our experiments show that these overlap-aware features remain insufficiently discriminative despite the task-specific fine-tuning. This motivated a thorough investigation of publicly available foundation models for geometric feature encoding. Fig. 3 summarizes the outcomes of this analysis in terms of average number of inliers on LM-O [3]. It compares various geometric encoders (center), appearance encoders (left), and dimensionality reduction techniques (right). Among the geometric feature extractors, the overlap-aware encoder achieves an average number of inliers of 3.48, whereas

Table 1: 6D pose accuracy on the BOP benchmark, evaluated in terms of AR.

	Method	Models	Params	LM-O	TUD-L	IC-BIN	YCB-V	Avg
1	HccePose(BF) [45]	34	41.5M	80.5	94.4	72.4	91.1	84.6
2	GDRNPP (BOP23) [26]	4	106.8M	79.4	96.4	73.7	92.8	85.6
3	Flose [11]	4	89.2M	86.1	98.8	74.8	92.9	88.2
4	FunFlow6D (ours)	4	43.1M	**87.1**	**98.9**	**75.3**	**93.6**	**88.7**
5	Improvement wrt row 3	-	-	+1.0	+0.1	+0.5	+0.7	+0.5

dGeDi [10] reaches 6.96, yielding approximately a twofold improvement despite requiring no task-specific fine-tuning. This is not observed for other foundation models, as the recent Utonia [48] encoder reaches only 2.5 inliers.

Among the appearance-aware feature extractors, DINOv2 outperforms both its follow up, DINOv3 [39], and two recent approaches, EUPE [49] and TIPSv2 [9]. Specifically, DINOv2 has an average number of inliers of 3.21, surpassing TIPSv2 (2.40), DINOv3 (1.20), and EUPE (0.70). We attribute this gap to the nature of the supervision. DINOv2's self-distilled patch features encode dense, spatially-localized semantics that remain stable across viewpoint and illumination changes, whereas the more recent variants optimize for global or task-aligned objectives that smooth out the fine-grained local distinctive details our point-level conditioning depends on. Since correspondence routing operates per point rather than per image, an encoder that preserves intra-object feature variation is preferable to one that maximizes inter-image semantic consistency. This motivates our choice of DINOv2 as an appearance encoder.

We discovered empirically that the specific technique used to reduce the dimensionality of the appearance encoder to match that of the geometric encoder plays an important role (Fig. 3, right). Among the different dimensionality reduction techniques tested, UMAP yields by far the highest discriminativity, raising the average number of inliers to 25.61, an order of magnitude above the next best alternative. The linear baselines fall well short. PCA reaches only 3.21, while applying no projection at all (No-PCA) gives 0.99 without normalization and 1.44 with it. Locality-preserving projection (LPP) [13], despite being designed to retain neighborhood structure, collapses, dropping to 0.54 when normalized and to 0.02 without normalization. Two observations follow. First, the choice of projection matters more than the choice of whether to normalize. The normalization toggle shifts the average number of inliers by a fraction of a point, whereas swapping PCA for UMAP improves it roughly eightfold. Second, the failure of PCA and LPP confirms our hypothesis. PCA preserves global variance but discards the local neighborhood structure that per-point conditioning relies on, and even an explicitly locality-aware linear method like LPP cannot recover the task-relevant manifold. UMAP's non-linear, neighborhood graph based embedding retains exactly these local relationships, keeping nearby points separable in the reduced space and thereby supplying the flow with a richer correspondence signal. We therefore adopt DINOv2 features reduced via UMAP with normalization as our appearance conditioning. This behavior is stable across runs: repeating the UMAP fit with five random seeds yields 25.4 ± 0.5 inliers on LM-O, against the 25.6 obtained with our fixed seed.

6D pose estimation accuracy. Tab. 1 reports a quantitative evaluation of 6D pose estimation accuracy in terms of AR. To keep the table compact, we compare FunFlow6D against the strongest indirect method (row 1, HccePose(BF) [45]), the strongest direct method (row 2, GDRNPP [26]), and the strongest flow matching-based approach (row 3, Flose [11]). The

"Models" column shows the number of neural networks trained by each method. HccePose(BF) trains one network for each object (34 models in total), whereas all other methods train a single network per dataset, requiring about 9× fewer models. The "Params" column shows the number of trainable parameters per model. FunFlow6D is the most parameter-efficient approach, requiring approximately 9× fewer parameters than HccePose(BF) and less than half the parameters of Flose. The remaining columns report AR on LM-O, TUD-L, IC-BIN, YCB-V, and their average. FunFlow6D consistently outperforms all competing methods on all four datasets. Row 5 shows the absolute improvement over Flose, which is both the closest approach to ours and the best performing competitor overall. Despite requiring substantially fewer trainable parameters and significantly weaker supervision, FunFlow6D achieves consistent gains across all datasets. Interestingly, the largest improvement is observed on LM-O, a dataset characterized by severe occlusions, suggesting that FunFlow6D is particularly effective in scenarios where robust geometric reasoning is required. Notably, these gains are achieved while conditioning on frozen encoders and without any per-instance training or finetuning, unlike all competing approaches.

4.3 Qualitative results

Fig. 4 shows qualitative comparisons on LM-O and YCB-V between FunFlow6D and Flose [11]. Each image shows a rendering of the object's 3D model in the predicted pose, overlaid on a grayscale version of input image. Prediction accuracy can be assessed by the alignment between the colored rendering and the visible portion of the object in the grayscale background. Rows 1–7 (LM-O) show objects under severe occlusion and clutter: a duck occluded by a drill, an ape severely occluded by a duck, a glue bottle partially hidden by a mechanical part and a cat highly occluded by the drill, a white egg box heavily occluded by multiple objects in the front, a smooth textured but geometrically distinct hole punch occluded by the duck, another instance of less occluded ape. In these cases, the overlap encoder in Flose provides unreliable point-to-point cues, whereas the discriminative geometric descriptor in FunFlow6D lets the flow route scene points to the model surface more accurately. Rows 1–7 (YCB-V) show objects where appearance is the dominant cue: a tomato soup can, a tuna fish can, a richly textured pudding box, a geometrically distinct clamp, a symmetric master chef can, a pair of scissors occluded by a sugar box, and a block heavily occluded by a white bottle. Flose misaligns the tomato soup can near its top rim and flips both the tuna fish can and the pudding box, errors caused by relying on near-symmetric geometry where the only discriminative cues are textural. On the master chef can, Flose misaligns the top of the can, again driven by its rotational near-symmetry. For the scissors, only the knuckle ends are clearly visible, and while both methods align that region, Flose places the pointed blades on the wrong side, producing a flipped pose; FunFlow6D recovers the correct orientation. On the block, Flose latches onto a single edge and aligns to it, leaving the rest of the object offset. In FunFlow6D, the brand text and logos supply distinctive semantic features, and the learned gating module up-weights the DINOv2 descriptors over geometry, resolving the symmetry and alignment ambiguities that the geometric branch alone cannot.

4.4 Inference time breakdown and ablation study

Inference time breakdown. We provide here the per-image inference time breakdown for LM-O (NVIDIA A100 SXM4 64GB, 32-core Intel Xeon CPU), which contains 8 objects across 200 images and on average features all of them per image. Timings are measured

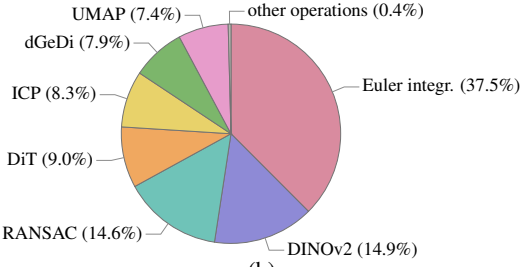
(a) LM-O (b) YCB-V

Figure 4: FunFlow6D (right) vs. Flose [11] (center) on LM-O (a) and YCB-V (b). FunFlow6D predicts more accurate poses under severe occlusions and symmetry ambiguities.

per image. For segmentation, we use precomputed ZebraPose [41] masks (NVIDIA RTX 2080Ti, Intel Xeon E-2146G CPU @ 3.50GHz) available on the BOP leaderboard, taking 0.08 s per image. For the rest of the pipeline, i.e. the geometric encoder, semantic encoder, and flow model are batched according to the number of segmented instances in the image, so that all instances of an image are processed in a single forward pass. For the dimensionality reduction we adopt the GPU implementation of UMAP from TorchDR [2]. Fig. 5(a) compares FunFlow6D average per-image runtime against the competing methods, where HccePose and GDRNPP's runtimes are taken from the public BOP leaderboard [1]. FunFlow6D runtime is comparable with that of Flose, evaluated on the same GPU type as ours, slightly slower

Method	GPU	Time (s)
HccePose [45]	RTX4090	0.923
GDRNPP [26]	RTX3090	2.579
Flose [11]	A100	1.018
FunFlow6D	A100	1.016

(a)



(b)

Figure 5: Inference time analysis on LM-O. (a) Average per-image runtime against competing methods. Object instance segmentation is excluded for all methods. (b) Per-image breakdown of FunFlow6D. The Euler integration (50 steps) dominates the pipeline.

Table 2: Ablation study on LM-O [3]. Key: 🔥 = trained, ❄ = frozen, - = not used. The default configuration of FunFlow6D is highlighted.

	Geom. Enc.	App. Enc.	Dim. Red.	Feat. fusion	Pose estim.	Pose refin.	AR
1	🔥 overlap-aware	-	-	-	SVD	ICP	83.5
2	🔥 overlap-aware	❄ DINOv2	PCA	❄ Point-level sum	RANSAC	ICP	86.1
3	❄ dGeDi	❄ DINOv2	PCA	❄ Point-level sum	RANSAC	ICP	81.8
4	❄ dGeDi	❄ DINOv2	UMAP	❄ Point-level sum	RANSAC	ICP	83.6
5	❄ dGeDi	❄ DINOv2	UMAP	❄ No attention	RANSAC	ICP	85.7
6	❄ dGeDi	❄ DINOv2	UMAP	🔥 Cross attention	RANSAC	ICP	86.3
7	❄ dGeDi	❄ DINOv2	UMAP	🔥 Gated attention	RANSAC	-	85.8
8	❄ dGeDi	❄ DINOv2	UMAP	🔥 Gated attention	RANSAC	ICP	**87.1**

than HccePose, and faster than GDRNPP. The detailed cost per image for FunFlow6D is as follows: DINOv2 forward pass takes 151.29 ms (14.89%), UMAP projection takes 75.10 ms (7.39%), the dGeDi geometric encoding takes 79.90 ms (7.87%), the DiT (flow model) forward pass takes 91.21 ms (8.98%), Euler integration at 50 steps takes 381.11 ms (37.52%), RANSAC registration (1k iters, Open3D) takes 148.45 ms (14.61%), ICP refinement (3k iters, Open3D) takes 84.59 ms (8.33%), and other operations (lifting point clouds, downsampling, and outlier removal) take 4.25 ms (0.42%), resulting in 1015.90 ms per image. The Euler integration accounts for the largest share of the total time (37.5%), which is inherent to iterative generative approaches. It is followed by the DINOv2 forward pass (14.9%) and RANSAC registration (14.6%). The full breakdown is summarized in Fig. 5(b).

Ablation study. Tab. 2 dissects each component of FunFlow6D on LM-O [3], isolating the contribution of geometric encoder, dimensionality reduction of semantic features, fusion strategy, and pose refinement stage. We start from an overlap-aware baseline and progressively replace each module with our proposed design. *Geometric conditioning.* Rows 1–2 establish the baseline, which conditions the flow on a pre-trained overlap-aware encoder [42]. Adding frozen DINOv2 [31] features reduced with PCA and fused by point-level summation raises AR from 83.5 to 86.1, confirming that semantic cues complement geometry. Replacing the overlap-aware encoder with *frozen* zero-shot dGeDi descriptor [10] under the same PCA and point-level sum scheme yields 81.8 AR. This configuration remains competitive while using no instance specific supervision for feature extraction. *Semantic dimensionality reduction.* Rows 3–4 isolate the effect of the reduction technique while keeping every other component fixed. Substituting PCA with UMAP [28] improves AR from 81.8 to 83.6 (+1.8). UMAP is

non-linear and explicitly preserves local neighborhoods, it retains task-relevant directions that PCA's linear, variance-driven projection drops below threshold. This is consistent with the correspondence-level evidence reported in our analysis, where DINOv2-UMAP raises the average number of inliers from 3 to 25 at the same target dimensionality. *Fusion strategy.* Rows 4–6 replace the static point-level summation with learned fusion. Our module consists of gated cross-attention followed by concatenation and MLP adaptation, and we isolate each of these in turn. Removing the attention entirely and retaining only the concatenation and MLP adaptation (row 5) already raises AR from 83.6 to 85.7, showing that a parameterized combination outperforms an equally weighted sum. Restoring the cross-modal interaction with standard, ungated cross-attention (row 6) adds a further $+0.6$, as every point can now attend to the complementary modality rather than being reweighted channel-wise. Adding the sigmoid gate (row 8) gives the best result, since each point can now suppress the unreliable modality instead of only mixing the two. This progression quantifies the limitation of treating semantics and geometry as equally informative. Textured regions benefit from appearance while geometrically salient regions rely on shape, and the per-point gates learn to route each cue accordingly. Even without any pose refinement, gated attention (row 7) reaches 85.8 AR, exceeding the summation variant that uses ICP refinement (row 4). *Pose refinement.* Rows 7–8 sweep the refinement stage on top of the full conditioning pipeline. The coarse RANSAC estimate alone yields 85.8 AR (row 7). ICP recovers an additional +1.3 AR (row 8) by correcting residual alignment errors. The proposed pipeline (row 8) attains 87.1 AR, a +3.6 improvement over the overlap-aware baseline (row 1), while eliminating the pre-training requirement. The three proposed components are complementary. Each contributes a measurable gain and their combination accounts for the full margin over the baseline.

5 Conclusion

We presented FunFlow6D, the first conditional flow matching method that conditions the flow matching process via fusion of appearance and geometric foundational features to estimate the 6D pose of known objects from RGBD observations of a scene. Compared to previous state-of-the-art methods, FunFlow6D conditions the generative denoising process on multimodal foundational features, employing a gated attention-based fusion strategy to merge them. Experiments demonstrate that FunFlow6D outperforms state-of-the-art competitors on four datasets from the BOP benchmark, achieving superior robustness to symmetries and occlusions while requiring a simplified training recipe and weaker supervision signals.

Limitations and future work. (1) Our pipeline relies on an external module to estimate 6D poses via RANSAC, as the output of the flow matching process is not guaranteed to be a valid rigid transformation. Enforcing rigidity constraints during denoising would make the approach fully end-to-end and more efficient. (2) During inference, FunFlow6D requires multiple denoising steps, which increases the latency of the method. In the future, we will explore single-step flow matching alternatives to improve the latency. (3) FunFlow6D relies on segmentation masks, this makes performance contingent on segmentation quality and introduces an additional pipeline stage that is not jointly optimized. A future direction can be removing the localization stage and predicting the flow directly at scene level point clouds.

References

[1] BOP benchmark leaderboard: 6D object localization, BOP-Classic-Core. https://bop.felk.cvut.cz/leaderboards/pose-estimation-bop19/bop-classic-core/. Accessed: 2026-08-26.

[2] Hugues Van Assel, Nicolas Courty, Rémi Flamary, Aurélien Garivier, Mathurin Massias, Titouan Vayer, et al. TorchDR: PyTorch dimensionality reduction, 2024. URL https://github.com/TorchDR/TorchDR.

[3] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In *ECCV*, 2014.

[4] Bingyi Cao, Koert Chen, Kevis-Kokitsi Maninis, Kaifeng Chen, Arjun Karpur, Ye Xia, et al. TIPSv2: Advancing vision-language pretraining with enhanced patch-text alignment. In *CVPR*, 2026.

[5] Andrea Caraffa, Davide Boscaini, Amir Hamza, and Fabio Poiesi. FreeZe: Training-free zero-shot 6D pose estimation with geometric and vision foundation models. In *ECCV*, 2024.

[6] Andrea Caraffa, Davide Boscaini, and Fabio Poiesi. Accurate and efficient zero-shot 6D pose estimation with frozen foundation models. *arXiv:2506.09784*, 2025.

[7] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, 2019.

[8] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global context aware local features for robust 3D point matching. In *CVPR*, 2018.

[9] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6D object pose and predicting next-best-view in the crowd. In *CVPR*, 2016.

[10] Amir Hamza, Andrea Caraffa, Davide Boscaini, and Fabio Poiesi. Distilling 3D distinctive local descriptors for 6D pose estimation. In *IROS*, 2025.

[11] Amir Hamza, Davide Boscaini, Weihang Li, Benjamin Busam, and Fabio Poiesi. Generative 6D pose estimation via conditional flow matching. In *ICIP*, 2026.

[12] Rasmus Laurvig Haugaard and Anders Glent Buch. SurfEmb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *CVPR*, 2022.

[13] Xiaofei He and Partha Niyogi. Locality preserving projections. In *NeurIPS*, 2003.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[15] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, et al. BOP: Benchmark for 6D object pose estimation. In *ECCV*, 2018.

[16] Tomas Hodan, Martin Sundermeyer, Yann Labbé, Van Nguyen Nguyen, Gu Wang, Eric Brachmann, et al. BOP Challenge 2023 on detection, segmentation and pose estimation of seen and unseen rigid objects. In *CVPRW*, 2024.

[17] Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Perspective flow aggregation for data-limited 6D object pose estimation. In *ECCV*, 2022.

[18] Yufeng Jin, Niklas Funk, Vignesh Prasad, Zechu Li, Mathias Franzius, Jan Peters, et al. SE(3)-PoseFlow: Estimating 6D pose distributions for uncertainty-aware robotic manipulation. In *ICRA*, 2026.

[19] Rebecca König and Bertram Drost. A hybrid approach for 6DoF pose estimation. In *ECCVW*, 2020.

[20] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. In *ECCV*, 2020.

[21] Sihang Li, Zeyu Jiang, Grace Chen, Chenyang Xu, Siqi Tan, Xue Wang, et al. GARF: Learning generalizable 3D reassembly for real-world fractures. In *ICCV*, 2025.

[22] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[23] Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled iterative refinement for 6D multi-object pose estimation. In *CVPR*, 2022.

[24] Jian Liu, Wei Sun, Chongpei Liu, Xing Zhang, and Qiang Fu. Robotic continuous grasping system by shape transformer-guided multiobject category-level 6-D pose estimation. *IEEE TII*, 2023.

[25] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[26] Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Gu Wang, Jiwen Tang, Zhigang Li, et al. GDRNPP: A geometry-guided and fully learning-based object pose estimator. *IEEE TPAMI*, 2025.

[27] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (PCA). *Computers & Geosciences*, 1993.

[28] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2018.

[29] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *CACM*, 2021.

[30] Mattia Nardon, Mikel Mujika Agirre, Ander González Tomé, Daniel Sedano Algarabel, Josep Rueda Collell, Ana Paola Caro, et al. CHIP: A multi-sensor dataset for 6D pose estimation of chairs in industrial settings. In *BMVC*, 2025.

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.

[32] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, et al. FoundPose: Unseen object pose estimation with foundation features. In *ECCV*, 2024.

[33] Yue Pan, Tao Sun, Liyuan Zhu, Lucas Nunes, Iro Armeni, Jens Behley, et al. Register Any Point: Scaling 3D point cloud registration by flow matching. In *ECCV*, 2026.

[34] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation. In *ICCV*, 2019.

[35] Fabio Poiesi and Davide Boscaini. Learning general and distinctive 3D local deep descriptors for point cloud registration. *IEEE TPAMI*, 2022.

[36] Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *NeurIPS*, 2025.

[37] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *ICRA*, 2009.

[38] Samuele Salti, Federico Tombari, and Luigi Di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *CVIU*, 2014.

[39] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, et al. DINOv3. *TMLR*, 2026.

[40] Yongzhi Su, Jason Rambach, Nareg Minaskan, Paul Lesur, Alain Pagani, and Didier Stricker. Deep multi-state object pose estimation for augmented reality assembly. In *ISMAR*, 2019.

[41] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, et al. ZebraPose: Coarse to fine surface encoding for 6DoF object pose estimation. In *CVPR*, 2022.

[42] Tao Sun, Liyuan Zhu, Shengyu Huang, Shuran Song, and Iro Armeni. Rectified Point Flow: Generic point cloud pose estimation. In *NeurIPS*, 2025.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, et al. Attention is all you need. In *NeurIPS*, 2017.

[44] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. In *CVPR*, 2021.

[45] Yulin Wang, Mengting Hu, Hongli Li, and Chen Luo. HccePose(BF): Predicting front & back surfaces to construct ultra-dense 2D-3D correspondences for pose estimation. In *ICCV*, 2025.

[46] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, et al. Point Transformer V3: Simpler, faster, stronger. In *CVPR*, 2024.

[47] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *RSS*, 2018.

[48] Yujia Zhang, Xiaoyang Wu, Yunhan Yang, Xianzhe Fan, Han Li, Yuechen Zhang, et al. Utonia: Toward one encoder for all point clouds. In *ICML*, 2026.

[49] Chenchen Zhu, Saksham Suri, Cijo Jose, Maxime Oquab, Marc Szafraniec, Wei Wen, et al. Efficient universal perception encoder. *arXiv:2603.22387*, 2026.

[50] Chungang Zhuang, Shaofei Li, and Han Ding. Instance segmentation based 6D pose estimation of industrial objects using point clouds for robotic bin-picking. *RCIM*, 2023.