EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation

Hansheng Chen[1,2,*] Pichao Wang[2,†] Fan Wang[2] Wei Tian[1,†] Lu Xiong[1] Hao Li[2]
[1]School of Automotive Studies, Tongji University [2]Alibaba Group
hanshengchen97@gmail.com {tian_wei, xiong_lu}@tongji.edu.cn
{pichao.wang, fan.w, lihao.lh}@alibaba-inc.com

Abstract

Locating 3D objects from a single RGB image via Perspective-n-Points (PnP) is a long-standing problem in computer vision. Driven by end-to-end deep learning, recent studies suggest interpreting PnP as a differentiable layer, so that 2D-3D point correspondences can be partly learned by backpropagating the gradient w.r.t. object pose. Yet, learning the entire set of unrestricted 2D-3D points from scratch fails to converge with existing approaches, since the deterministic pose is inherently non-differentiable. In this paper, we propose the EPro-PnP, a probabilistic PnP layer for general end-to-end pose estimation, which outputs a distribution of pose on the SE(3) manifold, essentially bringing categorical Softmax to the continuous domain. The 2D-3D coordinates and corresponding weights are treated as intermediate variables learned by minimizing the KL divergence between the predicted and target pose distribution. The underlying principle unifies the existing approaches and resembles the attention mechanism. EPro-PnP significantly outperforms competitive baselines, closing the gap between PnP-based method and the task-specific leaders on the LineMOD 6DoF pose estimation and nuScenes 3D object detection benchmarks.[3]

1. Introduction

Estimating the pose (*i.e.*, position and orientation) of 3D objects from a single RGB image is an important task in computer vision. This field is often subdivided into specific tasks, *e.g.*, 6DoF pose estimation for robot manipulation and 3D object detection for autonomous driving. Although they share the same fundamentals of pose estimation, the different nature of the data leads to biased choice of methods. Top performers [29, 42, 44] on the 3D object detection benchmarks [6, 14] fall into the category of direct 4DoF pose prediction, leveraging the advances in end-to-end deep learning. On the other hand, the 6DoF pose estimation benchmark [19] is largely dominated by geometry-based methods [20, 46], which exploit the provided 3D object models and achieve a stable generalization performance. However, it is quite challenging to bring together the best of both worlds, *i.e.*, training a geometric model to learn the object pose in an end-to-end manner.

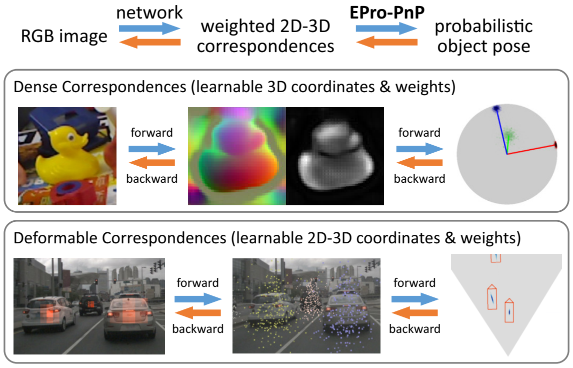
RGB image
network
weighted 2D-3D correspondences
EPro-PnP
probabilistic object pose
Dense Correspondences (learnable 3D coordinates & weights)
forward
backward
forward
backward
Deformable Correspondences (learnable 2D-3D coordinates & weights)
forward
backward
forward
backward

Figure 1. EPro-PnP is a general solution to end-to-end 2D-3D correspondence learning. In this paper, we present two distinct networks trained with EPro-PnP: (a) an off-the-shelf dense correspondence network whose potential is unleashed by end-to-end training, (b) a novel deformable correspondence network that explores new possibilities of fully learnable 2D-3D points.

There has been recent proposals for an end-to-end framework based on the Perspective-n-Points (PnP) approach [2, 4, 7, 10]. The PnP algorithm itself solves the pose from a set of 3D points in object space and their corresponding 2D projections in image space, leaving the problem of constructing these correspondences. Vanilla correspondence learning [9, 23, 24, 30, 30–32, 35, 40, 46] leverages the geometric prior to build surrogate loss functions, forcing the network to learn a set of pre-defined correspondences. End-to-end correspondence learning [2, 4, 7, 10] interprets the

[*] Part of work done during an internship at Alibaba Group.
[†] Corresponding authors: Pichao Wang, Wei Tian.
[3] Code: https://github.com/tjiiv-cprg/EPro-PnP

PnP as a differentiable layer and employs pose-driven loss function, so that gradient of the pose error can be backpropagated to the 2D-3D correspondences.

However, existing work on differentiable PnP learns only a portion of the correspondences (either 2D coordinates [10], 3D coordinates [2, 4] or corresponding weights [7]), assuming other components are given *a priori*. This raises an important question: why not learn the entire set of points and weights altogether in an end-to-end manner? The simple answer is: the solution of the PnP problem is inherently non-differentiable at some points, causing training difficulties and convergence issues. More specifically, a PnP problem can have ambiguous solutions [27,33], which makes backpropagation unstable.

To overcome the above limitations, we propose a generalized **e**nd-to-end **pro**babilistic **PnP** (EPro-PnP) approach that enables learning the weighted 2D-3D point correspondences entirely from scratch (Figure 1). The main idea is straightforward: deterministic pose is non-differentiable, but the probability density of pose is apparently differentiable, just like categorical classification scores. Therefore, we interpret the output of PnP as a probabilistic distribution parameterized by the learnable 2D-3D correspondences. During training, the Kullback-Leibler (KL) divergence between the predicted and target pose distributions is computed as the loss function, which is numerically tractable by efficient Monte Carlo pose sampling.

As a general approach, EPro-PnP inherently unifies existing correspondence learning techniques (Section 3.1). Moreover, just like the attention mechanism [38], the corresponding weights can be trained to automatically focus on important point pairs, allowing the networks to be designed with inspiration from attention-related work [8,43,48].

To summarize, our main contributions are as follows:

- We propose the EPro-PnP, a probabilistic PnP layer for general end-to-end pose estimation via learnable 2D-3D correspondences.
- We demonstrate that EPro-PnP can easily reach top-tier performance for 6DoF pose estimation by simply inserting it into the CDPN [24] framework.
- We demonstrate the flexibility of EPro-PnP by proposing *deformable correspondence learning* for accurate 3D object detection, where the entire 2D-3D correspondences are learned from scratch.

2. Related Work

Geometry-Based Object Pose Estimation In general, geometry-based methods exploit the points, edges or other types of representation that are subject to the projection constraints under the perspective camera. Then, the pose can be solved by optimization. A large body of work utilizes point representation, which can be categorized into sparse keypoints and dense correspondences. BB8 [32] and RTM3D [23] locate the corners of the 3D bounding box as keypoints, while PVNet [31] defines the keypoints by farthest point sampling and Deep MANTA [9] by handcrafted templates. On the other hand, dense correspondence methods [11, 24, 30, 40, 46] predict pixel-wise 3D coordinates within a cropped 2D region. Most existing geometry-based methods follow a two-stage strategy, where the intermediate representations (*i.e.*, 2D-3D correspondences) are learned with a surrogate loss function, which is sub-optimal compared to end-to-end learning.

End-to-End Correspondence Learning To mitigate the limitation of surrogate correspondence learning, end-to-end approaches have been proposed to backpropagate the gradient from pose to intermediate representation. By differentiating the PnP operation, Brachmann and Rother [4] propose a dense correspondence network where 3D points are learnable, BPnP [10] predicts 2D keypoint locations, and BlindPnP [7] learns the corresponding weight matrix given a set of unordered 2D/3D points. Beyond point correspondence, RePOSE [20] proposes a feature-metric correspondence network trained in a similar end-to-end fashion. The above methods are all coupled with surrogate regularization loss, otherwise convergence is not guaranteed due to the non-differentiable nature of deterministic pose. Under the probabilistic framework, these methods can be regarded as a Laplace approximation approach (Section 3.1) or a local regularization technique (Section 3.4).

Probabilistic Deep Learning Probabilistic methods account for uncertainty in the model and the data, known respectively as epistemic and aleatoric uncertainty [21]. The latter involves interpreting the prediction as learnable probabilistic distributions. Discrete categorical distribution via Softmax has been widely adopted as a smooth approximation of one-hot $\arg\max$ for end-to-end classification. This inspired works such as DSAC [2], a smooth RANSAC with a finite hypothesis pool. Meanwhile, simple parametric distributions (*e.g.*, normal distribution) are often used in predicting continuous variables [11,15,18,21,22,45], and mixture distributions can be employed to further capture ambiguity [1,3,26], *e.g.*, ambiguous 6DoF pose [5]. In this paper, we propose yet a unique contribution: backpropagating a complicated continuous distribution derived from a nested optimization layer (the PnP layer), essentially making the continuous counterpart of Softmax tractable.

3. Generalized End-to-End Probabilistic PnP

3.1. Overview

Given an object proposal, our goal is to predict a set $X = \left\{x_i^{\text{3D}}, x_i^{\text{2D}}, w_i^{\text{2D}} \,\middle|\, i = 1 \cdots N\right\}$ of N corresponding points, with 3D object coordinates $x_i^{\text{3D}} \in \mathbb{R}^3$, 2D image coordinates $x_i^{\text{2D}} \in \mathbb{R}^2$, and 2D weights $w_i^{\text{2D}} \in \mathbb{R}_+^2$, from which a weighted PnP problem can be formulated to estimate the object pose relative to the camera.

The essence of a PnP layer is searching for an optimal pose y (expanded as rotation matrix R and translation vector t) that minimizes the cumulative squared weighted reprojection error:

$$\arg\min_{y} \frac{1}{2} \sum_{i=1}^{N} \Big\| \underbrace{w_i^{\mathrm{2D}} \circ \left(\pi(Rx_i^{\mathrm{3D}} + t) - x_i^{\mathrm{2D}}\right)}_{f_i(y) \in \mathbb{R}^2} \Big\|^2, \qquad (1)$$

where $\pi(\cdot)$ is the projection function with camera intrinsics involved, $\circ$ stands for element-wise product, and $f_i(y)$ compactly denotes the weighted reprojection error.

Eq. (1) formulates a non-linear least squares problem that may have non-unique solutions, *i.e.*, pose ambiguity [27, 33]. Previous work [4, 7, 10] only backpropagates through a local solution y^*, which is inherently unstable and non-differentiable. To construct a differentiable alternative for end-to-end learning, we model the PnP output as a distribution of pose, which guarantees differentiable probability density. Consider the cumulative error to be the negative logarithm of the likelihood function $p(X|y)$ defined as:

$$p(X|y) = \exp -\frac{1}{2} \sum_{i=1}^{N} \|f_i(y)\|^2. \qquad (2)$$

With an additional prior pose distribution $p(y)$, we can derive the posterior pose $p(y|X)$ via the Bayes theorem. Using an *uninformative prior*, the posterior density is simplified to the normalized likelihood:

$$p(y|X) = \frac{\exp -\frac{1}{2} \sum_{i=1}^{N} \|f_i(y)\|^2}{\int \exp -\frac{1}{2} \sum_{i=1}^{N} \|f_i(y)\|^2 \,\mathrm{d}y}. \qquad (3)$$

Eq. (3) can be interpreted as a continuous counterpart of categorical Softmax.

KL Loss Function During training, given a target pose distribution with probability density $t(y)$, the KL divergence $D_{\mathrm{KL}}(t(y) \| p(y|X))$ is minimized as training loss. Intuitively, pose ambiguity can be captured by the multiple modes of $p(y|X)$, and convergence is ensured such that wrong modes are suppressed by the loss function. Dropping the constant, the KL divergence loss can be written as:

$$L_{\mathrm{KL}} = -\int t(y) \log p(X|y) \,\mathrm{d}y + \log \int p(X|y) \,\mathrm{d}y. \qquad (4)$$

We empirically found it effective to set a narrow (Dirac-like) target distribution centered at the ground truth y_{gt}, yielding the simplified loss (after substituting Eq. (2)):

$$L_{\mathrm{KL}} = \underbrace{\frac{1}{2} \sum_{i=1}^{N} \|f_i(y_{\mathrm{gt}})\|^2}_{L_{\mathrm{tgt}} \text{ (reproj. at target pose)}} + \underbrace{\log \int \exp -\frac{1}{2} \sum_{i=1}^{N} \|f_i(y)\|^2 \,\mathrm{d}y}_{L_{\mathrm{pred}} \text{ (reproj. at predicted pose)}}. \qquad (5)$$

The only remaining problem is the integration in the second term, which is elaborated in Section 3.2.

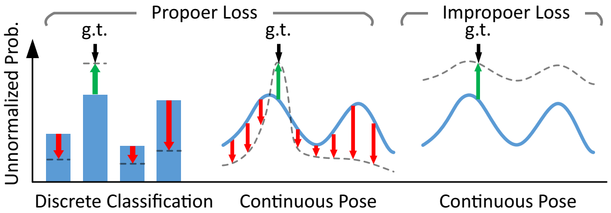

Figure 2. **Learning a discrete classifier *vs.* Learning the continuous pose distribution**. A discriminative loss function (left) shall encourage the unnormalized probability for the correct prediction as well as penalize for the incorrect. A one-sided loss (right) will degrade the distribution if the model is not well-regularized.

Comparison to Reprojection-Based Method The two terms in Eq. (5) are concerned with the reprojection errors at target and predicted pose respectively. The former is often used as a surrogate loss in previous work [4, 10, 11]. However, the first term alone cannot handle learning all 2D-3D points without imposing strict regularization, as the minimization could simply drive all the points to a concentrated location without pose discrimination. The second term originates from the normalization factor in Eq. (3), and is crucial to a discriminative loss function, as shown in Figure 2.

Comparison to Implicit Differentiation Method Existing work on end-to-end PnP [7, 10] derives a single solution of a particular solver $y^* = PnP(X)$ via implicit function theorem [16]. In the probabilistic framework, this is essentially the Laplace method that approximates the posterior by $\mathcal{N}(y^*, \Sigma_{y^*})$, where both y^* and Σ_{y^*} can be estimated by the PnP solver with analytical derivatives [11]. A special case is that, with Σ_{y^*} simplified to be homogeneous, the approximated KL divergence can be simplified to the L2 loss $\|y^* - y_{\mathrm{gt}}\|^2$ used in [7]. However, the Laplace approximation is inaccurate for non-normal posteriors with ambiguity, therefore does not guarantee global convergence.

3.2. Monte Carlo Pose Loss

In this section, we introduce a GPU-friendly efficient Monte Carlo approach to the integration in the proposed loss function, based on the Adaptive Multiple Importance Sampling (AMIS) algorithm [12].

Considering $q(y)$ to be the probability density function of a proposal distribution that approximates the shape of the integrand $\exp -\frac{1}{2} \sum_{i=1}^{N} \|f_i(y)\|^2$, and y_j to be one of the K samples drawn from $q(y)$, the estimation of the second term L_{pred} in Eq. (5) is thus:

$$L_{\mathrm{pred}} \approx \log \frac{1}{K} \sum_{j=1}^{K} \underbrace{\frac{\exp -\frac{1}{2} \sum_{i=1}^{N} \|f_i(y_j)\|^2}{q(y_j)}}_{v_j \text{ (importance weight)}}, \qquad (6)$$

where v_j compactly denotes the importance weight at y_j. Eq. (6) gives the vanilla importance sampling, where the choice of proposal $q(y)$ strongly affects the numerical stability. The AMIS algorithm is a better alternative as it iteratively adapts the proposal to the integrand.

In brief, AMIS utilizes the sampled importance weights from past iterations to estimate the new proposal. Then, all previous samples are re-weighted as being homogeneously sampled from a mixture of the overall sum of proposals. Initial proposal can be determined by the mode and covariance of the predicted pose distribution (see supplementary for details). A pseudo-code is given in Algorithm 1.

Choice of Proposal Distribution The proposal distributions for position and orientation have to be chosen separately in a decoupled manner, since the orientation space is non-Euclidean. For position, we adopt the 3DoF multivariate t-distribution. For 1D yaw-only orientation, we use a mixture of von Mises and uniform distribution. For 3D orientation represented by unit quaternion, the angular central Gaussian distribution [37] is adopted.

3.3. Backpropagation

In general, the partial derivatives of the loss function defined in Eq. (5) is:

$$\frac{\partial L_{\text{KL}}}{\partial(\cdot)} = \frac{\partial}{\partial(\cdot)}\frac{1}{2}\sum_{i=1}^{N}\|f_i(y_{\text{gt}})\|^2 - \mathop{\mathbb{E}}_{y\sim p(y|X)}\frac{\partial}{\partial(\cdot)}\frac{1}{2}\sum_{i=1}^{N}\|f_i(y)\|^2; \tag{7}$$

where the first term is the gradient of reprojection errors at target pose, and the second term is the expected gradient of reprojection errors over predicted pose distribution, which is approximated by backpropagating each weighted sample in the Monte Carlo pose loss.

Balancing Uncertainty and Discrimination Consider the negative gradient w.r.t. the corresponding weights w_i^{2D}:

$$-\frac{\partial L_{\text{KL}}}{\partial w_i^{\text{2D}}} = w_i^{\text{2D}} \circ \left(-r_i^{\circ 2}(y_{\text{gt}}) + \mathop{\mathbb{E}}_{y\sim p(y|X)} r_i^{\circ 2}(y)\right), \tag{8}$$

where $r_i(y) = \pi(Rx_i^{\text{3D}} + t) - x_i^{\text{2D}}$ (unweighted reprojection error), and $(\cdot)^{\circ 2}$ stands for element-wise square. The first bracketed term $-r_i^{\circ 2}(y_{\text{gt}})$ with negative sign indicates that correspondences with large reprojection error (hence high uncertainty) shall be weighted less. The second term $\mathbb{E}_{y\sim p(y|X)}\, r_i^{\circ 2}(y)$ is relevant to the variance of reprojection error over the predicted pose. The positive sign indicates that sensitive correspondences should be weighted more, because they provide stronger pose discrimination. The final gradient is thus a balance between the uncertainty and discrimination, as shown in Figure 3. Existing work [11,31] on learning uncertainty-aware correspondences only considers the former, hence lacking the discriminative ability.

3.4. Local Regularization of Derivatives

While the KL divergence is a good metric for the probabilistic distribution, for inference it is still required to estimate the exact pose y^* by solving the PnP problem in Eq. (1). The common choice of high precision is to utilize the iterative PnP solver based on the Levenberg-Marquardt (LM) algorithm – a robust variant of the Gauss-Newton (GN) algorithm, which solves the non-linear least squares by the first and approximated second order derivatives. To aid derivative-based optimization, we regularize the derivatives of the log density $\log p(y|X)$ w.r.t. the pose y, by encouraging the LM step Δy to find the true pose y_{gt}.

Algorithm 1: AMIS-based Monte Carlo pose loss

```
Input  : $X = \{x_i^{\text{3D}}, x_i^{\text{2D}}, w_i^{\text{2D}}\}$
Output: $L_{\text{pred}}$
$y^*, \Sigma_{y^*} \leftarrow PnP(X)$                                   // Laplace approximation
Fit $q_1(y)$ to $y^*, \Sigma_{y^*}$                                     // initial proposal
for $1 \le t \le T$ do
    Generate $K'$ samples $y_{j=1 \cdots K'}^t$ from $q_t(y)$
    for $1 \le j \le K'$ do
        $P_j^t \leftarrow \exp -\frac{1}{2}\sum_{i=1}^{N}\left\|f_i(y_j^t)\right\|^2$     // eval integrand
    for $1 \le \tau \le t$ and $1 \le j \le K'$ do
        $Q_j^\tau \leftarrow \frac{1}{t}\sum_{m=1}^{t} q_m(y_j^\tau)$                 // eval proposal mix
        $v_j^\tau \leftarrow P_j^\tau / Q_j^\tau$                                  // importance weight
    if $t < T$ then
        Estimate $q_{t+1}(y)$ from all weighted samples
          $\{y_j^\tau, v_j^\tau \mid 1 \le \tau \le t, 1 \le j \le K'\}$
$L_{\text{pred}} \leftarrow \log \frac{1}{TK'}\sum_{t=1}^{T}\sum_{j=1}^{K'} v_j^t$
```

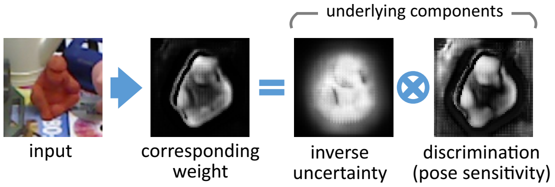

Figure 3. **The learned corresponding weight** can be factorized into inverse uncertainty and discrimination. Typically, inverse uncertainty roughly resembles the foreground mask, while discrimination emphasizes the 3D extremities of the object.

To employ the regularization during training, a detached solution y^* is obtained first. Then, at y^*, another iteration step is evaluated via the GN algorithm (which ideally equals 0 if y^* has converged to the local optimum):

$$\Delta y = -(J^{\text{T}}J + \varepsilon I)^{-1}J^{\text{T}}F(y^*), \tag{9}$$

where $F(y^*) = \left[f_1^{\text{T}}(y^*), f_2^{\text{T}}(y^*), \cdots, f_N^{\text{T}}(y^*)\right]^{\text{T}}$ is the concatenated weighted reprojection errors of all points, $J = \partial F(y)/\left.\partial y^{\text{T}}\right|_{y=y^*}$ is the Jacobian matrix, and ε is a small value for numerical stability. Note that Δy is analytically differentiable. We therefore design the regularization loss as follows:

$$L_{\text{reg}} = l(y^* + \Delta y, y_{\text{gt}}), \tag{10}$$

where $l(\cdot, \cdot)$ is a distance metric for pose. We adopt smooth L1 for position and cosine similarity for orientation (see supplementary materials for details). Note that the gradient is only backpropagated through Δy, encouraging the step to be non-zero if $y^* \neq y_{\text{gt}}$.

It is worth noting that this regularization loss is very similar to the loss function derived from implicit differentiation [7, 10], and it can be used for training pose refinement networks within a limited scope [20].

4. Attention-Inspired Correspondence Networks

As discussed in Section 3.3, the balance between uncertainty and discrimination enables locating important correspondences in an attention-like manner. This inspires us to take elements from attention-related work, *i.e.*, the Softmax layer and the deformable sampling [48].

In this section, we present two networks with EPro-PnP layer for 6DoF pose estimation and 3D object detection, respectively. For the former, EPro-PnP is incorporated into the existing dense correspondence architecture [24]. For the latter, we propose a radical deformable correspondence network to explore the flexibility of EPro-PnP.

4.1. Dense Correspondence Network

For a strict comparison against existing PnP-based pose estimators, this paper takes the network from CDPN [24] as a baseline, adding minor modifications to fit the EPro-PnP.

The original CDPN feeds cropped image regions within the detected 2D boxes into the pose estimation network, to which two decoupled heads are appended for rotation and translation respectively. The rotation head is PnP-based while the translation head uses direct regression. This paper discards the translation head to focus entirely on PnP.

Modifications are only made to the output layers. As shown in Figure 4, the original confidence map is expanded to two-channel XY weights with spatial Softmax and dynamic global weight scaling. Inspired by the attention mechanism [38], the Softmax layer is a vital element for stable training, as it translates the absolute corresponding weights into a relative measurement. On the other hand, the global weight scaling factors represent the global concentration of the predicted pose distribution, ensuring a better convergence of the KL divergence loss.

The dense correspondence network can be trained solely with the KL divergence loss L_{KL} to achieve decent performance. For top-tier performance, it is still beneficial to utilize additional coordinate regression as intermediate supervision, not to stabilize convergence but to introduce the geometric knowledge from the 3D models. Therefore, we keep the masked coordinate regression loss from CDPN [24] but leave out its confidence loss. Furthermore, the performance can be elevated by imposing the regularization loss L_{reg} in Eq. (10).

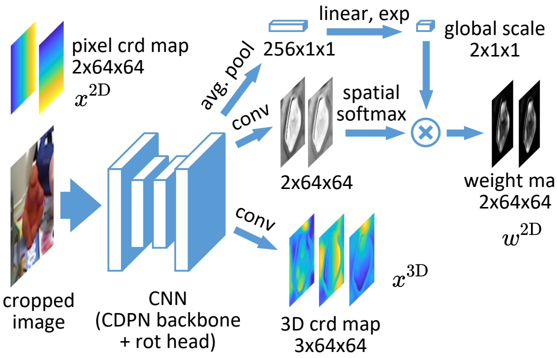

Figure 4. **The 6DoF pose estimation network** modified from CDPN [24]. with spatial Softmax and global weight scaling.

4.2. Deformable Correspondence Network

Inspired by Deformable DETR [48], we propose a novel deformable correspondence network for 3D object detection, in which the entire 2D-3D coordinates and weights are learned from scratch.

As shown in Figure 5, the deformable correspondence network is an extension of the FCOS3D [41] framework. The original FCOS3D is a one-stage detector that directly regresses the center offset, depth, and yaw orientation of multiple objects for 4DoF pose estimation. In our adaptation, the outputs of the multi-level FCOS head [36] are modified to generate object queries instead of directly predicting the pose. Also inspired by Deformable DETR [48], the appearance and position of a query is disentangled into the embedding vector and the reference point. A multi-head deformable attention layer [48] is adopted to sample the key-value pairs from the dense features, with the value projected into *point-wise features*, and meanwhile aggregated into the *object-level features*.

The point features are passed into a subnet that predicts the 3D points and corresponding weights (normalized by Softmax). Following MonoRUn [11], the 3D points are set in the normalized object coordinate (NOC) space to handle categorical objects of various sizes.

The object features are responsible for predicting the object-level properties: (a) the 3D score (*i.e.*, 3D localization confidence), (b) the weight scaling factor (same as in Section 4.1), (c) the 3D box size for recovering the absolute scale of the 3D points, and (d) other optional properties (velocity, attribute) required by the nuScenes benchmark [6].

The deformable 2D-3D correspondences can be learned solely with the KL divergence loss L_{KL}, preferably in conjunction with the regularization loss L_{reg}. Other auxiliary losses can be imposed onto the dense features for enhanced accuracy. Details are given in supplementary materials.

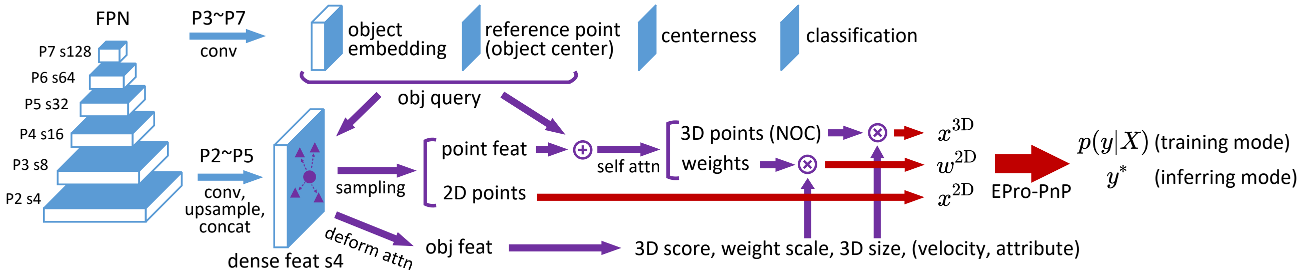

Figure 5. **The deformable correspondence network** based on the FCOS3D [41] detector. Note that the sampled point-wise features are shared by the point-level subnet and the deformable attention layer that aggregates the features for object-level predictions.

5. Experiments

5.1. Datasets and Metrics

LineMOD Dataset and Metrics The LineMOD dataset [19] consists of 13 sequences, each containing about 1.2K images annotated with 6DoF poses of a single object. Following [3], the images are split into the training and testing sets, with about 200 images per object for training. For data augmentation, we use the same synthetic data as in CDPN [24]. We use two common metrics for evaluation: ADD(-S) and $n°, n\,$cm. The ADD measures whether the average deviation of the transformed model points is less than a certain fraction of the object's diameter (*e.g.*, ADD-0.1d). For symmetric objects, ADD-S computes the average distance to the closest model point. $n°, n\,$cm measures the accuracy of pose based on angular/positional error thresholds. All metrics are presented as percentages.

nuScenes Dataset and Metrics The nuScenes 3D object detection benchmark [6] provides a large scale of data collected in 1000 scenes. Each scene contains 40 keyframes, annotated with a total of 1.4M 3D bounding boxes from 10 categories. Each keyframe includes 6 RGB images collected from surrounding cameras. The data is split into 700/150/150 scenes for training/validation/testing. The official benchmark evaluates the average precision with true positives judged by 2D center error on the ground plane. The mAP metric is computed by averaging over the thresholds of 0.5, 1, 2, 4 meters. Besides, there are 5 true positive metrics: Average Translation Error (ATE), Average Scale Error (ASE), Average Orientation Error (AOE), Average Velocity Error (AVE) and Average Attribute Error (AAE). Finally, there is a nuScenes detection score (NDS) computed as a weighted average of the above metrics.

5.2. Implementation Details

EPro-PnP Configuration For the PnP formulation in Eq. (1), in practice the actual reprojection costs are robustified by the Huber kernel $\rho(\cdot)$:

$$\arg\min_{y} \frac{1}{2}\sum_{i=1}^{N} \rho\Big(\|f_i(y)\|^2\Big). \tag{11}$$

The Huber kernel with threshold δ is defined as:

$$\rho(s) = \begin{cases} s, & s \le \delta^2, \\ \delta(2\sqrt{s} - \delta), & s > \delta^2. \end{cases} \tag{12}$$

We use an adaptive threshold as described in the supplementary materials. For Monte Carlo pose loss, we set the AMIS iteration count T to 4 and the number of samples per iteration K' to 128. The loss weights are tuned such that L_{KL} produces roughly the same magnitude of gradient as typical coordinate regression, while the gradient from L_{reg} are kept very low. The weight normalization technique in [11] is adopted to compute the dynamic loss weight for L_{KL}.

Training the Dense Correspondence Network General settings are kept the same as in CDPN [24] (with ResNet-34 [17] as backbone) for strict comparison, except that we increase the batch size to 32 for less training wall time. The network is trained for 160 epochs by RMSprop on the LineMOD dataset [19]. To reduce the Monte Carlo overhead, 512 points are randomly sampled from the 64×64 dense points to compute L_{KL}.

Training the Deformable Correspondence Network We adopt the same detector architecture as in FCOS3D [41], with ResNet-101-DCN [13] as backbone. The network is trained for 12 epochs by the AdamW [25] optimizer, with a batch size of 12 images across 4 GPUs on the nuScenes dataset [6].

5.3. Results on the LineMOD Benchmark

Comparison to the CDPN baseline with Ablations The contributions of every single modification to CDPN [24] are revealed in Table 1. From the results it can be observed that:

- The original CDPN heavily relies on direct position regression, and the performance drops greatly (-17.46) when reduced to a pure PnP estimator, although the LM solver partially recovers the mean metric (+6.29).
- Employing EPro-PnP with the KL divergence loss significantly improves the metric (+13.84), outperforming CDPN-Full by a clear margin (65.88 *vs.* 63.21).
- The regularization loss proposed in Eq. (10) further elevates the performance (+1.88).

- Strong improvement (+5.46) is seen when initialized from A1, because CDPN has been trained with the extra ground truth of object masks, providing a good initial state highlighting the foreground.
- Finally, the performance benefits (+0.97) from more training epochs (160 ep. from A1 + 320 ep.) as equivalent to CDPN-Full [24] (3 stages × 160 ep.).

The results clearly demonstrate that EPro-PnP can unleash the enormous potential of the classical PnP approach, without any fancy network design or decoupling tricks.

Comparison to the State of the Art As shown in Table 2, despite modified from the lower baseline, EPro-PnP easily reaches comparable performance to the top pose refiner RePOSE [20], which adds extra overhead to the PnP-based initial estimator PVNet [31]. Among all these entries, EPro-PnP is the most straightforward as it simply solves the PnP problem itself, without refinement network [20, 46], disentangled translation [24,39], or multiple representations [35].

Comparison to Implicit Differentiation and Reprojection Learning As shown in Table 3, when the coordinate regression loss is removed, both implicit differentiation and reprojection loss fail to learn the pose properly. Yet EPro-PnP manages to learn the coordinates from scratch, even outperforming CDPN without translation head (79.46 *vs.* 74.54). This validates that EPro-PnP can be used as a general pose estimator without relying on geometric prior.

Uncertainty and Discrimination In Table 3, *Reprojection vs. Monte Carlo* loss can be interpreted as uncertainty alone *vs.* uncertainty-discrimination balanced. The results reveal that uncertainty alone exhibits strong performance when intermediate coordinate supervision is available, while discrimination is the key element for learning correspondences from scratch.

Contribution of End-to-End Weight/Coordinate Learning As shown in Table 1, detaching the weights from the end-to-end loss has a stronger impact to the performance than detaching the coordinates (−8.69 *vs.* −3.08), stressing the importance of attention-like end-to-end weight learning.

On the Importance of the Softmax Layer Learning the corresponding weights without the normalization denominator of spatial Softmax (so it becomes exponential activation as in [11]) does not converge, as listed in Table 1.

5.4. Results on the nuScenes Benchmark

We evaluate 3 variants of EPro-PnP: (a) the basic approach that learns deformable correspondences without geometric prior (enhanced with regularization), (b) adding coordinate regression loss with sparse ground truth extracted from the available LiDAR points as in [11], (c) further adding test-time flip augmentation (TTA) for fair comparison against [41,42]. All results on the validation/test sets are presented in Table 4 with comparison to other approaches.

ID	Method	ADD(-S)			Mean
		0.02d	0.05d	0.1d	
A0	CDPN-Full [24]	29.10	69.50	91.03	63.21
A1	CDPN w/o trans. head	15.93	46.79	74.54	45.75 (−17.46)
A2	+ Batch=32, LM solver	21.17	55.00	79.96	52.04 (+ 6.29)
B0	Basic EPro-PnP	32.14	72.83	92.66	65.88 (+13.84)
B1	+ Regularize derivatives	35.44	74.41	93.43	67.76 (+ 1.88)
B2	+ Initialize from A1	42.92	80.98	95.76	73.22 (+ 5.46)
B3	+ Long sched. (320 ep.)	44.81	81.96	95.80	74.19 (+ 0.97)
C0	B0 → Detach coords.	29.57	68.61	90.23	62.80 (− 3.08)
C1	B0 → Detach weights	22.99	61.31	87.27	57.19 (− 8.69)
D0	B0 → No Softmax denom.	divergence			

Table 1. **Comparison to the CDPN baseline with Ablation Studies.** Results of CDPN are reproduced with the official code.[4] In C0/C1 either component is detached individually from the KL loss, while adding a surrogate mask regression loss [24] in C1.

Method	2°, 2 cm	5°, 5 cm	ADD(-S)		
			0.02d	0.05d	0.1d
CDPN [24]	-	94.31	-	-	89.86
HybridPose [35]	-	-	-	-	91.3
GDRNet* [39]	67.1	-	35.6	76.0	93.6
DPOD [46]	-	-	-	-	95.15
PVNet-RePOSE [20]	-	-	-	-	96.1
EPro-PnP	80.99	98.54	44.81	81.96	95.80

Table 2. **Comparison to the state-of-the-art geometric methods.** BPnP [10] is not included as it adopts a different train/test split. *Although GDRNet [39] only reports the performance in its ablation section, it is still a fair comparison to our method, since both use the same baseline (CDPN).

Main Loss	Coord. Regr.	2°	2 cm	2°, 2 cm	ADD(-S) 0.1d
Implicit diff. [10]		divergence			
Reprojection [11]		0.32	42.30	0.16	14.56
Monte Carlo (ours)		44.18	81.55	40.96	79.46
Implicit diff. [10]	✓	56.13	91.13	53.33	88.74
Reprojection [11]	✓	62.79	92.91	60.65	92.04
Monte Carlo (ours)	✓	65.75	93.90	63.80	92.66

Table 3. **Comparison between loss functions** by experiments conducted on the same dense correspondence network. For implicit differentiation, we minimize the distance metric of pose in Eq. (10) instead of the reprojection-metric pose loss in BPnP [10].

From the validation results it can be observed that:

- The basic EPro-PnP significantly outperforms the FCOS3D [41] baseline (NDS 0.425 *vs.* 0.372). Although it partially benefits from more parameters from the correspondence head, there is still good evidence that: with a proper end-to-end pipeline, PnP can outperform direct pose prediction on a large scale of data.

[4] https://git.io/JXZv6

Method	Data	NDS	mAP	True positive metrics (lower is better)				
				mATE	mASE	mAOE	mAVE	mAAE
CenterNet [47]	Val	0.328	0.306	0.716	0.264	0.609	1.426	0.658
FCOS3D [41]	Val	0.372	0.295	0.806	0.268	0.511	1.315	0.170
FCOS3D§† [41]	Val	0.415	0.343	0.725	0.263	0.422	1.292	**0.153**
PGD§ [42]	Val	0.422	**0.361**	0.694	0.265	0.442	1.255	0.185
Basic EPro-PnP	Val	0.425	0.349	0.676	0.263	0.363	1.035	0.196
+ coord. regr.	Val	0.430	0.352	0.667	0.258	0.337	1.031	0.193
+ TTA§	Val	**0.439**	**0.361**	**0.653**	**0.255**	**0.319**	**1.008**	0.193
MonoDIS [34]	Test	0.384	0.304	0.738	0.263	0.546	1.553	0.134
CenterNet [47]	Test	0.400	0.338	0.658	0.255	0.629	1.629	0.142
FCOS3D§† [41]	Test	0.428	0.358	0.690	0.249	0.452	1.434	**0.124**
PGD§ [42]	Test	0.448	**0.386**	0.626	0.245	0.451	1.509	0.127
EPro-PnP§	Test	**0.453**	0.373	**0.605**	**0.243**	**0.359**	**1.067**	**0.124**

Table 4. **3D object detection results** on the nuScenes benchmark. Methods with extra pretraining other than ImageNet backbone are not included for comparison. § indicates test-time flip augmentation (TTA). † indicates model ensemble.

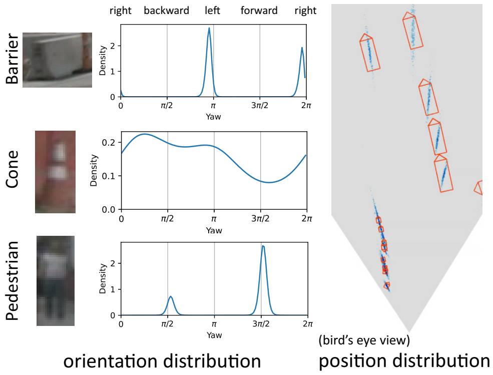

Figure 6. **Visualization of the predicted pose distribution**. The orientation density is clearly multimodal, capturing the pose ambiguity of symmetric objects (*Barrier*, *Cone*) and uncertain observations (*Pedestrian*).

- Regarding the mATE and mAOE metrics that reflect pose accuracy, the basic EPro-PnP already outperforms all previous methods, again demonstrating that EPro-PnP is a better pose estimator. The coordinate regression loss helps further reducing the orientation error (mAOE 0.337 *vs*. 0.363).
- With TTA, EPro-PnP outperforms the state of the art by a clear margin (NDS 0.439 *vs*. 0.422) on the validation set.

On the test data, with the advantage in pose accuracy (mATE and mAOE), EPro-PnP achieves the highest NDS score among other task-specific competitors.

5.5. Qualitative Analysis

As illustrated in Figure 7, the dense weight and coordinate maps learned with EPro-PnP generally capture less details compared to CDPN [24], as a result of higher uncertainty around sharp edges. Surprisingly, even though the learned-from-scratch coordinate maps seem to be a mess, the end-to-end pipeline gains comparable pose accuracy to the CDPN baseline (79.46 *vs*. 79.96). When initialized with pretrained CDPN, EPro-PnP inherits the detailed geometric profile, therefore confining the active weights within the foreground region and achieving the overall best performance. Also note that the weight maps of both derivative regularization and implicit differentiation [10] are more concentrated, biasing towards discrimination over uncertainty.

Figure 6 shows that the flexibility of EPro-PnP allows predicting multimodal distributions with strong expressive power, successfully capturing the orientation ambiguity without discrete multi-bin classification [28, 41] or complicated mixture model [5]. Owing to the ability to model orientation ambiguity, EPro-PnP outperforms other competitors by a wide margin in terms of the AOE metric in Table 4.

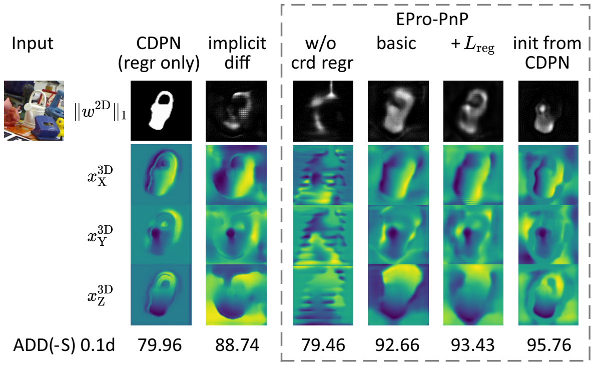

Figure 7. **Visualization of the inferred weight and coordinate maps** on LineMOD test data.

6. Conclusion

This paper proposes the EPro-PnP, which translates the non-differentiable deterministic PnP operation into a differentiable probabilistic layer, empowering end-to-end 2D-3D correspondence learning of unprecedented flexibility. The connections to previous work [4, 7, 10, 11] have been thoroughly discussed with theoretical and experimental proofs. For application, EPro-PnP can inspire novel solutions such as the deformable correspondence, or it can be simply integrated into existing PnP-based networks. Beyond the PnP problem, the underlying principles are theoretically generalizable to other learning models with nested optimization layer, known as declarative networks [16].

Acknowledgements This research was supported by Alibaba Group through Alibaba Research Intern Program, the National Natural Science Foundation of China (No. 52002285), the Shanghai Pujiang Program (No. 2020PJD075), the Natural Science Foundation of Shanghai (No. 21ZR1467400), and the Perspective Study Funding of Nanchang Automotive Institute of Intelligence & New Energy, Tongji University (TPD-TC202110-03).

References

[1] Christopher M. Bishop. Mixture density networks, 1994. 2

[2] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac - differentiable ransac for camera localization. In *CVPR*, 2017. 1, 2

[3] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and carsten Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *CVPR*, 2016. 2, 6

[4] Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3d surface regression. In *CVPR*, 2018. 1, 2, 3, 8

[5] Mai Bui, Tolga Birdal, Haowen Deng, Shadi Albarqouni, Leonidas Guibas, Slobodan Ilic, and Nassir Navab. 6d camera relocalization in ambiguous scenes via continuous multimodal inference. In *ECCV*, 2020. 2, 8

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1, 5, 6

[7] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In *ECCV*, 2020. 1, 2, 3, 5, 8

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[9] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *CVPR*, 2017. 1, 2

[10] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-end learnable geometric vision by backpropagating pnp optimization. In *CVPR*, 2020. 1, 2, 3, 5, 7, 8

[11] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *CVPR*, 2021. 2, 3, 4, 5, 6, 7, 8

[12] Jean-Marie Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2012. 3

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *CVPR*, 2017. 6

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1

[15] Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep orientation uncertainty learning based on a bingham loss. In *ICLR*, 2020. 2

[16] Stephen Gould, Richard Hartley, and Dylan John Campbell. Deep declarative networks. *IEEE TPAMI*, 2021. 3, 8

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[18] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, 2019. 2

[19] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*, 2011. 1, 6

[20] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M. Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *ICCV*, 2021. 1, 2, 5, 7

[21] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017. 2

[22] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[23] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *ECCV*, 2020. 1, 2

[24] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *ICCV*, 2019. 1, 2, 5, 6, 7, 8

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[26] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *CVPR*, 2019. 2

[27] Fabian Manhardt, Diego Martín Arroyo, Christian Rupprecht, Benjamin Busam, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. In *ICCV*, 2019. 2, 3

[28] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 8

[29] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, 2021. 1

[30] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*, 2019. 1, 2

[31] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 1, 2, 4, 7

[32] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *ICCV*, 2017. 1, 2

[33] Gerald Schweighofer and Axel Pinz. Robust pose estimation from a planar target. *IEEE TPAMI*, 28(12):2024–2030, 2006. 2, 3

[34] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *ICCV*, 2019. 8

[35] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *CVPR*, 2020. 1, 7

[36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *CVPR*, 2019. 5

[37] David E. Tyler. Statistical analysis for the angular central gaussian distribution on the sphere. *Biometrika*, 74(3):579–589, 1987. 4

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 5

[39] 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. 7

[40] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019. 1, 2

[41] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3d object detection. In *ICCV Workshops*, 2021. 5, 6, 7, 8

[42] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning (CoRL)*, 2021. 1, 7, 8

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2

[44] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning (CoRL)*, 2021. 1

[45] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *CVPR*, 2020. 2

[46] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *ICCV*, 2019. 1, 2, 7

[47] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019. 8

[48] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2, 5