Particle-based 6D Object Pose Estimation from Point Clouds using Diffusion Models

Christian Möller*
TU Darmstadt
christian.moller@abo.fi

Niklas Funk*
TU Darmstadt
niklas@robot-learning.de

Jan Peters
TU Darmstadt
jan@robot-learning.de

Abstract

Object pose estimation from a single view remains a challenging problem. In particular, partial observability, occlusions, and object symmetries eventually result in pose ambiguity. To account for this multimodality, this work proposes training a diffusion-based generative model for 6D object pose estimation. During inference, the trained generative model allows for sampling multiple particles, i.e., pose hypotheses. To distill this information into a single pose estimate, we propose two novel and effective pose selection strategies that do not require any additional training or computationally intensive operations. Moreover, while many existing methods for pose estimation primarily focus on the image domain and only incorporate depth information for final pose refinement, our model solely operates on point cloud data. The model leverages recent advancements in point cloud processing through an $\mathrm{SE}(3)$*-equivariant latent space that forms the basis for the selection strategies and improved inference times. Experimental results demonstrate the effectiveness of our design choices and competitive performance on the Linemod dataset.*

1. Introduction

Object pose estimation is a fundamental problem in many applications, including Robotics [18, 25], Autonomous Driving [6], and Virtual Reality [14]. Despite significant advances in recent years, mainly attributed to learning-based methods, the task remains challenging [9], especially when only a single scene view is available. A single perspective might hide distinct object characteristics, and partial observability leads to occlusion that intensifies with the level of clutter [10, 28]. Partial observability, occlusion, and symmetric objects eventually result in pose ambiguity and multiple pose hypotheses fitting the observation. To deal with this ambiguity, this work explores leveraging diffusion models for particle-based object pose estimation. Generative models [12, 21, 23, 24] have been shown to excel in learning multi-modal distributions and, therefore, hold promise in addressing the aforementioned challenges. Additionally, many existing methods for 6D object pose estimation primarily operate in the image domain [16, 20, 28] and only incorporate depth in the final refinement [13]. Herein, we directly work in the 3D point cloud domain, aligning the inherent three-dimensional nature of both the scene and its objects. Moreover, recent advancements in feature extraction from point clouds have opened up exciting possibilities for enhancing pose estimation from point clouds [4, 17, 27]. In particular, we exploit equivariant feature spaces to obtain more expressive encodings and improved inference times. Lastly, while particle-based approaches to pose estimation naturally yield multiple pose hypotheses, it is also crucial to rank the individual estimates and come up with a final pose estimate. To this end, we present and compare two novel, simple, but effective particle selection strategies.

This work therefore contributes a novel, particle-based approach for **(a) 6D Pose Estimation from Point Clouds.** Through leveraging information about the object's 3D model, the inference process of the diffusion-based generative model iteratively aligns the model with the observation. Moreover, the underlying generative model naturally captures the multimodality that arises from partial observability. This work also demonstrates effectiveness of **(b) utilizing** $\mathrm{SE}(3)$**-equivariant Vector Neurons.** We leverage vector neurons (VNs) to build a latent space that is equivariant to $\mathrm{SE}(3)$ transformations and yields expressive point cloud encodings. This property enables a substantial inference time improvement with small sacrifices in accuracy. Lastly, this work introduces **(c) Novel Pose Selection Methods** for deciding upon a single 6D pose from multiple pose hypotheses generated with the model. Both strategies show high success rates w.r.t. selecting particles resulting in an accurate pose estimation, are computationally efficient, and do not necessitate additional training. We evaluate our approach on the Linemod dataset [8], demonstrate its competitive performance, and the effectiveness of its individual components.

* Equal Contribution.

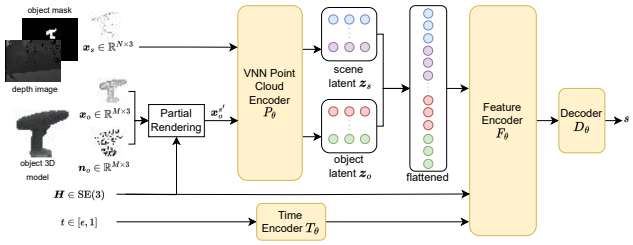

Figure 1. Noise Conditioned Score Model (NCSM) architecture for learning 6D pose distributions. The object point cloud $\boldsymbol{x}_o$ is sampled from a 3D model of the object whose pose we want to determine in the scene. Given a specific pose $\boldsymbol{H} \in \mathrm{SE}(3)$ and its normal vectors $\boldsymbol{n}_o$ the object point cloud is partially rendered. Scene point cloud $\boldsymbol{x}_s$ and partial rendered object point cloud $\boldsymbol{x}_o^{s'}$ are embedded through a shared encoder that leverages vector neurons (VNs). Finally, the scene latent $\boldsymbol{z}_s$ and object latent $\boldsymbol{z}_o$ are flattened and together with an encoded time step and the respective pose $\boldsymbol{H}$ fed through a Feature Encoder F_θ and Decoder D_θ to predict a score $\boldsymbol{s} \in \mathbb{R}^6$.

2. Method

We propose learning a point-cloud-based diffusion model for 6D pose estimation. Herein, a pose $\boldsymbol{H}$ is represented as an element of the group $\mathrm{SE}(3)$, with rotation $\boldsymbol{R} \in \mathrm{SO}(3)$, and translation $\boldsymbol{t} \in \mathbb{R}^3$. As observation, we consider a single depth image of the scene. Additionally, we assume to have access to the 3D object model.

2.1. Model Architecture

Our proposed model architecture is shown in Fig. 1. The network receives four inputs: (i) The depth image of a partial scene view together with a segmentation mask which is converted to point-cloud form $\boldsymbol{x}_s \in \mathbb{R}^{N\times 3}$. (ii) A current pose hypothesis $\boldsymbol{H} \in \mathrm{SE}(3)$, which, is to be refined based on the model's output. (iii) The time step $t \in [0, 1]$ that governs the diffusion process. (iv) A set of points sampled from the fully observable 3D object model $\boldsymbol{x}_o \in \mathbb{R}^{M\times 3}$ along with their normal vectors $\boldsymbol{n}_o \in \mathbb{R}^{M\times 3}$. Importantly, the object model's point cloud is pre-processed by a partial rendering module [15] to account for the fact that the scene is only partially observable. Both point clouds, i.e., the scene point cloud and the object point cloud (after passing the partial rendering module) are processed individually by a shared vector neural network (VNN) encoder to obtain the respective $\mathrm{SE}(3)$-equivariant latent representations $\boldsymbol{z}_s \in \mathbb{R}^{D\times 3}$ and $\boldsymbol{z}_o \in \mathbb{R}^{D\times 3}$ with dimensions D. Subsequently, the latents are flattened and concatenated with the rigid transformation $\boldsymbol{H}$, and the time encoding [25] and passed to a Feature Encoder F_θ and Decoder D_θ to predict the score $\boldsymbol{s} \in \mathbb{R}^6$ for updating the current pose estimate. For more information, see [15].

2.2. Training and Inference

We employ a diffusion-based approach for pose estimation and leverage a stochastic differentiable equation (SDE) for the diffusion process. Through exploiting the proposed $\mathrm{SE}(3)$-equivariant latent space, we additionally present a more time-efficient latent-space inference process.

Diffusion. The object poses $\boldsymbol{H}$ are diffused along infinite noise scales [24]. In our case, the SDE $dx{=}\sigma^t dw, t{\in}[0, 1]$ governs the diffusion process, with w being the standard Wiener process [24]. Practically, we sample from the perturbed distribution $q_t(\hat{\boldsymbol{H}})$ through first sampling a pose from the data distribution $\boldsymbol{H} \sim p_{\text{data}}(\boldsymbol{H})$ and composing it with a white noise vector $\boldsymbol{\epsilon} \in \mathbb{R}^6$ sampled from the respective noise scale distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \text{var}(\sigma, t)\boldsymbol{I}_6)$ [25]. Thus, the noise perturbed pose equates to $\hat{\boldsymbol{H}} = \boldsymbol{H}\text{Expmap}(\boldsymbol{\epsilon})$. We therefore follow [19, 25] in that we work within the vector space $\mathbb{R}^6$ instead of the Lie algebra. For moving the elements between Lie Group and vector space, we rely on the logarithmic and exponential maps, i.e., Logmap : SE(3) $\to \mathbb{R}^6$ and Expmap : $\mathbb{R}^6 \to$ SE(3) respectively [19].

Training. In line with [25], our proposed model is trained using denoising score matching (DSM) with loss term

$$\mathcal{L} = \mathbb{E}_{t\in[\epsilon,1]}\mathbb{E}_{q_t(\boldsymbol{H},\hat{\boldsymbol{H}})}\left[\left\| s_\theta(\hat{\boldsymbol{H}}, t) - \nabla_{\hat{\boldsymbol{H}}}\log(q_t(\hat{\boldsymbol{H}}|\boldsymbol{H}))\right\|_2^2\right], \quad (1)$$

with smallest time step ($\epsilon = 1e^{-5}$ in our case). We follow [22] and rescale the output of the NCSM by $1/\sqrt{\text{var}(\sigma, t)}$.

Inference. For inferring an object pose estimate, we deploy an inverse Langevin dynamics process, starting from the largest time step $t{=}1$ and a random initial pose $\boldsymbol{H}_0 = \text{Expmap}(\boldsymbol{\epsilon})$ sampled from $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \text{var}(\sigma, 1)\boldsymbol{I}_6)$. Then follows a sequence of L equally spaced time steps $(t_i)_{i=0}^{L-1}, t_i = (1 - (i/(L-1))(1-\epsilon))$ starting at $t_0{=}1$ and ending at $t_{L-1}{=}\epsilon$. At each time step we update the pose following the Langevin dynamics process $\boldsymbol{H}_{l+1} = \text{Expmap}(\alpha_l s_\theta(\boldsymbol{H}_l, t_l) + \sqrt{2\alpha_l}0.01\boldsymbol{\epsilon})\boldsymbol{H}_l$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_6)$, the score estimate by our NCSM $s_\theta(\boldsymbol{H}_l, t_l)$ and dynamic step size α_l. In the final five iterations of the inference process, this term is abolished leaving $\boldsymbol{H}_{l+1} = \text{Expmap}(\alpha_l s_\theta(\boldsymbol{H}_l, t_l))\boldsymbol{H}_l$.

Observation-encoding during Inference. As shown in Figure 1, our proposed architecture encodes the scene and object point cloud individually. Since the scene point cloud $\boldsymbol{x}_s$ does not change during inference, we only need to encode it once, i.e., $\boldsymbol{z}_s = P_\theta(\boldsymbol{x}_s)$. However, the object point cloud and this its encoding change in each iteration due to an update in the predicted pose. To facilitate a trade-off between accuracy and inference speed we introduce a hyperparameter k, which determines that the object point cloud is encoded and re-rendered only in every k^{th} iteration, and in between, we solely rotate the object latent.

2.3. Pose Estimation through Particle Selection

The inference process (cf. Sec. 2.2) can be used to generate multiple samples, i.e., a set of N pose hypotheses (particles) $\mathcal{H} = \{\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_N\}$. To condense these hypotheses to one pose prediction $\boldsymbol{H}^*$, we introduce two techniques.

Selection By Score. The inference process (cf. Sec. 2.2) generates a history of L score values $\{\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{L-1}\} \in \mathbb{R}^{L\times 6}$. As shown in [21], for score matching in Euclidean Spaces, i.e., with $q_t(\hat{\boldsymbol{x}}|\boldsymbol{x}) = \mathcal{N}(\hat{\boldsymbol{x}}|\boldsymbol{x}, \sigma_t)$, the objective for score matching equates to $\nabla_{\hat{\boldsymbol{x}}} \log(q_t(\hat{\boldsymbol{x}}|\boldsymbol{x})) = -(\hat{\boldsymbol{x}} - \boldsymbol{x})^2/\sigma_t^2$. For our case of score matching in SE(3), the Gaussian is defined according to $q_t(\hat{\boldsymbol{H}}|\boldsymbol{H}, \boldsymbol{\Sigma}) \propto \exp\left(-0.5 \left\|\text{Logmap}(\boldsymbol{H}_\mu^{-1}\boldsymbol{H})\right\|^2_{\boldsymbol{\Sigma}^{-1}}\right)$, centered around its mean $\boldsymbol{H} \in SE(3)$ and with covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{6\times 6}$ [3]. Therefore, for score matching in SE(3), the score should match $\nabla_{\hat{\boldsymbol{H}}} \log(q_t(\hat{\boldsymbol{H}}|\boldsymbol{H})) \propto \text{Logmap}(\boldsymbol{H}^{-1}\hat{\boldsymbol{H}})$, which essentially is a distance vector within the vector space $\mathbb{R}^6$. From these derivations, it is clear that the score, and in particular its 2-norm, i.e., $\|\boldsymbol{s}\|_2$ is an indicator of how close the current sample is to a sample from the dataset. Therefore, as a first heuristic to rank the particle-based pose hypotheses, we consider the last score value, as for precise pose estimates it should be smaller.

Selection By Latent. The partially rendered object point cloud and the cropped scene point cloud are encoded with the same point cloud encoder P_θ. Therefore, this strategy follows the geometric intuition, that the pose prediction quality correlates with the proximity of the object latent $\boldsymbol{z}_o \in \mathbb{R}^{D\times 3}$ to the scene latent $\boldsymbol{z}_s \in \mathbb{R}^{D\times 3}$ given a specific pose $\boldsymbol{H} \in \text{SE}(3)$. The output of the encoder are D 3-dimensional points, which we will compare in terms of euclidean distance. In particular, the scene latent consists of the vectors $\boldsymbol{z}_s = \{{}^{(0)}\boldsymbol{z}_s, {}^{(1)}\boldsymbol{z}_s, \cdots, {}^{(D-1)}\boldsymbol{z}_s\}$. The object latent on the other hand varies along the N different particles. To formalize this selection strategy we define a proximity function $\text{lprox}\colon \mathbb{R}^{D\times 3} \to \mathbb{R}$, $\text{lprox}(\boldsymbol{z}_o, \boldsymbol{z}_s) = \sum_{i=0}^{D-1} \left\|{}^{(i)}\boldsymbol{z}_s - {}^{(i)}\boldsymbol{z}_o\right\|_2$ that can be used to determine the proximity of the N pose hypotheses. Under this prediction strategy, the final predicted pose is the particle that minimizes this distance.

3. Experiments

We evaluate our 6D pose estimation method on the Linemod dataset [8], following [1, 2, 11, 15, 20, 26]. Accuracy is measured using ADD and ADD-S for symmetric objects [7] with correctness defined as ADD $< 10\%$ of the object's diameter. Our model has a 282-dimensional latent space ($D = 94$), a Feature Encoder F_θ with 7 fully connected (FC) layers (size 512, 20% dropout), and a Decoder D_θ with 3 FC layers (size 256) outputting the score value $\boldsymbol{s} \in \mathbb{R}^6$. Object point clouds (1024 points) and normal vectors are sampled from 3D face centroids and normals, while scene point clouds (1024 points) are cropped based on the segmentation mask of the visible object part. We train a single model for all 13 objects over 3500 epochs. Training takes 80 hours on an NVIDIA V100 GPU (32GB). Standard inference uses 100 Langevin iterations with 20 sampled particles, rendering object point clouds every iteration ($k = 1$). No synthetic data is used.

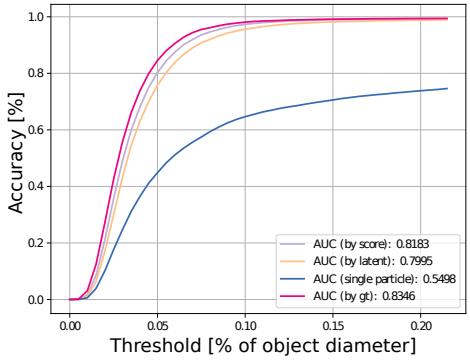

Figure 2. Accuracy Curve with AUC values for all selection methods for a model trained and tested on all objects. Single particle sampling, i.e. eliminating the need for any selection strategy yields the lower bound and selection by ground truth information the upper bound.

Particle Selection Strategies. To evaluate the particle selection strategies (Sec. 2.3), we consider two additional baseline strategies: 1) selection by ground truth (gt), and 2) single particle. While the selection by ground truth selects the pose hypothesis (particle) with the lowest average distance metric (ADD) to form an upper bound, the single particle baseline naturally eliminates the need for any particle selection strategy since it corresponds to inferring only a single pose using our proposed model and thus represents a lower bound. The results in Fig. 2 show that both our proposed selection strategies lead to comparable results, although the selection by score slightly outperforms the selection by latent with an accuracy (ACC-0.1) of 97.4% vs. 95.6% and a AUC of 81.8 vs. 80.0. Importantly, they are both much closer to the upper bound than to the lower bound and lead to accurate pose predictions. In ≈72% of the cases, the predicted pose by score is among the top 5 particles (compared to ≈60% for selection by latent). This leads to the conclusion that our particle selection strategies are effective in reliably selecting a pose amongst the best pose candidates and that inferring multiple pose candidates is crucial for obtaining good performance.

Model Performance Comparison. A comparison of our model's performance (selection by score & partial rendering) against other approaches on the Linemod dataset is provided in Tab. 1. It's important to note that our evaluation, as well as the evaluation for CloudAAE [5] was conducted with the advantage of having access to the ground truth segmentation

Approach	PVNet [16]	PoseCNN + DeepIm [13, 28]	DenseFusion [26]	HybridPose [20]	CloudAAE + ICP [5]	**Ours**
Modality	RGB	RGB	RGB-D	RGB	D	D
ape	43.6	77.0	92.3	63.1	92.5	92.3
bench v.	99.9	97.5	93.2	99.9	91.8	99.7
camera	86.9	93.5	94.4	90.4	88.9	94.0
can	95.5	96.5	93.1	98.5	96.4	98.7
cat	79.3	82.1	96.5	89.4	97.5	97.5
driller	96.4	95.0	87.0	98.5	99.0	98.6
duck	52.6	77.7	92.3	65.0	92.7	92.2
eggbox*	99.2	97.1	99.8	100.0	99.8	100.0
glue*	95.7	99.4	100.0	98.8	99.0	99.9
hole p.	81.9	52.8	92.1	89.7	93.7	96.4
iron	98.9	98.3	97.0	100.0	95.9	99.2
lamp	99.3	97.5	95.3	99.5	96.6	98.6
phone	92.4	87.7	92.8	84.9	97.4	98.4
MEAN	86.3	88.6	94.3	91.3	95.5	97.4

Table 1. Evaluation and comparison of our approach (using selection by score) with other state-of-the-art approaches for 6D pose estimation on the Linemod dataset. The reported metric is the ACC-0.1, i.e., accuracy, and colors indicate the three best ranked methods - blue indicates the best, orange the second best and violet the third best. * denotes symmetric objects.

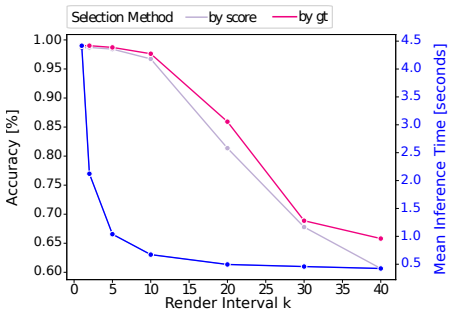

Figure 3. Relation between the rendering interval k, accuracy, and inference runtime. The interval k controls how often the point cloud is rendered and encoded during inference; in other iterations, only the latent vector $\boldsymbol{z}_o$ is transformed. Results are based on 100 iterations, 20 sampled particles, and the driller object.

masks for the objects in the test set. Comapred to CloudAAE, our approach yields an increased mean performance of almost 2.0 percentage points. However, this assumption is not made by the other methods listed in Tab. 1. Keeping this advantage in mind, we surpass the performance of DenseFusion [26] by 3.1 percentage points and the other baselines even more significantly.

Partial vs. Full Object Rendering. One design choice of our approach is to only feed the front facing points into the point cloud encoder to account for partial observability. An alternative would be to use the full transformed object model instead. This experiment, therefore, compares two models. We find that for both particle selection strategies, partial rendering leads to superior results. While for selection by score the difference in accuracy is 4.1 percentage points (97.4% vs. 93.3%), selection by latent exhibits a larger discrepancy of 6.7 percentage points (95.6% vs. 88.9%).

SE(3)- vs. SO(3)-Equivariant Latent. By aligning the cropped scene point cloud through centering before it enters the encoder and then reversing this operation in the latent space, we obtain a $\mathrm{SE}(3)$-equivariant latent representation. Omitting the centering and its reversal yields an only $\mathrm{SO}(3)$-equivariant latent. Deploying an $\mathrm{SE}(3)$-equivariant latent improves the performance for both selection strategies. For selection by score the accuracy drops by 9.2 percentage points when using the $\mathrm{SO}(3)$-equivariant latent (mean accuracy of 97.4% vs. 88.2%). In the case of selection by latent, the performance difference is more substantial. The $\mathrm{SO}(3)$-equivariant latent attains an accuracy of 52.0% compared to 95.6% for its $\mathrm{SE}(3)$-equivariant counterpart. This discrepancy can be explained by the fact that the selection by latent strategy relies on the assumption that the latent representations of the object and scene are comparable.

Inference and Runtime Analysis. The inference process encompasses various hyper-parameters. Here we investigate the influence of the rendering interval k on the pose estimation quality and the inference time. Experiments are conducted on an NVIDIA A100 GPU with 40GB. The first experiment considers the driller object and varying the render interval k between 1 and 40. As shown in Fig. 3, rendering the object point cloud every iteration yields an ACC-0.1 of 98.6% with an inference time of $4.41\,\mathrm{s}$ per pose prediction. Increasing the render interval to 5 and 10 leads to a similar accuracy while reducing the inference time to $1.04\,\mathrm{s}$ (-76%) and $0.67\,\mathrm{s}$ (-85%). Recognizing this favorable trade-off between inference time and accuracy at a rendering interval of $k = 10$, we compare this runtime-efficient setting with the default interval of $k = 1$ across all objects. Across all objects, the higher rendering interval only results in a performance decrease of -3%, while significantly improving the inference time by 85%.

4. Conclusion

This work introduced a novel approach for 6D pose estimation from single-perspective depth images. To account for the fact that partial observability and symmetric objects yield settings in which multiple pose hypotheses might fit the observation well, this work proposed to train a diffusion-based generative model for pose estimation. In terms of model architecture, we incorporated recent advancements in point cloud processing to obtain a $\mathrm{SE}(3)$-equivariant latent space. To decide upon a final pose estimate from multiple hypotheses generated during inference, we introduced two novel pose selection strategies. Our results demonstrated that sampling multiple pose hypotheses and selecting one of them is crucial and significantly outperforms solely inferring a single pose using the trained generative model. Moreover, the experiments underlined the importance of leveraging the $\mathrm{SE}(3)$ equivariant latent space. In the future, it would be interesting to extend our approach from object pose estimation to object pose tracking.

5. Acknowledgments

This work has received funding from the EU's Horizon Europe project ARISE (Grant no.: 101135959), and the AICO grant by the Nexplore/Hochtief Collaboration with TU Darmstadt. The authors also gratefully acknowledge the computing time provided to them on the high-performance computer Lichtenberg II at TU Darmstadt, funded by the German Federal Ministry of Education and Research (BMBF) and the State of Hesse.

References

[1] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372, 2016. 3

[2] Yannick Bukschat and Marcus Vetter. Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach, 2020. 3

[3] Gregory Chirikjian and Marin Kobilarov. Gaussian approximation of non-linear measurement models on lie groups. In *IEEE Conference on Decision and Control*, 2014. 3

[4] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12180–12189, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 1

[5] Ge Gao, Mikko Lauri, Xiaolin Hu, Jianwei Zhang, and Simone Frintrop. Cloudaae: Learning 6d object pose regression with on-line data synthesis on point clouds. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11081–11087. IEEE, 2021. 3, 4

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1

[7] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision – ACCV 2012*, pages 548–562, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 3

[8] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. 1, 3

[9] Sabera Hoque, Md Yasir Arafat, Shuxiang Xu, Ananda Maiti, and Yuchen Wei. A comprehensive review on 3d object detection and 6d pose estimation with deep learning. *IEEE Access*, 9:143746–143770, 2021. 1

[10] Tsu-Ching Hsiao, Hao-Wei Chen, Hsuan-Kung Yang, and Chun-Yi Lee. Confronting ambiguity in 6d object pose estimation via score-based diffusion on se(3), 2023. 1

[11] Jia Kang, Wenjun Liu, Wenzhe Tu, and Lu Yang. Yolo-6d+: Single shot 6d pose estimation using privileged silhouette information. In *2020 International Conference on Image Processing and Robotics (ICIP)*, pages 1–6, 2020. 3

[12] Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. *A tutorial on energy-based learning*. MIT Press, 2006. 1

[13] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. *International Journal of Computer Vision*, 128(3):657–678, 2020. 1, 4

[14] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: A hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, 22 (12):2633–2651, 2016. 1

[15] Christian Möller, Niklas Funk, and Jan Peters. Particle-based 6d object pose estimation from point clouds using diffusion models, 2024. 2, 3

[16] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4556–4565, 2019. 1, 4

[17] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016. 1

[18] Anthony Simeonov, Ankit Goyal, Lucas Manuelli, Lin Yen-Chen, Alina Sarmiento, Alberto Rodriguez, Pulkit Agrawal, and Dieter Fox. Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement. *Conference on Robot Learning*, 2023. 1

[19] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018. 2

[20] C. Song, J. Song, and Q. Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 428–437, Los Alamitos, CA, USA, 2020. IEEE Computer Society. 1, 3, 4

[21] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 1, 3

[22] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 2

[23] Yang Song and Diederik P. Kingma. How to train your energy-based models, 2021. 1

[24] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 2

[25] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se(3)-diffusionfields: Learning smooth cost functions

for joint grasp and motion optimization through diffusion. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 1, 2

[26] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3338–3347, Los Alamitos, CA, USA, 2019. IEEE Computer Society. 3, 4

[27] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5), 2019. 1

[28] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 1, 4