Democratising 2D Sketch to 3D Shape Retrieval Through Pivoting Supplemental Material

Pinaki Nath Chowdhury[1,2] Ayan Kumar Bhunia[1] Aneeshan Sain[1,2] Subhadeep Koley[1,2]
Tao Xiang[1,2] Yi-Zhe Song[1,2]
[1]SketchX, CVSSP, University of Surrey, United Kingdom.
[2]iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.
{p.chowdhury, a.bhunia, a.sain, s.koley, t.xiang, y.song}@surrey.ac.uk

A. Discussion on View Variance and Invariance

Naively mapping 2D sketches to 3D shapes is an ill-posed problem due to the *view variance* problem – there exists multiple drastically different 2D sketches drawn from different viewpoints for the same 3D shape. Prior literature [31] circumvents this problem by converting the retrieval task to the problem of matching a sketch to multiple ($n \geq 24$) 2D projections of a 3D shape, i.e., multi-view matching problem. In particular, an average representation from multiple 2D projections is computed using max-pool [27], triplet-center loss [17], Wasserstein barycenters [30], or attention-based pooling [23]. The objective is to compute a *view invariance* matching loss between 2D sketches and 3D shapes which helps overcome the *view variance* problem. However, forcing *view invariance* limits to only category-level sketch – shape matching [29, 11] thereby loosing crucial fine-grained visual cues depicted in sketches.

In this paper, we address the *view variance* problem without making unwanted assumption of using *view invariant* representations. In particular, we employ the Blind Perspective-n-Points algorithm [2, 3, 4, 6, 8] to solve for pose (rotation, translation) and construct 2D-3D correspondences. This helps "lift" [16] our 2D sketch to 3D space instead of lowering a 3D shape to 2D space by computing its *view invariant* representation.

B. Background of BPnP

The Blind Persoective-n-Points (BPnP) algorithm aims to solve the camera pose from a set of unordered 3D points in object space and their corresponding 2D points in image space. Specifically, camera pose $y = \{R, t\}$ is composed of rotation $R \in \mathbb{R}^{3\times3}$ and translation $t \in \mathbb{R}^{3\times1}$ that aligns the 3D points with 2D points, without knowledge of the true 2D-3D correspondence. This makes solving the BPnP problem quite challenging as the search space of correspondence and camera poses is significantly large with the non-convex objective function having many local optima, and outliers. BPnP is fundamental in several applications like computer vision, robotics, augmented reality, and visual localisation. Given an image $I_{\mathbf{g}}$, we compute its feature representation using a 2D encoder as, $f_{I_{\mathbf{g}}} = F_{2D}(I_{\mathbf{g}})$. Next, we use a dense correspondence network to predict a set of 3D points $\mathbf{z} = \{z_1, z_2, \ldots, z_N\}$ and weights $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ from sampled 2D points $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$, as $\{\mathbf{z}, \mathbf{w}\} = F_{dc}(f_{I_{\mathbf{g}}}, \mathbf{x})$ in Eq. 7.

For an optimal pose $y = \{R, t\}$, BPnP [4] solves for the second layer of optimisation that minimises the cumulative squared weighted re-projection error given by Eq. 8 as,

$$y_{pred} = \arg\min_{y} \frac{1}{2} \sum_{i=1}^{N} || \underbrace{w_i \circ (\pi(Rz_i + t) - x_i)}_{\Phi_i(y)} ||^2$$

However, Eq. 8 formulates a non-linear least squares problem that may have non-unique solutions, i.e., pose ambiguity. Hence, instead of backpropagating through unstable local solutions, we take an alternative approach to model the BPnP [4] output as a distribution of poses followed by computing KL-divergence with ground-truth target distribution $p(y_{gt})$. We choose this target distribution as a narrow Dirac delta distribution [8] to get unique solutions for pose using Eq. 11, where each integral (re-projection at GT pose, re-projection at predicted pose) follows a canonical solution.

C. Why non-unique in Eq. (8)?

To recap, the re-projection error is defined in Eq. 8 as,

$$y_{pred} = \arg\min_{y} \frac{1}{2} \sum_{i=1}^{N} || \underbrace{w_i \circ (\pi(Rz_i + t) - x_i)}_{\Phi_i(y)} ||^2$$

Existing works on BPnP [4, 6] derive a single solution of a particular solver $y^* = PnP(\theta)$, where $\theta = \{\mathbf{z}, \mathbf{x}, \mathbf{w}\}$ via implicit differentiation [15]. This is essentially the Laplace method that approximates the posterior by $\mathcal{N}(y^*, \sum_{y^*})$, where both y^* and $\sum_{y^*}$ can be estimated by the BPnP solver with analytical derivatives [7]. Simplifying $\sum_{y^*}$ to be homogeneous (i.e., ignoring the variance of projection

error $\mathcal{L}_{var}$ in Eq. 12), the approximated KL divergence in Eq. 11 can be simplified to the L2 loss as,

$$\mathcal{L}_{KL} = \frac{1}{2}\sum_{i=1}^{N} ||\Phi_i(y_{gt})||^2 \tag{15}$$

This leads to non-unique solutions since Laplace approximation (Levenberg-Marquardt BPnP solver) only guarantees local convergence. Hence, non-normal posteriors (with ambiguous multiple modes) leads to inaccuracies when using LM solvers for global convergence.

D. Domain Adaptation versus Pivoting

Since the goal of pivoting is to bridge the domain gap between 2D sketches and 3D shapes, it is important to understand the subtle difference between pivoting and Domain Adaptation. Using DA instead of pivoting is challenging when the target domain is significantly different from the source domain [12] e.g., sparse 2D sketch and 3D shapes. Nevertheless, we compare pivoting with alternative sub-space mapping like (i) Gradient Reversal Layer (GRL) [1] that allows features to be indistinguishable between 2D sketch and 3D shapes, aligning the two. Replacing pivoting with GRL drops Acc.@1/Acc.@5 for 'Chairs' in Qi *et al*. [23] to 32.73/65.13. Using kernel-based methods for like Optimal Transport [10], and MMD [28] to minimise the distribution gap between 2D sketch and 3D shape features drops Acc.@1/Acc.@5 to 30.56/64.92 and 27.72/63.45 respectively. Instead of directly aligning 2D sketches with 3D shapes, as in naive DA, pivoting takes a cascaded alignment process using a third (shared) domain as pivot: (i) it aligns 2D sketch with 2D photos via training with triplet loss ($\mathcal{L}^{\mathbf{src}\rightarrow\mathbf{piv}}$), (ii) aligns 2D rendered photos and its 3D shapes via triplet loss ($\mathcal{L}^{\mathbf{piv}\rightarrow\mathbf{trg}}$), and (iii) aligns the two metric spaces ($\mathcal{L}^{\mathbf{src}\rightarrow\mathbf{piv}}$, $\mathcal{L}^{\mathbf{piv}\rightarrow\mathbf{trg}}$) via KL-divergence term ($\mathcal{L}_{dist}$) facilitating $\mathbf{src} \rightarrow \mathbf{trg}$, i.e., 2D sketch to 3D shape retrieval.

E. Why we report upper-bound/all-shot?

Our goal is *zero-shot setup* for fine-grained sketch-based shape retrieval that overcomes the ill-posed problem of collecting paired 2D sketches for 3D shapes. However, to inform and encourage future research, it is important to know the upper-bound performance when using paired 2D sketches and 3D shapes as training data. Hence, we report the upper-bound/all-shot performance in the experimental sections. While our proposed pivoting + lifting can reach competitive performance to the fully supervised counterparts, we hope future works will close the remaining performance gap between Ours (zero-shot) and upper-bound/all-shot methods.

F. PyTorch-like pseudo-code for training.

Algorithm 1: PyTorch code to solve $\mathcal{L}_{reg}$

```
import math.log as log
import torch

# Download ops/pnp/* and models/* from
 github.com/tjiiv-cprg/EPro-PnP

from camera import PerspectiveCamera
from cost_fun import AdaptiveHuberPnPCost
from epropnp import EProPnP6DoF
from levenberg_marquardt import LMSolver
import MonteCarloPoseLoss

def compute_loss(f_Ig, x, y_gt, **kwargs):
    # f_Ig:  Tensor of shape [nbatch, d]
    # x:  Tensor of shape [nbatch, K, 2]
    # y_gt:  Tensor of shape [nbatch, 7]

    # Predict dense correspondence in Eq.7
    z, w, scale = F_dc(f_Ig, x)
    w = (w - w.mean(dim=1) - log(K))
    w = w.exp()*scale

    # Set Camera and PnP Loss parameters
    cam = PerspectiveCamera(**kwargs)
    mcposefn = MonteCarloPoseLoss(**kwargs)
    costfn = AdaptiveHuberPnPCost(**kwargs)
    costfn.set_param(x, z)
    pnpobj = EProPnP6DoF(
        mc_samples=512,
        num_iter=4,
        solver=LMSolver(dof=6, num_iter=5)

    # Use AIMS and LM algorithm, Eq.12,13
    _, _, y, _, logw, ctgt =
        pnpobj.monte_carlo_forward(
            z, x, w, cam, pose_init=y_gt)

    # Get losses
    scaleD = scale.detach().mean()
    loss_mc = mcposefn(logw, ctgt, scaleD)

    lt = (y[:, :3] - y_gt[:, :3]).norm(2, -1)
    beta = 0.05
    loss_t = torch.where(lt<beta,
     0.5*lt.square()/beta, lt-0.5*beta)
    loss_t = loss_t.mean()

    quat = y[:, None, 3:]  @ y_gt[:, 3:, None]
    quat = quat.squeeze(-1).squeeze(-1)
    loss_r = ((1 - quat.square())*2).mean()

    L_reg = loss_mc + 0.1*(loss_r + loss_t)
    return L_reg
```

G. Performance on 'Lamps' is lower than 'Chairs'

While our proposed method (pivoting + lifting) reaches competitive performance for 'Chairs' category (Tab. 1, it is comparatively lower for 'Lamps'. This is because the training data [32] used in the pivoting ($\mathbf{src} \rightarrow \mathbf{piv}$) step only has 'Chairs' category and not 'Lamps'. Achieving cross-

category generalisation (trained on 'Chairs' and evaluated on 'Lamps') for fine-grained retrieval is still an open problem for multiple research fields like FG-SBIR (2D sketch to 2D photos) and ours FG-SBSR (2D sketch to 3D shape). Future work exploiting foundation models (having open-set generalisation [21]) like CLIP [25] can help improve performance fine-grained retrieval performance on both 'Chairs' and 'Lamps'.

H. Evaluation on unseen 3D 'Chairs' sub-categories

Although our zero-shot setup gives competitive performance for FG-SBSR without using paired 2D sketch and 3D shapes, we further investigate if the proposed method generalises to unseen 'Chairs' sub-categories. Accordingly, we manually remove 5 'Chairs' sub-categories from the training set of pivoting [32] + lifting [5] and report their Acc.@1 as: armchair (55.01), X-chair (55.51), ladder-back (55.67), bean chair (54.97), lawn-chair (55.84). This is comparable to the zero-shot Acc.@1 in Tab. 1 of 55.79. This shows that our proposed (pivoting + lifting) can generalise to unseen 3D 'Chairs' sub-categories.

I. Lifting versus directly regress 3D coordinates

While directly regressing the 3D coordinates to learn 2D-3D correspondence might seem like an alternative, it does not leverage the geometric priors [8]. Introducing the applications of the geometry-based Blind Perspective-n-Points algorithm [4, 3] for fine-grained retrieval shape retrieval helps to have a stable generalisation [8, 6]. We additionally compare directly regressing 3D coordinates/shapes using baselines like SDFSketch and PSGNSketch in Tab. 1,2.

J. Summarising our Contributions

While our utmost contribution lie with democratising 2D sketch to 3D shape fine-grained retrieval, as a new problem setup – this was not possible before due to a lack of large-scale datasets, and the large domain gap between 2D amateur sketches and geometrically well-defined 3D shapes. The neat bit is this was all achieved via a clever use of pivoting. We however still needed to (i) extend pivoting to complex multi-modal setup with $\mathbf{src}$ (2D sketch), $\mathbf{piv}$ (2D photo), and $\mathbf{trg}$ (3D shape) all from different modalities. Also, neural machine translation (NMT) literature [9] used pivoting in generative tasks whereas we adapt for discriminative tasks; and (ii) reformulate BPnP, a technique as an auxiliary task to inject 3D aware knowledge in 2D encoder (sketch, and photo), thereby solving for data scarcity. In contrast, prior works limited BPnP to pose prediction [6] and 3D object detection [8].

K. Clarification on Lifting and Fig. 6

Our lifting loss ($\mathcal{L}_{reg}$) is only used as an auxiliary loss during training and not during inference, i.e., FG-SBSR. Although not our primary goal (fine-grained retrieval), to verify that our 2D sketch and 3D shape latent space are indeed aligned and 3D aware, we additionally pass the encoded sketch feature $f_{\mathbf{s}}$ through the dense prediction network $F_{dc}(\cdot)$ and visually examine the predicted 3D points $\mathbf{z}$. A reasonable prediction of 3D coordinates in Fig.6 from 2D sketches verifies that the 2D sketch and 3D shapes are indeed aligned (due to pivoting) and 3D aware (due to auxiliary lifting loss).

L. Category-level versus Fine-Grained retrieval

Ours (i.e., zero-shot) method comprising of pivoting + lifting gives competitive performance for fine-grained sketch-based shape retrieval (Tab. 1) but not so much for category-level sketch-based shape retrieval (Tab. 2). This is because (i) the valuable geometric constraints provided by our lifting module – necessary for fine-grained retrieval [16] – have limited use in category-level sketch-based shape retrieval. (ii) Collecting category-level sketch–shape data is easier than for fine-grained setups. This led to large-scale category-level sketch–shape datasets like SHREC'13 [18] and SHREC'14 [19]. Hence, our pivoting module (that bridges 2D sketch and 3D shape domain gaps in data scarcity scenarios) also has limited use.

M. Details of Point Sampling for BPnP

For images, we evenly sample (64×64) points from which 512 Monte Carlo samples are selected for faster training. This sampling strategy works for images, hence used in PnP loss between the 3D model and its renderings (images). Albeit not our goal, we use the same sampling strategy on rasterised sketches to visually examine if the reconstructed 3D shape is aligned with our 2D sketch. While our primitive exploration of alternative sampling strategies like 2D landmarks [6], minimal set [2] did not significantly improve overall retrieval performance, future work could further conduct a detailed analysis of alternative sampling strategies and their effects on 2D images and sparse sketches.

N. Additional Ablation on Lifting

The motivation of lifting loss in Eq. 13 is to provide valuable geometric constraints necessary for fine-grained retrieval – "lift" 2D sketch to 3D space using the Blind Perspective-n-Points (BPnP) algorithm as an additional geometric constraint. Replacing BPnP in our proposed method, we ablate using alternative methods to inject 3D-aware knowledge, as (i) 2D sketch-to-3D Reconstruction (*3D Recon.*) – using SDF [22] (as in SDFSketch), PSGN [14] (as in PSGNSketch), and Voxel [13] representations. (ii) Since directly predicting 3D shape from 2D sketch feature does not leverage geometric priors [8], we supervise by predicting 3D pose from 2D sketch (*Pose Pred.*) – via classifying (Class.) [20] input 2D sketch into 36 poses, or

directly regressing (Reg.) the 4DoF [26]. (iii) Finally, we compare with our geometry (Geo.) based BPnP loss as,

Table 4. Ablative study for Lifting on 'Chairs' in [24].

Acc.@1	3D Recon.			Pose Pred.		Ours
	SDF	PSGN	Voxels	Reg.	Class.	Geo.
zero-shot	52.7	53.5	51.3	52.7	51.2	55.8
all-shot	57.4	58.0	56.4	57.3	56.2	58.5

O. Additional Clarification on Tab. 3

In Row-1 of Tab. 3, we use 804 paired sketch-3D shapes for all-shot setting, but without pivoting or lifting to train 2D-3D encoder. This gives a low Acc.@1 by 10.5. Adding pivoting + paired sketch-3D shapes (all-shot) in Row-2 (thereby scaling train data with easy-to-collect 2D sketch-photos) greatly improves Acc.@1 by 38.2. This shows the impact of scaling train data using easy-to-collect sketch-photo datasets and removing the dependency on 3D annotations. Finally, pivoting + lifting + sketch-3D shapes in Row-6 further improves Acc.@1 by 9.8.

References

[1] Ayan Kumar Bhunia, Aneeshan Sain, Parth Hiren Shah, Animesh Gupta, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Adaptive Fine-Grained Sketch-Based Image Retrieval. In *ECCV*, 2022. 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, 3

[3] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 1, 3

[4] 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, 3

[5] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[6] 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, 3

[7] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *CVPR*, 2021. 1

[8] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation. In *CVPR*, 2022. 1, 3

[9] Yong Chen, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. Joint Training for Pivot-based Neural Machine Translation. *IJCAI*, 2017. 3

[10] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal Transport for Domain Adaptation. *IEEE TPAMI*, 2016. 2

[11] Guoxian Dai, Jin Xie, Fan Zhu, and Yi Fang. Deep Correlated Metric Learning for Sketch-Based 3D Shape Retrieval. In *AAAI*, 2017. 1

[12] Shuyang Dai, Kihyuk Sohn, Yi-Hsuan Tsai, Lawrence Carin, and Manmohan Chandraker. Adaptation Across Extreme Variations using Unlabelled Bridges. In *BMVC*, 2020. 2

[13] Johanna Delanoy, Mathieu Aubry, Phillip Isola, Alexei A Efros, and Adrien Bousseau. 3D Sketching using Multi-View Deep Volumetric Prediction. *CGIT*, 2018. 3

[14] Haoqiang Fan, Hao Su, and Leonidas Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *CVPR*, 2017. 3

[15] Stephen Gould, Richard Hartley, and Dylan John Campbell. Deep declarative networks. *IEEE TPAMI*, 2021. 1

[16] Yulia Gryaditskaya, Felix Hähnlein, Chenxi Liu, Alla Sheffer, and Adrien Bousseau. Lifting Freehand Concept Sketches into 3D. *ACM Trans. Graph*, 2020. 1, 3

[17] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-Center Loss for Multi-View 3D Object Retrieval. In *CVPR*, 2018. 1

[18] Bo Li, Yijuan Lu, Afzal Godil, Tobias Schreck, Masaki Aono, Henry Johan, Jose M. Saavedra, and Shoki Tashiro. SHREC'13 track: large scale sketch-based 3D shape retrieval. In *3DOR*, 2013. 3

[19] Bo Li, Y. Lu, Chen-Feng Li, Afzal A. Godil, Tobias Schreck, Aono, Martin Burtscher, Hongbo Fu, Takahiko Furuya, H. Johan, J. Liu, Ryutarou Ohbuchi, A. Tatsuma, and Changqing Zou. SHREC'14 track: Extended Large Scale Sketch-Based 3D Shape Retrieval. In *3DOR*, 2014. 3

[20] Siddharth Mahendran, Haider Ali, and Rene Vidal. A Mixed Classification-Regression Framework for 3D Pose Estimation from 2D Images. In *BMVC*, 2018. 3

[21] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple Open-Vocabulary Object Detection with Vision Transformers. In *ECCV*, 2022. 3

[22] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *CVPR*, 2019. 3

[23] Anran Qi, Yulia Gryaditskaya, Jeifei Song, Yongxin Yang, Yonggang Qi, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song. Toward Fine-Grained Sketch-Based 3D Shape Retrieval. *TIP*, 2021. 1, 2

[24] Anran Qi, Yi-Zhe Song, and Tao Xiang. Semantic Embedding for Sketch-Based 3D Shape Retrieval. In *BMVC*, 2018. 4

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 3

[26] Justin Solomon and Yue Wang. Object DGCNN: 3D Object Detection using Dynamic Graphs. In *NeurIPS*, 2021. 4

[27] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In *ICCV*, 2015. 1

[28] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous Deep Transfer Across Domains and Tasks. In *ICCV*, 2015. 2

[29] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-Based Convolutional Neural Networks for 3D Shape Analysis. *ACM Trans. Graph*, 2017. 1

[30] Jin Xie, Guoxian Dai, and Yi Zhu, Fan Fang. Learning Barycentric Representations of 3D Shapes for Sketch-based 3D Shape Retrieval. In *CVPR*, 2017. 1

[31] Rui Xu, Zongyan Han, Le Hui, Jianjun Qian, and Jin Xie. Domain Disentangled Generative Adversarial Network for Zero-Shot Sketch-Based 3D Shape Retrieval. In *AAAI*, 2022. 1

[32] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Chen Change Loy. Sketch Me That Shoe. In *CVPR*, 2016. 2, 3