Category-Agnostic Pose Estimation on MP-100 Dataset Using Raster2Seq Framework

Theodoros Chronopoulos
Cornell Tech
NYC, NY, USA
tc796@cornell.edu

Pavlos Rousoglou
Cornell Tech
NYC, NY, USA
pr484@cornell.edu

Niki Karanikola
Cornell Tech
NYC, NY, USA
nk699@cornell.edu

1 Problem Statement and Related Work

The goal of this project is to investigate the problem of category-agnostic pose estimation. In traditional 2D pose-estimation models, training is usually done on a single object category (e.g., humans, specific animals, cars, furniture, etc.). These models work well for the categories they were trained on, but tend to not generalize well to new, unseen object categories without retraining.

Category-agnostic pose estimation (CAPE) tries to solve this limitation. The goal in CAPE is to predict the 2D keypoints of objects that the model has never seen before, using only a very small amount of information that defines the pose for that new category. Specifically, given a query image (the image whose keypoints we want to predict) and a pose definition (which keypoints exist, how they connect, and their rough geometric layout), the model should output the 2D locations of those keypoints in the query image, even if the model never saw that category during training.

Our investigation is based on three main papers:

1. **Pose for Everything: Towards Category-Agnostic Pose Estimation (POMNet)**
   This paper introduces the MP-100 dataset (which we use in our project) and proposes the original CAPE setup. The model is given a support image with labeled keypoints from an unseen category and must transfer that pose information to a query image from the same unseen category by predicting the image's keypoints.
2. **CapeX: Category-Agnostic Pose Estimation from Textual Point Explanation**
   CapeX extends the original CAPE idea by removing the need for a support image. Instead, it uses textual descriptions of keypoints and the skeleton graph to guide pose prediction on the query image. By replacing labeled images with text and a graph structure, the paper shows that pose estimation can be performed using a more abstract representation (i.e. text instead of labelled keypoints).
3. **Raster2Seq: Polygon Sequence Generation for Floorplan Reconstruction**
   This paper introduces an autoregressive sequence model that converts floorplan images into sequences of polygon vertices. It shows that structured 2D geometric sequences can be generated token-by-token from an image.

Our aim in this project is to combine the ideas discussed in these papers and study CAPE on the MP-100 dataset. Instead of using labeled support images (as in POMNet) or textual descriptions (as in CapeX), we will adapt the Raster2Seq autoregressive framework to predict the keypoints of an image using a support pose graph and a query image. The support pose graph is represented as 2D coordinates and edges. In summary, we want to understand whether a geometric, sequence-based representation of the pose graph can achieve CAPE performance comparable to the existing methods so far.

2 Problem Formulation

The category-agnostic pose estimation (CAPE) problem is a task that aims to predict the 2D keypoint locations of categories that a model hasn't been trained on. Unlike traditional pose-estimation approaches, where a separate model is trained for each category (human pose, animal pose, furniture pose, etc.), CAPE requires a single model that can generalize across many different object categories without retraining. This is a challenging problem, because there differences in keypoints and skeletal structures across categories. We want the model to infer the correct pose for a new category using only the pose graph provided at inference time.

Formally, let C_{seen} be the set of categories used during training and C_{unseen} be the categories the model only encounters during inference. For each category c, we are given:

- A pose graph $G_c = (V_c, E_c)$, where:
  - V_c is the set of keypoints, and
  - E_c is the set of edges (the skeletal connections) between them.
- A set of query images I_q, each depicting an object from category c.
- The corresponding ground-truth 2D keypoint coordinates:

$$K_q = \{(x_i, y_i)\}_i, \quad i = 1, \ldots, |V_c|.$$

At inference time, we give the model:

- The pose graph G_c for an unseen category $c \in C_{\text{unseen}}$, represented as a sequence of 2D template coordinates and their skeletal edges, and
- A query image I_q for that unseen category.

The model must then produce a set of predicted keypoints:

$$\hat{K}_q = \{(\hat{x}_i, \hat{y}_i)\}_i, \quad i = 1, \ldots, |V_c|,$$

that correspond to the object in the query image I_q. The challenge is that the model has never seen category c during training and the only information it has about the new category is the abstract pose definition G_c at test time.

3 Dataset

We use the MP-100 dataset introduced in the paper Pose for Everything. MP-100 contains 20,000 images across 100 object categories, including animals, vehicles, furniture, tools, and clothing. Each category has its own:

- Number of keypoints (roughly 8–20)
- Skeletal structure describing how those keypoints connect

- A variety of images with different structural variations (rotated, bent, folded, etc.) and visual differences (textures, lighting, shape, etc.)

The dataset is pre-divided into five training/validation/testing splits. In each split, about 70% of the categories serve as the seen categories for training, while the remaining categories serve as unseen for evaluation. Unfortunately, we were only able to collect 86 out of the 100 categories that make up the full MP-100 dataset. Specifically, we successfully recovered all categories except for 13 clothing categories from DeepFashion2 and the human hand category from the OneHand10K dataset. Even though we didn't manage to reconstruct MP-100 completely, we still follow the same predefined splits as the original paper. This allows us to maintain the closest possible comparison with prior work such as POMNet and CapeX.

4 Evaluation Metric

We evaluate our model using the Probability of Correct Keypoint (PCK) metric. PCK is the standard metric for category-agnostic pose estimation, and it measures the proportion of predicted keypoints that fall within a small, normalized distance from their corresponding ground-truth locations.

A predicted keypoint $\hat{k}_i$ is considered correct if:

$$\text{PCK} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left(\frac{\|\hat{k}_i - k_i\|_2}{D_{\text{norm}}} < \alpha\right).$$

Where:

- k_i is the ground-truth coordinate of keypoint i,
- d_{norm} the normalization factor,
- α is a fixed threshold (e.g., 0.05),
- 1 the indicator function

PCK represents the percentage of correctly predicted key points across all images and categories in the test set. Similarly to CapeX and Pose For Everything, the normalization factor is the bounding box size. By using the same exact metric we can directly benchmark our method against already existing approaches.

5 Method

5.1 Input and Output

Our method adapts the Raster2Seq framework to category-agnostic pose estimation conditioned on a support pose graph. The input is (i) a query RGB image $I_q \in \mathbb{R}^{512\times512\times3}$ containing a single instance of category c, and (ii) a support pose graph that specifies the canonical layout of V_c semantic keypoints for that category. Since MP-100 does not provide canonical template coordinates, we compute a category-specific template pose by averaging normalized training keypoints for each category. This produces a consistent support pose graph $G_c = (V_c, E_c)$ with template coordinates (x_i^s, y_i^s) for each semantic keypoint.The model outputs a set of query keypoints $\{(x_i^q, y_i^q)\}_{i=1}^{V_c}$ localized on the object in I_q, where index i is aligned with the same semantic keypoint in G_c.

5.2 Sequence Representation

Instead of regressing coordinates directly, we follow Raster2Seq and represent keypoints as a discrete token sequence. For a category with N_c keypoints, the sequence has the form

$$[\langle coord\rangle, x_1, y_1, \langle sep\rangle, \ldots, \langle coord\rangle, x_{N_c}, y_{N_c}, \langle sep\rangle, \langle eos\rangle],$$

where $\langle coord\rangle$ marks the start of a keypoint, $\langle sep\rangle$ separates keypoints, and $\langle eos\rangle$ terminates the sequence. This converts coordinate prediction into token classification and allows us to use a transformer decoder for sequence modeling. In addition to coordinate tokens, each keypoint token is also augmented with a learned embedding that encodes its adjacency in the pose graph $G_c = (V_c, E_c)$, so that the transformer can exploit the skeletal connectivity E_c.

5.3 Coordinate Tokenization

Continuous coordinates $(x, y) \in [0, 512]^2$ are mapped to tokens via bilinear interpolation over a learned embedding table. We learn embeddings e_k for integer coordinates $k \in \{0, \ldots, V-1\}$. For a real-valued coordinate u, we take the neighboring integers and represent u as a convex combination of them, with weights proportional to proximity. This provides a smooth, differentiable bridge between continuous image space and discrete token space.

5.4 Model Architecture

The architecture has four components: a query image encoder, a support pose encoder, a cross-modal transformer decoder, and dual prediction heads.

Query image encoder. A ResNet-50 backbone pretrained on ImageNet is used to encode I_q into a feature map $F_q \in \mathbb{R}^{2048\times16\times16}$. As a result the query image is downsampled by a factor of 32. The resulted map captures both low-level and high-level object-part information.

Support pose encoder. The support pose graph is first flattened into a coordinate sequence. Each coordinate is embedded and passed through a transformer encoder with 3 layers and 8 attention heads, producing contextual support embeddings $E_s \in \mathbb{R}^{N_c\times256}$ that encode both absolute positions and spatial relationships between support keypoints.

Cross-modal transformer decoder. A 6-layer transformer decoder (8 heads, hidden dimension 256) autoregressively generates the token sequence. At each layer, masked self-attention operates over previously generated tokens, deformable cross-attention attends to a sparse set of informative locations in F_q, and cross-attention to E_s injects structural information from the support pose graph. A feed-forward block with residual connections and layer normalization completes each decoder layer.

Prediction heads. For every decoder state, two linear heads are applied in parallel. The classification head outputs logits over the token vocabulary (coordinate tokens and special tokens such as $\langle coord\rangle$, $\langle sep\rangle$, $\langle eos\rangle$). The regression head predicts a continuous (x, y) pair, which is passed through a sigmoid and scaled to $[0, 512]$.

5.5 Autoregressive Generation

At inference time, generation starts from a start token and proceeds one token at a time. At each step the decoder conditions on all previously generated tokens, the image features F_q, and the support embeddings E_s, until the $\langle eos\rangle$ token is produced.

5.6 Loss Functions

We train the model end-to-end with a multi-task loss

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{ce} + \lambda_2 \mathcal{L}_{coords},$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss over tokens and $\mathcal{L}_{coords}$ is the L1 loss between predicted and ground-truth continuous coordinates. We use $\lambda_1 = 1.0$ and $\lambda_2 = 5.0$, and apply the same loss at intermediate decoder layers as auxiliary supervision to stabilize training. During training, the model only observes categories in the seen split. At test time, the model is evaluated on unseen categories using only the support pose graph and the query image. This ensures category-agnostic generalization consistent with the MP-100 and CAPE protocols.

6 Preliminary Results

After adapting the Raster2Seq framework to the category pose estimation problem using the MP-100, we trained a tiny version of the model with 5 epochs for our preliminary results using NVIDIA A100 GPUs.

Parameter	Value
Mode	tiny
Total Epochs	5
Batch Size	8
Learning Rate	0.0001
Backbone Model	resnet50
Train Samples	11,665
Validation Samples	1,241
Device	cuda_0
Seed	42

Table 1: Training Run Hyperparameters excluding the DeepFashion2 categories.

We can observe in Figure 1 that the training loss consistently decreased over the 5 epochs, whereas the validation loss plateaued after epoch 2. This difference between training and validation could be an early sign of overfitting. However, we need to train the full model on 300 epochs to draw this conclusion.

In Figure 2, we can see that the overfitting could be driven by the Validation Class Loss, which peaked after epoch 4. However, it's still too early to make this assumption.

The Mean PCK peaked at epoch 4 (0.5980) before dropping at epoch 5 (0.5506). Interestingly, the highest PCK@0.2 score was recorded at epoch 2 (0.8605). So far, our model is correctly learning to find the appropriate area of the keypoints.

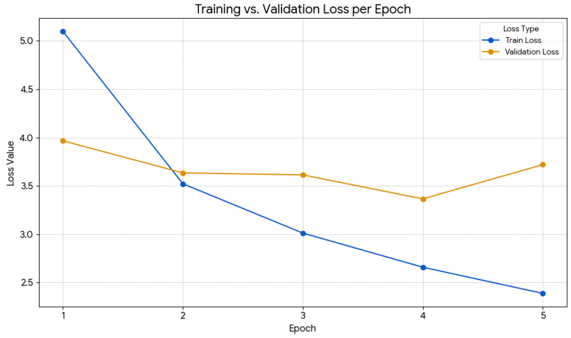

Figure 1: Train and validation losses per epoch

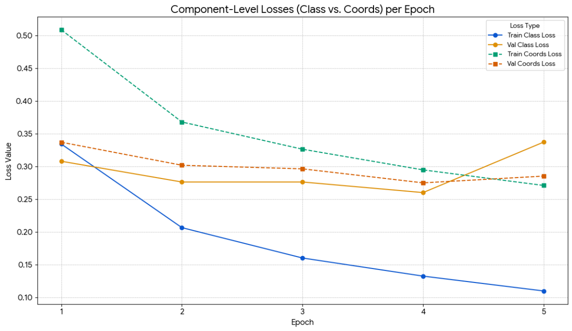

Figure 2: Component-Level Losses per epoch

Epoch	PCK@0.2	Mean PCK
1	0.8306	0.5273
2	0.8605	0.5858
3	0.8392	0.5725
4	0.8531	0.5980
5	0.8197	0.5506

Table 2: Validation PCK Metrics per Epoch.

7 Next Steps

Our preliminary results show the potential of our model, but it's too early to form an well-rounded opinion on its performance. By December first, we plan to:

(1) **Collect Full Dataset:** We hope to hear back from the DeepFashion2 dataset's stakeholders, so that our final model is benchmarked against the complete MP-100 dataset.
(2) **Train full model:** We will train the model on 300 epochs and monitor convergence. This will help us identify the best performing model.
(3) **Model evaluation:** We will compare our final PCK results against the MP-100 benchmark baselines.

8 Conclusion

We have made substantial progress toward adapting the Raster2Seq framework for category-agnostic pose estimation on the MP-100 dataset. We will use this work and build upon it to improve the model performance by December 1st. This work will provide insights into the effectiveness of the Raster2Seq approaches for category-agnostic pose estimation.

Acknowledgments

We gratefully acknowledge the author of Raster2Seq for releasing their code. Our approach builds upon CapeX for the category pose estimation and draws inspiration from Raster2Seq for the architecture design.

References

[1] M. Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020.

[2] O. Hirschorn and S. Avidan. Edge weight prediction for category-agnostic pose estimation. *arXiv preprint arXiv:2411.16665*, 2024.

[3] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li, and K. Chen. RTMPose: Real-time multi-person pose estimation based on MMPose. *arXiv preprint arXiv:2303.07399*, 2023.

[4] D. Maji, S. Nagori, M. Mathew, and D. Poddar. YOLO-Pose: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2637–2646, 2022.

[5] M. Rusanovsky, O. Hirschorn, and S. Avidan. CapeX: Category-agnostic pose estimation from textual point explanation. *arXiv preprint arXiv:2406.00384*, 2024.

[6] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang. ApolloCar3D: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5452–5462, 2019.

[7] L. Xu, S. Jin, W. Zeng, W. Liu, C. Qian, W. Ouyang, P. Luo, and X. Wang. Pose for everything: Towards category-agnostic pose estimation. In *European conference on computer vision*, pages 398–416. Springer, 2022.

[8] H. Yu, Y. Xu, J. Zhang, W. Zhao, Z. Guan, and D. Tao. AP-10K: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617*, 2021.

[9] A. M. Bataineh. *Monocular 3D Human Pose Estimation for REBA Ergonomics*. ScienceDirect, 2025.