CapeX: Category-Agnostic Pose Estimation from Textual Point Explanation

Matan Rusanovsky   Or Hirschorn   Shai Avidan

Tel Aviv University

https://github.com/matanr/capex

Abstract

Conventional 2D pose estimation models are constrained by their design to specific object categories. This limits their applicability to predefined objects. To overcome these limitations, category-agnostic pose estimation (CAPE) emerged as a solution. CAPE aims to facilitate keypoint localization for diverse object categories using a unified model, which can generalize from minimal annotated support images. Recent CAPE works have produced object poses based on arbitrary keypoint definitions annotated on a user-provided support image. Our work departs from conventional CAPE methods, which require a support image, by adopting a text-based approach instead of the support image. Specifically, we use a pose-graph, where nodes represent keypoints that are described with text. This representation takes advantage of the abstraction of text descriptions and the structure imposed by the graph. Our approach effectively breaks symmetry, preserves structure, and improves occlusion handling. We validate our novel approach using the MP-100 benchmark, a comprehensive dataset spanning over 100 categories and 18,000 images. Under a 1-shot setting, our solution achieves a notable performance boost of 1.07%, establishing a new state-of-the-art for CAPE. Additionally, we enrich the dataset by providing text description annotations, further enhancing its utility for future research.

1 Introduction

Pose estimation deals with the prediction of semantic parts' positions within objects depicted in images, a task crucial for applications like zoology, autonomous driving, virtual reality, and robotics [36]. Previous pose estimation methods were typically constrained by their reliance on category-specific datasets for training. Consequently, when confronted with novel objects, these methods often exhibit limited efficacy due to their lack of adaptability.

To address this challenge, recent research has introduced category-agnostic pose estimation (CAPE) [36], a paradigm capable of localizing semantic parts across diverse object categories, based on a single or few support examples. All previous CAPE works require a small set of support images annotated with the keypoints of interest. These support images are used in order to find the best spatial arrangement of the keypoints in the query image, based on latent visual correspondence to the annotated support keypoints.

This raises two challenges. First, the need to provide annotated support image(s) is cumbersome. Second, relying solely on visual correspondence between keypoints in different images, even from the same category, may lead to suboptimal results. This is because no two distinct images share parts with the exact same appearance. Still, both images should share parts with the same semantic
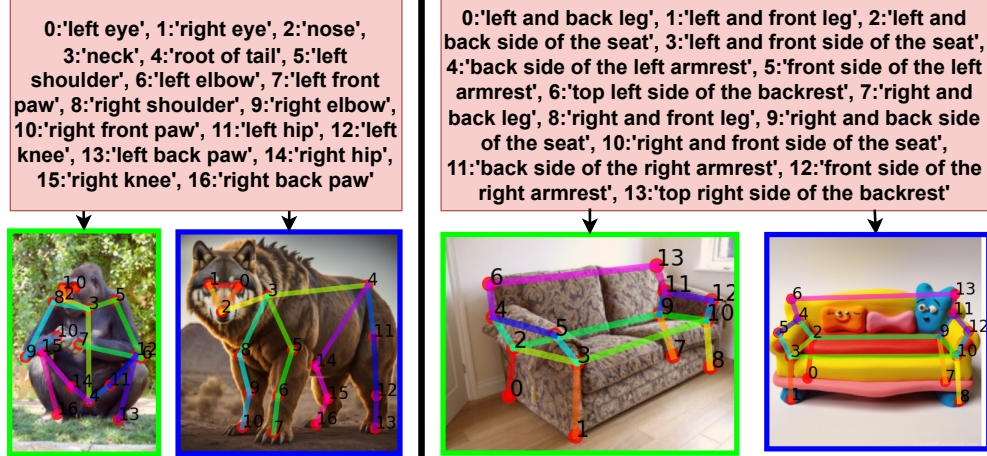
Figure 1: **CapeX in action:** Given support keypoints text descriptions (in pink) and a corresponding skeleton (not shown), our model localizes the skeleton on query images. In the first row, there are few input support text descriptions, and below each support input, there is a query image from the test set on the left (green), and an AI generated query image on the right (blue). Our approach does not require a support image. Instead, it utilizes the abstraction power of text to improve keypoint localization.

meaning. For example, all cats have a head, legs, and a tail, but they never look the same. This idea is even more crucial when the objective is to estimate poses of objects in images from novel categories (i.e., dogs), as in CAPE.

We cope with these limitations by adopting a holistic approach to pose estimation, based on a support graph as input, with open-vocabulary textual descriptions on its nodes. No support images are needed. Instead of exclusively relying on visual support data, we leverage the abstraction power of textual data. This comprehensive view enables us to match the query keypoints' appearance to the textual description of the support keypoints, eliminating the need for support images altogether. Furthermore, following Pose Anything [10], instead of treating the input keypoints as isolated entities, we treat them as structure-aware connected nodes of a graph. By doing so, we effectively leverage the inherent relationships and dependencies between keypoints, enhancing the overall performance, breaking symmetry, preserving structure, and better handling occlusions. Figure 1 demonstrates our approach.

To evaluate the efficacy of our proposed method, we utilize the extended version [10] of the CAPE benchmark, MP-100 [36]. This dataset consists of more than 18,000 images spanning 100 categories, encompassing diverse subjects such as animals, vehicles, furniture, and clothes. As some of the categories miss the keypoints' text descriptions, we collected and unified the text descriptions of the keypoints in all categories. Our method is evaluated against previous CAPE methodologies. Notably, our approach surpasses the performance of existing methods, showcasing a new state-of-the-art performance under the 1-shot setting.

In summary, our contributions can be outlined as follows:

- We propose modeling the support keypoints using connected graph nodes coupled with text descriptions as opposed to previous methods that rely on visual signals. This methodology matches the support to the query keypoints, thanks to the abstraction power of text and graphs. Furthermore, this approach does not require support images for either training or inference.

- We provide an enhanced version of the MP-100 dataset with textual annotations for the keypoints in all categories, enriching the benchmarking capabilities for category-agnostic pose estimation.

- We establish new benchmarks in category-agnostic pose estimation, showcasing state-of-the-art performance on the MP-100 dataset, without finetuning the support feature extraction.

2 Related Work

2.1 Category-Agnostic Pose Estimation

The primary aim of pose estimation is to localize the semantic keypoints of objects or instances precisely. Traditionally, pose estimation methods have been largely tailored to specific categories, such as humans [6, 2, 38], animals [41, 40], or vehicles [28, 23]. However, these prior works are constrained to object categories encountered during training.

An emerging aspect in this field is category-independent pose estimation, as introduced by POM-Net [36]. This few-shot approach predicts keypoints by comparing support keypoints with query images in the embedding space, addressing the challenge of object categories not seen during training. POMNet employs a transformer to encode the support keypoints and query images. It uses a regression head to predict similarity from the extracted features. CapeFormer [25] extends this matching paradigm to a two-stage framework, refining unreliable matching outcomes to improve prediction precision. Pose Anything [10] presented a significant departure from previous CAPE methods, which refer to keypoints as isolated entities, by treating the input pose data as a graph. It utilizes Graph Convolutional Networks (GCNs) to leverage the inherent object's structure to break symmetry, preserve the structure, and better handle occlusions. However, similar to previous CAPE models, it relies solely on visual features. Our work builds upon Pose Anything, utilizing its structure-aware architecture, while introducing the abstraction power of text.

2.2 Open-Vocabulary Models

A growing area in computer vision called Open-Vocabulary learning is being explored in various vision tasks. These new methods aim to localize and recognize categories beyond the labeled space. The open-vocabulary approach is broader, more practical, and more efficient compared to weakly supervised setups [33]. Large-scale vision-language models (VLMs) like CLIP [22] and ALIGN [11] have shown promise in handling both visual and text data, and proved useful for open-vocabulary tasks.

Open-vocabulary object detection (OVOD) using VLMs was utilized by performing object-centric alignment of language embeddings from the CLIP model [1]. Zang et al. [42] suggested a DETR (common transformer-based architecture) based detector, able to detect any object given its class name. In addition, LLMs were also used to generate informative language descriptions for object classes and construct powerful text-based annotations [12]. Another task that recently achieved significant progress is open-vocabulary semantic segmentation (OVSS), which aims to segment objects with arbitrary text. One line of research [5, 37, 34] combines powerful segmentation models like MaskFormer [3] with CLIP [35] while others [44] utilize foundation segmentation models like SAM [14]. Recently, Wei et al. [30] suggested a new benchmark for Open-Vocabulary Part Segmentation, to further enhance open-vocabulary capabilities.

Yet, there's still limited exploration into open-vocabulary keypoint detection. Recently, CLAMP [43] leveraged CLIP to prompt animal keypoints. They found that establishing effective connections between pre-trained language models and visual animal keypoints is challenging due to the substantial disparity between text-based descriptions and keypoint visual features. CLAMP attempts to narrow this gap by using contrastive learning to align the text prompts with the animal keypoints during training. Our approach aims for general keypoint estimation of any category while taking advantage of structure as a prior for localization by treating the input prompts as a graph.

3 Method

3.1 Open-Vocabulary Keypoint Detection

Open-vocabulary learning seeks to localize and recognize categories beyond those included in annotated labels. While open-vocabulary object detection and segmentation have gained attraction, keypoint detection has largely been overlooked. Segmentation offers pixel-level details about semantic regions, whereas object detection identifies specific objects and their locations. Keypoint detection lies between these two, offering finer semantic localization than object detection, yet being more
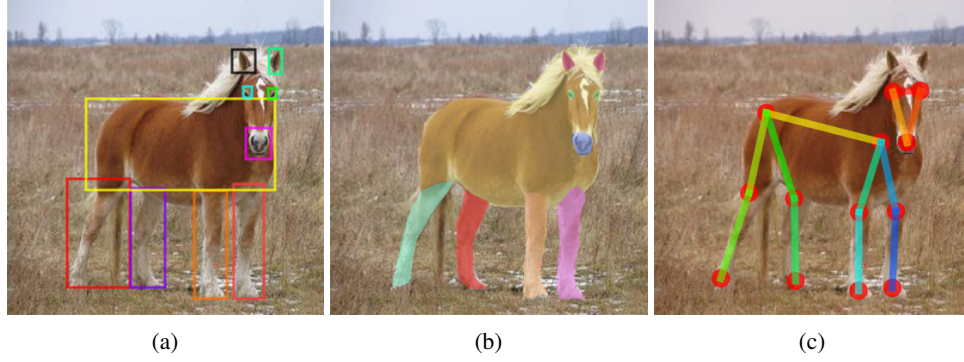
Figure 2: **Different Open-Vocabulary Tasks:** We show three different open-vocabulary tasks: (a) object detection, (b) part segmentation, and (c) keypoint detection. Object detection identifies objects and locations, segmentation provides pixel-level semantic details, and keypoint detection offers finer localization than object detection while being more practical for localization than segmentation.
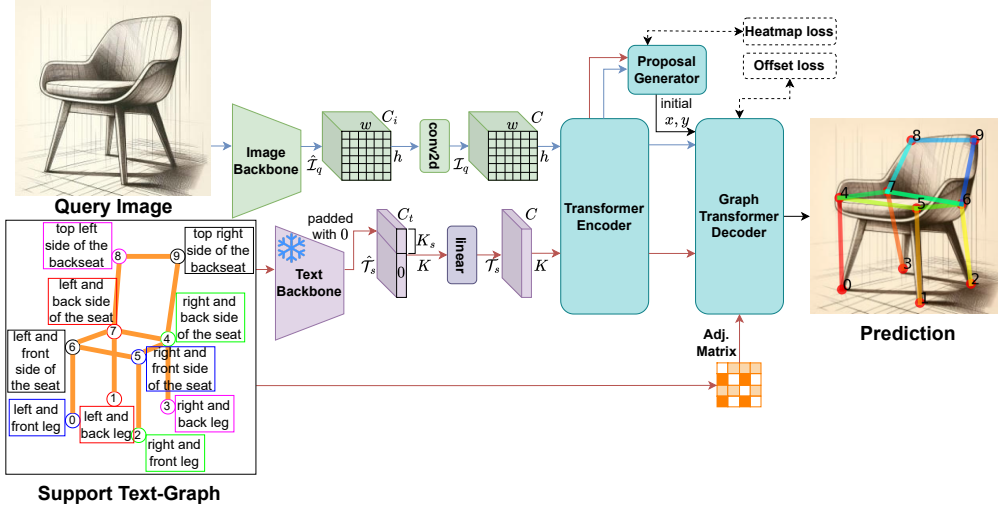

Figure 3: **Architecture overview:** Our framework uses image and text backbones benefiting from both accurate and abstract descriptions respectively. The extracted feature descriptors are forwarded into the transformer encoder that refines them. The refined features are passed into the proposal generator alongside the graph transformer decoder, utilizing the graph structure within the data.

lightweight and practical for parts localization compared to segmentation. Figure 2 demonstrates the differences between the three tasks.

Open-vocabulary keypoint detection aims to use natural language to identify any keypoints in images, even if those key categories were not part of the training data. Advances in vision-language models such as CLIP allow keypoint detectors to harness powerful language models to perform language-driven tasks. We introduce a new open-vocabulary keypoint detector inspired by CAPE, a few-shot task of localizing keypoints in unseen categories using a few annotated images. The core idea of our work is that for the task of CAPE, it is more beneficial to describe the searched points in the query image using text description instead of relying only on the visual features of the support images. This is because text allows a higher level of abstraction and offers a looser restriction to the support request. This is true even when the support and query images are from the same category - for example, no two cats share visually the exact same *front left leg*, but both cats have a part within them that follows the same text description: *front left leg*. This distinction is even more significant when dealing with images from different categories as in CAPE. We present in the supplemental Figure 10 how a support

4

image-based CAPE solution might suffer from incorrect pose estimation due to visually inconsistent support images. Bottom left in Figure 3 is an example of a support text-graph that our system utilizes.

3.2 Text Prompts as Visual Queues

Our framework extracts visual features from the query image and matches them to the textual features that are extracted from the support text-graph. We incorporated this notion by introducing text comprehension into Pose Anything's framework [10].

A pre-trained and fine-tuned SwinV2-S [19] is utilized for extracting image features from the input query image producing the feature map $\hat{\mathcal{I}}_q \in \mathbb{R}^{hw \times C_i}$, where hw is the total number of patches and C_i is the image embedding dimension. Then $\hat{\mathcal{I}}_q$ is passed through a 1x1 convolutional layer, resulting in $\mathcal{I}_q \in \mathbb{R}^{hw \times C}$.

The support keypoint text descriptions are embedded in our model using a pre-trained gte-base-v1.5 [17]. The text embeddings of all K_s keypoints of the provided support sample are then normalized. The normalized keypoints are padded with zeros, effectively resulting in K keypoints, where K is defined to be the maximum amount of possible keypoints in the dataset. The final text feature map is of the form $\hat{\mathcal{T}}_s \in \mathbb{R}^{K \times C_t}$ where C_t is the text embedding dimension. Then $\hat{\mathcal{T}}_s$ is passed through a linear layer resulting in $\mathcal{T}_s \in \mathbb{R}^{K \times C}$. During training, the text backbone is frozen. This approach also offers a lighter optimization procedure, as the gradients of the text features are ignored. An architecture overview is presented in Figure 3.

The extracted query image features and the support descriptions features are then refined using the transformer encoder. This encoder comprises three transformer blocks. Since the embedding spaces of the support text and query image differs, the support and query features are first fused together and then separated again. This practice aids in closing the gap between their representations [25] using self-attention layers. Then, similarity heatmaps between the query and support features are formed, using the proposal generator. The proposal generator utilizes a trainable inner-product mechanism [26] to explicitly represent similarity. Peaks are then chosen from these maps to act as the basis for similarity-aware proposals. A graph transformer decoder network receives these initial proposals, processes them using a combination of attention and Graph Convolutional Network (GCN) layers, and predicts the final estimated keypoints locations. Utilizing GCN layers allows for the explicit consideration of semantic connections between keypoints, thereby benefiting CAPE tasks. We visualize cross-attention maps from the decoder trained with text prompts compared to visual prompts in the supplemental (Figure 11).

To train our end-to-end method we use two loss terms: $\mathcal{L}_{heatmap}$ and $\mathcal{L}_{offset}$. The former penalizes the similarity metric while the latter penalizes the localization output:

$$\mathcal{L}_{heatmap} = \frac{1}{(K \cdot H \cdot W)} \sum_{i=1}^{K} ||\sigma(M_i) - H_i|| \tag{1}$$

$$\mathcal{L}_{offset} = \frac{1}{L} \sum_{i=1}^{L} \sum_{i=1}^{K} |P_i^l - \hat{P}_i| \tag{2}$$

where σ is the sigmoid function, and for each point i, M_i is the output similarity heatmap of the proposal generator, H_i is the ground truth heatmap, P_i^l is the output location from layer l and $\hat{P}_i$ is the ground truth location. The overall loss is:

$$\mathcal{L} = \lambda_{heatmap} \cdot \mathcal{L}_{heatmap} + \mathcal{L}_{offset} \tag{3}$$

4 Experiments

In line with prior CAPE studies, we utilize the MP-100 dataset [36] as both our training and evaluation dataset, which comprises samples sourced from existing category-specific pose estimation datasets [18, 24, 15, 29, 8, 41, 16, 21, 9, 31, 23, 13, 32]. This dataset consists of over 18K images spread across 100 distinct sub-categories and 8 super-categories (human hand & face & body, animal face & body, clothes, furniture and vehicle), featuring varying numbers of keypoints, ranging from 8 to 68 keypoints.

Table 1: **MP-100 results:** $PCK_{0.2}$ performance under the 1-shot setting. Our approach outperforms other methods on average.

Model	Split 1	Split 2	Split 3	Split 4	Split 5	Avg
ProtoNet [27]	46.05	40.84	49.13	43.34	44.54	44.78
MAML [7]	68.14	54.72	64.19	63.24	57.20	61.50
Fine-tuned [20]	70.60	57.04	66.06	65.00	59.20	63.58
POMNet [36]	84.23	78.25	78.17	78.68	79.17	79.70
CapeFormer [25]	89.45	84.88	83.59	83.53	85.09	85.31
CapeFormer-S [10]	92.88	89.11	89.16	87.19	88.73	89.41
Pose Anything-S [10]	93.66	90.42	**89.79**	88.68	89.61	90.43
CapeX	**95.62**	**90.94**	88.95	**89.43**	**92.57**	**91.50**

The dataset is divided into five separate splits for training and evaluation. Importantly, each split ensures that the categories used for training, validation, and testing are mutually exclusive, ensuring that the categories used for evaluation are unseen during the training phase.

The original dataset comes with partial skeleton annotations in different formats, including variations in the keypoint indexing. We use the updated version of Pose Anything [10] that includes unified skeleton definitions for all categories. The updated version predominantly featured brief text sentences describing each point within most categories. However, certain categories exhibited text descriptions with distinct characteristics, such as the use of underscores between words instead of spaces, while others lacked any text descriptions altogether. We annotated and standardized the text descriptions of all points in all categories, offering a new supervision capability to the updated version of [10] of the original MP-100.

To assess our model's performance, we employ the Probability of Correct Keypoint (PCK) metric [39], setting a PCK threshold of 0.2, following the conventions established by Pose Anything [10], POMNet [36] and CapeFormer [25]. More design choices and evaluations are in the supplementary.

Implementation Details   To ensure a fair comparison, except for the text backbone, the configuration settings remain consistent with Pose Anything [10] and CapeFormer [25]. The trainable features of the framework remain exactly the same as in Pose Anything (except for the new linear layer) since the text backbone is frozen during training in our framework. However, we also evaluate and present the performance of the framework with an unfrozen text backbone in the supplemental Table 2. C_i is 768 in SwinV2-S, C_t is 768 in gte-base-v1.5. C and K are set to 256 and 100, respectively. The architecture is implemented within the MMPose framework [4], trained using the Adam optimizer for 200 epochs with a batch size of 16. The initial learning rate is 10^{-5}, reducing by a factor of 10 at the 160th and 180th epochs. All experiments in our work were carried out using a machine equipped with an NVIDIA RTX A5000 GPU. Our model required 10 GB of GPU memory and took roughly 20 hours to train for each split.

4.1   Benchmark Results

We conduct a comparative analysis of our approach with gte-base-v1.5 [17] as the freezed text backbone, against Pose Anything [10], as well as prior CAPE methodologies such as CapeFormer [25] and its enhanced version CapeFormer-T from [10], POMNet [36], ProtoNet [27], MAML [7], and Fine-tuned [20]. For a comprehensive understanding of these models' performance, additional details can be found in [36].

Our evaluation is based on the MP-100 dataset, considering the 1-shot scenario. While traditionally 1-shot refers to a single required support image, our framework uses a single text-graph instead. We do not report the 5-shot results, because we do not use 5 different support images. The results are presented in Table 1. Notably, our text-based approach outperforms Pose Anything on most splits, with an average improvement of 1.07% under the 1-shot setting. These results establish a new state-of-the-art result, showcasing the efficacy of utilizing text-graphs for CAPE.
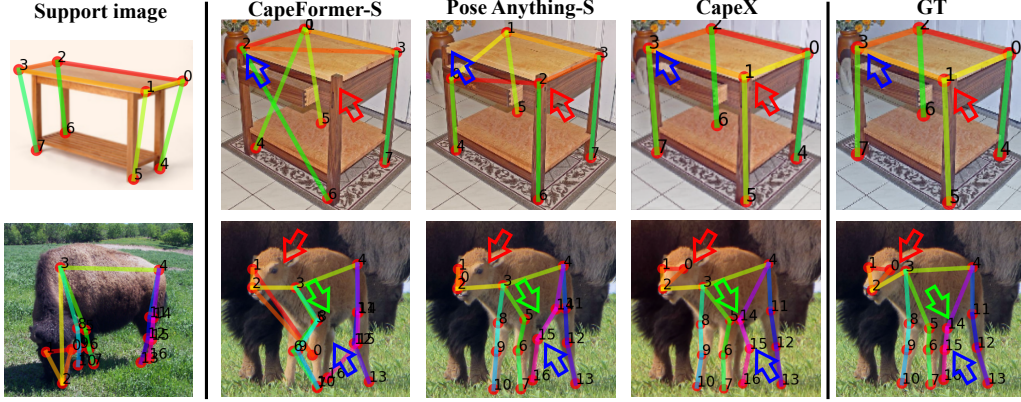
Figure 4: **Qualitative results:** From left to right: support images that are used by the competitors, CapeFormer-S, Pose Anything-S, our model, and the GT. Support text descriptions used by our model are not shown. Main differences are pointed out using arrows.
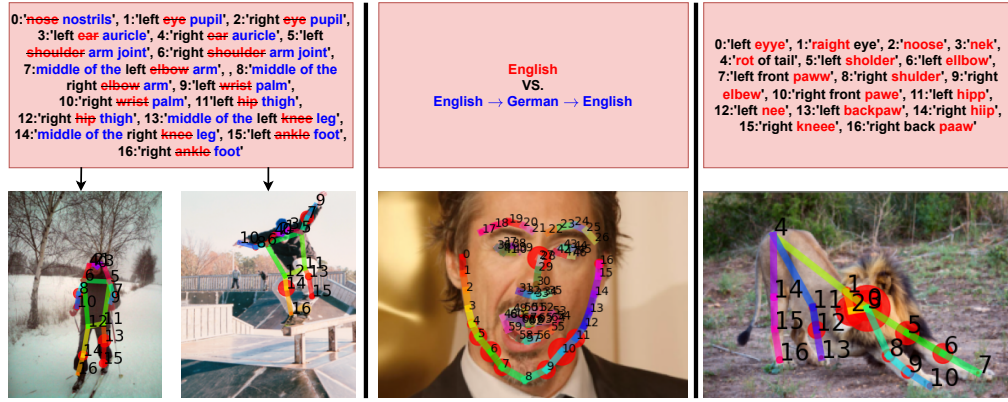

Figure 5: **Modified text descriptions:** Top is the support keypoints text descriptions. Left is a **synonym** words test, middle is a **translation** test and right is **typo** test. Below each description, query output(s) are presented. Each node in the presented graph is the average positions between the original and modified text descriptions. The diameter represents the distance between the positions.

A qualitative comparison of our model against CapeFormer-S and Pose Anything-S in presented in Figure 4. Our model performs well given the support text-graph input (not shown), while the support image-based techniques are sensitive to the inconsistencies between the support and query images.

4.2 Ablation Study

Text Modifications The fact that the text backbone was not fine-tuned during the training of our model, keeps it from overfitting to text descriptions from the training set. On the contrary, the model demonstrates its effectiveness across modified text inputs, while preserving similar estimated poses overall. We test the robustness of our model on different types of modifications for the keypoint descriptions in Figure 5. Specifically, we test the adaptability of the model to synonym descriptions (left), to translation to another language and back to English (middle), and to typos (right). Notably, all average keypoints are placed in acceptable positions. The main differences in the two synonym test examples are in keypoints 7, 8 ('elbow' → 'middle of the arm') and 13, 14 ('knee' → 'middle of the leg'). In the translation test, the main differences are in keypoints 5-7 and 9-11 ('top/bottom side of the right/left jaw/cheek' → 'upper/lower side of the right/left jaw/cheek') and 27 ('top side of the nose' → 'upper side of the nose'). In the typos test, the significant inconsistencies are in the head (keypoints 0,2 and 3), while minor differences are spotted also in the leg joints (keypoints 5,6 and 12). All these differences are compatible with the discrepancy imposed by the different descriptions.

7

Figure 6: **Text Abstractions:** Model performance over different levels of text-pose abstractions.

Figure 8: **Out of distribution performance:** Top is the support keypoints text descriptions. Below each description, we present the query output.

Occlusions and Levels of Abstraction   We test the robustness of our model on keypoints that are described using different levels of text and pose-graphs abstractions. Results are in Figure 6. Although not trained with the prompted text descriptions and pose-graphs, the model presents satisfactory results in both examples.

Furthermore, we assess the effectiveness of text-graphs in handling occlusions within query images by applying random masks to them before estimating the support keypoints. Quantitative results are in Figure 7, and a qualitative comparison is presented in the supplemental Figure 13. Our method demonstrates superior performance over Pose Anything-S and CapeFormer-S in the entire presented occlusion range while maintaining similar degradation levels between 0% to 25%. The improved performance at lower masking percentages can be attributed to the text-

Figure 7: **Masking the query image:** $PCK_{0.2}$ performance as a function of the masking percentage.

graphs' abstraction capability and their ability to estimate missing keypoints relative to the visible ones. However, as our approach does not utilize a support image as input, performance significantly drops and matches the competitors, when a substantial portion of the image is occluded (50%). This is because the absence of the query image leaves the model with insufficient information to operate effectively. This stands in contrast to traditional CAPE methodologies that incorporate a support image, which provides crucial structural cues. In such frameworks, the support image aids the model in hallucinating and extrapolating matching keypoints within the query, particularly when considering graph structures as in Pose Anything.
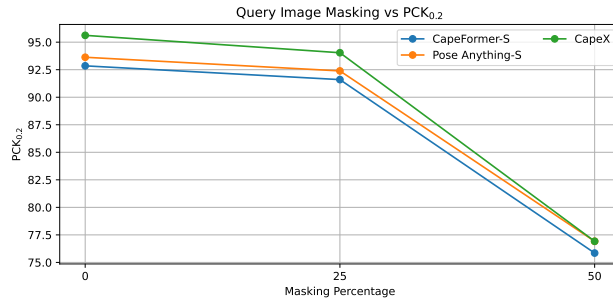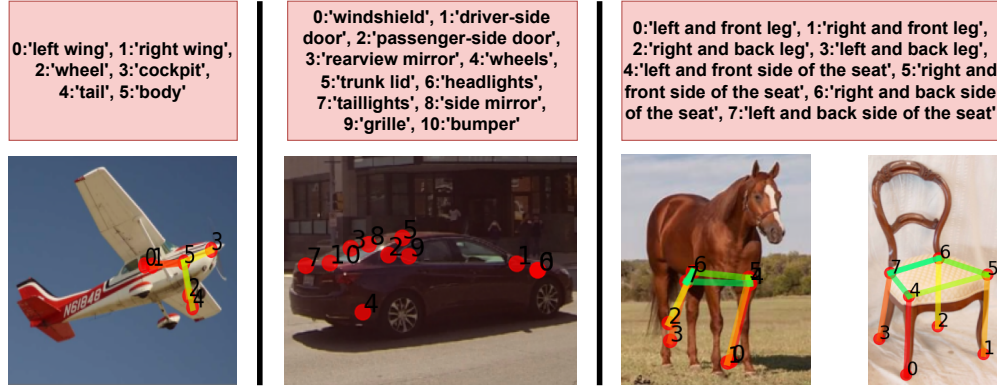
8

Figure 9: **Failure cases:** From left to right: a category outside of the dataset, introducing vastly new keypoint descriptions, and cross-category descriptions.

Out of Distribution Query Images We evaluate the resilience of our model to out-of-distribution query images generated via diffusion models. In Figure 8, we examine novel styles, categories, poses, and even imaging methods. While the estimated poses generally align coherently with both the query and the support text-graph, there are notable inconsistencies. For example, the model appears to inaccurately localize 'knees' in both two left query outputs and fails to localize the 'front paws' in the zebra query output.

5 Limitations

We stress that although our model strives for full open-vocabulary performance, it is still trained on a relatively small training set over arguably a short time period, compared to the state-of-the-art large vision-language foundation models. We present in Figure 9 a few failure cases that may be addressed in future research. Our model does not handle new categories with novel text-graphs well, as can be seen in the plane example on the left. In addition, prompting with vastly new parts may lead to incorrect localizations as can be seen in the car example on the middle (for example, driver/passenger-side door). Lastly, the model incorrectly executes semantically challenging descriptions. For example, the model can not localize a 'seat' in a horse, even though riders may seat on it. Instead, it hallucinates a pose of a chair that it has seen in the training set.

6 Broader impact

Advancements in pose estimation technology can revolutionize fields such as autonomous driving, smart cities, sports, etc., by enabling precise movement analysis. However, when a pose estimation tool is used, specifically in fields such as surveillance, it is crucial to address privacy concerns and establish ethical guidelines to protect sensitive personal data and ensure responsible use.

7 Conclusions

CapeX is a Category Agnostic Pose Estimation (CAPE) approach that is based on text input. In particular, CapeX takes a pose-graph, where text descriptions are attached to its nodes, and finds these keypoints in a query image. This stands in contrast to previous CAPE approaches that require support image with annotated pose-graph as part of the input. CapeX can be viewed as an Open Vocabulary Kepoint Detection algorithm, closing the gap between Open Vocabulary Object Detection and Open Vocabulary Segmentation.

CapeX was tested on the standard MP-100 dataset and achieves a new state of the art result, surpassing previous CAPE methods that rely on support image as input instead of text.

Acknowledgments and Disclosure of Funding

References

[1] Bangalath, H., Maaz, M., Khattak, M.U., Khan, S.H., Shahbaz Khan, F.: Bridging the gap between object and image-level representations for open-vocabulary detection. Advances in Neural Information Processing Systems **35**, 33781–33794 (2022)

[2] Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)

[3] Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in neural information processing systems **34**, 17864–17875 (2021)

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

[5] Ding, J., Xue, N., Xia, G.S., Dai, D.: Decoupling zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11583–11592 (2022)

[6] Fang, H.S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.L., Lu, C.: Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)

[7] Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017)

[8] Ge, Y., Zhang, R., Wang, X., Tang, X., Luo, P.: Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5337–5345 (2019)

[9] Graving, J.M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B.R., Couzin, I.D.: Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. Elife **8**, e47994 (2019)

[10] Hirschorn, O., Avidan, S.: Pose anything: A graph-based approach for category-agnostic pose estimation. arXiv preprint arXiv:2311.17891 (2023)

[11] Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021)

[12] Kaul, P., Xie, W., Zisserman, A.: Multi-modal classifiers for open-vocabulary object detection. In: International Conference on Machine Learning. pp. 15946–15969. PMLR (2023)

[13] Khan, M.H., McDonagh, J., Khan, S., Shahabuddin, M., Arora, A., Khan, F.S., Shao, L., Tzimiropoulos, G.: Animalweb: A large-scale hierarchical dataset of annotated animal faces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6939–6948 (2020)

[14] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026 (2023)

[15] Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: 2011 IEEE international conference on computer vision workshops (ICCV workshops). pp. 2144–2151. IEEE (2011)

[16] Labuguen, R., Matsumoto, J., Negrete, S.B., Nishimaru, H., Nishijo, H., Takada, M., Go, Y., Inoue, K.i., Shibata, T.: Macaquepose: a novel "in the wild" macaque monkey pose dataset for markerless motion capture. Frontiers in behavioral neuroscience **14**, 581154 (2021)

[17] Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., Zhang, M.: Towards general text embeddings with multi-stage contrastive learning. arXiv preprint arXiv:2308.03281 (2023)

[18] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)

[19] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12009–12019 (2022)

[20] Nakamura, A., Harada, T.: Revisiting fine-tuning for few-shot learning. arXiv preprint arXiv:1910.00216 (2019)

[21] Pereira, T.D., Aldarondo, D.E., Willmore, L., Kislin, M., Wang, S.S.H., Murthy, M., Shaevitz, J.W.: Fast animal pose estimation using deep neural networks. Nature methods **16**(1), 117–125 (2019)

[22] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)

[23] Reddy, N.D., Vo, M., Narasimhan, S.G.: Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1906–1915 (2018)

[24] Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: Database and results. Image and vision computing **47**, 3–18 (2016)

[25] Shi, M., Huang, Z., Ma, X., Hu, X., Cao, Z.: Matching is not enough: A two-stage framework for category-agnostic pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7308–7317 (2023)

[26] Shi, M., Lu, H., Feng, C., Liu, C., Cao, Z.: Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9529–9538 (2022)

[27] Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. NIPS (2017)

[28] Song, X., Wang, P., Zhou, D., Zhu, R., Guan, C., Dai, Y., Su, H., Li, H., Yang, R.: Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5452–5462 (2019)

[29] Wang, Y., Peng, C., Liu, Y.: Mask-pose cascaded cnn for 2d hand pose estimation from single color image. IEEE Transactions on Circuits and Systems for Video Technology **29**(11), 3258–3268 (2018)

[30] Wei, M., Yue, X., Zhang, W., Kong, S., Liu, X., Pang, J.: Ov-parts: Towards open-vocabulary part segmentation. Advances in Neural Information Processing Systems **36** (2024)

[31] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200 (2010)

[32] Wu, J., Xue, T., Lim, J.J., Tian, Y., Tenenbaum, J.B., Torralba, A., Freeman, W.T.: Single image 3d interpreter network. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14. pp. 365–382. Springer (2016)

[33] Wu, J., Li, X., Xu, S., Yuan, H., Ding, H., Yang, Y., Li, X., Zhang, J., Tong, Y., Jiang, X., et al.: Towards open vocabulary learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

[34] Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2955–2966 (2023)

[35] Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2955–2966 (June 2023)

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

[37] Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., Bai, X.: A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In: European Conference on Computer Vision. pp. 736–753. Springer (2022)

[38] Yang, S., Quan, Z., Nie, M., Yang, W.: Transpose: Keypoint localization via transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11802–11812 (2021)

[39] Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE transactions on pattern analysis and machine intelligence 35(12), 2878–2890 (2012)

[40] Yang, Y., Yang, J., Xu, Y., Zhang, J., Lan, L., Tao, D.: Apt-36k: A large-scale benchmark for animal pose estimation and tracking. Advances in Neural Information Processing Systems 35, 17301–17313 (2022)

[41] Yu, H., Xu, Y., Zhang, J., Zhao, W., Guan, Z., Tao, D.: Ap-10k: A benchmark for animal pose estimation in the wild. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021)

[42] Zang, Y., Li, W., Zhou, K., Huang, C., Loy, C.C.: Open-vocabulary detr with conditional matching. In: European Conference on Computer Vision. pp. 106–122. Springer (2022)

[43] Zhang, X., Wang, W., Chen, Z., Xu, Y., Zhang, J., Tao, D.: Clamp: Prompt-based contrastive learning for connecting language and animal pose. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23272–23281 (2023)

[44] Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J., Lee, Y.J.: Segment everything everywhere all at once. Advances in Neural Information Processing Systems 36 (2024)

A   Appendix / Supplemental Material
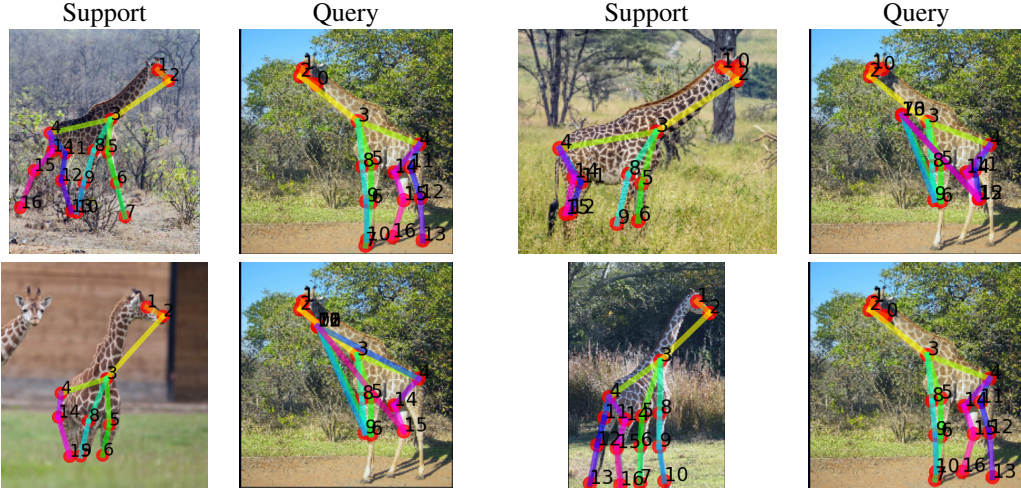
A.1   Disadvantages of Visual Prompts in CAPE

Figure 10: **Visual Prompts Inconsistencies:** We show different results using Pose Anything model, for the same query image using different support images. Keypoints definitions and skeletons are the same. Using visual features impairs the ability to describe abstract semantic parts.
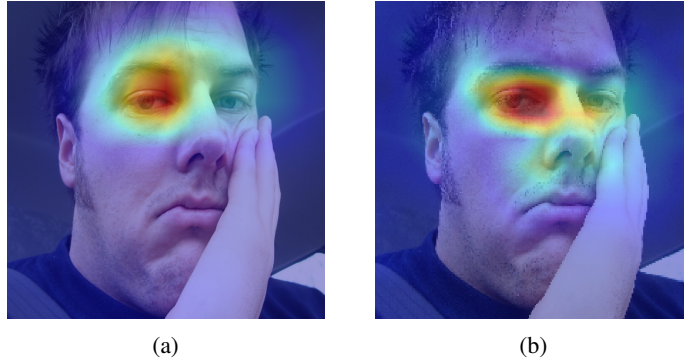

Figure 11: **Cross-attention maps:** comparison between the query image and the 'top right side of the left eye' keypoint. (a) is our model , and (b) is PoseAnything-S. Our model demonstrates in (a) better performance at breaking symmetry and distinguishing between left and right, compared to Pose Anything-S in (b), that attends more to the right eye and the nose.

We exemplify the key disadvantage of support image-based CAPE approaches in Figure 10. Specifically, Pose Anything-S, suffers from incorrect pose estimations when prompted with visually inconsistent support images.

We also compare our models with regards to localization and symmetry breaking. We visualize cross-attention maps from the decoder trained with text prompts compared to visual prompts in Figure 11. Our model breaks symmetry and distinguishes better between left and right, compared to Pose Anything. This can be seen by the lower attention to the right eye, when prompted with the keypoint 'top right side of the left eye'.

A.2   Additional Experiments

A.2.1   Different Architectures

We assess the framework's performance with or without fine-tuning applied to the text backbone. We explore two potential text backbones: gte-base-v1.5 [17] and CLIP ViT-B/32 [22]. The optimal

Table 2: **Ablation experiments:** Tuning (T) VS. Freezing (F) the text backbone in model training, utilizing the graph transformer decoder or the original mlp transformer decoder. $PCK_{0.2}$ performance under 1-shot setting, with gte-base-v1.5 or CLIP ViT-B/32 as the text backbone.

Model	Split 1	Split 2	Split 3	Split 4	Split 5	Avg
CapeX-CLIP-T-graph	94.55	88.71	87.29	88.54	91.65	90.15
CapeX-CLIP-F-graph	95.17	88.88	87.72	88.24	91.81	90.37
CapeX-gte-T-graph	**96.28**	89.15	**89.17**	87.66	92.62	90.98
CapeX-gte-F-mlp	94.69	89.99	89.08	**89.55**	**92.79**	91.22
CapeX-gte-F-graph	95.62	**90.94**	88.95	89.43	92.57	**91.50**

configuration appears to be the frozen gte-base-v1.5 as the text backbone, yielding superior performance. Interestingly, although gte-base-v1.5 boasts approximately 139 million trainable parameters compared to the 63 million parameters in the text module of CLIP ViT-B/32 (totaling 150 million parameters), training with either as a frozen text backbone consumed similar execution times, lasting roughly 20 hours. Memory usage for loading both models required a similar volume of 10 GB. However, fine-tuning both models incurred substantial costs in terms of memory: 15 in gte and 30 GB in CLIP, as well as in execution time: 35 hours for both architectures, without yielding any performance improvements in both text backbones. The drop in performance can possibly be attributed to the fact that the text backbones suffer from overfitting during their tuning. This is somewhat expected as language models usually train on larger datasets over longer training sessions.

We also tested the original MLP transformer decoder architecture as in [25] with the best performing setting. Memory consumption and execution time using this transformer decoder were comparable to the graph transformer decoder. We find that utilizing the graph structure via the graph transformer decoder as in [10] slightly boosts the performance. Full results are presented in Table 2.
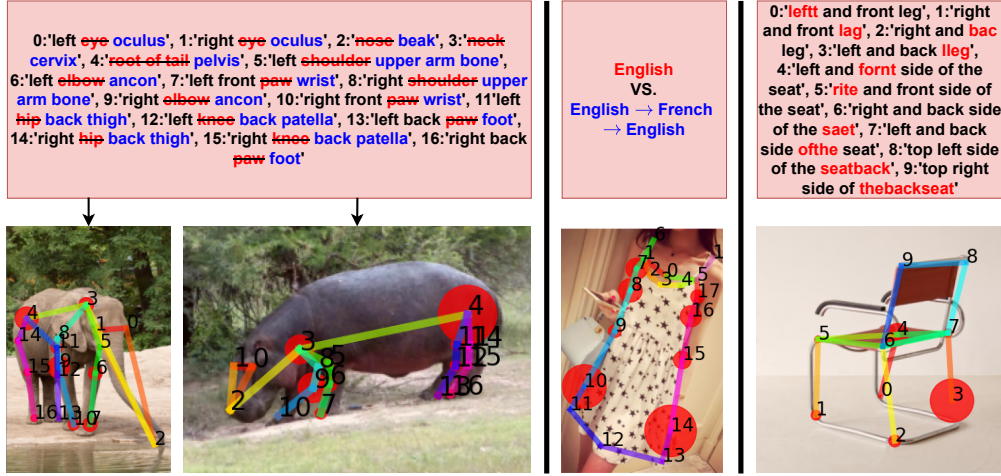

Figure 12: **Modified text descriptions:** Top is the support keypoints text descriptions. Left is a synonym words test, middle is a translation test and right is typo test. Below each description, query output(s) are presented. Each node in the presented graph is the average positions of the original and modified text descriptions. The diameter represents the distance between the positions.

A.2.2 Adaptability to Support Text Modifications

We provide additional examples of the ability of our model to adapt to different types of text modifications in Figure 12.
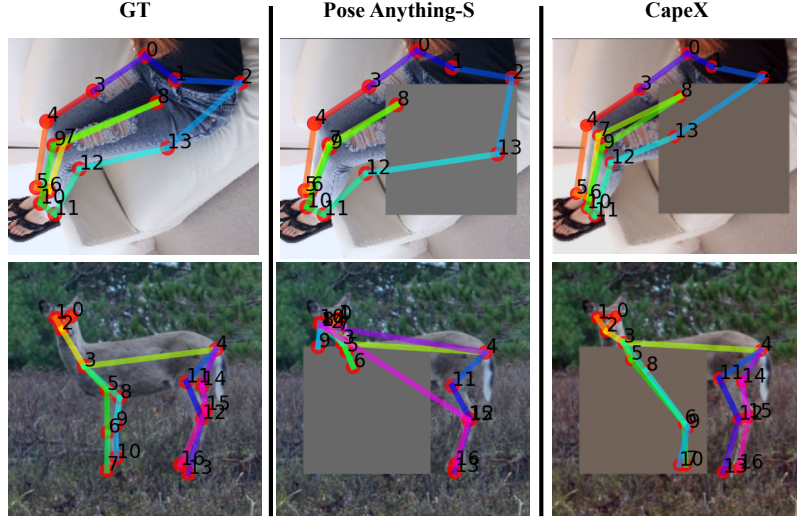
Figure 13: **Comparison to Pose Anything-S:** Qualitative comparison between our model and Pose Anything-S on masked queries. CapeX does not require support images and can handle masked occlusions. Support images and text-graphs are not shown.

Figure 14: **Text Abstractions:** Model performance over different levels of text-pose abstractions.

A.2.3 Occlusions and levels of abstraction

We present qualitative comparison between our support text-based framework and Pose Anything-S's support image-based framework in Figure 13. Our model demonstrates better performance due to the abstraction power of text-graphs, compared to the use of support image which may be more restrictive.

We include additional results of our model's performance on different levels of abstractions in Figure 14.