

Text-Guided Dynamic Structure Prediction for Category-Agnostic Pose Estimation

Khanh-Duy Nguyen

National Central University

Guan-Bin Lin

National Central University

Quoc-Viet Nguyen

National Central University

Huy-Hoang Vo

National Central University

Min-Te Sun

`msun@csie.ncu.edu.tw`

National Central University

Kazuya Sakai

Tokyo Metropolitan University

Wei-Shinn Ku

Auburn University

Research Article

Keywords: CAPE, Pose Estimation, Text Guidance, Graph Transformer, Skeleton Modeling, Keypoints

Posted Date: May 13th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9101076/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Text-Guided Dynamic Structure Prediction for Category-Agnostic Pose Estimation

Khanh-Duy Nguyen^{1,4}, Guan-Bin Lin¹, Quoc-Viet Nguyen¹,
Huy-Hoang Vo¹, Min-Te Sun^{1*}, Kazuya Sakai², Wei-Shinn Ku³

¹Department of Computer Science and Information Engineering,
National Central University, Taoyuan, Taiwan.

²Department of Computer Science, Tokyo Metropolitan University,
Hino, Tokyo, Japan.

³Department of Computer Science and Software Engineering, Auburn
University, Auburn, AL, USA .

⁴Artificial Intelligence Center, Tra Vinh University, Vinh Long,
VietNam .

*Corresponding author(s). E-mail(s): msun@csie.ncu.edu.tw;
Contributing authors: nkduy@tvu.edu.vn; binxlin1102@gmail.com;
vietnq@uit.edu.vn; hoangvohuy851@gmail.com; ksakai@tmu.ac.jp;
weishinn@auburn.edu;

Abstract

In the field of computer vision, keypoint detection is a crucial area of research for understanding the structure of an object. This has led to the emergence of Category-Agnostic Pose Estimation (CAPE), a novel task that utilizes a single model to localize keypoints across diverse object categories. To broaden its application, recent research has incorporated text descriptions into the CAPE task. However, when extending text descriptions to the CAPE task, shortcomings in skeleton information learning and graph information propagation persist, making it difficult for the model to utilize comprehensive structural information, which in turn affects the precision of keypoint localization. In this paper, we propose a model that integrates a Semantic Skeleton Refiner and optimizes the Graph Transformer Decoder architecture, using features from text descriptions as a guidance for dynamic structure prediction. We conducted experiments on the MP-100 dataset and compared our model with existing state-of-the-art models in the CAPE domain. The experimental results indicate that our method outperforms current leading models in the field of CAPE.

Keywords: CAPE, Pose Estimation, Text Guidance, Graph Transformer, Skeleton Modeling, Keypoints

1 Introduction

Category-Agnostic Pose Estimation (CAPE) is a pivotal task in computer vision that aims to detect and localize the keypoints of objects from any category, often in a few-shot learning manner. The primary goal of CAPE is to develop a unified model capable of performing pose estimation on previously unseen object classes, which significantly reduces the need for extensive training data and manual annotation costs compared to traditional Category-Specific Pose Estimation (CSPE) methods [1–3]. This capability is of great importance in various domains, particularly in robotics, where it can enable more precise object tracking.

The evolution of CAPE has seen a significant change in methodology. Early approaches [4–7] in this domain predominantly adopted a paradigm based on support images. This methodology typically involves extracting visual features from an exemplar support image and then leveraging these features to guide the location of corresponding keypoints on the query image, often through a feature matching or correlation process. To further advance the field’s applicability and flexibility, recent research has expanded to incorporate natural language. Methods like CapeX [8] began to use textual descriptions to guide the query image.

Despite these advances, CAPE tasks that incorporate natural language still face significant hurdles. First, a significant limitation is the ineffective processing of skeletal information. This issue stems from relying on the object’s skeleton merely as a static prior, which restricts the model’s ability to dynamically adapt its structural understanding to diverse object poses. Second, models often face the challenge of insufficient propagation of graph information. This bottleneck can hinder the effective exchange of spatial cues among keypoint nodes, limiting the model’s capacity to perform holistic structural refinement for precise localization.

To solve these challenges, we propose a novel framework for text-guided dynamic structure prediction in Category-Agnostic Pose Estimation. Our architecture employs a transformer to effectively fuse visual features from the query image with textual features from the text description. Following this fusion, we integrate Semantic Skeleton Refiner module which dynamically infers the object’s skeletal structure, moving beyond static priors. Subsequently, we introduce the Spatial-Aware Graph Transformer Decoder, utilizing a multi-layer GCN architecture coupled with a novel Spatial Gate mechanism to facilitate a more profound and geometrically-aware exchange of information between keypoint nodes. This allows the model to perform more sophisticated structural reasoning by making feature updates sensitive to their precise spatial relationships, ultimately enriching the final skeletal representation.

In summary, main contributions of this paper are listed as follows:

- We proposed a text-guided multi-module framework for category-agnostic pose estimation. This framework dynamically predicts object structure by effectively fusing

text features and image features, thereby enhancing the localization precision of keypoints for a more accurate pose estimation.

- We integrated a Semantic Skeleton Refiner module that dynamically learns the relationships between skeletal structure and keypoint features, resolving the challenge of static skeletal information processing.
- We implemented a Spatial-Aware Graph Transformer Decoder, which utilizes a multi-layer GCN architecture and a novel Spatial Gate mechanism to improve the propagation and integration of spatial information within the graph structure.
- We conducted comprehensive experiments. The results demonstrate that our proposed model surpasses existing state-of-the-art methods in the field of Category-Agnostic Pose Estimation.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 provides the background on key technologies on which our work is based. Section 4 elaborates on the design of our proposed model. Section 5 details the experiments and analysis to validate our proposed method. Finally, Section 6 concludes the paper and provides future research directions.

2 Related Work

2.1 2D Pose Estimation

2.1.1 Category-Specific Pose Estimation

CSPE primarily focuses on a specific category, with common examples including human pose estimation [1–3] or vehicle skeleton prediction [9]. These methods represent keypoints as learnable tokens, enabling them to develop a universal representation for specific keypoints. They also capture the inherent relationships among keypoints. However, a major drawback of these approaches is that they are limited to the categories present in the training data and cannot handle unseen categories. Li et al. [10] introduced a regression framework based on Residual Log-likelihood Estimation that models distributional shifts to improve training stability. Luo et al. [11] presented a scale-adaptive heatmap approach that learns keypoint-specific standard deviations, enabling more tolerant pose estimation across varying scales and uncertain regions.

2.1.2 Category-Agnostic Pose Estimation

To address the limitations of conventional category-specific pose estimation (CSPE), Xu et al. [4] introduced Category-Agnostic Pose Estimation (CAPE), which extends CSPE [1, 2, 12] and multi-category pose estimation [13] to unseen object categories. CAPE enables a single model to predict keypoints for arbitrary categories in a few-shot setting, substantially reducing annotation and training costs. As the first CAPE method, POMNet [4] employs Transformer-based image encoding but tends to overfit due to its reliance on a small number of support samples. Building on this idea, CapeFormer [7] introduces a two-stage framework that refines initial keypoint matches, improving accuracy at the cost of increased computational complexity and model size. CapeLLM [14] introduces a fully support-free framework that leverages multimodal

large language models with detailed keypoint descriptions, enabling stronger spatial reasoning and state-of-the-art performance on unseen categories without auxiliary supports.

In order to enhance different aspects of CapeFormer, Hirschorn and Avidan [6] introduce a method that differs significantly from traditional CAPE techniques. It treats the input pose data as a graph rather than as independent entities, leveraging the geometric relationships and connectivity between keypoints to improve prediction accuracy, while also effectively preserving the structural integrity of keypoints. Furthermore, CapeX [8] integrates this technique with natural language. By combining features from the query image and supported text-graph, CapeX builds a richer semantic understanding. When visual information is insufficient, textual descriptions improve detection accuracy, but the approach struggles with complex semantics and missing keypoint edge relationships, causing errors under large-scale occlusion.

2.2 Open-Vocabulary Models in Computer Vision

Recent studies have increasingly integrated open-vocabulary models into traditional computer vision tasks, particularly object detection [15], object counting [16] and semantic segmentation [17, 18]. These approaches enable models to recognize objects beyond predefined categories, marking a key advance in open-vocabulary detection (OVD). Early OVD methods mainly rely on pre-trained vision-language models to exploit image-text alignment, but their region-level alignment remains limited. More recent works directly train detectors to model region-text relationships more effectively. For example, Mamba-YOLO-World [15] introduces a YOLO-based OVD framework with an efficient feature fusion design that improves detection performance while keeping the model lightweight.

In the domain of Open-Vocabulary Semantic Segmentation (OVSS), initial research leverages primarily the inherent localization properties of CLIP [19] for task prediction. These methods refine noisy attention maps by reformulating query-key attention as self-attention, but their performance is still limited by CLIP’s weak localization in dense prediction tasks. To overcome these constraints, ProxyCLIP [17] integrates Vision Foundation Models, which demonstrates robust spatial consistency and feature correspondence characteristics, with CLIP.

Compared to OVD and OVSS, Open-Vocabulary Keypoint Detection (OVKD) methods are relatively scarce. The approach proposed by Zhang et al. [20] integrates vision and language models, enhancing the correlation between language and local keypoint visual features through Domain Distribution Matrix Matching and specialized modules. However, this method was constrained by its applicability to animal datasets exclusively. Subsequently, CapeX [8] integrates language models into a vision-based CAPE framework, extending keypoint detection from animal-specific to category-agnostic settings with notable performance gains.

2.3 Graph Transformer encoding

Conventional Graph Neural Network approaches directly encode the graph structure through sparse message passing processes [21], which have inherent limitations such

as oversmoothing [22] and over-squashing [23]. However, Graph Transformers [24, 25] address these limitations by generalizing the traditional Transformer neural network architecture to handle arbitrary graph structures. In Graph Transformer tasks, how to encode graph structural positions is a key focus of many research methods. Dwivedi and Bresson [25] utilize Laplacian eigenvectors for positional encodings, whereas Kreuzer et al. [26] extend this approach by harnessing the complete spectrum of the Laplacian matrix to develop more expressive positional representations. Meanwhile, GraphCape [6] combines Transformers with Graph Convolutional Networks [27] to learn positional encodings. Building upon this approach, Hirschorn and Avidan [5] propose a method that incorporates Markovian Attention Bias into the self-attention matrix and leverages learned residual adjacency matrices with Transformers for positional encoding, enabling the Transformer to naturally understand and utilize node connectivity.

3 Preliminary

3.1 Category-Agnostic Pose Estimation from Textual Point Explanation

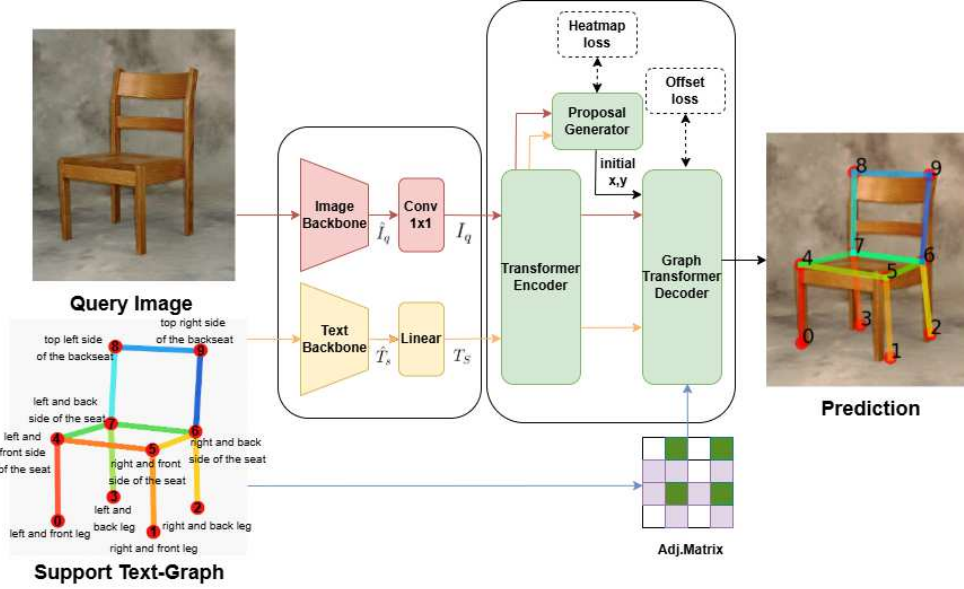


Fig. 1 The CapeX’s model architecture

Category-Agnostic Pose Estimation from Textual Point Explanation (CapeX) [8] is an approach to the challenging task of CAPE. Unlike conventional CAPE methods that need support image as input, CapeX can localize semantic keypoints across diverse

object categories using only query image and support text-graph, without requiring support images. The CapeX architecture is shown in Figure 1, which is divided into several key components that work in sequence to perform keypoint detection for category-agnostic objects and connect these keypoints to perform pose estimation. These components include Swin Transformer V2 [28], General Text Embeddings (GTE-base-en-v1.5) [29], and CapeX’s framework [6]. They are elaborated in the following sections.

3.2 Swin Transformer V2

The Vision Transformer (ViT) [30] brings Transformer architectures to computer vision and has influenced many subsequent models [28, 31]. Building on ViT, the Swin Transformer [31] adopts a hierarchical design that extracts multi-scale image features. It processes images through shifted window attention and patch merging, enabling efficient local attention while preserving global context dimensions, forming a robust multi-scale feature representation.

Compared to the original Swin Transformer, Swin Transformer V2 [28] introduces three key improvements to enhance image feature extraction. First, it adopts residual post-normalization by moving Layer Normalization to the end of each residual block, which stabilizes training across deep layers. Second, it replaces dot-product attention with scaled cosine attention, where the similarity between two pixels i and j is computed as

$$\text{Sim}(\mathbf{q}_i, \mathbf{k}_j) = \frac{\cos(\mathbf{q}_i, \mathbf{k}_j)}{\tau} + B_{ij}, \quad (1)$$

where B_{ij} denotes the relative position bias and τ is a learnable temperature parameter for each attention head. Finally, Swin Transformer V2 introduces a log-spaced continuous position bias using a meta-network over logarithmic coordinates, enabling robust transfer of pre-trained models across different window resolutions.

Fig. 2 illustrates the architecture of Swin Transformer V2. Its attention module incorporates the Log-CPB mechanism, where logarithmic relative coordinates Δx and Δy are processed by an MLP to generate a bias term b , and attention weights are computed using scaled cosine similarity between queries q and keys k . The resulting attention scores are normalized by a softmax function and applied to values v to produce transformed features. Along the main pathway, features pass through attention, layer normalization, and MLP blocks with residual connections, enabling efficient modeling of both local details and global context across multiple scales.

3.3 General Text Embeddings

Text embeddings map textual descriptions into numerical vectors that capture semantic meaning, enabling few-shot keypoint detection by linking language (e.g., “left shoulder”) with visual features [32]. Models such as CLIP [19] learn semantic relationships between joint descriptions, allowing text features to guide visual understanding even for unseen keypoints. GTE-base [29] is a text embedding model built on BERT [33], enhanced with Rotary Position Embedding [34] and Gated Linear Units [35] to improve representation capacity. It is trained using a multi-stage

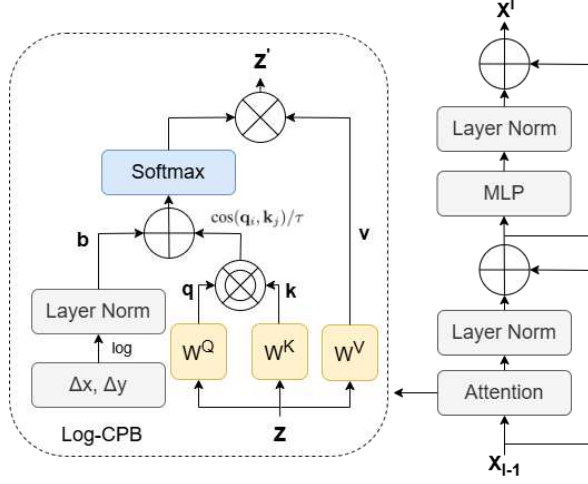


Fig. 2 The Swin Transformer V2 Architecture

contrastive learning strategy, combining large-scale unsupervised pretraining with supervised fine-tuning on annotated language tasks. At its core, GTE-base relies on the InfoNCE loss [36] (Equation 2) to pull semantically related text embeddings closer while pushing irrelevant ones apart.

$$L_{cl} = -\log \frac{e^{s(q, d^+)/\tau}}{e^{s(q, d^+)/\tau} + \sum_{i=1}^n e^{s(q, d_i^-)/\tau}}, \quad (2)$$

GTE-base-en-v1.5 is an upgraded version of the original GTE-base embedding model. Building upon the model architecture of GTE-base, GTE-base-en-v1.5 extends the context length capacity, increasing it to support up to 8192 tokens, enabling the processing of much longer documents and complex semantic relationships. The model has been specifically optimized for English content, providing better performance in English text processing.

3.4 CapeX

The CapeX model is modified from the GraphCape [6] model, adopting GraphCape’s structure-aware architecture while introducing text comprehension capabilities. In the CapeX approach, features initially obtained from the image and text backbones are fed into a Transformer encoder. This encoder refines these features and uses self-attention mechanisms to fuse the image and text representations. Subsequently, the refined features are sent to the Proposal Generator [7], which calculates similarity heatmaps between the query image and support text features to generate initial keypoint proposals. Finally, these initial proposals, along with the refined query image features, are passed to the Graph Transformer decoder. Crucially, this decoder also utilizes the

adjacency matrix from the support text-graph, integrating the explicit structural relationships between keypoints through GCN [27] layers and attention mechanisms. This allows the model to predict precise keypoint locations.

4 Method

4.1 Proposed Model Architecture

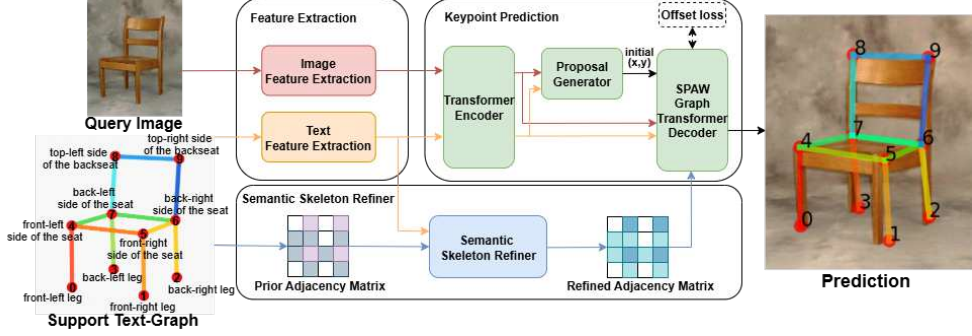


Fig. 3 Model architecture

The architecture of the proposed model is illustrated in Fig. 3, which is divided into three modules: Feature Extraction, Semantic Skeleton Refiner, and Keypoint Prediction. First, in the Feature Extraction module, the input query image and auxiliary text-graph information is processed in parallel to extract image and text features compatible with the Transformer architecture. Next, in the Semantic Skeleton Refiner module, the skeleton of the support text-graph is processed to extract skeleton features. Then, the skeleton and text features are used to analyze the relationships between keypoint features, providing structural guidance for the Graph Transformer Decoder. Finally, in the Keypoint Prediction module, a Transformer-based method is employed to fuse image and text features, subsequently understanding the feature content and integrating spatial awareness and graph structural information to predict keypoint locations. The key components and their detailed mechanisms will be elaborated on in subsequent sections.

4.1.1 Feature Extraction

Given a query image and a support text-graph, we perform parallel feature extraction on both. The detailed architecture is shown in Fig. 4. For the query image, we utilize SwinTransformerV2 [28] as the image backbone to extract the initial image features $I_q \in \mathbb{R}^{B \times C \times H \times W}$, where B , C , H and W represent the batch size, the number of output channels from the image backbone, and the height and width of the feature map, respectively. For the textual description, we use gte-base-en-v1.5 [37] as our text backbone. We freeze the parameters of the gte backbone, thereby extracting the initial

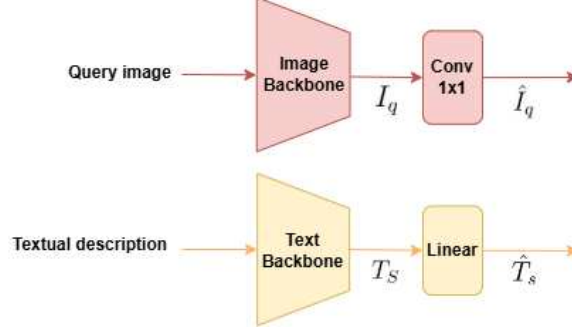


Fig. 4 Feature Extraction architecture

embedding features of the text $T_s \in \mathbb{R}^{B \times N_{kp} \times D_{text}}$, where B is the batch size, N_{kp} represents the maximum number of keypoint descriptions per sample in the batch, and D_{text} is the dimensionality of the embeddings of the feature output by the text backbone. To maintain consistency in the dimensionality of the initial image and text embedding features, we use a 1×1 convolutional layer for the image features and a linear layer for the text features, to produce features $\hat{I}_q$ and $\hat{T}_s$ of the same dimension. This approach ensures that the dimensions of both feature sets are aligned with the architecture of the Transformer encoder.

4.1.2 Semantic Skeleton Refiner

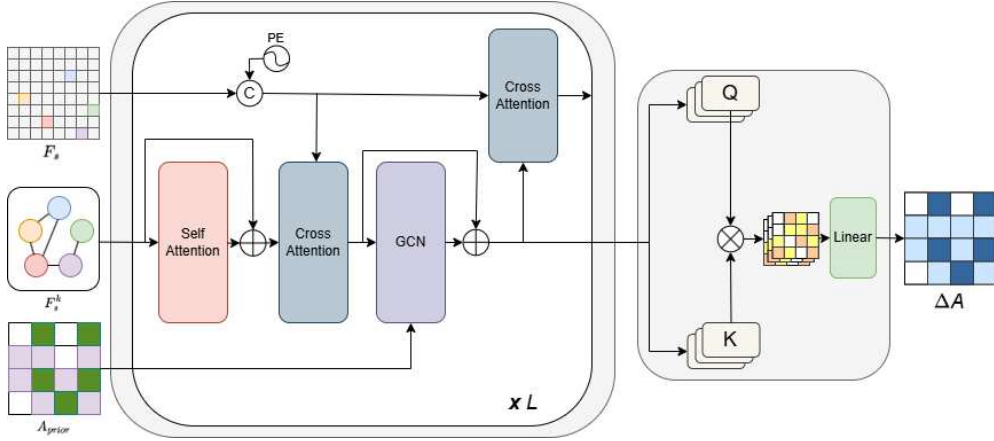


Fig. 5 The skeleton predictor model architecture in EdgeCape [5].

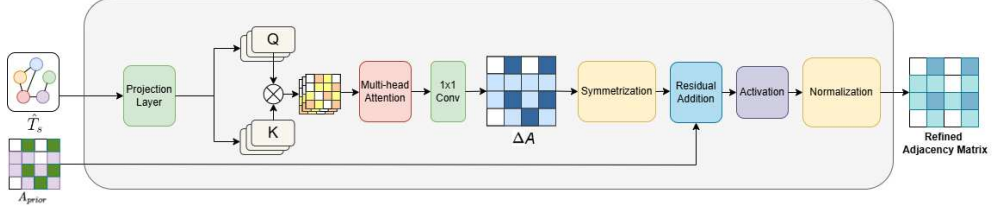


Fig. 6 The modified semantic skeleton refiner module.

The design of our Semantic Skeleton Refiner module is inspired by the EdgeCape [5] (Fig. 5). The architecture of the Semantic Skeleton Refiner module is illustrated in Fig. 6. First, the skeleton from the support text-graph is processed by translating its defined keypoints and their interconnecting edges into a binary adjacency matrix, which is then normalized, to generate the ground truth adjacency matrix A_{prior} . Besides, text features $\hat{T}_s$ from Feature Extraction are served as another input in Semantic Skeleton Refiner module. Within the module, text features $\hat{T}_s$ are passed through the query and key projection layers simultaneously, and matrix multiplication is then performed to obtain an attention score matrix, where each element represents the relevance between corresponding keypoints. The results are fed into Multi-head attention. Multi-Head Attention [7] expands upon Self-Attention by utilizing a parallel collection of Self-Attention modules, with each head employing different learning parameters, allowing the model to focus on relationships within sequences from various perspectives. The expression for each head is shown in Equation 3.

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

Finally, these outputs are concatenated and integrated through another linear transformation W^O to combine information from all heads. Its mathematical expression is shown in Equation 4.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O, \quad (4)$$

where the projection matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_k}$ and $W_i^O \in \mathbb{R}^{d_{model} \times d_k}$. The mathematical expression of Seft-attention is shown in Equation 5.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (5)$$

where Q, K, V represent the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors. Through a multi-head attention mechanism, we capture complex relationships between keypoints from different perspectives, and utilize 1×1 convolution to fuse multiple attention heads into a initial predicted adjacency matrix,

represented by ΔA . To ensure the undirected nature of the skeleton, we subsequently apply symmetrization to the predicted matrix as shown in Equation 6.

$$A' = \frac{\Delta A + \Delta A^T}{2}, \quad (6)$$

where A' denotes the predicted adjacency matrix. Using a residual learning strategy, A' is added to A_{prior} . The activation function then performs a non-linear transformation on the resulting sum, yielding the predicted unnormalized adjacency matrix, denoted as $A_{unnormalized}$, as shown in Equation 7

$$A_{unnormalized} = \text{Activation}(A' + A_{prior}). \quad (7)$$

Subsequently, all connections corresponding to invalid keypoints in $A_{unnormalized}$ are set to zero. Finally, we perform row normalization to convert the adjacency matrix into a probability distribution. The refined adjacency matrix is ultimately used in the Spatial-Aware (SPAW) Graph Convolutional Network for feature propagation, enabling the model to dynamically adjust the skeletal structure according to different object types and poses.

4.1.3 Keypoint Prediction

The design of the Keypoint Prediction module is based on CapeX [8]. In Keypoint Prediction, the precise coordinates for predefined keypoints within a given query image are generated. The module consists of three main components: Transformer Encoder, Proposal Generator, and Spatial-Aware (SPAW) Graph Transformer Decoder. Each component is described as follows.

4.1.4 Transformer Encoder

The Transformer Encoder consists of three stacked transformer blocks that jointly process image and text features. The image features $\hat{I}_q$ are flattened into a sequence and concatenated with the text features $\hat{T}_s$, together with their corresponding positional and order encodings. Through self-attention, the unified sequence enables cross-modal interaction, allowing visual features to be guided by textual cues and textual features to be grounded in visual context. After encoding, the enriched image and text features are separated and forwarded to the Proposal Generator and the SPAW Graph Transformer Decoder for subsequent processing.

4.1.5 Proposal Generator

Refined image features and refined text features, originating from the Transformer Encoder, are jointly supplied to the Proposal Generator. A computation of similarity between various image locations and the textual descriptions is performed by Proposal Generator. Through a combination of said similarity scores with spatial coordinate information, a preliminary position estimate for the keypoints is inferred, leading to

the generation of an initial 2-dimensional coordinate proposal (x, y) for each keypoint description.

4.1.6 Spatial-Aware Graph Transformer Decoder

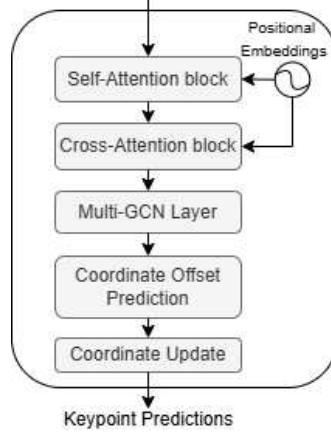


Fig. 7 SPAW Graph Transformer Decoder architecture

The SPAW Graph Transformer Decoder component combines the input features, refines the keypoint locations, and predicts the final keypoint coordinates. The architecture diagram of SPAW Graph Transformer Decoder is shown in Fig. 7. Its details will be introduced as follows.

1. **Inputs:** The decoder takes as input support features T_s representing encoded keypoint descriptions from the Feature Extraction, query features I_q from the Feature Extraction, initial coordinate proposals from the Proposal Generator, and the refined adjacency matrix from the Semantic Skeleton Refiner that encodes structural relationships between keypoints. In addition, the initial coordinate proposals are converted to positional embeddings from the positional encoding, which are provided to incorporate spatial awareness.
2. **SPAW Graph Transformer Decoder Layer:** The SPAW Graph Transformer Decoder Layer is composed of three transformer decoder blocks. Every block is composed of five items.
 - **Self-Attention on Keypoint Features:** Textual features corresponding to keypoints are oriented to each other, conditioned by their current positional embeddings and image features. This self-attention mechanism allows keypoints to update their features based on the state of other related keypoints, facilitating intra-keypoint relationship modeling.
 - **Cross-Attention to Image Features:** Keypoint features, served as queries and concatenated with their positional embeddings, attend to the image features. For this attention mechanism, the image features, also concatenated with their

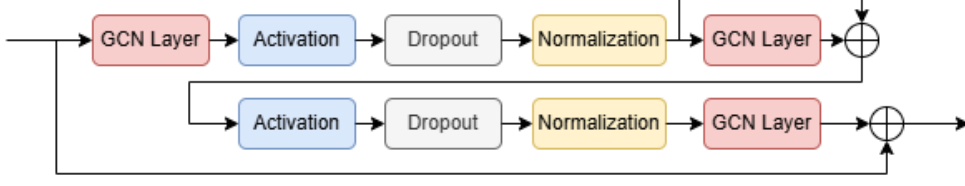


Fig. 8 Multi-GCN architecture

positional embeddings, provide the corresponding keys, while the original image features serve as values. This enables each keypoint to gather relevant visual information from the image, grounding its representation in visual evidence.

- **Spatial-Aware Multi-GCN Layer:** As shown in Fig. 8, the multi-GCN module consists of three stacked GCN layers with both local and global residual connections. The first layer expands feature dimensions, the second refines features with a residual shortcut for stability, and the third compresses them back to the original dimension while adding a global residual from the input. This design enables deeper graph feature learning with stable optimization. Each GCN layer aggregates neighborhood information via a one-dimensional convolution and Einstein summation with the adjacency matrix, while a spatial gate incorporates 2D keypoint coordinates to introduce spatial awareness (Equation 8).

$$W_s = \text{Sigmoid}(\text{Linear}_{\text{down}}(\text{GELU}(\text{Linear}_{\text{up}}(x, y)))), \quad (8)$$

where (x, y) represents the 2-dimensional coordinates of a keypoint. The spatial gate first expands the 2-dimensional coordinates to 32 dimensions via $\text{Linear}_{\text{up}}$, applies GELU activation, then compresses back to a single weight value through $\text{Linear}_{\text{down}}$ and sigmoid activation. The resulting spatial weight W_s is then applied to modulate the input features through element-wise multiplication in Equation 9.

$$\mathbf{x}' = \mathbf{x} \odot W_s, \quad (9)$$

where $\mathbf{x}$ represents the original node features and $\odot$ denotes element-wise multiplication. This spatial gating mechanism enables the decoder layers to adaptively emphasize or suppress features based on their geometric positions.

- **Coordinate Offset Prediction:** Each decoder layer employs a dedicated MLP head to predict an offset for the current keypoint coordinates. This offset represents the predicted adjustment needed to move the keypoint proposal closer to its true location.
- **Coordinate Update:** The predicted coordinate offsets are used to update the current coordinate proposals. An update to the coordinates typically involves first applying an inverse sigmoid transformation to the current coordinate values. The predicted offset is then added to these unsigmoided coordinate values. Finally,

Table 1 Hardware specifications.

Component	Specification
CPU	Intel Xeon E5-2698 v4 @ 2.20 GHz
GPU	NVIDIA Tesla V100-SXM2
GPU memory	16 GB
VRAM	16 GB
Shared memory	40 GB

a sigmoid transformation is applied to the resulting sum. Such a sequence of operations ensures coordinates remain normalized, thereby iteratively refining the keypoint locations layer by layer.

3. **Final Keypoint Predictions:** The decoder ultimately produces the final predicted (x, y) coordinates for each keypoint. These coordinates represent the model’s best estimate of the location for each predefined keypoint in the input query image.

4.1.7 Loss Calculation

The training objective employs an L1 coordinate regression loss between the predicted and ground-truth keypoints. To account for occluded or non-visible keypoints, each term is weighted by a target mask so that only valid keypoints contribute to the loss. Since the SPAW Graph Transformer Decoder produces intermediate predictions across its L layers, a deep supervision strategy is adopted. Specifically, a proposal loss ($\mathcal{L}_{\text{proposal}}$) is first computed for the initial keypoint proposals, while the final coordinate loss ($\mathcal{L}_{\text{coord}}$) aggregates the L1 losses from all decoder layers, as defined in Equation 10.

$$\mathcal{L}_{\text{coord}} = \sum_{l=1}^L \mathcal{L}_{\text{L1}}^{(l)}. \quad (10)$$

The total training loss combines the proposal loss and the coordinate loss, calculated as Equation 11.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{proposal}} + \mathcal{L}_{\text{coord}}. \quad (11)$$

5 Experiment

5.1 Experimental Environment

The experimental environment includes both software and hardware configurations. For the hardware setup, we used the server which consists of an Intel(R) Xeon(R) CPU E5-2698 v4 @2.20GHz along with Tesla V100-SXM2 GPU with 16GB of VRAM and 40GB of shared memory. Detailed summary of hardware specifications is provided in Table 1. For the software configuration, the operating system was Ubuntu 22.04.5 LTS. The model was implemented using PyTorch 2.2.1 and CUDA 12.6.

5.2 Dataset

To evaluate our model for pose estimation, we follow previous CAPE task studies [4–6, 8] and use the MP-100 dataset, which was proposed by Xu et al. [4]. This dataset is a large-scale benchmark for category-level object pose estimation, composed of over 18,000 images across 100 distinct object categories under 8 super-categories (human hand, human face, human body, animal body, animal face, clothes, furniture, and vehicle). These images are aggregated from several existing 2D pose estimation datasets, including COCO [4], 300W [4], AFLW [4], DeepFashion2 [4], OneHand10K [4], AP-10K [12], MacaquePose [4], Desert Locust [4], Vinegar Fly [4], CUB-200 [4], AnimalWeb [4], CarFusion [4], and Keypoint-5 [4]. A key characteristic of MP-100 is that the keypoint definitions are not fixed but vary across categories, challenging a model’s ability to generalize to novel object classes.

We evaluate generalization on the MP-100 dataset using a five-split protocol, where each split contains all 100 categories and uses disjoint sets of 70, 10, and 20 categories for training, validation, and testing, respectively. This ensures that testing is performed on unseen categories while covering all objects across splits. Skeleton definitions follow an updated CapeX specification, providing unified structures and concise text descriptions that are further standardized for consistency.

5.3 Evaluation Metrics

To evaluate the performance of our model, we use the Probability of Correct Keypoint (PCK) metric, a widely adopted standard for the category-level object pose estimation proposed by Yang and Pavone [38]. Following their definition, a predicted keypoint is deemed correct if it falls within a distance criterion of its corresponding ground-truth location. The distance D is defined as Equation 12.

$$D = \alpha \cdot \max(h, w), \quad (12)$$

where α is a tolerance threshold and $\max(h, w)$ serves as a normalization factor corresponding to the object scale, defined by the maximum between the height and width of the bounding box. Let $\mathbf{p}_i$ and $\mathbf{p}_i^*$ be the predicted and ground-truth coordinates, respectively, for the i -th keypoint. The final PCK score is the percentage of correctly predicted keypoints. As formulated in Equation 13.

$$\text{PCK} = \frac{\sum_{i=1}^N \mathbf{1}(\|\mathbf{p}_i - \mathbf{p}_i^*\|_2 \leq D)}{N} \quad (13)$$

In this equation, the indicator function $\mathbf{1}(\cdot)$ is used to count each keypoint where the Euclidean distance between prediction $\mathbf{p}_i$ and ground truth $\mathbf{p}_i^*$ is less than or equal to the normalized distance D . This sum is then divided by the total number of keypoints, N . This metric, through its tunable threshold α , allows for a nuanced analysis of performance across a spectrum of precision requirements.

5.4 Parameter Configuration

All five splits of the MP-100 dataset share the same training configuration. The model is trained for 200 epochs with a batch size of 14 per GPU using the AdamW optimizer, an initial learning rate of 1×10^{-5} , and a weight decay of 0.04. To reduce overfitting, a dropout rate of 0.1 is applied in the Keypoint Prediction and Semantic Skeleton Refiner modules. The learning rate follows a step decay schedule, reduced by a factor of 0.3 at epochs 140, 160, and 180, with a linear warmup applied during the first 1000 iterations to ensure stable convergence.

5.5 Experimental Results and Analysis

5.5.1 Model Performance

In this section, we will present and analyze the experimental results of our proposed model in comparison with several state-of-the-art methods on the category-agnostic pose estimation task. Our evaluation is conducted using the comprehensive MP-100 dataset. The comparison across its five test splits follows the setup detailed in Section 5.2. The performance is measured using PCK metric at a distance threshold of 0.2, a common convention established in previous works.

Table 2 PCK@0.2 results on the MP-100 benchmark under the 1-shot setting. Best results are shown in bold, and second-best results are underlined.

Model	Split1	Split2	Split3	Split4	Split5	Avg
Capeformer [7]	89.45	84.88	83.59	83.53	85.09	85.31
X-Pose [39]	89.07	85.05	85.26	85.52	85.79	86.14
PPM+CPT [40]	91.03	88.06	84.48	86.73	87.40	87.54
EdgeCape [5]	93.69	89.27	87.85	86.67	87.59	89.01
GraphCape [6]	94.73	89.79	90.69	88.09	90.11	90.68
CapeX [8]	<u>95.54 $\pm$ 0.03</u>	<u>90.98 $\pm$ 0.05</u>	88.71 $\pm$ 0.02	<u>88.71 $\pm$ 0.29</u>	<u>92.76 $\pm$ 0.02</u>	<u>91.34</u>
Our work	96.00 $\pm$ 0.00	91.42 $\pm$ 0.00	<u>89.72 $\pm$ 0.02</u>	89.78 $\pm$ 0.09	93.51 $\pm$ 0.05	92.09

Table 2 presents the PCK@0.2 results of various CAPE models on the MP-100 benchmark under a 1-shot setting, across five different data splits and their average performance. Note that the results of CapeX [8] are obtained by training and running the model from their source code. For the other methods [5–7, 39, 40], we cite the results from their original publications.

Our proposed model consistently demonstrates superior performance across all splits, achieving the highest average PCK@0.2 score of 92.09. This represents a notable improvement over existing state-of-the-art methods. Given that the object categories evaluated in each split were unseen during training, our method demonstrates superior keypoint localization precision and excellent generalization ability. Specifically, our model outperforms CapeX [8], which previously held the leading position, by a margin of approximately 0.75% on average.

Table 3 PCK@0.1 results on the MP-100 benchmark under the 1-shot setting. Best results are shown in bold, and second-best results are underlined.

Model	Split1	Split2	Split3	Split4	Split5	Avg
Capeformer [7]	75.17	68.53	67.42	66.98	71.13	69.85
X-Pose [39]	—	—	—	—	—	—
PPM+CPT [40]	—	—	—	—	—	—
EdgeCape [5]	83.92	75.01	74.36	73.27	75.19	76.35
GraphCape [6]	85.06	77.45	<u>77.48</u>	76.39	79.89	79.25
CapeX [8]	<u>87.19 ± 0.09</u>	<u>78.54 ± 0.10</u>	76.85 ± 0.01	<u>78.71 ± 0.15</u>	<u>83.02 ± 0.12</u>	<u>80.86</u>
Our work	88.65 ± 0.04	79.99 ± 0.01	78.51 ± 0.06	80.60 ± 0.04	84.57 ± 0.52	82.46

Analyzing the individual splits, our model achieved the highest score in the split1, reaching 96.00. Similarly, it led in Split2 with 91.42, in Split4 with 89.78, and in Split5 with 93.51. These results highlight our model’s superior performance across most object categories represented in the MP-100 dataset. However, in Split3, GraphCape [6] achieved the highest score of 90.69, while our model secured the second-best performance with 89.72, a minor difference from the PCK score of 0.97%. Despite this specific characteristic in Split3, the performance of our model in the object categories of the other splits was notably superior to GraphCape [6]. Furthermore, our model offers the key advantages of not requiring support images and seamlessly integrating textual information, differentiating it from GraphCape [6]. The potential reason for this specific performance characteristic in Split3 could be that in certain specific data categories, when encountering particularly challenging or complex nominal descriptions for keypoints, the fusion of our method’s textual and image features might result in less robust spatial localization capability compared to purely image-assisted models. Nevertheless, our method maintains strong and competitive results even in such challenging splits.

Table 3 presents the PCK@0.1 results of various CAPE models on the MP-100 benchmark under a 1-shot setting, across five different data splits and their average performance. As presented in the table, the scores for X-Pose [39] and PPM+CPT [40] are left blank, as the code for X-Pose does not support retraining on the MP-100 dataset, and the code for PPM+CPT was not provided. To focus our comparison on the current state-of-the-art method, we obtained the results for CapeX [8] by training and running their publicly available source code. Our model achieves an average accuracy of 82.46% under the strict PCK@0.1 metric. This result marks a significant improvement of 1.60% over the previous state-of-the-art model, CapeX [8], which scored 80.86%, with consistent gains across every split. Outperforming the leading method on this more demanding evaluation, which uses a tighter tolerance for correct keypoint localization, highlights our model’s enhanced capability for fine-grained precision.

In summary, this comprehensive evaluation confirms the effectiveness of our proposed model. Its consistent superiority across different precision thresholds, from the standard PCK@0.2 to the more stringent PCK@0.1, highlights its robustness in

advancing category-agnostic pose estimation and its strong generalization ability in 1-shot settings.

5.5.2 Visual Comparison

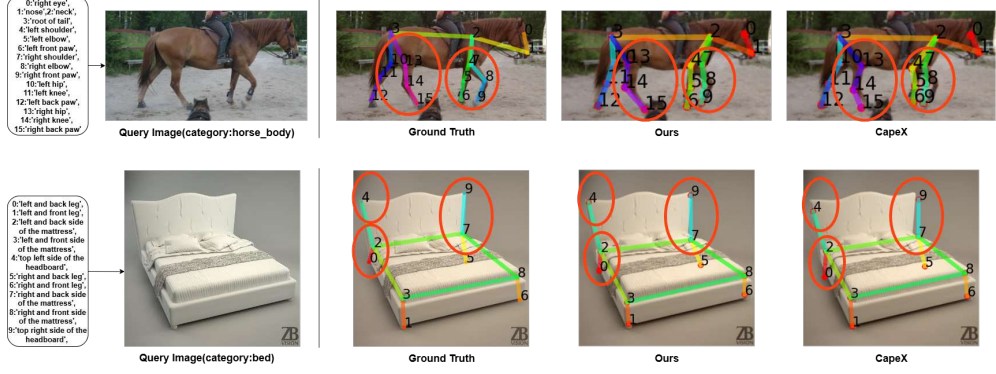


Fig. 9 Visual comparison on MP-100

Fig. 9 presents a visual comparison between our model and CapeX [8] on two examples from the MP-100 dataset, representing different categories of objects unseen during training. As highlighted by the red circles in the figures, it is clear that our method, guided by supplementary text, achieves more precise localization for the corresponding keypoints on the image. This enhanced precision is particularly evident in challenging areas for both animate and inanimate objects, such as the horse’s limbs and the legs and headboard of the bed, where each keypoint is clearly identified. Consequently, the model’s grasp of the object’s overall fine-grained structure and positioning is more faithful to the ground truth. This result underscores our model’s enhanced robustness in interpreting complex skeletal structures, leading to more precise and reliable keypoint detection.

5.6 Ablation Studies

In this subsection, we validate the effectiveness of our proposed technique by performing ablation studies on split1 of the MP-100 dataset, using PCK@0.2 as the evaluation metric.

To evaluate each component, we construct a baseline model with all three modules disabled: the multi-GCN is replaced by a single GCN layer in each Graph Transformer Decoder, and both the Semantic Skeleton Refiner and Spatial Gate are removed. The results, detailed in Table 4, demonstrate the contribution of each component to the overall performance of the model. The Semantic Skeleton Refiner proves to be the most critical module, increasing performance to 95.81%, highlighting the vital role of dynamically refining the graph structure. The Spatial Gate is the second most impactful component, providing a 0.21% gain to 95.75% and underscoring the value of

Table 4 Ablation study on Split1 under the 1-shot setting. Best performance is shown in bold.

Ablation settings			MP-100 (Split1)
Semantic skeleton refiner	Multi-GCN	Spatial gate	PCK@0.2 (%)
—	—	—	95.54 $\pm$ 0.03
—	✓	—	95.75 $\pm$ 0.01
—	—	✓	95.75 $\pm$ 0.04
—	✓	✓	95.78 $\pm$ 0.01
✓	—	—	95.81 $\pm$ 0.01
✓	✓	—	95.86 $\pm$ 0.01
✓	—	✓	95.92 $\pm$ 0.00
✓	✓	✓	96.00 $\pm$ 0.00

spatially-aware feature refinement. In particular, it shows the strongest synergy when combined with the refiner, achieving the best two-component performance of 95.92%. Although the multi-GCN is still beneficial, it provides a smaller performance increase of 0.21% to reach 95.75%. These results clearly demonstrate a hierarchy of importance among the components.

6 Conclusion

In this paper, we propose a text-guided framework for dynamic structure prediction in category-agnostic pose estimation. The proposed Semantic Skeleton Refiner enables the model to infer object skeletons directly from textual cues, overcoming the limitations of static structural priors. We further introduce a Spatial-Aware Graph Transformer Decoder that enhances geometric reasoning among keypoints through multi-layer GCNs and a Spatial Gate mechanism. Extensive experiments on the MP-100 benchmark demonstrate state-of-the-art performance, and ablation studies confirm the effectiveness of each component. Future work will extend this framework to 3D pose estimation and explore real-world applications such as robotics, augmented reality, and video-based pose tracking.

References

- [1] 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 TPAMI* **45**, 7157–7173 (2023)
- [2] Li, W., Wang, Z., Yin, B., Peng, Q., Du, Y., Xiao, T., Yu, G., Lu, H., Wei, Y., Sun, J.: Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv: 1901.00148* (2019)
- [3] Geng, Z., Sun, K., Xiao, B., Zhang, Z., Wang, J.: Bottom-up human pose estimation via disentangled keypoint regression. *CVPR* (2021)

- [4] 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: ECCV, pp. 398–416 (2022). Springer
- [5] Hirschorn, O., Avidan, S.: Edge weight prediction for category-agnostic pose estimation. arXiv preprint arXiv: 2411.16665 (2024)
- [6] Hirschorn, O., Avidan, S.: A graph-based approach for category-agnostic pose estimation. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O. (eds.) Computer Vision – ECCV 2024. Lecture Notes in Computer Science, vol. 15093, pp. 469–485. Springer, ??? (2024). https://doi.org/10.1007/978-3-031-73036-8_27
- [7] Shi, M., Huang, Z., Ma, X., Hu, X., Cao, Z.: Matching is not enough: A two-stage framework for category-agnostic pose estimation. In: CVPR, pp. 7308–7317 (2023)
- [8] Rusanovsky, M., Hirschorn, O., Avidan, S.: Capex: Category-agnostic pose estimation from textual point explanation. In: The TACL (2025)
- [9] Wang, B., Wu, Q., Wang, H., Hu, L., Li, B.: 3d surface reconstruction of car body based on any single view. IEEE Access **12**, 74903–74914 (2024)
- [10] Li, J., Bian, S., Zeng, A., Wang, C., Pang, B., Liu, W., Lu, C.: Human pose regression with residual log-likelihood estimation. In: Proc. CVPR, pp. 11025–11034 (2021)
- [11] Luo, Z., Wang, Z., Huang, Y., Wang, L., Tan, T., Zhou, E.: Rethinking the heatmap regression for bottom-up human pose estimation. In: Proc. CVPR, pp. 13264–13273 (2021)
- [12] Yu, H., Xu, Y., Zhang, J., Zhao, W., Guan, Z., Tao, D.: Ap-10k: A benchmark for animal pose estimation in the wild. NeurIPS Datasets and Benchmarks (2021)
- [13] Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose++: Vision transformer for generic body pose estimation. arXiv preprint arXiv: 2212.04246 (2022)
- [14] Kim, J., Chung, H., Kim, B.-H.: Capellm: Support-free category-agnostic pose estimation with multimodal large language models. In: IEEE/CVF ICCV, pp. 22889–22898 (2025)
- [15] Wang, H., He, Q., Peng, J., Yang, H., Chi, M., Wang, Y.: Mamba-yolo-world: Marrying yolo-world with mamba for open-vocabulary detection. arXiv preprint arXiv: 2409.08513 (2024)
- [16] Yang, S.-D., Su, H.-T., Hsu, W.H., Chen, W.-C.: Class-agnostic few-shot object counting. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 870–878 (2021)

- [17] Lan, M., Chen, C., Ke, Y., Wang, X., Feng, L., Zhang, W.: Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. ECCV (2024)
- [18] Wysoczańska, M., Siméoni, O., Ramamonjisoa, M., Bursuc, A., Trzeciński, T., Pérez, P.: Clip-dinoiser: Teaching clip a few dino tricks for open-vocabulary semantic segmentation. arXiv preprint arXiv: 2312.12359 (2023)
- [19] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. ICML (2021)
- [20] Zhang, H., Xu, L., Lai, S., Shao, W., Zheng, N., Luo, P., Qiao, Y., Zhang, K.: Open-vocabulary animal keypoint detection with semantic-feature matching. International Journal of Computer Vision, 1–18 (2024)
- [21] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Message passing neural networks. In: Machine Learning Meets Quantum Physics. LNIP, vol. 968, pp. 199–214. Springer, Online via Zoom (2020). https://doi.org/10.1007/978-3-030-40245-7_11
- [22] Zhang, X., Xu, Y., He, W., Guo, W., Cui, L.: A comprehensive review of the over-smoothing in graph neural networks. In: Computer Supported Cooperative Work and Social Computing, pp. 451–465. Springer, Singapore (2024)
- [23] Topping, J., Giovanni, F.D., Chamberlain, B.P., Dong, X., Bronstein, M.M.: Understanding over-squashing and bottlenecks on graphs via curvature. In: The Tenth ICLR, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, ??? (2022)
- [24] Yuan, C., Zhao, K., Kuruoglu, E.E., Wang, L., Xu, T., Huang, W., Zhao, D., Cheng, H., Rong, Y.: A Survey of Graph Transformers: Architectures, Theories and Applications (2025). <https://arxiv.org/abs/2502.16533>
- [25] Dwivedi, V.P., Bresson, X.: A generalization of transformer networks to graphs. arXiv preprint arXiv: 2012.09699 (2020)
- [26] Kreuzer, D., Beaini, D., Hamilton, W.L., Létourneau, V., Tossou, P.: Rethinking graph transformers with spectral attention. arXiv preprint arXiv: 2106.03893 (2021)
- [27] Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y.: Simple and deep graph convolutional networks. In: Proc. ICML. Proceedings of Machine Learning Research, vol. 119, pp. 1725–1735. PMLR, Vienna, Austri (2020)
- [28] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., Guo, B.: Swin transformer v2: Scaling up capacity and resolution. arXiv preprint arXiv: 2111.09883 (2021)

- [29] Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., Zhang, M.: Towards General Text Embeddings with Multi-stage Contrastive Learning (2023)
- [30] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2020)
- [31] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv: 2103.14030 (2021)
- [32] Lu, C., Liu, Z., Koniusz, P.: Openkd: Opening prompt diversity for zero- and few-shot keypoint detection. In: ECCV 2024. Lecture Notes in Computer Science (LNCS). Springer, Cham, Switzerland (2024)
- [33] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. NAACL (2019)
- [34] Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
- [35] Shazeer, N.: Glu variants improve transformer. arXiv preprint arXiv: 2002.05202 (2020)
- [36] Wang, Z., Xu, B., Yuan, Y., Shen, H., Cheng, X.: Infonce is a free lunch for semantically guided graph contrastive learning. In: ACM SIGIR 25, pp. 719–728. Association for Computing Machinery, New York, NY, USA (2025). <https://doi.org/10.1145/3726302.3730007>
- [37] Zhang, X., Zhang, Y., Long, D., Xie, W., Dai, Z., Tang, J., Lin, H., Yang, B., Xie, P., Huang, F., Zhang, M., Li, W., Zhang, M.: mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval (2024)
- [38] Yang, H., Pavone, M.: Object pose estimation with statistical guarantees: Conformal keypoint detection and geometric uncertainty propagation. In: CVPR, pp. 8947–8958 (2023)
- [39] Yang, J., Zeng, A., Zhang, R., Zhang, L.: X-pose: Detecting any keypoints. ECCV (2023)
- [40] Peng, D., Zhang, Z., Hu, P., Ke, Q., Yau, D.K.Y., Liu, J.: Harnessing text-to-image diffusion models for category-agnostic pose estimation. In: ECCV (13), pp. 342–360 (2024)