PoseRAC: Enhancing Repetitive Action Counting with Salient Poses

Ziyu Yao(✉) and Yuexian Zou

School of Electronic and Computer Engineering, Peking University, Beijing, China
yaozy@stu.pku.edu.cn, zouyx@pku.edu.cn

Abstract. Repetitive action counting aims to count the number of periodic actions in a video, offering significant application values for human activities. However, this task has not been extensively explored, and previous methods struggle to address the periodic representation. Through an analysis of the relationship between human poses and actions, we present a novel concept called **Salient Pose**, which effectively represents each action. By further linking these salient poses and repetitive actions, we introduce a new approach for repetitive action counting called **PoseRAC**, to model the relationship between salient pose and actions and complete the counting task based on salient poses. Leveraging the foundation generative models, our model can perform zero-shot predictions without using any training set. Furthermore, by incorporating an off-the-shelf text encoder, our model can count unseen actions in an open-set setting. Our approach achieves state-of-the-art performance on three mainstream benchmarks: RepCount, UCFRep, and Countix.

Keywords: Repetitive Action Counting · Human Pose Estimation · Foundation Generative Model

1 Introduction

Periodic movement is widespread in nature, encompassing various human activities. In computer vision, the detection of periodic human actions plays a crucial role, such as counting physical exercise movements for effective fitness planning. Additionally, counting repetitive/periodic movements is essential for analyzing human actions, including tasks such as pedestrian detection [24], camera calibration [14], and 3D reconstruction [18]. Given the significance of accurately counting periodic movements, academia proposes a task named "repetitive action counting", which outputs the number of any repetitive action in a video.

Despite its importance, the field of repetitive action counting has not yet been fully explored. Existing methods [10,13,31] have mainly relied on intricate and high-cost temporal feature modeling to capture periodic patterns in videos. Typically, they pre-calculate the heat map on each frame across a video, then regress the counting value, often by counting the peak points in these heat

M. Mahmud et al. (Eds.): ICONIP 2024, LNCS 15290, pp. 255–270, 2025.
https://doi.org/10.1007/978-981-96-6588-4_18

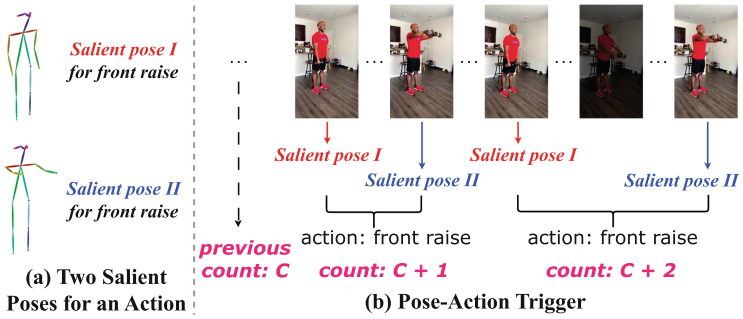


Fig. 1. Our proposed Pose-Action Trigger can perform action counting. Specifically, if two salient poses of the current action appear sequentially, it means that this action is occurring, and the counting value can be incremented by one.

maps. However, these methods often overlook two key characteristics of repetitive action counting, leading to suboptimal results. 1) Videos vary in length. For longer videos, which means more frames, calculating the heatmap frame-by-frame incurs a significant computational cost. Additionally, global modeling can introduce a lot of noise, leading to inaccurate counting. 2) The cycle lengths of different types of actions also vary. When determining whether an action has completed a cycle, the network struggles to accommodate all types of actions, resulting in inaccurate counts. These two factors render temporal modeling less efficient in accurately representing periodic features in videos, suggesting the need for alternative approaches to tackle this task.

Meanwhile, research in human pose estimation [2,30] is advancing rapidly. We observe that poses are the most essential factors in an action and can remain unaffected by contextual nuisances, such as background and lighting changes. Thus, *an intuitive idea is linking human pose with action-related tasks*. While recent studies have utilized human poses in action recognition [8,34], it has not been explored in repetitive action counting task, due to the distinct differences between the two tasks. Action recognition requires coarse-grained classification results, whereas action counting demands the fine-grained repetitive counts of the action. One feasible solution is to extract the poses from each frames and then perform temporal modeling on these poses. Although computational costs may be reduced when working with poses compared to directly on video frames, such methods are still ineffective in addressing the characteristics of varying cycle lengths of actions. Thus, this inspires us to ask: *How to effectively link pose with action to achieve a breakthrough in repetitive action counting task?*

By observing the relationship between human poses and actions, we notice that each action consistently involves its own distinctive poses at certain moments, serving to differentiate it from other actions. We define these distinctive poses as **salient poses**, and pre-define them for each action by analyzing their pose characteristics and capturing their most distinct moments. For instance, as illustrated in Fig. 1(a), the front raise action includes salient pose

raising the arms, which distinguishes it from other actions, such as pull up or situp. To take a step further, considering all actions (more than fifty) within three mainstream benchmarks in the field of counting, we find each action can be distinguished by manually assigning two salient poses and using salient poses can significantly reduce the learning complexity. Moreover, once we have explicitly linked the two salient poses to each action, we can train the model to implicitly learn more intrinsic relationships between different actions and poses. When the model encounters new, unseen actions, it can still identify the salient poses of these actions, allowing it to generalize to unseen action classes.

Based on these findings, we propose a new perspective to represent and count actions based on the salient poses, called **Pose-Action Trigger**. Within this framework, as shown in Fig. 1(b), an action is counted when two salient poses appear in sequence, increasing the action counting value by one. In contrast to common methods that represent actions using redundant RGB frames, the key advantage of our framework lies in its selective use of specific frames to capture salient poses, simplifying action representation. This simple yet effective mechanism not only allows us to ignore irrelevant backgrounds and focus on the essential poses, but also significantly reduces computational costs.

Furthermore, we introduce the first pose-level network called **Pose** Saliency Network for **R**epetitive **A**ction **C**ounting (**PoseRAC**). It decouples the video-level action counting into two stages: **1) Pose-Action Modeling**, and **2) Pose-Action Trigger**. The second stage has been discussed in Fig. 1. In the first stage, we want the model to capture the relationship between the pose of each frame and its corresponding action, and infer each importance score of each frame on the action. To this end, we introduce Stable Diffusion [25] and ControlNet [32] to generate training data conditioned on our pre-defined salient poses and encourage the model to give higher scores on salient poses. This further allows us to achieve "zero-shot"[1] prediction, as we can train the model without using any real training set. Moreover, after training with a sufficient number of manually designed salient poses on specific (seen) actions, we want the model to have the potential to generalize to unseen actions. To achieve this, we additionally introduce the CLIP [22] Text Encoder to obtain semantic information about unseen actions and incorporate it into our model.

We summarize our contributions in three-fold:

- To the best of our knowledge, we are the first to introduce salient pose into the action counting task, and propose a two-stage framework, called PoseRAC, including Pose-Action Modeling and Pose-Action Trigger to efficiently represent and count actions with two salient poses.
- As for the Pose-Action Modeling, we only employ the generative model to generate pose-related training images, thus enabling zero-shot prediction. And we integrate the CLIP text encoder to incorporate additional action semantic information to facilitate the generalization of PoseRAC for open-set unseen actions.

[1] Starting from the CLIP [22], the term of zero-shot is used in a broader sense to study generalization to unseen datasets.

– Our PoseRAC achieves state-of-the-art performance on three mainstream action counting benchmarks: RepCount, UCFRep, and Countix, far outperforming all current methods.

2 Related Work

Repetitive Action Counting. Early works focus on compressing the motion field into one-dimensional signals to recover the repetition, such as Fourier analysis [3,21], peak detection [28], classification [7,16]. However, these methods cannot tackle non-stationary scenarios in the context of repetitive actions.

Recently, Context [31] proposes a context-aware regression network accompanied by a coarse-to-fine refinement strategy for varying action cycles. Meanwhile, RepNet [10] integrates a temporal self-similarity matrix into the process of counting. Moreover, [13] encodes multi-scale temporal correlations to handle both high- and low-frequency actions. In contrast to previous works, we introduce the pose modality into this task, opening up a new avenue for action counting.

Pose in Action-Related Tasks. Existing studies have explored various modalities for action-related tasks, such as RGB frames [6], optical flows [26], and audio waves [29]. While human pose has received growing interest due to its action-focusing nature and resistance to background nuisances, such as skeleton-based action recognition [8,34]. In this paper, we explore the relationship between pose and action to effectively address the action counting task.

Foundation Generative Model. Based on the diffusion model, Stable Diffusion [25], DALL-E2 [23], and others showcase impressive abilities in generating images based on text prompts. Recently, ControlNet [32] proposes a practical image editing solution, showing controllable image generation from various conditioning signals, such as depth, segmentation, human pose, etc.

3 Methodology

Given a video $\mathcal{V} = \{x_i\}_1^T \in \mathbb{R}^{C\times H\times W\times T}$ with T RGB frames, action counting model aims to predict a value $\mathcal{M}$, which is the number of repetitive actions.

3.1 Model Overview

Previous works mainly relied on high-cost temporal feature modeling, which ignores the action-focused nature of the human pose. So our starting point is to introduce poses to better represent actions and reduce complexity. As demonstrated in Fig. 2, our PoseRAC decouples the counting into two stages:

Stage 1. Pose-Action Modeling (Sect. 3.2 to Sect. 3.4). We model the relationship between the pose of each frame with their corresponding action classes. Specifically, the pose of each frame is extracted and then encoded to generate the

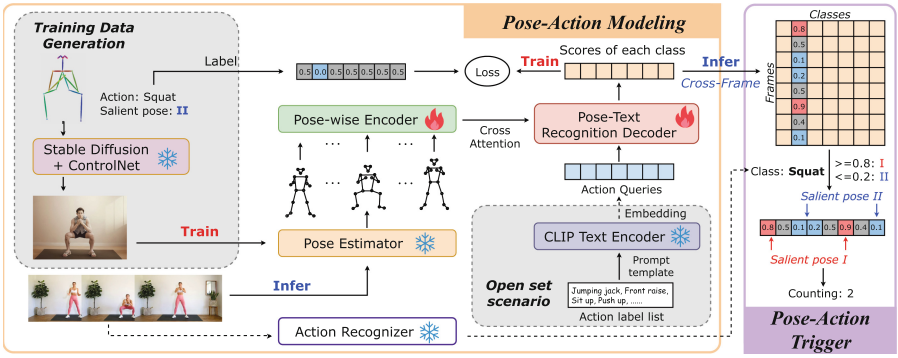


Fig. 2. PoseRAC is a two-stage action counter. In first stage, it models the relationship between salient poses and actions, where Stable Diffusion and ControlNet allow us to perform *zero-shot* prediction without using the training set, and the CLIP Text Encoder facilitates *open-set* generalization. In second stage, there is a training-free Pose-Action Trigger to perform video-level action counting.

latent embedding. The embedding is subsequently decoded using Action Queries via the cross-attention process, resulting in scores for each action class.

To obtain the training data containing salient poses, we use Stable Diffusion and ControlNet to generate sufficient data for training. This process allows us to train the model without accessing any samples from the training set, thus achieving *zero-shot prediction*. Moreover, we incorporate the semantic information of unseen action classes into the Action Queries to facilitate pose recognition for any unseen actions. This approach enhances the *open-set generalization*.

Stage 2. Pose-Action Trigger (Sect. 3.5). We apply an pre-trained Action Recognizer to obtain video-level action class, and combine pose recognition scores of each frame corresponding this class. We set two threshold to distinguish the salient pose I and II, and scan through all the frames. Each sequential appearance of two salient poses indicates an increment in the count value by one.

3.2 Pose-Action Modeling

Pose Estimator. For the given RGB frame x, a pre-trained pose estimator [4] converts it into pose keypoints sequence $p \in \mathbb{R}^{D\times K}$, where K represents the number of keypoints, and D is the dimension of each keypoint.

Pose-Wise Encoder. We encode each pose into latent embedding using a L-layer Transformer. For the pose $p \in \mathbb{R}^{D\times K}$, we further define it as $\{k_j\}_1^K$, where $k_j \in \mathbb{R}^D$, and embed it to obtain richer information via a MLP network $\mathbf{E}$:

$$\mathbf{Z}^0 = [\mathbf{E}(k_1), \mathbf{E}(k_2), \ldots, \mathbf{E}(k_K)]^T, \tag{1}$$

where $\mathbf{E}(k_j) \in \mathbb{R}^{D'}$ is the embedding. $\mathbf{Z}^0 \in \mathbb{R}^{K\times D'}$ is further encoded through L layers of self-attention in the Transformer to obtain the features $\mathbf{Z}^L \in \mathbb{R}^{K\times D'}$.

Pose-Text Recognition Decoder. Suppose that the number of actions is C. For each pose, the Decoder takes $\mathbf{Z}^L \in \mathbb{R}^{K \times D'}$ and C Action Queries with learnable embeddings $\hat{z}^0 \in \mathbb{R}^{C \times D'}$ as input. It then transforms these queries into action scores $S \in \mathbb{R}^C$, which represents the scores for each action.

To achieve decoding, we employ cross attention to connect the Encoder features $\mathbf{Z}^L$ with Action Queries $\hat{z}^0$, where the query ($\mathbf{Q}$) originates from $\hat{z}^0$, and the key ($\mathbf{K}$) and value ($\mathbf{V}$) are derived from $\mathbf{Z}^L$. After L_{dec} layers of decoding, we obtain the output $\hat{z}^{L_{dec}} \in \mathbb{R}^{C \times D'}$. Finally, we get the score of each action class $S \in \mathbb{R}^C$ using the following module:

$$S = \sigma(\text{Squeeze}(\text{Linear}(\hat{z}^{L_{dec}}))), \tag{2}$$

where the output channel of Linear is 1, and σ represents Sigmoid activation.

3.3 Training Data Generation

To obtain the data containing salient pose, we apply generative models, particularly Stable Diffusion [25] and ControlNet [32], to generate training data.

After pre-defining two salient poses for each action class, we have $2 \times C$ salient poses, which also serve as the pose conditions $\{c_i\}_{i=1}^{2\times C}$ used to control Stable Diffusion in generating k training images for each salient pose. For a single pose condition c, we can generate the synthetic image x_i using the ControlNet Stable Diffusion G as follows: $x_i = G(z_i, p, c)$, where $z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is random noise, and p represents the text prompt. Finally, we can obtain our training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{2\times C\times k}$, which contains $2 \times C \times k$ images. Here, y_i represents the label for each image, specifically indicating both the action and salient pose.

3.4 Open-Set Scenario Generalization

Besides the close-set setting, we aim to facilitate the generalization to previously unseen action classes in an open-set setting. To achieve this, we apply an off-the-shelf text encoder to encode the action classes and obtain rich semantic information, which is then assigned to the Action Queries.

Specifically, for the unseen action classes $\{c_i\}_{i=1}^{C_{open}}$, where C_{open} represents the number of unseen actions, we utilize the template such as "there is a human doing {`action name`} action" to create each prompt. Subsequently, we apply the CLIP Text Encoder [22] to encode these prompts and generate the embedding $\hat{z}^0_{open} \in \mathbb{R}^{C_{open} \times D'}$, capturing the semantic information of these open-set action classes. These embeddings are assigned to the Action Queries in the open-set scenario. We use the obtained $\hat{z}^0_{open}$ to conduct cross-attention with the pose feature $\mathbf{Z}^L$, enabling open-set pose-action modeling.

3.5 Pose-Action Trigger

The above modules are both in the first stage, which means that the output S represents the classification for a single pose, as well as a single frame. To

perform video-level counting, we aggregate the scores of all frames to obtain the video score matrix $\hat{S} \in \mathbb{R}^{C \times T}$, where T represents the number of frames in the current video. Meanwhile, we apply an off-the-shelf Action Recognizer [17] to conduct video-level action recognition, and obtain the action class $\mathcal{C}$. We then extract the scores $S_{\mathcal{C}} \in \mathbb{R}^{T}$ for $\mathcal{C}$ from $\hat{S}$.

We design a lightweight Pose-Action Trigger, which has no trainable parameters and a time complexity of only $\mathcal{O}(n)$. First, we define upper and lower bounds to separate the scores of two salient poses. This method groups non-salient poses in the middle and classifies the salient poses at either end. Next, we scan all frames to count the sequential occurrences of the two salient poses for this action class. Each sequential occurrences indicates an increment in the count by one. Through this module, we can obtain the final action count $\mathcal{M}$.

3.6 Training Strategies

The modules that require training include the Pose-wise Encoder and the Pose-Text Recognition Decoder. In terms of **Action Queries**, we have two settings: **1) Close-set**, where the Action Queries are randomly initialized and can be tuned during training, eliminating the need for text encoder; and **2) Open-set**, where we utilize the embedding of the CLIP Text Encoder to assign the Action Queries and freeze them throughout the training process.

First, we transform the label y_i of each image into the vector $y'_i \in \mathbb{R}^C$. In our PoseRAC, the target for the salient pose I is set to 1, while the target for salient pose II is set to 0. The targets for all other negative samples are set to 0.5. For example, if $C = 5$ and y_i represent the salient pose I of the second action, then $y'_i = [0.5, \mathbf{1.0}, 0.5, 0.5, 0.5]$. Similarly, if y_i represents the salient pose II of the third action, then $y'_i = [0.5, 0.5, \mathbf{0.0}, 0.5, 0.5]$. Then, we calculate the Binary Cross Entropy Loss between $y'_i \in \mathbb{R}^C$ and its corresponding output $S_i \in \mathbb{R}^C$ in the Decoder, with the batch size set to N. This can be defined as follows:

$$\mathcal{L}_{bce} = -\frac{1}{N}\sum_{i=1}^{N}(\frac{1}{C}\sum_{j=1}^{C} loss(i,j)), \tag{3}$$

$$loss(i,j) = y'_{ij}\log S_{ij} + (1 - y'_{ij})\log(1 - S_{ij}). \tag{4}$$

Furthermore, we employ the Triplet Margin Loss to train only the Encoder, with the aim of enhancing its ability to produce more representative features $\mathbf{Z}^L$ given a pose. We select anchors a, same salient pose positive samples p, and different salient poses negative samples n in a batch. Cosine Similarity (CS) is used to measure the distance between features. It can be expressed as:

$$\mathcal{L}_{tri} = \max(\mathrm{CS}(a,p) - \mathrm{CS}(a,n) + \mathrm{margin}, 0). \tag{5}$$

We pay more attention to hard samples, where the distances between anchors and negative samples are even smaller than those of positive samples. After training, the salient poses of each action can be distinguishable, and can cluster in the high-level space.

Table 1. Detailed information for three benchmarks.

Dataset	Categories	Number of Videos		
		Train	Val	Test
RepCount [13]	9	758	129	152
UCFRep [31]	23	421	105	—
Countix [10]	57	4414	1406	2555

At last, our overall training combines two losses:

$$\mathcal{L} = \mathcal{L}_{bce} + \alpha \mathcal{L}_{tri}, \tag{6}$$

where α represents the weight factor.

4 Experiments

4.1 Experimental Setup

Datasets and Close and Open-Set Setting. We conduct comprehensive experiments on three mainstream benchmarks for repetitive action counting, namely RepCount, UCFRep, and Countix. The detailed information is shown in Table 1.

In the **close-set setting**, we compare with previous state-of-the-art methods on the test set of RepCount and Countix, and the validation set of UCFRep.

In the **open-set setting**, we evaluate the generalization of our method when *handling unseen actions*. In this scenario, we use Countix dataset and re-split the training set and validation set to ensure the classes in them are disjoint.

Evaluation Metrics. We demonstrate the superiority of our method on two widely used metrics in this task, which are **Off-By-One (OBO) count error** and **Mean Absolute Error (MAE)**. OBO measures the error rate of repetition count over the entire dataset, while MAE represents the normalized absolute error between the ground truth and the prediction. They can be defined as:

$$\textbf{OBO} = \frac{1}{N}\sum_{i=1}^{N}[|\tilde{c_i} - c_i| \leq 1] \tag{7}$$

$$\textbf{MAE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\tilde{c_i} - c_i|}{\tilde{c_i}} \tag{8}$$

where $\tilde{c}$ is the ground truth, c_i is our prediction, and N is the video number.

Implementation Detail. We compare with some state-of-the-art action counting methods: RepNet [10], Context [31], Sight & Sound [33], and TransRAC [13]. Additionally, we use video understanding methods [1,11,19] as baselines, adapting their output layers according to the TransRAC. Except for RepNet and Sight

Table 2. Pre-defined salient poses for each action in datasets. Due to space limitations, we take the RepCount dataset as an example.

Action	Salient Pose I	Salient Pose II
battle rope	raising left hand, lowering right hand, and performing a battle rope action	raising right hand, lowering left hand, and performing a battle rope action
bench pressing	curling up legs	stretching legs
front raise	putting down the arms	raising the arms
jumping jacks	standing upright and arms-down	jumping up and arms-upward
pommel horse	body leaning to the left	body leaning to the right
pull up	hanging on the arms	pulling up the arms
push up	lying prone with arms straight	lying prone and bending arms
situp	lying down	sitting
squat	standing upright	squatting

& Sound, all other methods cannot deal with the Countix dataset as it do not provide annotations for each action cycle. For consistency, we follow the optimal hyperparameter settings mentioned in the papers of each method.

In our method, we utilize 3D OpenPose [4] for Pose Estimation. We design a 6-layer Transformer for Encoder and a 2-layer Transformer for Decoder. We select UniFormerV2-L [17] as Action Recognizer, which is pretrained on Kinetics-700 [5]. We pre-define two salient poses for each action in three datasets. Due to space limitations, we take the RepCount dataset as an example to show the pre-defined salient poses of each action in Table 2. During training data generation, we use OpenPose Editor[2] to edit these salient poses, and use Stable Diffusion and ControlNet to generate $1k$ images for each salient pose.

We report both **zero-shot** and **few-shot** performance. In zero-shot, we generate synthetic data for training *without using any real training set*. In few-shot, along with the synthetic data, we extract 200 frames containing pre-defined salient poses for each action from the training set, to fine-tune our model.

4.2 Comparisons with Previous Methods

Close-Set Performance. As shown in Table 3, PoseRAC outperforms existing methods in terms of performance. First, when we only pretrain the PoseRAC with synthetic images, the zero-shot performance exceeds that of previous fully-supervised methods on the RepCount dataset, with an OBO metric of 0.43 compared to the 0.29 of TransRAC. When fine-tuning with a few of samples from the training set, the few-shot performance is even higher, with the OBO metric surpassing 0.5 for the first time. Similar superiority can be observed on the UCFRep and Countix datasets.

Here, we delve into why our approach is effective. Previous methods primarily extract intricate temporal features from a video clip, which are challenging

[2] https://github.com/ZhUyU1997/open-pose-editor.

Table 3. Comparison on three datasets in both *close-set* and *open-set* settings. The best results are highlighted in **bold**, and the second best is underlined. Our approach shows superiority in both zero-shot (not see the training set) and few-shot (only a few frames) settings. †: As RepCount and UCFRep solely consist of RGB frames, this approach is a sight-only model for handling these datasets.

Method	Method	RepCount		UCFRep		Countix		Countix (Open)	
		MAE ↓	OBO ↑	MAE ↓	OBO ↑	MAE ↓	OBO ↑	MAE ↓	OBO ↑
X3D [11]	fully	0.910	0.106	0.882	0.126	—	—	—	—
TANet [19]		0.662	0.099	0.691	0.103	—	—	—	—
ViViT [1]		0.676	0.103	0.655	0.098	—	—	—	—
RepNet [10]		0.995	0.013	0.998	0.009	0.364	0.303	0.723	0.195
Context [31]		0.879	0.155	0.147	0.790	—	—	—	—
Sight & Sound† [33]		0.732	0.196	**0.143**	0.800	0.307	0.511	0.760	0.188
TransRAC [13]		0.443	0.291	0.441	0.430	—	—	—	—
PoseRAC (Ours)	zero-shot	0.328	0.425	0.319	0.526	0.403	0.339	0.692	0.226
	few-shot	**0.226**	**0.570**	0.146	**0.803**	**0.305**	**0.530**	**0.616**	**0.317**

to be trained well to represent periodic movements. On the other hand, our approach introduces salient pose into this task and further decouples this complex temporal process into single-frame pose recognition. In our approach, there is no need for training temporal modeling, making it considerably less challenging to achieve effective training. Moreover, our Pose-Action Trigger can complete counting based on salient poses, which is robust to interruptions during actions and inconsistent action cycles of different action classes.

Open-Set Performance. Moreover, we also evaluate the open-set action counting ability of different methods on Countix. As demonstrated in Table 3, due to the more challenging open-set setting, the performance of RepNet and Sight & Sound is much lower compared to the regular setting. However, our method consistently outperforms previous methods, achieving an OBO metric of 0.32 compared to the 0.20 of RepNet. This is attributed to the ability of our method to link salient poses with actions. Specifically, PoseRAC learns the relationship between manually pre-defined salient poses and their corresponding actions during training, enabling it to model the relationship between salient poses and **unseen actions**. By further incorporating the semantic information of actions, our PoseRAC can effectively recognize the salient poses of those unseen actions.

4.3 Ablation Studies

Because the RepCount dataset is currently the highest quality dataset, we conduct ablation studies on the validation set of RepCount, to analyze some core ideas of PoseRAC. Here we focus on the close-set and few-shot setting.

Table 4. Performance of different baselines. **Pose-level**: Replace the Encoder. **Image-level**: Replace both the Pose Estimator and the Encoder

Baselines		MAE ↓	OBO ↑
Pose-level	MLP	0.316	0.486
	1DCNN	0.357	0.439
	Transformer	**0.221**	**0.576**
Image-level	ResNet-50 [12]	0.762	0.103
	ViT-32 [9]	0.723	0.116

Table 5. Comparison of different α for training losses

Loss	α	MAE ↓	OBO ↑
$\mathcal{L}_{bce}$ only	—	0.339	0.450
$\mathcal{L}_{bce} + \alpha\mathcal{L}_{tri}$	**0.01**	**0.221**	**0.576**
	0.05	0.259	0.542
	0.1	0.276	0.516

Table 6. Comparing different assignments of Action Queries on the RepCount and UCFRep datasets in the close-set setting

Action Queries	RepCount		UCFRep	
	MAE ↓	OBO ↑	MAE ↓	OBO ↑
Randomly	**0.221**	**0.576**	**0.146**	**0.803**
Text Embedding	0.319	0.483	0.287	0.639

Table 7. Comparing different assignments of Action Queries on Countix in both close and open set settings

Action Queries	Scenario	Countix	
		MAE ↓	OBO ↑
Randomly	Close-set	**0.310**	**0.516**
	Open-set	—	—
Text Embedding	Close-set	0.427	0.409
	Open-set	**0.619**	**0.305**

Additional Baselines for Encoder. We consider two dimensions for additional baselines. 1) We use different structures in the Pose-wise Encoder to observe changes in performance. We select MLP, CNN, and Transformer, and *ensure that their parameters are set to be close to each other for a fair comparison.* 2) We replace both the Pose Estimator and Pose-wise Encoder with image-classification baselines. This involves directly extracting features from each RGB frame without estimating human pose. Through this, we aim to *validate the effectiveness of introducing pose information into repetitive action counting.*

As demonstrated in Table 4, image-classification baselines perform poorly because the RGB frame contains much irrelevant information, leading to high difficulty in extracting the action information. This further demonstrates the effectiveness of linking pose with action. On the other hand, in pose-level baselines, we observe that the performance of Transformer is slightly higher. We attribute it to the self-attention between different pose keypoints, while common MLP and CNN cannot model the relationship between two distant points effectively.

The Number of Training Data. Here we consider the impact of the amount of training data, especially both the synthetic image and the extracted frames from training set. To facilitate comparison, we set different numbers of synthetic data while keeping the number of real data fixed at 200 for each pose. Simultaneously, we maintain the number of synthetic data at $1k$ for each pose while varying the number of real data. As shown in Fig. 3, as the number of both synthetic and real data increases, the OBO metric consistently improves. Especially, increasing the number of synthetic data from $0.25k$ to $1k$ results in a significant improvement,

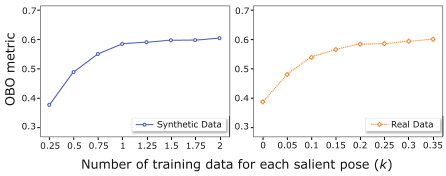
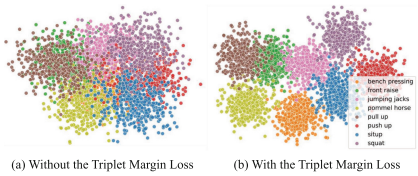


Fig. 3. OBO metric with various training data volume. **Left**: the number of real data is fixed at 200. **Right**: the number of synthetic data is fixed at $1k$.



(a) Without the Triplet Margin Loss (b) With the Triplet Margin Loss

Fig. 4. T-SNE visualization of the embedding features extracted by the Encoder trained **(a) without** or **(b) with** triplet margin loss.

Table 8. Comparison among different pose estimators. In the end, we select 3D OpenPose for its superiority

Pose Estimator	MAE ↓	OBO ↑
HRNet [27]	0.269	0.496
BlazePose [2]	0.240	0.561
Vitpose [30]	0.243	0.551
RTMPose [15]	0.241	0.562
3D OpenPose [4]	**0.221**	**0.576**

Table 9. Comparison of all baselines for the Pose-wise Encoder in Pose-Action Modeling accuracy

Datasets	Image-level		Pose-level		
	ResNet	ViT	MLP	CNN	**Trans**
RepCount	0.52	0.61	0.83	0.69	**0.93**
UCFRep	0.43	0.47	0.68	0.59	**0.82**
Countix (close)	0.45	0.51	0.71	0.60	**0.83**
Countix (open)	0.09	0.09	0.30	0.26	**0.49**

from around 0.4 to around 0.6. The performance can be further enhanced with more data, such as $0.35k$ real data. In this way, our method can significantly boost performance by generating more synthetic samples without the need for complex design. Considering both efficiency and performance, we set the number of synthetic and real data per salient pose to $1k$ and $0.2k$, respectively.

Effectiveness of Training Losses. We employ two loss functions to train our model. As shown in Table 5, we compare the performance with and without the triplet margin loss using different values of α. While our model can be effectively trained with binary cross-entropy loss alone, the addition of the triplet margin loss leads to improvement. We observe that the optimal value for α is 0.01, primarily because the two losses have different numeric scales.

Moreover, we select those frames in which the salient pose of each action appears (here we take salient pose II as the example), and use Pose Estimator and Pose-wise Encoder to extract their embedding features. We employ t-SNE [20] to visualize the first two principal components of these features. As shown in Fig. 4, after training with the triplet margin loss, the encoder enhances its ability to distinguish the salient poses of each class.

Effectiveness of Text Encoder. We conduct experiments to observe the effectiveness of text encoder. In close-set setting, we replace the randomly initialized Action Queries with the text embeddings from CLIP Text Encoder and retrain the model. As shown in Table 6 and Table 7, directly assigning the Action Queries to the text embeddings leads to a slight decrease in performance on all datasets because the trainable queries can be more representative. However, random ini-

tialization cannot handle the open-set scenario, as shown in Table 7. The text encoder can encode any unseen class, facilitating open-set pose recognition, give our PoseRAC the ability to generalize to any unseen actions in open-set scenario.

Choice of Pose Estimator. Accurate action counting of PoseRAC relies on accurate pose estimation, so we compare several excellent pose estimatiors. Here, we replace the Pose Estimator in our PoseRAC with different algorithms and report the performance. As shown in Table 8, 3D OpenPose surpasses other algorithms, which we attribute to its capability in 3D keypoint reconstruction. By extracting richer pose information compared to others, it enhances the learning capabilities of PoseRAC. Thus, we choose 3D OpenPose as the Pose Estimator.

4.4 More Analyses of the Pose-Action Modeling

In our method, the accurate Pose-Action Trigger is highly dependent on accurate Pose-Action Modeling. Therefore, we validate the salient pose recognition ability of Pose-Action Modeling. Specifically, we extract frames where the two salient poses of all repetitive actions appear in each video and input them into the network to obtain outputs from the Decoder. We also annotate the classification label of these frames. We compare our model with additional baselines mentioned in Sect. 4.3 and evaluate the recognition accuracy on all datasets. Furthermore, we assess the recognition accuracy on Countix for the open-set scenario.

As shown in Table 9, pose-level baselines outperform the image-level baselines, further demonstrating the effectiveness of introducing pose. Moreover, using Transformer as the Encoder yields the highest accuracy across all datasets (0.93 on RepCount, 0.82 on UCFRep, and 0.83 on Countix), enabling the following module to successfully complete the counting task. Similar conclusions can be drawn from the qualitative evaluation shown in Fig. 5, where we utilize our model to recognize poses in each frame. It is evident that our model accurately recognizes the salient poses. In the open-set scenario, the accuracy of image-level baselines is almost zero, while the accuracy of our method is 0.49. However, it

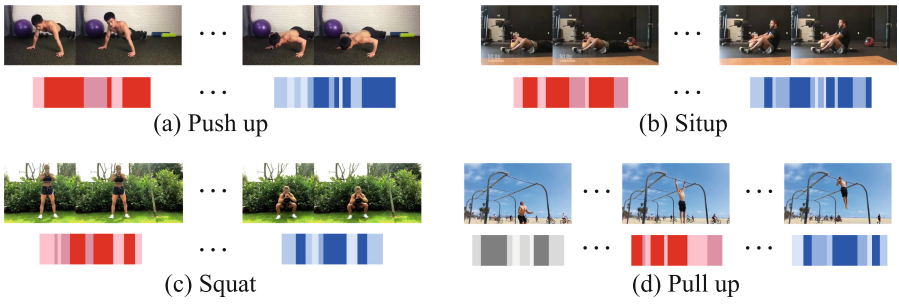


Fig. 5. Qualitative evaluation. The histogram represents the scores of pose recognition, where the red histogram and the blue histogram represent that the detection of the current frame corresponds to salient poses I and II, respectively. (Color figure online)

is still far below that of the close-set performance, showing the challenge of open-set pose recognition, which merits further research in the future.

5 Conclusion

In this paper, we focus on the valuable yet recently underexplored repetitive action counting task. Considering the shortcomings of previous methods on addressing the periodic representation, we introduce pose modality into this task for the first time. By analyzing the relationship between human poses and actions, we present a novel concept called **Salient Pose** to effectively represent each action. Furthermore, we propose a new approach called **PoseRAC**, which includes Pose-Action Modeling and Pose-Action Trigger to efficiently count actions based on our salient poses. Leveraging generative models and an off-the-shelf text encoder enables our model to perform zero-shot and open-set counting, respectively, showcasing further innovation in this area. Comprehensive experiments demonstrate that our approach yields promising results and opens up new avenues for future research in the field of repetitive action counting.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: a video vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6836–6846 (2021)
2. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., Grundmann, M.: Blazepose: on-device real-time body pose tracking. arXiv preprint arXiv:2006.10204 (2020)
3. Briassouli, A., Ahuja, N.: Extraction and analysis of multiple periodic motions in video sequences. IEEE Trans. Pattern Anal. Mach. Intell. **29**(7), 1244–1261 (2007)
4. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 172–186 (2021). https://doi.org/10.1109/TPAMI.2019.2929257
5. Carreira, J., Noland, E., Hillier, C., Zisserman, A.: A short note on the kinetics-700 human action dataset. arXiv preprint arXiv:1907.06987 (2019)
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017)
7. Davis, J., Bobick, A., Richards, W.: Categorical representation and recognition of oscillatory motion patterns. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662), vol. 1, pp. 628–635. IEEE (2000)
8. Dong, J., Sun, S., Liu, Z., Chen, S., Liu, B., Wang, X.: Hierarchical contrast for unsupervised skeleton-based action representation learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 525–533 (2023)
9. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

10. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: Counting out time: Class agnostic video repetition counting in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10387–10396 (2020)
11. Feichtenhofer, C.: X3D: expanding architectures for efficient video recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 203–213 (2020)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
13. Hu, H., Dong, S., Zhao, Y., Lian, D., Li, Z., Gao, S.: Transrac: encoding multi-scale temporal correlation with transformers for repetitive action counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19013–19022 (2022)
14. Huang, S., Ying, X., Rong, J., Shang, Z., Zha, H.: Camera calibration from periodic motion of a pedestrian. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3025–3033 (2016)
15. Jiang, T., et al.: Rtmpose: real-time multi-person pose estimation based on mmpose. arXiv preprint arXiv:2303.07399 (2023)
16. Levy, O., Wolf, L.: Live repetition counting. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3020–3028 (2015)
17. Li, K., et al.: Uniformerv2: spatiotemporal learning by arming image ViTs with video uniformer. arXiv preprint arXiv:2211.09552 (2022)
18. Li, X., Li, H., Joo, H., Liu, Y., Sheikh, Y.: Structure from recurrent motion: from rigidity to recurrency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3032–3040 (2018)
19. Liu, Z., Wang, L., Wu, W., Qian, C., Lu, T.: TAM: temporal adaptive module for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13708–13718 (2021)
20. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(11) (2008)
21. Pogalin, E., Smeulders, A.W., Thean, A.H.: Visual quasi-periodicity. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
22. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
23. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents, 2022. arXiv:2204.06125 **7** (2022)
24. Ran, Y., Weiss, I., Zheng, Q., Davis, L.S.: Pedestrian detection via periodic motion analysis. Int. J. Comput. Vision **71**, 143–160 (2007)
25. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)
26. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
27. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5693–5703 (2019)

28. Thangali, A., Sclaroff, S.: Periodic motion detection and estimation via space-time sampling. In: 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1, vol. 2, pp. 176–182. IEEE (2005)
29. Xiao, F., Lee, Y.J., Grauman, K., Malik, J., Feichtenhofer, C.: Audiovisual slowfast networks for video recognition. arXiv preprint arXiv:2001.08740 (2020)
30. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose: simple vision transformer baselines for human pose estimation. arXiv preprint arXiv:2204.12484 (2022)
31. Zhang, H., Xu, X., Han, G., He, S.: Context-aware and scale-insensitive temporal repetition counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 670–678 (2020)
32. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836–3847 (2023)
33. Zhang, Y., Shao, L., Snoek, C.G.: Repetitive activity counting by sight and sound. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14070–14079 (2021)
34. Zhou, H., Liu, Q., Wang, Y.: Learning discriminative representations for skeleton based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10608–10617 (2023)