D^2-STX: Decoupling Spatial-temporal Cross-attention for Dual-branch Repetitive Action Counting

Supplementary Material

6. Method Appendix

6.1. Adjustment of key points of the human body

As shown in Figure S4, to meet the requirements of pose feature extraction for the CTR-GCN [7] pre-trained model, this study converted the 33-keypoint human pose estimation framework of BlazePose [3] to a streamlined 25-keypoint structure. This 25-keypoint structure is similar to the human topology structure of the Kinetics system.

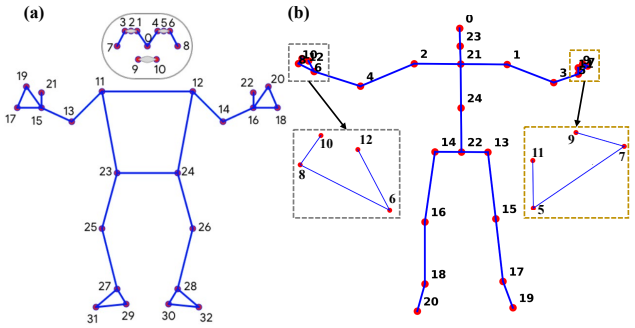

Figure 4. Comparison of human body key point topology before and after adjustment. (a) Topology of 33 human body key points before adjustment (b) Topology of 25 human body key points after adjustment

The relationship between the index and the corresponding keypoints in the original BlazePose architecture is as follows: 0. Nose 1. Left eye inner 2. Left eye 3. Left eye outer 4. Right eye inner 5. Right eye 6. Right eye outer 7. Left ear 8. Right ear 9. Mouth left 10. Mouth right 11. Left shoulder 12. Right shoulder 13. Left elbow 14. Right elbow 15. Left wrist 16. Right wrist 17. Left pinky 18. Right pinky 19. Left index 20. Right index 21. Left thumb 22. Right thumb 23. Left hip 24. Right hip 25. Left knee 26. Right knee 27. Left ankle 28. Right ankle 29. Left heel 30. Right heel 31. Left foot index 32. Right foot index

The relationship between the index and the corresponding keypoints in the proposed 25-keypoint architecture is as follows: 0. head, 1. left shoulder, 2. right shoulder, 3. left elbow, 4. right elbow, 5. left wrist, 6. right wrist, 7. left hand, 8. right hand, 9. tip of left hand, 10. tip of right hand, 11. left thumb, 12. right thumb, 13. left hip, 14. right hip, 15. left knee, 16. right knee, 17. left ankle, 18. right ankle, 19. left foot, 20. right foot, 21. spine, 22. middle of spine, 23. neck, 24. middle of spine.

The coordinate mapping preserves spatial relationships and maintains compatibility with standard pose estimation pipelines while offering improved processing efficiency for motion-focused applications.

6.2. Dual-Branch Video and Pose Representations

This section outlines the process used to extract and store spatial-temporal feature representations from video data and pose data, which serve as input for downstream repetitive action analysis tasks. We support three datasets: RepCount-A [15], Countix [13], and UCFRep [36]. For video data, three backbone encoders are used: R(2+1)D-18, Swin-T and 3D-ResNeXt101. For pose data, a backbone encoder: CTR-GCN is used.

We support both convolutional and Transformer-based backbones. Specifically, R(2+1)D-18 is a spatiotemporal convolutional neural network that factorizes traditional 3D convolutions into separate spatial and temporal components. This model is initialized from pretrained weights on Kinetics-400 and is used strictly as a frozen feature extractor during inference. In addition, we utilize Swin-T, a hierarchical vision Transformer architecture pretrained on Kinetics-400, and 3D-ResNeXt101, a deep convolutional model designed for action recognition tasks.

The three datasets are handled using task-specific PyTorch $'Dataset'$ classes ($'RepCount'$, $'Countix'$, and $'UCFRep'$), all configured with $'encode\_only = True'$ to extract features without computing loss values. For each dataset, videos are sampled at fixed frame rates to reduce temporal redundancy. The data loading pipeline is configured to handle training, validation, and test splits appropriately. Notably, for UCFRep, the validation split is reused for testing due to the lack of a dedicated test set.

We implement two types of feature encoding strategies. The first, video-level token encoding ($'save\_tokens'$), processes full videos by dividing them into overlapping 64-frame windows. Each window is then subsampled to 16 frames and passed through the encoder. The resulting features, typically shaped as $S \times C \times T \times H \times W$, are saved as $'.npz'$ files in designated directories. This strategy effectively captures the global spatial-temporal dynamics of the video. The second strategy, exemplar-level encoding ($'save\_exemplar'$), focuses on extracting features from individual repetitions. Based on the ground-truth annotations, 16 frames are uniformly sampled from within each repetition segment. The extracted features are reshaped appropriately and also saved as $'.npz'$ files, allowing fine-grained analysis of single-cycle patterns.

In addition to video-based encoding, we also support

pose-level feature extraction, where the model input consists of human joint coordinates extracted from video frames. The pose encoder CTR-GCN is used to capture spatial and temporal dependencies between joints and across time. This model is also used in frozen mode during the encoding process. Similarly to video encoding, pose sequences are processed using both full-length and exemplar-level strategies: sliding windows over the entire pose sequence are used for video-level encoding, while ground-truth repetition segments are sampled to extract pose exemplars. These complete pose sequences are represented as tensors of shape $S \times C \times T \times J \times 1$. The extracted CTR-GCN features are saved as $'.npz'$ files, matching the naming convention and structure of the video-based features.

To ensure the robustness of the feature extraction process, several safeguards are implemented. The script automatically skips over corrupted or empty video files and filters out invalid annotations. It also checks for numerical stability by verifying that all feature tensors are free of NaN or infinite values. Furthermore, careful attention is paid to maintaining consistent tensor shapes for operations involving batch normalization and other inputs of the model.

The final output of the encoding process is a set of $'.npz'$ files, each filename corresponding to a complete video feature, a complete pose feature, and a video and pose feature for a repeating segment. These files store intermediate feature representations from the video and pose modalities and serve as input for training and evaluating models for the repetition counting and density estimation tasks.

6.3. Decoupled Spatial-Temporal Cross-Attention

While conventional cross-attention fusion approaches (Fig. S5(a)) flatten multi-modal features into unified token sequences, this design inherently conflates spatial and temporal dimensions, losing the hierarchical structure critical for modeling repetitive actions. Specifically, when video features of shape $[H, W, T]$ and pose features of shape $[J, T]$ are directly flattened into sequences $[\mathrm{thw}_1, \ldots, \mathrm{thw}_n]$ and $[\mathrm{tj}_1, \ldots, \mathrm{tj}_n]$, the resulting cross-attention mechanism treats all spatial-temporal positions equivalently, failing to capture the distinct roles of spatial configurations and temporal dynamics in cross-modal interaction.

To address this limitation, we propose a Decoupled Spatial-Temporal Cross-attention Fusion mechanism (Fig. S5(b)) that explicitly separates spatial and temporal reasoning. Our key insight is that cross-modal dependencies exhibit fundamentally different characteristics across spatial and temporal dimensions: spatial cross-attention captures how body joint configurations correspond to visual regions within each frame, while temporal cross-attention models how motion patterns evolve synchronously across modalities. By decoupling these operations, our approach enables spatial-temporal coupling at the feature level while maintaining spatial-temporal decoupling in the attention computation. This design preserves the inherent hierarchical structure of video and pose data, enabling more interpretable cross-modal reasoning while reducing computational complexity from $O\big((HWT + JT)^2\big)$ to $O(HW \cdot J \cdot T + T^2)$, making it scalable to longer sequences. Furthermore, the decoupled architecture facilitates independent optimization of spatial alignment and temporal synchronization, leading to more robust fusion for repetitive action counting where both precise spatial correspondence and consistent temporal periodicity are essential.

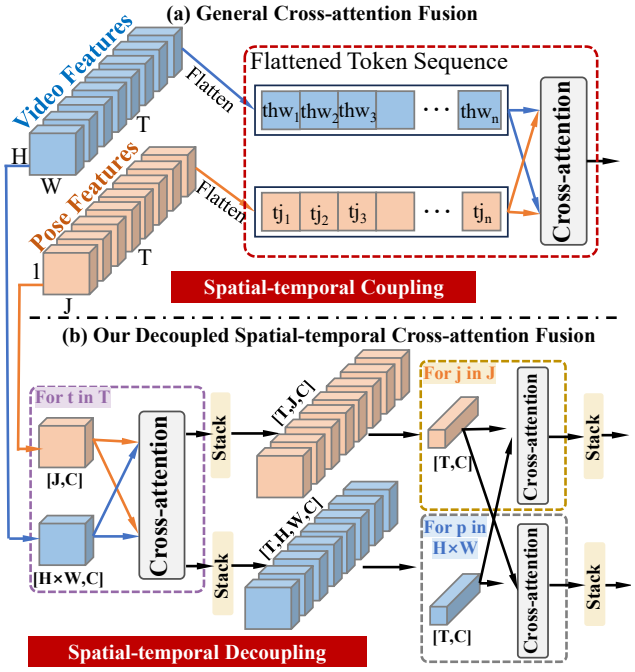

Figure 5. Comparison of cross-attention fusion mechanisms for video and pose feature fusion. (a) General cross-attention fusion directly processes flattened token sequences from video features and pose features through spatial-temporal coupling. (b) Our proposed decoupled spatial-temporal cross-attention fusion separately handles spatial and temporal dimensions

The computation process of our spatial-temporal decoupled attention mechanism is shown in Algorithms 1-4[1]. Algorithm 1 demonstrates spatial pose-guided cross-attention, Algorithm 2 demonstrates temporal pose-guided cross-attention, Algorithm 3 demonstrates spatial video-guided cross-attention, and Algorithm 4 demonstrates temporal video-guided cross-attention. E_{spatial} and E_{temporal} are positional codes used in the processing to prevent the loss of positional information in the video and pose sequences during interaction.

[1]The code and models will be made publicly available.

Algorithm 1 Spatial Pose-guided Cross-attention

Require: Pose features $\mathbf{P} \in \mathbb{R}^{B\times T\times J\times C}$
Require: Video features $\mathbf{V} \in \mathbb{R}^{B\times T\times H\times W\times C}$
Ensure: Fused features $\mathbf{F} \in \mathbb{R}^{B\times T\times J\times C}$
$\mathcal{L}_{\text{fused}} \leftarrow [\,]$, $\mathcal{L}_{\text{attn}} \leftarrow [\,]$, $\mathcal{L}_{\text{output}} \leftarrow [\,]$
for $t = 1$ to T **do**
  $\mathbf{P}_t \leftarrow \mathbf{P}[:, t, :, :]$ // $\in \mathbb{R}^{B\times J\times C}$
  $\mathbf{V}_t \leftarrow \text{reshape}(\mathbf{V}[:, t, :, :, :], (B, HW, C))$
  $\mathbf{V}_t^{\text{pos}} \leftarrow \mathbf{V}_t + \mathbf{E}_{\text{spatial}}$
  $\mathbf{O}_t, \mathbf{A}_t \leftarrow \text{Attention}(\mathbf{P}_t, \mathbf{V}_t^{\text{pos}}, \mathbf{V}_t^{\text{pos}})$
  $\mathbf{F}_t \leftarrow \text{LayerNorm}(\mathbf{P}_t + \mathbf{O}_t)$
  Append $\mathbf{F}_t$ to $\mathcal{L}_{\text{fused}}$
end for
$\mathbf{F} \leftarrow \text{stack}(\mathcal{L}_{\text{fused}}, \text{dim} = 1)$
return $\mathbf{F}$

Algorithm 2 Temporal Pose-guided Cross-attention

Require: Pose features $\mathbf{P} \in \mathbb{R}^{B\times T\times J\times C}$
Require: Video features $\mathbf{V} \in \mathbb{R}^{B\times T\times H\times W\times C}$
Require: Spatial attention weights $\mathbf{W}_{\text{spatial}} \in \mathbb{R}^{B\times T\times J\times (HW)}$
Ensure: Temporal fused features $\mathbf{F}_{\text{temporal}} \in \mathbb{R}^{B\times T\times J\times C}$
$\mathcal{L}_{\text{fused}} \leftarrow [\,]$, $\mathcal{L}_{\text{attn}} \leftarrow [\,]$, $\mathcal{L}_{\text{output}} \leftarrow [\,]$
$\mathbf{V}_{\text{flat}} \leftarrow \text{reshape}(\mathbf{V}, (B, T, HW, C))$
$\mathbf{V}_{\text{joint}} \leftarrow \mathbf{W}_{\text{spatial}} \times \mathbf{V}_{\text{flat}}$
$\mathbf{E}_{\text{temporal}} \leftarrow \text{get\_temporal\_pos\_embed}(T)$
for $j = 1$ to J **do**
  $\mathbf{P}_j \leftarrow \mathbf{P}[:, :, j, :]$
  $\mathbf{V}_j \leftarrow \mathbf{V}_{\text{joint}}[:, :, j, :]$
  $\mathbf{P}_j^{\text{pos}} \leftarrow \mathbf{P}_j + \mathbf{E}_{\text{temporal}}$
  $\mathbf{V}_j^{\text{pos}} \leftarrow \mathbf{V}_j + \mathbf{E}_{\text{temporal}}$
  $\mathbf{O}_j, \mathbf{A}_j \leftarrow \text{Attention}(\mathbf{P}_j^{\text{pos}}, \mathbf{V}_j^{\text{pos}}, \mathbf{V}_j^{\text{pos}})$
  $\mathbf{F}_j \leftarrow \text{LayerNorm}(\mathbf{P}_j + \mathbf{O}_j)$
  Append $\mathbf{F}_j$ to $\mathcal{L}_{\text{fused}}$
end for
$\mathbf{F}_{\text{temporal}} \leftarrow \text{stack}(\mathcal{L}_{\text{fused}}, \text{dim} = 2)$
return $\mathbf{F}_{\text{temporal}}$

Algorithm 3 Spatial Video-guided Cross-attention

Require: Pose features $\mathbf{P} \in \mathbb{R}^{B\times T\times J\times C}$
Require: Video features $\mathbf{V} \in \mathbb{R}^{B\times T\times H\times W\times C}$
Ensure: Fused features $\mathbf{F} \in \mathbb{R}^{B\times T\times H\times W\times C}$
$\mathcal{L}_{\text{fused}} \leftarrow [\,]$
for $t = 1$ to T **do**
  $\mathbf{P}_t \leftarrow \mathbf{P}[:, t, :, :]$
  $\mathbf{V}_t \leftarrow \text{reshape}(\mathbf{V}[:, t, :, :, :], (B, HW, C))$
  $\mathbf{V}_t^{\text{pos}} \leftarrow \mathbf{V}_t + \mathbf{E}_{\text{spatial}}$
  $\mathbf{O}_t \leftarrow \text{Attention}(\mathbf{V}_t^{\text{pos}}, \mathbf{P}_t, \mathbf{P}_t)$
  $\mathbf{F}_t^{\text{flat}} \leftarrow \text{LayerNorm}(\mathbf{V}_t + \mathbf{O}_t)$
  $\mathbf{F}_t \leftarrow \text{reshape}(\mathbf{F}_t^{\text{flat}}, (B, H, W, C))$
  Append $\mathbf{F}_t$ to $\mathcal{L}_{\text{fused}}$
end for
$\mathbf{F} \leftarrow \text{stack}(\mathcal{L}_{\text{fused}}, \text{dim} = 1)$
return $\mathbf{F}$

Algorithm 4 Temporal Video-guided Cross-attention

Require: Pose features $\mathbf{P} \in \mathbb{R}^{B\times T\times J\times C}$
Require: Video features $\mathbf{V} \in \mathbb{R}^{B\times T\times H\times W\times C}$
Require: Spatial attention weights $\mathbf{W}_{\text{spatial}} \in \mathbb{R}^{B\times T\times HW\times J}$
Ensure: Temporal fused features $\mathbf{F}_{\text{temp}} \in \mathbb{R}^{B\times T\times H\times W\times C}$
$\mathcal{L}_{\text{fused}} \leftarrow [\,]$
$\mathbf{P}_{\text{pixel}} \leftarrow \mathbf{W}_{\text{spatial}} \times \mathbf{P}$
$\mathbf{V}_{\text{flat}} \leftarrow \text{reshape}(\mathbf{V}, (B, T, HW, C))$
$\mathbf{E}_{\text{temporal}} \leftarrow \text{get\_temporal\_pos\_embed}(T)$
for $s = 1$ to HW **do**
  $\mathbf{V}_s \leftarrow \mathbf{V}_{\text{flat}}[:, :, s, :]$
  $\mathbf{P}_s \leftarrow \mathbf{P}_{\text{pixel}}[:, :, s, :]$
  $\mathbf{V}_s^{\text{pos}} \leftarrow \mathbf{V}_s + \mathbf{E}_{\text{temporal}}$
  $\mathbf{P}_s^{\text{pos}} \leftarrow \mathbf{P}_s + \mathbf{E}_{\text{temporal}}$
  $\mathbf{O}_s \leftarrow \text{Attention}(\mathbf{V}_s^{\text{pos}}, \mathbf{P}_s^{\text{pos}}, \mathbf{P}_s^{\text{pos}})$
  $\mathbf{F}_s \leftarrow \text{LayerNorm}(\mathbf{V}_s + \mathbf{O}_s)$
  Append $\mathbf{F}_s$ to $\mathcal{L}_{\text{fused}}$
end for
$\mathbf{F}_{\text{flat}} \leftarrow \text{stack}(\mathcal{L}_{\text{fused}}, \text{dim} = 2)$
$\mathbf{F}_{\text{temporal}} \leftarrow \text{reshape}(\mathbf{F}_{\text{flat}}, (B, T, H, W, C))$
return $\mathbf{F}_{\text{temporal}}$

6.4. Evaluation metrics

Following previous RAC works [13, 15, 38], we use Mean Absolute Error (MAE) and Off-By-One accuracy (OBO) as evaluation metrics, calculated as Eq . 16 and 17 respectively. Inspired by image counting methods, we introduce the root mean square error (RMSE) in Eq . 18 for RAC providing a more robust metric for diverse counts compared to MAE's bias toward small counts. We also report the off-by-zero accuracy (OBZ) in Eq . 19 as a tighter metric than the corresponding OBO for precise counts.

$$\text{MAE} = \frac{1}{|\Omega|}\sum_{i\in\Omega}\frac{|c_{pred} - c_{gt}|}{c_{gt}} \tag{16}$$

$$\text{OBO} = \frac{1}{|\Omega|}\sum_{i\in\Omega}(|c_{pred} - c_{gt}| \le 1) \tag{17}$$

$$\text{RMSE} = \sqrt{\frac{1}{|\Omega|}\sum_{i\in\Omega}(c_{pred} - c_{gt})^2} \tag{18}$$

$$\text{OBZ} = \frac{1}{|\Omega|}\sum_{i\in\Omega}(|c_{pred} - c_{gt}| = 0) \tag{19}$$

Method	Encoder	RMSE ↓	MAE ↓	OBZ ↑	OBO ↑
ESCounts[25]	ViT-B	4.455	0.213	0.245	0.563
Reproduced Results	ViT-B	13.606	0.642	0.147	0.300

Table 5. Comparison of SOTA results published by ESCounts with reproduced results on RepCount-A.

7. Comparison with State-of-the-art Appendix

While ESCounts [25] reports superior performance using ViT-B as the video encoder (MAE 0.213 on RepCount-A benchmark, as shown in Tab. S5), we found significant challenges in reproducing these results. Following the implementation details provided in [25], we conducted extensive experiments to replicate their ViT-B configuration, including matching the preprocessing pipeline, training schedule, and hyperparameters. However, our reproduction attempts consistently fell short of their reported metrics by a substantial margin (details in Tab. S5). Given the discrepancy between reported and reproduced results, we believe the most fair and rigorous comparison should be based on methods whose results we can reliably verify. Therefore, our primary comparisons focus on methods using consistent encoder architectures (Swin-T) where reproduction is feasible and results are verifiable. Under this fair comparison protocol with reproducible baselines, our D²-STX demonstrates state-of-the-art performance across all three datasets (RepCount-A, UCFRep, and Countix), achieving the best MAE and OBO metrics. We emphasize that our contribution lies in the novel spatial-temporal inter-modality cross-attention mechanism for multi-modal fusion, which provides consistent improvements regardless of the specific encoder backbone employed.

For a fair, apple-to-apple comparison with SkimFocusNet[40], Tab.S 6 reports efficiency and cross-dataset inference results. Our D^2-STX achieves comparable accuracy on RepCount-A while using fewer parameters, lower FLOPs, and significantly faster inference. Moreover, D^2-STX demonstrates stronger generalization ability in cross-dataset scenarios, including RepCount-A → UCFRep and RepCount-A → Countix. These results demonstrate that D^2-STX achieves a better trade-off between accuracy, efficiency, and generalization compared to SkimFocusNet.

8. Ablation Experiment Appendix

As shown in Tab. S7, we conduct an ablation study to evaluate the contribution of the proposed decision fusion mechanism. The baseline model ("Bidirectional without decision fusion") simply averages the predictions from the video and pose branches using fixed, uniform weights of 0.5 for each. In contrast, Our complete Bidirectional mode employs a dynamic decision fusion module. The comparison clearly demonstrates the critical importance of this adaptive fusion. Incorporating the proposed decision fusion leads to consistent improvements across all metrics, reducing the RMSE from 6.832 to 6.811 and the MAE from 0.252 to 0.250. More significantly, it substantially enhances the counting accuracy, evidenced by the notable increase in the OBZ from 0.200 to 0.233 and the OBO from 0.473 to 0.493. These results validate that our decision fusion strategy effectively integrates complementary information from bidirectional temporal contexts, moving beyond a simple static average to produce more robust and accurate repetition counts.

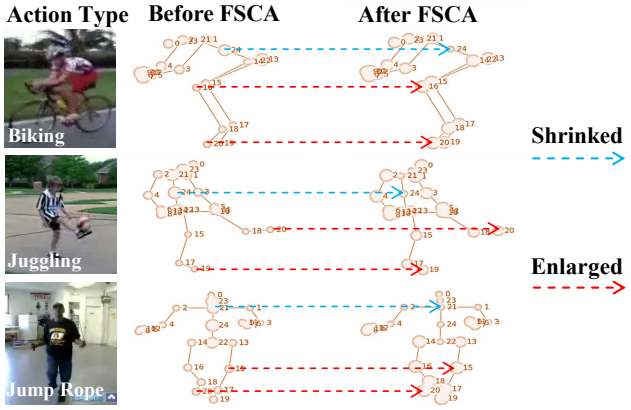

Figure 6. Visualization and comparison of pose features before and after the FSCA module

9. Visualization Appendix

The FSCA module, a key component of our proposed framework, integrates pose and video modalities to enhance pose-based feature representations. To validate the module's effectiveness and interpret its functionality, we employ a visualization technique that highlights the attention-driven feature enhancements across joint coordinates. This section details the interpretability insights derived from the visualization process, demonstrating how the module strengthens pose modality features.

The visualization results in Fig. S6. demonstrate clear evidence of pose feature enhancement through the FSCA module across three representative activities: JumpRope, Biking, and Soccerjuggling. Skeleton joint visualization uses the size of key points to distinguish weights, where larger key points represent higher attention weights and feature importance, while smaller key points represent lower attention weights and feature importance.

In the JumpRope sequence, the Before FSCA visualization displays uniform attention across joints, lacking emphasis on motion-critical regions. After applying FSCA, attention becomes concentrated in functionally relevant areas. The upper limbs (shoulders, elbows, wrists) show increased

Method	Params	FLOPs	Infer. Time	RepCount-A → UCFRep		RepCount-A → Countix	
	(M)	(G)	(ms)	MAE	OBO	MAE	OBO
TransRAC (CVPR'22)	42.28	146.48	380.44	0.640	0.324	0.593	0.364
HTRM-Net (TMM'25)	42.60	146.55	382.70	0.521	0.391	-	-
SkimFocusNet (IJCV'25)	61.50	-	2918.20	0.502	0.391	-	-
Our D^2-STX	52.12	**20.32**	**65.67**	**0.297**	**0.643**	**0.359**	**0.605**

Table 6. Efficiency and cross-dataset inference comparison.

Fusion Method	**RMSE** ↓	**MAE** ↓	**OBZ** ↑	**OBO** ↑
Bidirectional without decision fusion	6.832	0.252	0.200	0.473
Bidirectional	**6.811**	**0.250**	**0.233**	**0.493**

Table 7. Ablation of decision fusion dynamic fusion parameter on RepCount-A.

attention, effectively capturing the rope-handling motion. The lower limbs, particularly the ankles and knees, also exhibit stronger activation. Notably, in this sample, the subject performs leg-opening and closing movements, requiring more precise modeling of dynamic limb motion. These enhancements highlight FSCA's ability to leverage video context to amplify pose features essential for the specific action. In the Biking activity, the Before FSCA attention is scattered and lacks motion specificity. After FSCA, the hips and knees receive increased emphasis, aligning with pedaling, while the shoulders and elbows show moderate enhancement, reflecting their role in steering and balance.