Exploiting VLM Localizability and Semantics for Open Vocabulary Action Detection (Supplementary Material)

Wentao Bao [1], Kai Li[2], Yuxiao Chen[3], Deep Patel[2], Martin Renqiang Min[2], Yu Kong[1]
[1]Department of Computer Science and Engineering, Michigan State University
[2]Machine Learning Group, NEC Laboratories America
[3]Department of Computer Science, Rutgers University
{baowenta,yukong}@msu.edu, yc984@cs.rutgers.edu,
{kaili,dpatel,renqiang}@nec-labs.com

A. Prompts for Query-Text Alignment

To generate text prompts for each action category, we send a request to GPT [9] by using the template: "`For the action type {CLS}, what are the visual descriptions? Please respond with a list of 16 short sentences.`" where the placeholder "`{CLS}`" is replaced by the action class name from the vocabulary. Thus, we obtained multiple caption-like sentence descriptions of the action. Eventually, the text feature for each class is computed by mean pooling of features from the VLM text encoder given the text prompts. In Fig. 1 and Fig. 2, we show a few pieces of the prompt examples on the J-HMDB and UCF101-24 datasets, respectively. We will release all the prompts we used in this work.

B. Explanation of the Reversed Attention

As discussed in the main paper, the seemly counterintuitive phenomenon of the reversed visual-text attention has been studied in [4,5] and we also observed this in our video-based experiments. For CLIP-based models, `[CLS]` token in ViT is aligned to the text semantics so that its attention weight corresponds to the foreground, while the rest L visual token weights are complementary after softmax over $L+1$ tokens before attention pooling. Therefore, due to the attention pooling, high similarity between text feature (or visual `[CLS]` token feature) and L visual tokens could indicate the background.

C. Implementation Details

Positional Embedding Interpolation. When using the pre-trained VLM without fine-tuning, an immediate challenge is that the input videos have different spatiotemporal resolutions from the data in VLM pre-training. For example, the CLIP-ViP is pre-trained on input videos with size $12 \times 224 \times 224$ while videos from J-HMDB can be in any resolution after random augmentations in training. A simple solution is to resize the input video size to match with the pre-trained ones. But for the action detection subtask, person localization is sensitive to the input resolution. To handle this challenge, we instead keep the raw resolution as input, but interpolate the pre-trained spatial and temporal positional embeddings. For example, given the CLIP-ViP B16 VLM and an input video with size $T \times H \times W$, we interpolate the 12 temporal positional embeddings $\text{PE}_t \in \mathbb{R}^{12\times D}$ to $\hat{\text{PE}}_t \in \mathbb{R}^{T\times D}$, and interpolate the 196 $(= \frac{224}{16} \times \frac{224}{16})$ spatial positional embeddings $\text{PE}_s \in \mathbb{R}^{196\times D}$ to $\hat{\text{PE}}_s \in \mathbb{R}^{L\times D}$ where $L = \frac{H}{16} \times \frac{W}{16}$. This technique is found useful for the action detection problem.

4D Feature Pyramid. Following the line of detection literature [3, 13], the pre-trained patch token features are transformed into a 4D feature pyramid before the detection head. Let the $\mathbf{H} \in \mathbb{R}^{h\times w\times T\times D}$ be the pre-trained patch token features from the VLM video encoder, where $h \times w$ is the number of patches for each frame, T is the number of video frames, and D is the Transformer dimension. We use deconvolution or convolution to produce hierarchical feature maps $\tilde{\mathbf{H}}^{(l)}$ by spatial strides $s^{(l)} \in \{1/4, 1/2, 1, 2\}$ where the fractional strides are deconvolutional stides and l indexes the pyramid level. Different from [3, 13] that fully fine-tunes the visual encoder, our VLM visual encoder has to be frozen. Therefore, to allow pre-trained features better utilized by OpenMixer head, we propose to add residual connection at each level of the 4D feature pyramid by spatial interpolation: $\hat{\mathbf{H}}^{(l)} = \phi(\mathbf{H}, s^{(l)}) + \tilde{\mathbf{H}}^{(l)}$. The function ϕ is to spatially interpolate the feature map from the size $h \times w$ to the same resolution of $\tilde{\mathbf{H}}^{(l)}$.

Table 1. **Effect of VLMs.** We implement the OpenMixer by CLIP-ViP and CLIP with the same ViT-B/16 transformer.

VLMs	Modality	Mean	Base	Novel
CLIP [10]	image	71.60	79.46	64.44
CLIP-ViP [15]	video	**86.34**	**90.75**	**82.33**

Table 2. **GPT help temporal localization.** We compute mAP by only using temporal IoU on J-HMDB dataset.

	Mean(t)	Base(t)	Novel(t)
w/o. GPT	83.57	90.74	77.06
w. GPT	**91.62**	**93.63**	**89.79**

Table 3. **Impact of person detectors.** For E2E setting, predicted boxes from OpenMixer are replaced with boxes from Mask RCNN [1] or G-DINO [7], and their classification scores are assigned by maximum IoU with OpenMixer boxes that have scores.

models	person boxes	J-HMDB			UCF101-24		
		Mean	Base	Novel	Mean	Base	Novel
ZSR+ZSL	MaskRCNN [1]	66.73	64.61	68.66	35.01	34.59	35.43
	G-DINO [7]	69.72	67.09	72.12	45.43	44.82	46.04
E2E	Mask RCNN [1]	83.51	87.45	79.92	42.31	48.48	36.13
	G-DINO [7]	85.06	87.76	82.60	46.56	47.00	46.11

D. Additional Results

Impact of VLMs. We note there is a line of literature [2, 6, 8, 11, 12] built on image CLIP for open-vocabulary video understanding. Therefore, it is interesting to see whether image CLIP also works for the OVAD task. In Tab. 1, we compare OpenMixer with its variants using video-based CLIP-ViP [15] and image-based CLIP [10] under the same ViT-B/16 architecture. The results show that the OpenMixer with CLIP performs way worse than the model with CLIP-ViP, because of the limited capacity of image CLIP in capturing video actions.

Can GPT help temporal action localization? This question is interesting as how textual prompts from language models like GPT could help temporal localization has not been explored in literature. In Tab. 2, we show that by evaluating the temporal action localization performance, GPT prompts could significantly help.

Impact of person detectors. In Tab. 3, we compare the impact of using external person boxes from off-the-shelf person detectors, *i.e.*, G-DINO [7] and Mask RCNN [1], in test time on the two best-performed models under the ZSR+ZSL and E2E settings, respectively. It shows that the high-quality boxes from G-DINO could consistently outperform those from Mask RCNN. With the same external test-time boxes, the results of OpenMixer model are consistently better than those of the strongest ZSR+ZSL baseline (Video+GPT). The relatively smaller gains on UCF101-24 than the gains on J-HMDB can be explained by the background bias in UCF videos that restricts VLMs in action recognition.

Table 4. **Impact of the location priors noise.** We analyze the performance impact from the noise level aspect of the location prior for initializing the box queries of the first S-OMB block.

	priors from	noise level	Mean	Base	Novel
(a)	G.T. (UB)	clean	91.19	93.23	89.34
(b)	detection	moderate	83.92	88.19	80.03
(c)	random (LB)	serious	54.15	56.50	52.02
ours	attention map	slight	86.34	90.75	82.33

Table 5. **Generalized zero-shot testing**. A complete vocabulary of base and novel categories is given in testing.

Models	J-HMDB			UCF101-24		
	Mean	Base	Novel	Mean	Base	Novel
STMixer [13]	36.26	55.71	18.57	28.72	53.42	4.02
OpenMixer	74.28	77.72	71.16	40.07	54.00	26.14

Impact of location prior noise. In Table 4, we compare ours with 3 variants that use location priors from **(a)** ground truth (G.T.) boxes which can be regarded as clean without noise and upper-bound (UB) the performance, **(b)** detected person boxes that may be moderately noisy, and **(c)** uniform random boxes that are completely noisy and lower-bounds (LB) the performance. The results show our location priors, which are sampled from the text-patch attention map, perform much better than the baselines (b)(c), and are close to the upper-bound performance in (a).

Generalized zero shot testing. In our main paper, the base and novel categories are individually given in testing. Thus, in Table 5, we additionally present the results of the generalized zero-shot testing, in which a complete vocabulary of base and novel categories is given for each testing video. This is more challenging but our OpenMixer still keeps outperformance than the STMixer baseline [13]. Moreover, according to [16, 17], the rankings of models are stable by the two testing protocols, and only the scales of numbers are different. Therefore, the efficacy of models can still be validated by individual testing in our main paper.

Results on Different Splits. We experiment with five random 50%-50% seen-unseen class splits on both the J-HMDB and UCF101-24 datasets. Full results of video mAP are summarized in Tab. 6 and 7. The split (0) is used in all experiments of the main paper. We also experiment with five random 75%-25% seen-unseen class splits on the two

Table 6. **Results on 50%-50% J-HMDB splits.**

Metrics	(0)	(1)	(2)	(3)	(4)	avg
Mean	86.34	86.29	85.50	86.73	83.40	85.65
Base	90.75	89.89	89.20	87.70	85.36	88.58
Novel	82.33	83.02	82.13	85.85	81.61	82.99

Table 7. **Results on 50%-50% UCF101-24 splits.**

Metrics	(0)	(1)	(2)	(3)	(4)	avg
Mean	46.42	46.28	45.45	47.32	48.30	46.75
Base	59.10	61.11	55.85	62.33	61.25	59.93
Novel	33.73	31.45	35.05	32.31	35.34	33.58

Table 8. **Results on 75%-25% J-HMDB splits.**

Metrics	(0)	(1)	(2)	(3)	(4)	avg
Mean	75.96	79.43	79.77	81.88	86.56	80.72
Base	74.73	75.21	78.34	82.14	85.46	79.17
Novel	79.03	89.98	83.34	81.23	89.30	84.57

Table 9. **Results on 75%-25% UCF101-24 splits.**

Metrics	(0)	(1)	(2)	(3)	(4)	avg
Mean	55.78	55.83	57.04	57.19	61.85	57.54
Base	64.85	61.83	60.16	58.74	61.82	61.48
Novel	28.55	37.80	47.69	52.55	61.96	45.71

datasets, and report results in Tab. 8 and 9. As some of human actions are much harder to detect than others and they could be included into either base or novel categories, it is normal that the overall performances on different splits vary significantly. Following the existing literature, we will release all splits.

E. Visualizations

We present more visualizations on the J-HMDB dataset and UCF101-24 in Fig. 3 and 4, respectively. They show that our method could detect human actions with precise bounding boxes for both seen and unseen actions. Specifically, in scenarios where multiple persons exist, for the examples of the seen action `Volleyball Spiking` and the unseen action `Ice Dancing` on the UCF101-24 dataset, our method could still localize the action-relevant persons on most frames. Referring to single-person action detection, there is still room to improve the performance of multi-person action detection in the future.

F. Comparison with Concurrent Work [14]

The prior work [14] defines the same task setting and identifies similar challenges as ours. However, there are several important differences in terms of technical motivations and design. First, for the roadmap, [14] focuses on large-scale video region-text pre-training followed by downstream fine-tuning, while we emphasize the model adaptation to small downstream datasets in one-time training. Second, for model design, [14] is a two-stage method with region proposal generation and action detection refinement, while we adopt DETR-like end-to-end design. As for empirical comparison, currently, this is not feasible because (1) the [14] is a concurrent work as ours without releasing any code, data, and models (during the submission period), and (2) it is not an apple-to-apple comparison since the data splits and evaluation metrics of the benchmarks in [14] are different from ours as indicated in the paper [14].

G. Limitations and Future Work.

The recent large-scale action detection dataset AVA [**?**] is not included in this paper, as we emphasize the adaptation of existing pre-trained VLMs for downstream small datasets. In the future, similar to the concurrent work [**?**], we will explore how to effectively pre-train on AVA to benefit for a more general audience.

References

[1] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV. pp. 2961–2969 (2017) 2

[2] Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: ECCV. pp. 105–124 (2022) 2

[3] Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: ECCV. pp. 280–296 (2022) 1

[4] Li, Y., Wang, H., Duan, Y., Li, X.: Clip surgery for better explainability with enhancement in open-vocabulary tasks. arXiv preprint arXiv:2304.05653 (2023) 1

[5] Li, Y., Wang, H., Duan, Y., Xu, H., Li, X.: Exploring visual interpretability for contrastive language-image pre-training. arXiv preprint arXiv:2209.07046 (2022) 1

[6] Liu, R., Huang, J., Li, G., Feng, J., Wu, X., Li, T.H.: Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In: CVPR. pp. 6555–6564 (2023) 2

[7] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023) 2

[8] Nag, S., Zhu, X., Song, Y.Z., Xiang, T.: Zero-shot temporal action detection via vision-language prompting. In: ECCV. pp. 681–697 (2022) 2

[9] OpenAI: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 1

[10] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021) 2

[11] Rathod, V., Seybold, B., Vijayanarasimhan, S., Myers, A., Gu, X., Birodkar, V., Ross, D.A.: Open-vocabulary temporal action detection with off-the-shelf image-text features. In: BMVC (2022) 2

[12] Wang, M., Xing, J., Liu, Y.: Actionclip: A new paradigm for video action recognition. arXiv preprint arXiv:2109.08472 (2021) 2

[13] Wu, T., Cao, M., Gao, Z., Wu, G., Wang, L.: Stmixer: A one-stage sparse action detector. In: CVPR. pp. 14720–14729 (2023) 1, 2

[14] Wu, T., Ge, S., Qin, J., Wu, G., Wang, L.: Open-vocabulary spatio-temporal action detection. arXiv preprint arXiv:2405.10832 (2024) 3

[15] Xue, H., Sun, Y., Liu, B., Fu, J., Song, R., Li, H., Luo, J.: Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. In: ICLR (2022) 2

[16] Zareian, A., Rosa, K.D., Hu, D.H., Chang, S.F.: Open-vocabulary object detection using captions. In: CVPR. pp. 14393–14402 (2021) 2

[17] Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L.H., Zhou, L., Dai, X., Yuan, L., Li, Y., et al.: Regionclip: Region-based language-image pretraining. In: CVPR. pp. 16793–16803 (2022) 2

```
{
"brush_hair": "Brush Hair: A person is brushing their hair using hand movements with a hairbrush or their fingers.",
"catch": "Catch: Someone is catching an object, such as a ball or a frisbee, with their hands.",
"clap": "Clap: A person is bringing their hands together to create a clapping sound.",
"climb_stairs": "Climb Stairs: Someone is ascending or descending a set of stairs, using alternating leg movements.",
"golf": "Golf: A person is swinging a golf club to hit a golf ball.",
"jump": "Jump: Someone is propelling themselves off the ground using both feet simultaneously.",
"kick_ball": "Kick Ball: A person is striking a ball with their foot.",
"pick": "Pick: Someone is picking up an object from the ground, usually using their hands.",
"pour": "Pour: A person is pouring liquid from one container to another.",
"pullup": "Pull Up: Someone is lifting their body upwards using their arms, typically performed on a horizontal bar.",
"push": "Push: A person is exerting force on an object away from their body, using their hands or body.",
"run": "Run: Someone is moving quickly on their feet, usually in a straight line.",
"shoot_ball": "Shoot Ball: A person is shooting a ball towards a target or a goal using their hands or feet.",
"shoot_bow": "Shoot Bow: Someone is using a bow to shoot an arrow.",
"shoot_gun": "Shoot Gun: A person is firing a gun, typically aimed at a target.",
"sit": "Sit: Someone is in a seated position with their weight supported by a surface, such as a chair.",
"stand": "Stand: A person is upright on their feet, with their body fully supported by their legs.",
"swing_baseball": "Swing Baseball Bat: Someone is swinging a baseball bat to hit a ball.",
"throw": "Throw: A person is propelling an object through the air using their hand or arm.",
"walk": "Walk: Someone is moving on their feet with a regular, steady pace, but slower than running.",
"wave": "Wave: A person is moving their hand or arm back and forth in a greeting or farewell gesture, usually with an
    open palm."
}
```

Figure 1. Generated prompts for J-HMDB action categories. For each category, we generate one prompt sentence.

```
{"Basketball": [
   "Basketball: A player dribbles the ball swiftly down the court amidst cheers from the crowd.",
   "Basketball: An athlete performs a high jump and slam dunks the ball into the net with confidence.",
   "Basketball: Teammates pass the ball around the court, strategizing their next move.",
   "Basketball: A player precision shoots the ball from the three-point line and scores.",
   "Basketball: A tense one-on-one standoff as a player attempts to steal the ball.",
   "Basketball: Players execute deft maneuvers around opponents on the court.",
   "Basketball: A player displays impressive footwork while maintaining control of the ball.",
   "Basketball: Following a whistle blow, a player steps up to take a free throw.",
   "Basketball: The coach calls a timeout to relay new strategies to the team.",
   "Basketball: A swift breakaway leads to a stunning layup and two points on the board.",
   "Basketball: Thorny defense put up by players trying to prevent the opposing team from scoring.",
   "Basketball: The player manages to steal the ball, intercepting a pass and turning the game around.",
   "Basketball: In the sound of the last buzzer, players celebrate a well-earned victory.",
   "Basketball: Spectators erupt in cheers as the ball swishes through the net.",
   "Basketball: A captivating display of agility and teamwork witnessed on the court.",
   "Basketball: A player makes a long, arching shot from the half-court line, electrifying the crowd."
],
.
.
.
"TrampolineJumping": [
   "Trampoline Jumping: A joyful child is leaping high on a trampoline in their backyard.",
   "Trampoline Jumping: A gymnast is skillfully performing somersaults on a trampoline.",
   "Trampoline Jumping: A group of friends are competing in tricks while bouncing on a trampoline.",
   "Trampoline Jumping: A professional athlete is executing a perfect backflip on a trampoline. ",
   "Trampoline Jumping: Enthralled family members are enjoying a trampoline jump session on a sunny day.",
   "Trampoline Jumping: Excited children are bouncing and laughing on a trampoline at a birthday party.",
   "Trampoline Jumping: A fitness enthusiast is getting an intense workout by jumping on a trampoline.",
   "Trampoline Jumping: An acrobat rehearses complicated maneuvers on a large trampoline. ",
   "Trampoline Jumping: A fearless teenager is executing high jumps on a trampoline in a skate park.",
   "Trampoline Jumping: An adventurous person is defying gravity with bounces on a massive trampoline.",
   "Trampoline Jumping: A young girl confidently performs flips and twists on a trampoline. ",
   "Trampoline Jumping: A trampoline athlete practices precise landings in a professional gym.",
   "Trampoline Jumping: An aspiring gymnast is perfecting their routine on a trampoline.",
   "Trampoline Jumping: A boy exhilaratingly jumps towards the sky on a trampoline, his laughter filling the air.",
   "Trampoline Jumping: A daring young woman is doing mid-air splits on a trampoline in an indoor park.",
   "Trampoline Jumping: A man is reaching extreme heights, all while being propelled off a trampoline."
]}
```

Figure 2. Generated prompts for UCF101-24 action categories. For each category, we generate 16 prompt sentences.

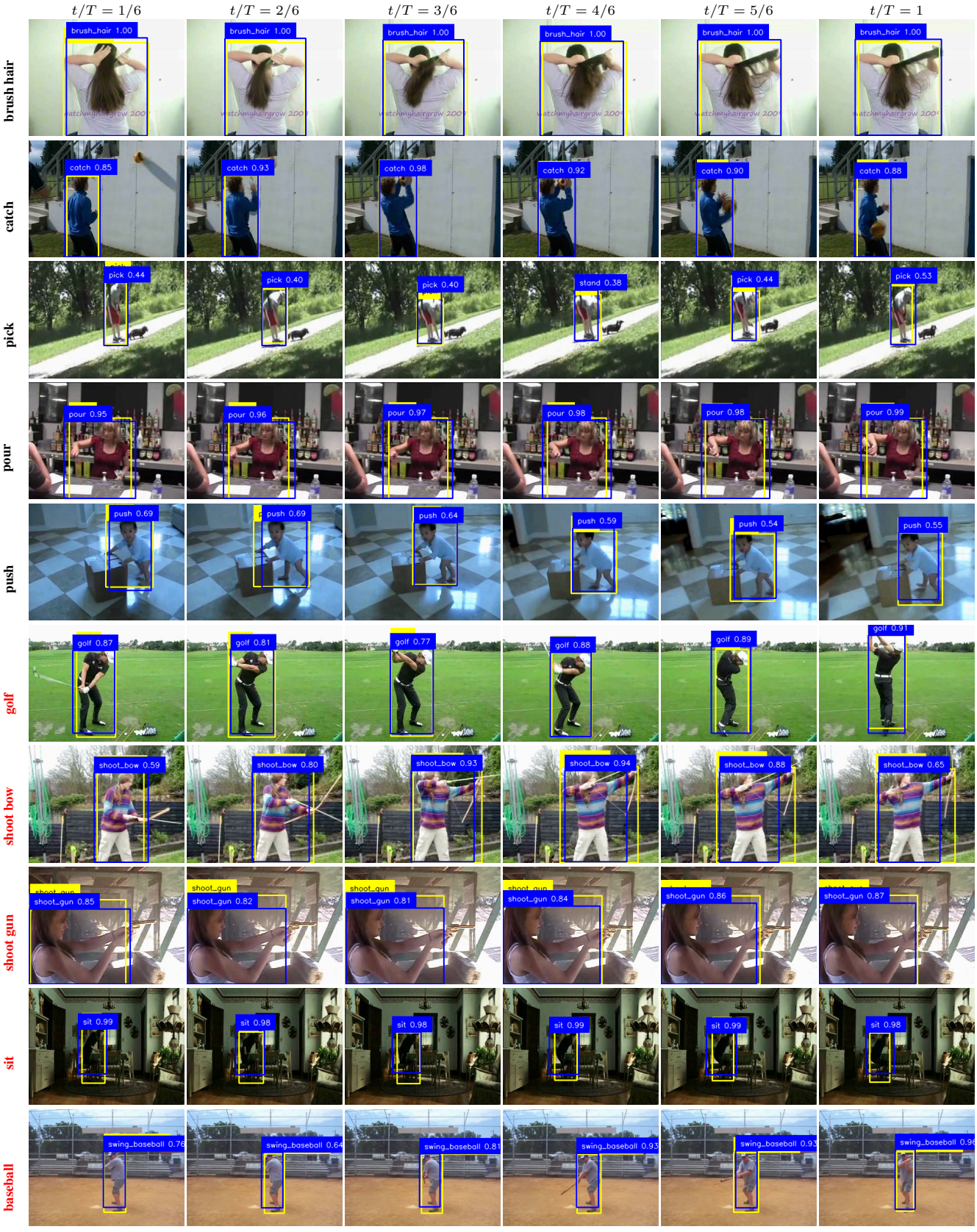

Figure 3. **Visualization on J-HMDB dataset.** We visualize our OpenMixer detections (in blue boxes) and ground truth (in yellow boxes) on five base classes (in **black font**) and five novel classes (in **red font**). Class names are shortened for brevity. The numbers after class names are confidence scores.

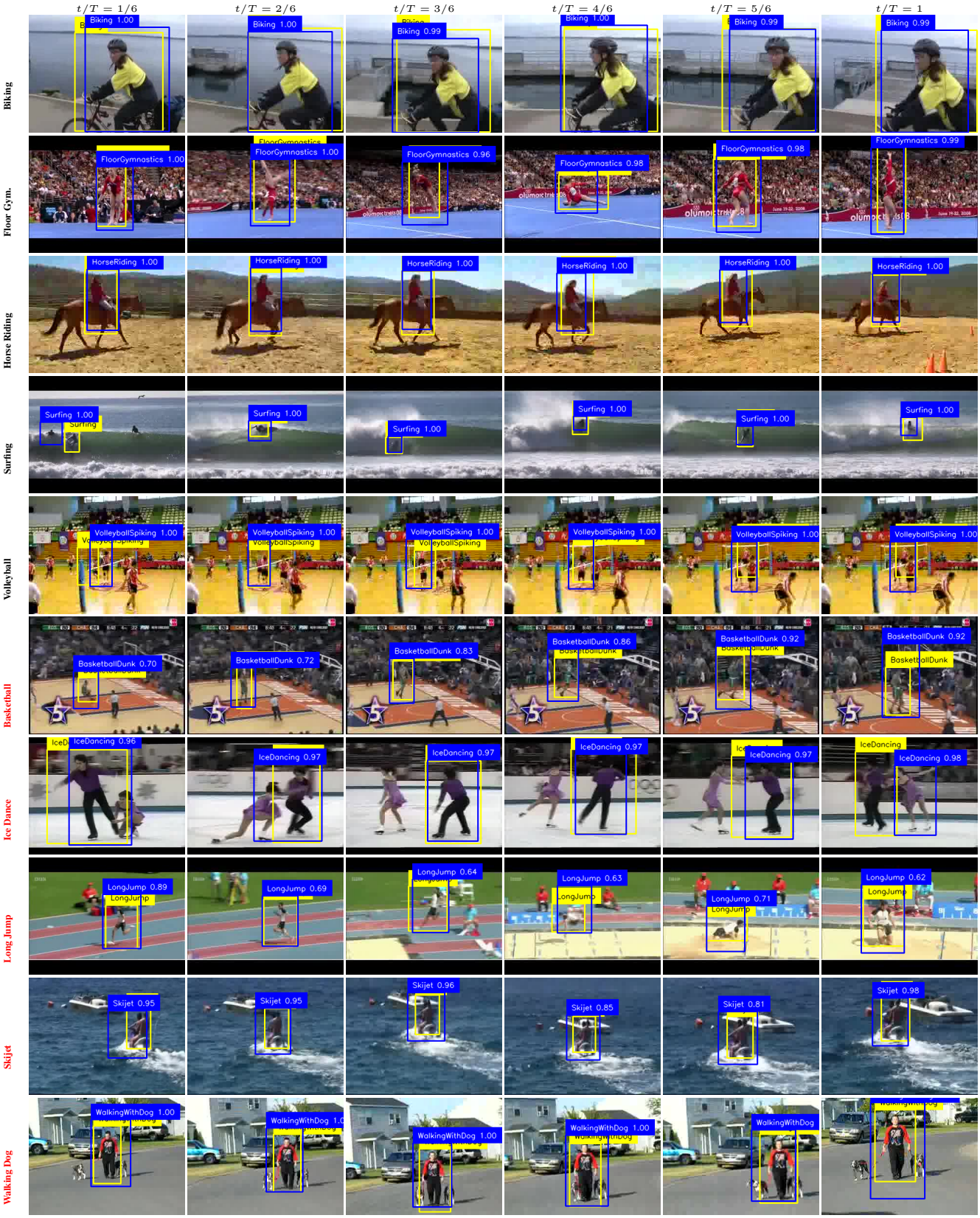

Figure 4. **Visualization on UCF101-24 dataset.** We visualize our OpenMixer detections (in blue boxes) and ground truth (in yellow boxes) on five base classes (in **black font**) and five novel classes (in **red font**). Class names are shortened for brevity. The numbers after class names are confidence scores.