Supplementary Material for
RCL: Recurrent Continuous Localization for Temporal Action Detection

Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan
DAMO Academy, Alibaba Group
{qishi.wq, yanhao.zyh, zhengyun.zy, panpan.pp}@alibaba-inc.com

Appendix
In this supplementary material, we provide additional implementation details for our method (Section 1) and show additional qualitative results (Section 2).

1. Implementation Details

In this section, we present some implementation details that were omitted in the main paper for brevity.

The RCL module is general, and can be applied to other action detection frameworks [4, 7]. BMN [4] is a grid-based detector which utilizes the boundary-matching network to improve the efficiency for retrieving proposals [5]. Our RCL module can be directly applied to replace the proposal evaluation module (as Figure 1). G-TAD [7] is a state-of-the-art action detector which employs a well-designed graph module, GCNeX, to improve the temporal representation. Since there is no special optimization objects for the architecture, we use the improved feature in our RCL framework. Other settings are maintained the same as those for BMN.

1.1. General components for our baseline

Base Feature is the part that enhances the original snippet features with more context. BMN adopts stacks of local convolutions to capture local-range context. G-TAD utilize structured graphs to model long-range dependencies. TEM is the part that predicts the starting and ending probabilities for all temporal locations. BMN use these boundary probability to generate more reliable confidence scores. G-TAD and our RCL utilize this module to regularize the training process. PEM is the part that predicts tIoU and classification scores on dense locations of feature maps. Specifically, BMN use Boundary-Matching Layer [4] to sample all segment features, and G-TAD utilizes SGAlign [7] to extract sub-graph features.

By forwarding the snippet features into a convolutional network (see Figure 2), the proposal confident scores are computed which can be represented to the $2 \times D \times T_s$ feature maps. For THUMOS14 [3] and ActivityNet v1.3 [2], we set $D = 64, T_s = 256$ for THUMOS14 and $D = 100, T_s = 100$ for ActivityNet v1.3 for both baseline [4, 7].

Feature alignment for CAR. As shown in Figure 3, we use the nearest regular grid feature as the temporal representation and add two additional input channel for temporal coordinates. We achieve continuity by densely sampling coordinates. For the regular grid feature, we sample 32 points and extend the boundary region with 0.5 segment length, following the original implementations [4].

Network for RRM. Inspired by the recent success in
Recurrence All-Pairs Field Transforms for Optical Flow (RAFT [6]), we utilize the SmallUpdateBlock to refine the confidence maps. The update operator is a gated activation unit based on the variant of GRU cell:

\[
\begin{align*}
    z_t &= \sigma \left( \text{Conv}_{3 \times 3} \left( [h_{t-1}, x_t], W_z \right) \right) \\
    r_t &= \sigma \left( \text{Conv}_{3 \times 3} \left( [h_{t-1}, x_t], W_r \right) \right) \\
    \tilde{h}_t &= \tanh \left( \text{Conv}_{3 \times 3} \left( [r_t \odot h_{t-1}, x_t], W_h \right) \right) \\
    h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t
\end{align*}
\]

where \( x_t \) is the concatenation of confidence scores \( G_t \), temporal features, and context features. The main forward code are shown in Figure 4.

2. Additional Results

**Scale imbalance:** In Figure 5 we show the sensitivity average-mAP\(_N\) analysis, generated by the DETAD toolbox [1], for BMN baseline with different sample densities.

First, we compared three sampling methods: (1) regular grid sampling, (2) reducing the sample ratio of 0.5 \times for region with the duration greater than 0.5. (3) reducing the sample ratio of 0.5\(\times\) for region with the duration parameter \( l = \lceil \text{duration}/0.2 \rceil \). These sparse sample strategy are proposed in 2D-TAN [8]. By reducing the sampling density for the long-term segment, we can see that the overall performance has not dropped too much and the mAP\(_N\) for Coverage-XS is improve.

Note that while the sparse sampling method uses fewer anchors, the amount of calculation remains the same as the regular one. The discretized grid structure inherently limits the flexibility of dynamic sampling. To maximally excavate redundancy in the grid representation, our RCL can dynamically instantiate any segments. Compared with regular grid sample strategy and the sparse sample strategy, our continuous representation and scale-invariant sample strategy has a great advantage in predicting short-term and also performs better for long segments.

**Iterative steps for RRM:** As shown in Table 1, our recurrent refine module (RRM) achieves the best results in 10 iterations, which only adds 17% FLOPs. Our proposed model obtains a good trade-off between accuracy and computational cost. More iterations usually improve mAP and improvements saturate at 10 steps, which may be related to overfitting.
References


