A. More training details

We train our models in 16 or 64 GPUs (NVIDIA Tesla V100), using Stochastic Gradient Descent (SGD) with momentum 0.9 and cosine learning rate schedule. In order to prevent overfitting, we add a dropout layer before the classification layer of each pathway in the BQN model. Following the experimental settings in [10, 13], the learning rate and weight decay parameters for the classification layers are 5 times of the convolutional layers. Meanwhile, we only apply L2 regularization to the weights in the convolutional and classification layers to avoid overfitting.

Hyperparameters for models based on ResNet. For Kinetics400 [3], the initial learning rate, batch size, total epochs, weight decay and dropout ratio are set to 0.08, 512 (8 samples per GPU), 100, 2e-4 and 0.5, respectively. For Something-Something V1 [5], these hyperparameters are set to 0.12, 256, 50, 8e-4 and 0.8, respectively. We use linear warm-up [11] for the first 7 epochs to overcome early optimization difficulty. When fine-tuning the Kinetics models on UCF101 [12] and HMDB51 [9], we freeze all of the batch normalization [8] layers except for the first one to avoid overfitting, following the recipe in [13]. The initial learning rate, batch size, total epochs, weight decay and dropout ratio are set to 0.001, 64 (4 samples per GPU), 10, 1e-4 and 0.8, respectively.

Hyperparameters for models based on X3D-M. For Kinetics400, the initial learning rate, batch size, total epochs, weight decay and dropout ratio are set to 0.4, 256 (16 samples per GPU), 256, 5e-5 and 0.5, respectively. For Something-Something V1, the models trained from scratch use the followings hyperparameters: learning rate 0.2, batch size 256, total epochs 100, weight decay 5e-5 and dropout ratio 0.5. When fine-tuning the Kinetics models, the initial learning rate, batch size, total epochs, weight decay and dropout ratio are set to 0.12, 256 (16 samples per GPU), 60, 4e-4 and 0.8, respectively.

B. More Studies for the MBPM settings

We search for the optimal settings of the scale $\sigma$ and kernel size $k \times k$ of MBPM on UCF101. The results are presented in Figure 1. We observe that the experimental results vary greatly under different settings. Nevertheless, the optimal scale is $\sigma = 1.1$ when setting the kernel size as $9 \times 9$, which is the same as that on Something-Something dataset. Furthermore, we try a larger kernel ($11 \times 11$), but it shows a performance drop. We speculate that this is caused by insufficient training.

C. Implementation details of Efficiency and Effectiveness of the MBPM

These additional explanations are useful for Section 5.2 from the main paper. We provide the implementation details for the comparative experiments of MBPM with other mainstream motion representation methods [2, 4, 7, 13, 14, 15]. We follow the experimental settings on PA [15] for fair comparison. The backbone network for all the methods is ResNet50 [6]. We use the computer code provided by the original authors for these methods to generate the network inputs. For any kind of motion representation, we divide the representation of a video into 8 segments and randomly select one frame of the representation for each segment. Following the practices in TSN [13] and PA [15], the output activations of 8 segments are averaged for the final predic-
Figure 2: Visualization of the first channels of the 64 conv1 filters of BQN after training on Kinetics400. All the 64 filters have a size of $7 \times 7$. From left to right, in (a), (b) and (c), we respectively present the trained conv1 filters in the Busy pathway, Quiet pathway and TSM ResNet50. We observe that the kernels of the 64 filters in the Busy pathway have a similar line-like shape, while those for the filters in the Quiet pathway are more like larger blobs. The conv1 in TSM ResNet50 (baseline) contains both types of filters from the Busy and Quiet pathways. Best viewed in color and zoomed in.

Figure 3: Visualization of the spatial channel-wise convolution $LoG_{1 \times k \times k}$ of MBPM in the Busy pathway before and after training on Kinetics400. The $9 \times 9$ channel-wise convolution is initialized with a Laplacian of Gaussian with the scale parameter $\sigma = 1.1$. Best viewed in color and zoomed in.

D. Visualization examples

These additional explanations and results are useful for Section 5.2 from the main paper. In order to visually observe the difference between our MBPM and other motion representation method, in Figure 4, we show some example video frames and their corresponding motion representations generated by different methods. For a better view, we use the optical flow visualization approach used in [1] to vi-
ualize the output of MBPM. The optical flow estimates the instantaneous velocity and direction of movement in every position (The color represents the direction of movement while the brightness represents the absolute value of instantaneous velocity in a position). In contrast, TVNet [4], PA [15] and MBPM are more absorbed in the visual information presented in boundary regions where motion happens. Given that TVNet [4] is based on the optical flow estimations, which lose the color information, the information of different channels is still unknown. Meanwhile, PA [15] generates the motion representations of a single channel, which does not preserve the RGB color information. However, when simply passing the RGB frames to the proposed MBPM, the color information in these motion boundaries is still perfectly preserved. Figures 5–6 display the motion representation extracted by the MBPM for eight different sequences from various video datasets used for the experiments.

E. Kernel visualization

In Figure 3, we visualize the kernel of the spatial convolution $LoG_{2\times k \times k}$ of MBPM in the Busy pathway. Interestingly, before and after training, kernels always present a similar shape to Mexican hats in 3-dimensional space. In Figure 2, we visualize the first channel of the 64 filters in the first layers of the BQN and the baseline (TSM ResNet50). We can observe that the Busy and Quiet pathways’ filters have quite distinct shapes in their kernels, suggesting that the Busy and Quiet pathways learned different types of features after training.

References

Figure 5: Examples of video and the corresponding motion representations extracted by MBPM.
“Playing poker” from Kinetics400

“Moving something away from something” from Something-Something V1

“Playing basketball” from Kinetics400

“Holding something next to something” from Something-Something V1

Figure 6: Examples of video and the corresponding motion representations extracted by MBPM.