Active Inference for Micro-Gesture Recognition: EFE-Guided Temporal Sampling and Adaptive Learning

Supplementary Material

1. Implementation Details

All models are implemented using PyTorch ($version \geq 1.11$). All experiments are conducted on a single NVIDIA A100 GPU (80 GB memory). The overall system follows a multi-stage pipeline consisting of feature extraction, temporal selection, spatial reasoning, uncertainty-aware augmentation, and final VFE-based optimization.

1.1. Feature Extraction

Two types of features are extracted from the backbone and used in different stages of our framework. The global 2D feature vectors obtained through the average pooling layer of the backbone are used to estimate the temporal uncertainty and train the initial classifier. The 4D convolutional feature maps, extracted by taking the output before the global pooling layer, preserve the spatial structure and are used for spatial selection and final classification. Each 4D feature map has a shape [B, 2048, 7, 7].

1.2. Temporal Uncertainty Estimation

To enable active frame selection, the initial classifier is trained on 2D features and used to compute predictive uncertainty via Monte-Carlo dropout. During evaluation, dropout layers are kept active, and each feature is forward-propagated through the classifier for T=5 times. The averaged prediction forms the approximate posterior distribution for each frame, on which the Expected Free Energy (EFE) is computed to guide frame selection.

1.3. Spatial Attention Module

Once key frame has been selected by the temporal module, spatial inference is performed on its 4D convolutional feature map using a lightweight spatial attention mechanism. Given an input tensor of shape [B,2048,7,7], we first compute the channel-wise average map and maximum map, which encode complementary global and salient responses. These two maps are concatenated to form a tensor of shape [B,2,7,7] and passed through a 7×7 convolution with padding 3, followed by a sigmoid activation to generate a spatial attention mask of size [B,1,7,7]. The mask is applied to the original feature map via element-wise multiplication, yielding an attended representation that highlights gesture-relevant regions. A global average pooling operation then reduces the attended tensor to a 2048-dimensional vector, which is subsequently fed into a linear classifier to obtain the final gesture prediction.

1.4. Uncertainty Estimation for UMIX

To construct uncertainty-aware mixup samples, we compute predictive uncertainty via Monte-Carlo dropout. Dropout layers remain active at test time, and each feature is evaluated through T=5 stochastic forward passes. The variance across the predicted class probabilities is used as the uncertainty score for UMIX sample re-weighting.

2. Ablation Studies

2.1. Ablation Study on Batch Size

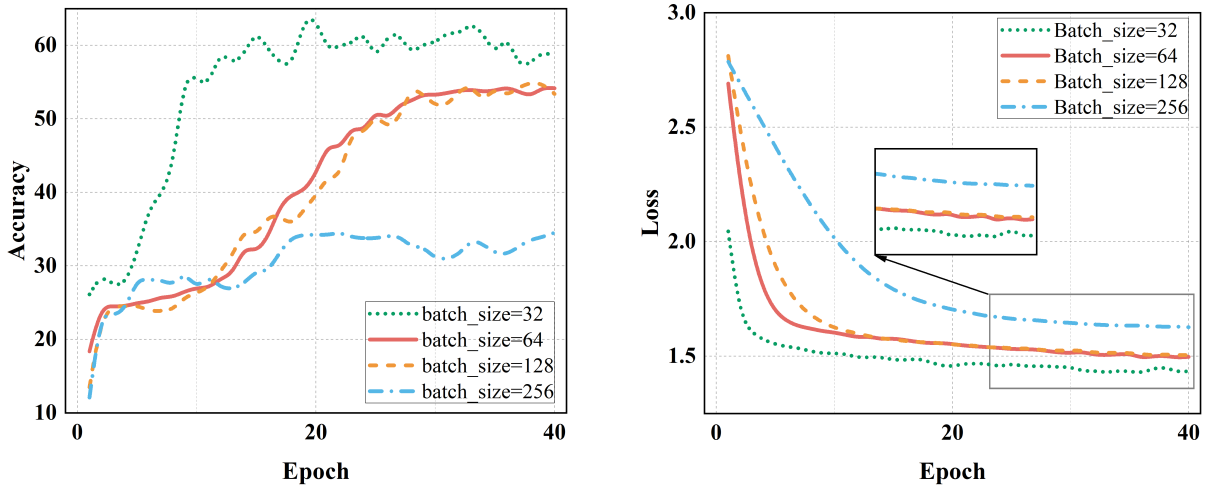
To evaluate the influence of batch size, we trained our model under four configurations: 32, 64, 128, and 256. As illustrated in Figure 1, a batch size of 32 achieves the fastest convergence and the highest final accuracy, benefiting from the moderate stochasticity introduced by smaller batches. This stochasticity acts as an implicit regularizer and allows the optimizer to escape sharp minima, leading to better generalization.

A batch size of 64 yields relatively stable convergence but slightly inferior accuracy compared with 32. Larger batch sizes of 128 and 256 converge significantly more slowly and exhibit noticeable degradation in final performance. This behavior is consistent with prior findings that excessively large batches reduce gradient noise, push the optimization toward overly deterministic trajectories, and lead to sharp minima that generalize poorly.

Therefore, we adopt batch size = 32 for all main experiments as it offers the best trade-off between training stability and generalization.

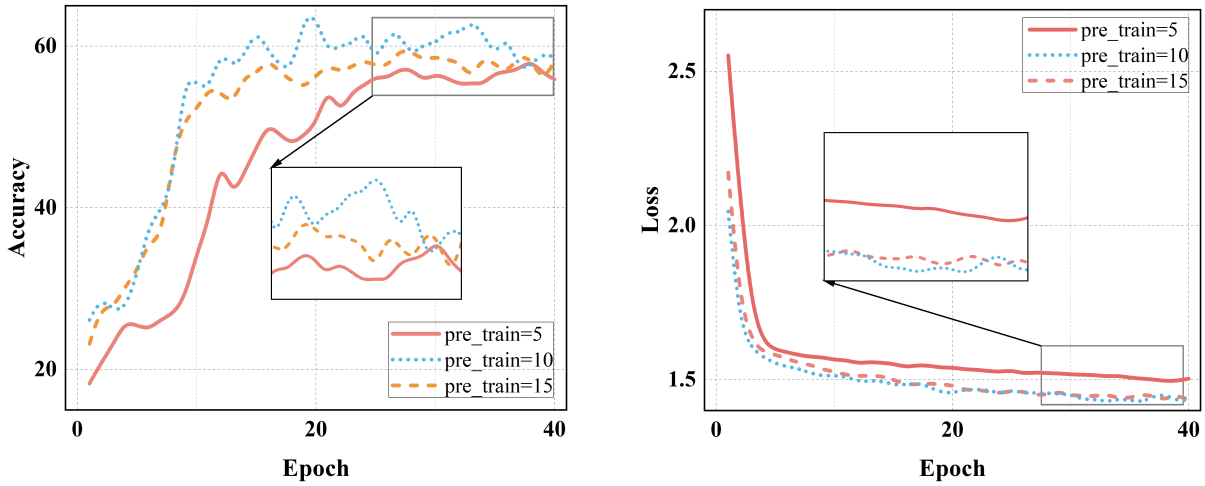
2.2. Ablation Study on Pre-training Epoch

We further investigate how the number of pre-training epochs for the initial classifier influences the final performance of the full pipeline. As shown in Figure 2, pre-training for only 5 epochs leads to rapid early convergence but the overall accuracy remains suboptimal, indicating that the initial classifier does not provide sufficiently calibrated predictions for the subsequent EFE-based temporal selection. Increasing the pre-training to 10 epochs yields the highest accuracy and the most stable learning trajectory, suggesting that the classifier at this stage captures discriminative patterns without overfitting. Extending the pre-training to 15 epochs produces slightly weaker results, likely because an overly confident initial classifier reduces the diversity and usefulness of the posterior distributions employed in active inference. Based on these observations,



(a) Accuracy Curve (b) Loss Curve

Figure 1. Accuracy and Loss Curves under Different batch sizes



(a) Accuracy Curve (b) Loss Curve

Figure 2. Accuracy and Loss Curves under Different pre-training epochs

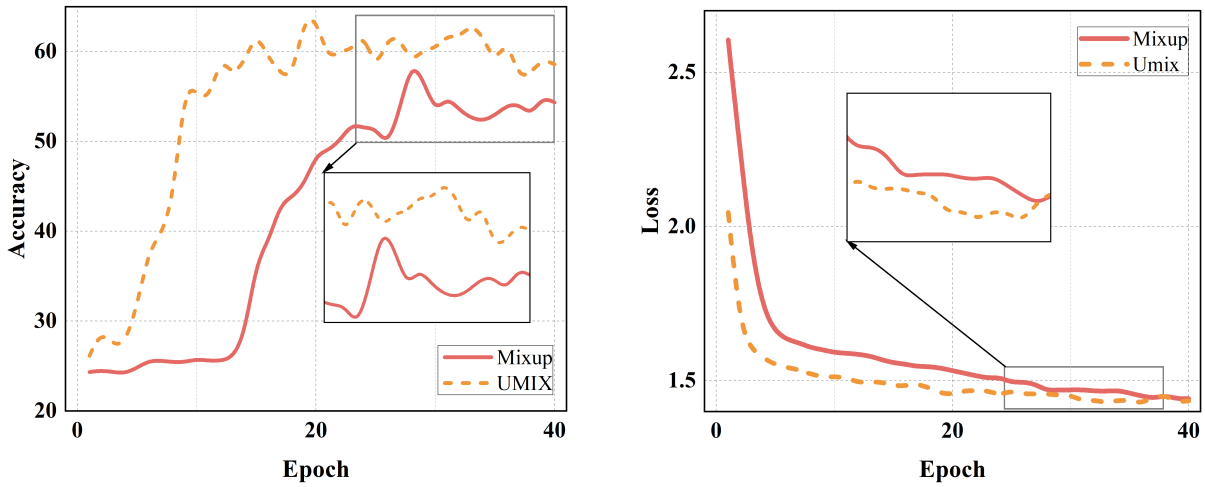
we use 10 epochs of pre-training as the default configuration.

2.3. Ablation Study on Mixup and UMIX

We compare our uncertainty-guided UMIX strategy with the standard Mixup augmentation to evaluate the effect of incorporating epistemic uncertainty into sample interpolation. As shown in Figure 3, Mixup improves model stability during early training but its accuracy plateaus at a relatively low level. In contrast, UMIX achieves consistently higher accuracy throughout the training process and exhibits more stable convergence after the midpoint of training. This improvement arises from the fact that UMIX adaptively adjusts the contribution of each mixed sample based on its uncertainty, thereby reducing the influence of unreliable or noisy samples while enhancing the utility of informative ones. The loss curves further confirm this behavior: UMIX maintains a lower training loss across epochs, indicating a more effective optimization trajectory and better calibrated gradients. These results demonstrate that incorporating uncertainty into the data mixing process leads to more robust representations and yields superior performance compared to conventional Mixup.

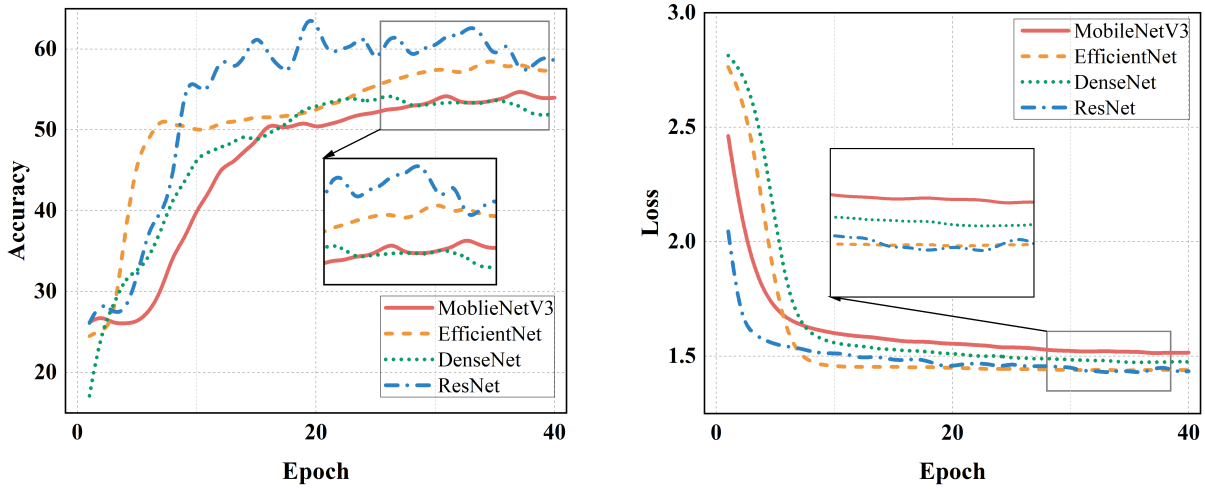
2.4. Ablation Study on Backbone

We further conducted an ablation study to examine how different backbone architectures influence the overall performance of our pipeline. Under identical training configurations, we compared four commonly used CNN backbones: MobileNetV3, EfficientNet-B0, DenseNet-121, and



(a) Accuracy Curve (b) Loss Curve

Figure 3. Accuracy and Loss Curves under Mixup and UMIX



(a) Accuracy Curve (b) Loss Curve

Figure 4. Accuracy and Loss Curves under Different Backbones

ResNet-50. As shown in Figure 4, lightweight models such as MobileNetV3 and EfficientNet converge quickly but suffer from limited representation capacity, leading to clearly lower final accuracy. DenseNet-121 achieves better performance than EfficientNet but exhibits unstable convergence and higher sensitivity to noisy temporal frames. In contrast, ResNet-50 achieves the best balance between convergence speed and final accuracy. Its deeper architecture produces more discriminative and more stable spatio-temporal representations, leading to smoother predictive distributions and more reliable frame-level uncertainty estimates within our active-inference framework, which ultimately explains its consistently robust performance. Therefore, we adopt ResNet-50 as the default backbone for all subsequent experiments, including temporal selection, spatial attention, and the UMIX augmentation strategy.