Dear Reviewer pfah,

We sincerely thank the reviewer for such careful attention in our paper, and we express our genuine respect for the reviewer's thorough and responsible approach. The latest version of the paper(marked orange) has been submitted. Please let us know if there are any further issues.

Best,

Paper 1571 Authors

W1. The interpolation module is not logical.

A1. Specifically, the "**Latent Space Temporal Logical Interpolation**" module is composed of "Observation Encoder", "Latent Space Interpolator" and "Transformer Encoder Blocks" in Figure 3(a).

1. Observation Encoder: This part is used to extract the features into latent space, to better capture abstract representations and learn meaningful feature embeddings that are more suitable for our tasks, corresponding to "**Latent Space**".

2. Latent Space Interpolator: The function of this part is to create new visual features based on V_s and V_g, which not only add more diverse temporal characteristics but also preserve the original information, similar to the interpolation of video frames[1]. According to the temporal logic of actions (TLA)[2], which is a logic for specifying and reasoning about concurrent systems, temporal relationships can be considered as a type of logical relationships. This part corresponds to the preliminary modeling of **Temporal** logical relationships using **Interpolation**.

3. Transformer Encoder Blocks: According to Hahn et al.[3], our transformer encoder blocks learn the underlying semantics of these relationships and enhance the logical modeling by incorporating both temporal and causal dependencies into the features, corresponding to **Logical**.

In fact, actions have not only temporal relationships but also causal and other relationships. Therefore, we use "logical" to emphasize that our model considers both temporal and causal dependencies between actions.

[1] Danier, D., Zhang, F., & Bull, D. (2024). LDMVFI: Video frame interpolation with latent diffusion models. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 2).

[2] Lamport, Leslie. "The temporal logic of actions." ACM Transactions on Programming Languages and Systems (TOPLAS) 16.3 (1994): 872-923.

[3] Hahn, C., Schmitt, F., Kreber, J. U., Rabe, M. N., & Finkbeiner, B. (2020). Transformers generalize to the semantics of logics. arXiv preprint arXiv:2003.04218.

W2. The experiment settings and results need clarification.

A2. In the latest version of the paper, we have clarified the experimental settings and results.

For CrossTask, the results in Table 1 (main paper at line 324) are correct under the experiment settings of PDPP, demonstrating the effectiveness of our approach on this dataset.

For COIN and NIV datasets, we have unified the experimental settings across all methods using KEPP's setting and presented our results in Table 3 (main paper at line 384). The evaluation results demonstrate that our approach consistently outperforms the best-performing methods on both datasets, highlighting our model's robust performance across datasets of different sizes.

We also present the results on COIN and NIV under the settings of PDPP in Appendix D, section "Comparison with PDPP" at line 1044. For NIV, our performance is slightly lower, which can be attributed to two main factors:

1. Dataset size: NIV is significantly smaller than CrossTask and COIN, which leads to the model excessively learning detailed patterns from the training data, resulting in reduced generalization ability.

2. Experimental setting differences: PDPP defines states as the window between start and end times, while KEPP uses a 2-second window around start/end times. This difference allows PDPP to access more step information, especially for short-term actions, potentially weakening the impact of our interpolation feature supplementation.

Despite these challenges with NIV under PDPP settings, it's important to note that our model demonstrates strong capabilities on larger CrossTask and COIN, showcasing its effectiveness in temporal logic and memory utilization.

W3. The paper needs ablation studies to show how each proposed component contributes to the performance.

A3. We have added the relevant experiments in Appendix C under the section "Ablation for Our Different Methods" at line 971. The experimental results in Table 1 demonstrate that our proposed components significantly improve the model's performance. Other experiments on COIN and NIV are also provided in Appendix C at line 972.

Table 1: Ablation study with our proposed components on CrossTask.

ID	M	K	L	SR↑	mAcc↑	mIoU↑
1				37.20	64.67	66.57
2	✓			39.03	66.49	68.26
3		✓		38.88	66.36	68.35
4			✓	38.57	66.02	68.17
5	✓	✓		39.64	66.74	68.77
6	✓		✓	39.71	66.65	68.83
7		✓	✓	39.17	66.49	68.38
8	✓	✓	✓	**40.45**	**67.19**	**69.17**

Note: M: Our latent space temporal logical interpolation module, K: mask projection, L: task-adaptive masked proximity loss. The results of ID 1 are from PDPP.

Q1. Clarify which components in MTID are inherited from PDPP.

A4. Our method builds upon the base PDPP model, inheriting several components while addressing its limitations. Here are the inherited components and our improvements:

1. **Input Matrix**:
    - Inherited: We maintain the same input matrix design.
    - Drawback: This design leads to the model predicting actions that are less likely to fall within the current task label's action space.
    - Improvement: We introduce masked projection during initialization to limit the action space and reduce the probability of predicted actions falling outside the action space.

2. **U-Net Architecture**:
    - Inherited: We adopt the base U-Net model from PDPP.

- Drawback: This architecture is too simple to capture the temporal logic of actions and lacks intermediate visual features for supervision.
- Improvement: We extend it by incorporating a latent space temporal logical interpolation module and a cross-attention module, enhancing temporal reasoning and feature integration.

3. **Loss Function**:

- Inherited: We retain PDPP's base MSE loss while removing the weights on both sides.
- Drawback: This loss is more likely to cause predicted actions to fall outside the current task's action space. Additionally, the real information provided by visual features gradually weakens - the farther from the endpoints, the worse the effect becomes, as shown in Figure 6d. Weighting only both ends imposes excessive attention on endpoints, causing over-reliance while neglecting the middle portions.
- Improvement: We introduce additional constraints, including gradient weighting and masking techniques, to enhance the model's performance and training stability.

4. **Diffusion Process**:

- Inherited: We retain the denoising diffusion process.
- Drawback: This diffusion process is time-consuming.
- Improvement: We replace DDPM with DDIM, which accelerates both training and inference while maintaining high-quality predictions.

5. **Task Classification**:

- Inherited: We maintain a task classifier component.
- Drawback: MLP classifier is too simple to model complex spatial-temporal relationships between actions for long instructional videos to obtain accurate task labels.
- Improvement: We enhance it by integrating a transformer module, improving the model's ability to understand and classify complex tasks through transformer's long-term temporal relationship modeling capabilities.

These improvements address PDPP's limitations in temporal reasoning, feature integration, and computational efficiency, resulting in a more powerful and versatile model for procedural planning in instructional videos.

Q2. In equation 1, why introduce M?

A5. Our MTID diffusion model takes as input a matrix containing action sequences with T timesteps and is based on U-Net, which contains **M residual temporal blocks** in the downsampling, upsampling, and middle layers for directly diffusing and generating T intermediate target actions. To ensure that each intermediate layer contains valid auxiliary information, our Latent Space Temporal Logical Interpolation Module needs to generate M intermediate auxiliary features. Subsequently, we apply cross-attention in residual temporal blocks across the M interpolated features and the entire input matrix rather than individual timesteps, enabling better temporal integration.

We also conducted experiments to demonstrate the effect of M. Our results showed that using interpolated features only for T steps led to suboptimal performance. This also supports our decision to use interpolated features across all M modules.

Table 2: Ablation study on M on CrossTask when T=3.

Method	**SR↑**	**mAcc↑**	**mIoU↑**
T	38.64	66.13	68.05
M	**40.45**	**67.19**	**69.17**

About more information, we have added the relevant experiments in Appendix D, section "More Explanation of M" at line 990.

Q3. Some mistakes in the paper.

A6. We have corrected the mistakes in the latest version of the paper.