Hierarchical Action Learning for Weakly-Supervised Action Segmentation

Supplementary Material

9. Extended Related Work

9.1. Weakly-Supervised Action Segmentation

Weakly-supervised action segmentation approaches [3, 7, 8, 23, 39, 40, 46, 54, 64, 66, 85] seek to reduce the dependence on detailed frame-level annotations. In the context of action segmentation, four primary forms of weak supervision have been studied: transcripts, action sets, timestamps, and textual descriptions (e.g., narrations). Among them, transcripts provide an action sequence without frame-level alignment, while action sets offer an unordered collection of actions appearing in the video. Timestamps indicate the occurrence of specific actions at particular time points.

A transcript provides a sequential list of actions occurring in a video, but does not include precise frame-level timing information. This type of supervision offers substantial advantages in terms of annotation efficiency, as it removes the requirement for detailed, frame-wise labeling. Approaches utilizing transcripts generally adopt either a single-stage design or an iterative two-step learning paradigm. **Iterative two-stage methods** first generate initial frame-level labels from the action transcript, then refine them through repeated updates. HTK [39] adapts a supervised HMM-GMM model [38] to weak supervision by uniformly initializing segments and refining them using transcripts. Richard et al. [40, 64] replace GMMs with RNNs and introduce latent sub-actions for finer motion modeling. ISBA [12] starts with uniform segmentation and gradually adjusts boundaries using soft labels. TASL [54] iteratively aligns video frames with transcripts to improve segmentation. **Single-stage methods** aim to overcome the initialization sensitivity of two-stage approaches. ECTC [23] aligns transcripts with video frames using temporal classification while enforcing consistency. NN-Viterbi [66] integrates visual, context, and length models, generating pseudo-labels via Viterbi decoding. D3TW [7] introduces a differentiable loss to distinguish between correct and incorrect transcripts. Building on this, CDFL [46] uses a segmentation graph to model valid/invalid alignments and compares their energies. Despite strong performance, NN-Viterbi and CDFL incur high computational costs. To address this, MuCon [72] reduces training time with a dual-branch design enforcing mutual consistency. DP-DTW [8] learns class-specific prototypes and improves action discrimination for temporal recognition.

Action sets specify only the presence of actions in a video, without indicating their order or frequency, making them a weaker supervision form than transcripts. Early work by Richard et al. [65] models temporal segmentation under action set supervision by integrating contextual, duration, and action models to infer the most probable action sequence for a given video, formulated as a maximum likelihood problem solvable using the Viterbi algorithm. Later methods, such as SCT [16], directly learn segmentation from action sets through region-wise predictions and frame-level consistency constraints. SCV [44] enhances pseudo-label generation via a constrained Viterbi approach and an n-pair loss. ACV [45] extends SCV by proposing a differentiable formulation that enables end-to-end training and eliminates the reliance on post-processing steps. POC [55] proposes a loss formulation that leverages the cross-video consistency of action pair ordering to enforce temporal alignment between the predicted segments and the extracted templates.

Timestamp supervision provides sparse frame annotations instead of dense labels, offering a cost-effective alternative to full supervision. Some methods [31, 48] assume one timestamp per action, while others [62] support arbitrary sampling. Typically, these approaches generate pseudo frame-wise labels that are refined iteratively. Li et al. [48] detect action transitions and apply a confidence loss to guide predictions around timestamps. EM-TSS [62] employs an EM framework to infer full sequences from sparse labels, offering greater flexibility in annotation and outperforming approaches constrained to one timestamp per action. GCN-TSS [31] introduces a GNN-based approach where video frames are modeled as graph nodes, with edges weighted by feature similarity. Labels from sparsely annotated frames are propagated across the graph to infer full segmentations, assuming all action instances are timestamped. In contrast, RAS-TSS [73] relaxes this assumption by allowing incomplete annotations and expands segment boundaries around timestamps to reduce ambiguity and better handle missing labels.

Narrations and subtitles, often available alongside instructional videos, offer a valuable but weak supervisory signal for temporal understanding. Prior work has leveraged such text for alignment [4, 57] or step localization [2, 88] and recent efforts in multi-modal learning [61, 70] explore joint embeddings across vision, language, and occasionally audio. However, a key limitation is the frequent misalignment between textual and visual content. To address this, Sener et al. [69] propose a hybrid generative model combining visual and textual vocabularies across videos for shared action discovery, while Fried et al. [17] assume a canonical transcript per activity class and use a semi-Markov model for structured alignment. More recent approaches [20] focus on aligning narrations to specific segments within in-

structional videos.

9.2. Causal Representation Learning

To recover latent variables with theoretical guarantees of identifiability [18, 53, 68, 80], Independent Component Analysis (ICA) has been widely employed for uncovering causal representations [50, 58, 63, 76]. Traditional ICA approaches typically rely on the assumption that observed variables are generated through a linear mixture of latent variables [10, 25, 43, 83]. However, identifying such a linear mixing function is nontrivial in practical, real-world settings. To establish identifiability in more general nonlinear ICA models, additional assumptions are introduced, such as sparsity in the data generation process and the incorporation of auxiliary variables [28, 30, 33, 49, 87].

In particular, the work of Aapo et al. provides the first theoretical foundation for identifiability in such settings [32]. When latent sources are assumed to follow distributions from the exponential family, identifiability can be achieved by integrating auxiliary information such as domain indices, temporal markers, or class labels [26, 27, 29, 32]. Moreover, recent studies [34, 35, 77, 79] demonstrate that the identifiability of individual components in nonlinear ICA can be achieved without relying on exponential family assumptions.

To enable identifiability in unsupervised scenarios, many approaches have introduced structural assumptions such as sparsity in the generative mechanisms [28, 30, 33, 49, 87]. For instance, Lachapelle et al. [41, 42] propose mechanism sparsity regularization as an inductive bias to isolate distinct causal latent factors. Zhang et al. [84] utilize sparse latent structures and establish identifiability under conditions of distributional shifts, without requiring explicit supervision. Besides, nonlinear ICA is extended to the time-series setting for learning identifiable representations from temporal data [19, 24, 26, 52, 79]. For example, Aapo et al. [26] propose a theoretical framework based on nonstationary variances across time segments to identify latent sources in temporally varying data. In the stationary case, researchers utilize permutation-based contrastive learning methods to disentangle latent components.

More recent advances incorporate independent noise and variability history information. Importantly, TDRL [82] and LEAP [81] exploit such features to facilitate identifiability. CHiLD [51], a recently proposed framework for hierarchical temporal causal representation learning, leverages temporal contextual observations and hierarchical sparsity to achieve identifiability of multi-layer latent dynamics. By introducing a nonstationary sparse transition assumption, CtrlNS [71] establishes identifiability from a theoretical perspective and empirically demonstrates its framework's effectiveness in uncovering latent factors and distribution shifts. Although CtrlNS can also be applied to the action segmentation task, its latent variables are limited to a single layer, which restricts its ability to capture the varying dynamics across different levels of latent factors.

10. Proof

10.1. Useful Theorems and Lemmas

Theorem 2. *(Theorem XV 4.5 in [14] Part III) A bounded operator T is a spectral operator if and only if it is the sum $T = S + N$ of a bounded scalar type operator S and a quasi-nilpotent operator N commuting with S. Furthermore, this decomposition is unique and T and S have the same spectrum and the same resolution of the identity.*

Lemma 2. *(Lemma 1 in [22]) Under Assumption A2, if $L_{z|x}$ is injective, then $L_{x|z}^{-1}$ exists and is densely defined over $\mathcal{G}(\mathcal{X})$ (for $\mathcal{G} = \mathcal{L}^1, \mathcal{L}^1_{\text{bnd}}$).*

10.2. Proof of Block-wise Identifiability of Latent Action and Visual Variables.

Lemma 3. ***(Block-wise Identification of** $(\mathbf{v}_t, \mathbf{c}_t)$**.)** *Suppose the observed, latent visual, and latent action variables follow the augmented data generation process in Figure 2(a). By matching the true joint distribution of 5 numbers of adjacent video frames, i.e., $\{\mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t+2}\}$, we further make the following assumptions:*

- *A1 (**Bounded and Continuous Density**): The joint distribution of $\mathbf{x}_t$,$\mathbf{v}_t$,$\mathbf{c}_t$, and their marginal and conditional densities are bounded and continuous.*
- *A2 (**Injective Linear Operators**): The linear operators $L_{\mathbf{x}_{t+1}|\mathbf{v}_t,\mathbf{c}_t}$ and $L_{\mathbf{x}_{t-1}|\mathbf{x}_{t+1}}$ are injective for bounded function space.*
- *A3 (**Positive Density**):For all $[\mathbf{v}_t, \mathbf{c}_t], [\mathbf{v}_t, \mathbf{c}_t]' \in \mathcal{V}_t \cup \mathcal{C}_t$ with $[\mathbf{v}_t, \mathbf{c}_t] \neq [\mathbf{v}_t, \mathbf{c}_t]'$ the set $\{\mathbf{x}_t : p(\mathbf{x}_t|\mathbf{v}_t, \mathbf{c}_t) \neq p(\mathbf{x}_t|\mathbf{v}'_t, \mathbf{c}'_t)\}$ has positive probability.*

Suppose that the learned $(\hat{g}, \hat{f}, p_{\hat{\epsilon}})$ to achieve Equation (1) - (3), then the latent variables $(\mathbf{v}_t, \mathbf{c}_t)$ are block-wise identifiable.

Proof. We first follow Hu et al [22] framework to prove that $(\mathbf{v}_t, \mathbf{c}_t)$ is block-wise identifiable given sufficient observation. Sequentially, we prove that we require at least 5 adjacent observed variables to achieve block-wise identifiability.

Given time series data with T timesteps $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T\}$, 2-layers of latent variables $V = \{\mathbf{v}_1, \cdots, \mathbf{v}_t, \cdots, \mathbf{v}_T\}$ and $C = \{\mathbf{c}_1, \cdots, \mathbf{c}_t, \cdots, \mathbf{c}_T\}$. To simplify the notation, we let $\mathbf{x}_{<t}$, $\mathbf{x}_{>t}$ and $\mathbf{z}_t$ be $\{\mathbf{x}_{t-2}, \mathbf{x}_{t-1}\}$, $\{\mathbf{x}_{t+1}, \mathbf{x}_{t+2}\}$ and $(\mathbf{v}_t, \mathbf{c}_t)$, respectively. Sequentially, according to the data generation process in Figure 3(a), we have:

$$\begin{aligned} P(\mathbf{x}_{<t}|\mathbf{x}_t, \mathbf{z}_t) &= P(\mathbf{x}_{<t}|\mathbf{z}_t), \\ P(\mathbf{x}_{>t}|\mathbf{x}_t, \mathbf{x}_{<t}, \mathbf{z}_t) &= P(\mathbf{x}_{>t}|\mathbf{z}_t). \end{aligned} \tag{12}$$

Sequentially, the observed $P(\mathbf{x}_{t-1})$ and joint distribution $P(\mathbf{x}_{>t}, \mathbf{x}_t, \mathbf{x}_{<t})$ directly indicates $P(\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t})$, and we have:

$$\begin{aligned}
&P(\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}) \\
&= \underbrace{\int_{\mathcal{Z}_t} P(\mathbf{x}_{>t}, \mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}) d\mathbf{z}_t}_{\text{Integration over } \mathcal{Z}_t} \\
&= \underbrace{\int_{\mathcal{Z}_t} P(\mathbf{x}_{>t}|\mathbf{x}_t, \mathbf{z}_t, \mathbf{x}_{<t}) P(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}) d\mathbf{z}_t}_{\text{Factorization of joint conditional probability}} \\
&= \underbrace{\int_{\mathcal{Z}_t} P(\mathbf{x}_{>t}|\mathbf{z}_t) P(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}) d\mathbf{z}_t}_{\text{Conditional Independence}} \\
&= \underbrace{\int_{\mathcal{Z}_t} P(\mathbf{x}_{>t}|\mathbf{z}_t) P(\mathbf{x}_t|\mathbf{z}_t, \mathbf{x}_{<t}) P(\mathbf{z}_t|\mathbf{x}_{<t}) d\mathbf{z}_t}_{\text{Bayes Law}} \\
&= \int_{\mathcal{Z}_t} P(\mathbf{x}_{>t}|\mathbf{z}_t) P(\mathbf{x}_t|\mathbf{z}_t) P(\mathbf{z}_t|\mathbf{x}_{<t}) d\mathbf{z}_t.
\end{aligned} \tag{13}$$

We further incorporate the integration over $\mathcal{X}_{<t}$ as follows:

$$\begin{aligned}
&\int_{\mathcal{X}_{<t}} P(\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}) P(\mathbf{x}_{<t}) d\mathbf{x}_{<t} \\
&= \int_{\mathcal{X}_{<t}} \int_{\mathcal{Z}_t} P(\mathbf{x}_{>t}|\mathbf{z}_t) P(\mathbf{x}_t|\mathbf{z}_t) P(\mathbf{z}_t|\mathbf{x}_{<t}) P(\mathbf{x}_{<t}) d\mathbf{z}_t d\mathbf{x}_{<t}.
\end{aligned} \tag{14}$$

According to the definition of linear operator, we have:

$$\begin{aligned}
\int_{\mathcal{X}_{<t}} P(\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}) P(\mathbf{x}_{<t}) d\mathbf{x}_{<t} &= [L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}} \circ P](\mathbf{x}_{<t}), \\
\int_{\mathcal{X}_{<t}} P(\mathbf{z}_t|\mathbf{x}_{<t}) P(\mathbf{x}_{<t}) d\mathbf{x}_{<t} &= [L_{\mathbf{z}_t|\mathbf{x}<t} \circ P](\mathbf{x}_{<t}) \\
\int_{\mathcal{Z}_t} P(\mathbf{x}_{>t}|\mathbf{z}_t) d\mathbf{z}_t &= L_{\mathbf{x}_{>t}|\mathbf{z}_t}.
\end{aligned} \tag{15}$$

By combining Equation (14) and (15), we have:

$$[L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}} \circ P](\mathbf{x}_{<t}) = [L_{\mathbf{x}_{>t}|\mathbf{z}_t} D_{\mathbf{x}_t|\mathbf{z}_t} L_{\mathbf{z}_t|\mathbf{x}<t} \circ P](\mathbf{x}_{<t}), \tag{16}$$

which implies the operator equivalence:

$$L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}} = L_{\mathbf{x}_{>t}|\mathbf{z}_t} D_{\mathbf{x}_t|\mathbf{z}_t} L_{\mathbf{z}_t|\mathbf{x}<t}. \tag{17}$$

Sequentially, we further integrate out $\mathbf{x}_t$ and have:

$$\int_{\mathcal{X}_t} L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}} d\mathbf{x}_t = \int_{\mathcal{X}_t} L_{\mathbf{x}_{>t}|\mathbf{z}_t} D_{\mathbf{x}_t|\mathbf{z}_t} L_{\mathbf{z}_t|\mathbf{x}<t} d\mathbf{x}_t, \tag{18}$$

and it results in:

$$L_{\mathbf{x}_{>t}|\mathbf{x}_{<t}} = L_{\mathbf{x}_{>t}|\mathbf{z}_t} L_{\mathbf{z}_t|\mathbf{x}<t}. \tag{19}$$

According to assumption A2, the linear operator $L_{\mathbf{x}_{>t}|\mathbf{z}_t}$ is injective, Equation (19) can be rewritten as:

$$L^{-1}_{\mathbf{x}>t|\mathbf{z}_t} L_{\mathbf{x}_{>t}|\mathbf{x}_{<t}} = L_{\mathbf{z}_t|\mathbf{x}<t}. \tag{20}$$

By combining Equation (17) and (20), we have

$$L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}<t} = L_{\mathbf{x}>t|\mathbf{z}_t} D_{\mathbf{x}_t|\mathbf{z}_t} L^{-1}_{\mathbf{x}>t|\mathbf{z}_t} L_{\mathbf{x}>t|\mathbf{x}<t}. \tag{21}$$

By leveraging Lemma 2, if $L_{\mathbf{x}_{<t}|\mathbf{x}_{>t}}$ is injective, then $L^{-1}_{\mathbf{x}_{>t}|\mathbf{x}_{<t}}$ exists. Therefore, we have:

$$L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}} L^{-1}_{\mathbf{x}>t|\mathbf{x}<t} = L_{\mathbf{x}>t|\mathbf{z}_t} D_{\mathbf{x}_t|\mathbf{z}_t} L^{-1}_{\mathbf{x}>t|\mathbf{z}_t}. \tag{22}$$

Then we can leverage assumption A3 and the linear operator is bounded. Consequently, $L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}} L^{-1}_{\mathbf{x}>t|\mathbf{x}<t}$ is also bounded, which satisfies the condition of Theorem 2, and hence the the operator $L_{\mathbf{x}>t|\mathbf{z}_t} D_{\mathbf{x}_t|\mathbf{z}_t} L^{-1}_{\mathbf{x}>t|\mathbf{z}_t}$ have a unique spectral decomposition, where $L_{\mathbf{x}>t|\mathbf{z}_t}$ and $D_{\mathbf{x}_t|\mathbf{z}_t}$ correspond to eigenfunctions and eigenvalues, respectively.

Since both the marginal and conditional distributions of the observed variables are matched, the true model and the estimated model yield the same distribution over the observed variables. Therefore, we also have:

$$L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}} L^{-1}_{\mathbf{x}>t|\mathbf{x}<t} = L_{\hat{\mathbf{x}}_{>t}, \hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{<t}} L^{-1}_{\hat{\mathbf{x}}>t|\hat{\mathbf{x}}<t}, \tag{23}$$

where the L.H.S corresponds to the true model and the R.H.S corresponds to the estimated model. Moreover, $L_{\hat{\mathbf{x}}_{>t}, \hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{<t}} L^{-1}_{\hat{\mathbf{x}}>t|\hat{\mathbf{x}}<t}$ also have the unique decomposition, so the L.H.S of the Equation (23) can be written as:

$$L_{\mathbf{x}_{>t}, \mathbf{x}_t|\mathbf{x}_{<t}} L^{-1}_{\mathbf{x}>t|\mathbf{x}<t} = L_{\hat{\mathbf{x}}_{>t}|\hat{\mathbf{z}}_t} D_{\hat{\mathbf{x}}_t|\hat{\mathbf{z}}_t} L^{-1}_{\hat{\mathbf{x}}_{>t}|\hat{\mathbf{z}}_t}, \tag{24}$$

Integrating Equation (22) and Equation (24), and noting that their L.H.S. are identical, it follows that they share the same spectral decomposition. This yields

$$L_{\mathbf{x}_{>t}|\mathbf{z}_t} = C L_{\hat{\mathbf{x}}_{>t}|\hat{\mathbf{z}}_t} P, \quad D_{\mathbf{x}_t|\mathbf{z}_t} = P^{-1} D_{\hat{\mathbf{x}}_t|\hat{\mathbf{z}}_t} P, \tag{25}$$

where C is a scalar accounting for scaling indeterminacy and P is a permutation on the order of elements in $D_{\hat{\mathbf{x}}_t|\hat{\mathbf{z}}_t}$, as discussed in [14]. These forms of indeterminacy are analogous to those in eigendecomposition, which can be viewed as a finite-dimensional special case. P is a mapping from distribution to distribution

Since the normalizing condition

$$\int_{\hat{\mathcal{X}}_{t+1}} p_{\hat{\mathbf{x}}_t|\hat{\mathbf{z}}_t} \, d\hat{\mathbf{x}}_t = 1 \tag{26}$$

must hold for every $\hat{\mathbf{z}}_t$, one only solution is to set $C = 1$.

Hence, $D_{\hat{\mathbf{x}}_t|\hat{\mathbf{z}}_t}$ and $D_{\mathbf{x}_t|\mathbf{z}_t}$ are identical up to a permutation on their repsective elements. We use unordered sets to express this equivalence:

$$\{p(\mathbf{x}_t \mid \mathbf{z}_t)\} = \{p(\mathbf{x}_t \mid \hat{\mathbf{z}}_t)\}, \quad \text{for all } \mathbf{z}_t, \hat{\mathbf{z}}_t. \tag{27}$$

Due to the set being unordered, the only way to match the R.H.S. with the L.H.S. in a consistent order is to exchange the conditioning variables, that is,

$$\begin{aligned} &\left\{p(\mathbf{x}_t|\mathbf{z}_t^{(1)}), p(\mathbf{x}_t|\mathbf{z}_t^{(2)}), \cdots\right\} \\ &= \left\{p(\mathbf{x}_t|\hat{\mathbf{z}}_t^{(\pi(1))}), p(\mathbf{x}_t|\hat{\mathbf{z}}_t^{(\pi(2))}), \cdots\right\}, \end{aligned} \tag{28}$$

where superscript $(\cdot)$ denotes the index of a conditioning variable, and π is reindexing the conditioning variables. We use a relabeling map h to represent its corresponding value mapping:

$$p(\mathbf{x}_t|\mathbf{z}_t) = p(\mathbf{x}_t|h(\hat{\mathbf{z}}_t)), \text{ for all } \mathbf{z}_t, \hat{\mathbf{z}}_t \tag{29}$$

Since $K_{\hat{\mathbf{z}}_t,\mathbf{z}_t}$, $L^{-1}_{\mathbf{x}_{>t}|\hat{\mathbf{z}}_t}$, and $L_{\mathbf{z}_t|\mathbf{x}_{>t}}$ are continuous, h is continuous and differentiable. Moreover, by leveraging assumption A3, different values of $\mathbf{z}$, i.e., $\mathbf{z}_t^{(1)}, \mathbf{z}_t^{(2)}$ imply $p(\mathbf{x}_t|\mathbf{z}_t^{(1)}) \neq p(\mathbf{x}_t|\mathbf{z}_t^{(2)})$. So we can construct a function $F : \mathcal{Z} \to p(\mathbf{x}_t|\mathbf{z}_t)$, and we have:

$$\mathbf{z}_t^{(1)} \neq \mathbf{z}_t^{(2)} \longrightarrow F(\mathbf{z}_t^{(1)}) \neq F(\mathbf{z}_t^{(2)}), \tag{30}$$

implying that F is injective. Moreover, by using Equation (29), we have $F(\mathbf{z}_t) = F(h(\hat{\mathbf{z}}_t))$, which implies $\mathbf{z}_t = h(\hat{\mathbf{z}}_t)$.

The aforementioned result leverage $\mathbf{x}_{<t}, \mathbf{x}_t$, and $\mathbf{x}_{>t}$ as three different measurement of $\mathbf{z}_t$, where $|\mathbf{x}_{<t}|\}|\mathbf{z}_t|$, $|\mathbf{x}_{>t}|\}|\mathbf{z}_t|$ and $|\mathbf{x}_t| < |\mathbf{z}_t|$. It may imply that when the $\mathbf{x}_t$ cannot provide enough information to recover $\mathbf{z}_t$, we can seek more information from $\mathbf{x}_{<t}$ and $\mathbf{x}_{>t}$.

Sequentially, we further prove that when the observed and 2-layer latent variables follow the data generation process in Equation (1) and (2), we require at least 5 adjacent observed variables, i.e., $\{\mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t+2}\}$ to make $(\mathbf{v}_t, \mathbf{c}_t)$ block-wise identifiable. We prove it by contradiction as follows.

Suppose we have 4 adjacent observations, which can be divided into two cases: 1) $\{\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t+2}\}$ and $\{\mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}\}$. In the first case, suppose the dimension of $\mathbf{x}_t$ and that of latent variables $\mathbf{z}_t = (\mathbf{v}_t, \mathbf{c}_t)$ are $2n$, the dimension of $\mathbf{x}_{t-1}$ is n and the dimension of $\mathbf{z}_t$ is $2n$, conflicting with the assumption that $L_{\mathbf{x}_{t+1},\mathbf{x}_{t+2}|\mathbf{z}_t}$ is injective. In the second case, the dimensions of $\mathbf{x}_{t-2}, \mathbf{x}_{t-1}$ and $\mathbf{x}_{t+1}$ are $2n$ and n, respectively, conflicting with the assumption that $L_{\mathbf{x}_{t-2},\mathbf{x}_{t-1}|\mathbf{x}_{t+1}}$ is injective. As a result, we require at least 5 adjacent observations, i.e., $\{\mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t+2}\}$ to make $(\mathbf{v}_t, \mathbf{c}_t)$ block-wise identifiable. In fact, as long as the total dimension of the observed variable $\mathbf{x}_{t-\tau}, \cdots, \mathbf{x}_{t-1}$ is greater than or equal to the total dimension of the latent variable $(\mathbf{v}_t, \mathbf{c}_t)$, the mapping from the latent variable to the observed variable is injective. Then the latent variable $(\mathbf{v}_t, \mathbf{c}_t)$ is identifiable. □

10.3. Proof of Block-wise Identifiability of Latent Action variables.

In this section, we present the proof for Theorem 1. We first give a general identifiability theory (i.e., Theorem 3) for the generating process in Figure 3(b) and then make the connection to the proof of Theorem 1.

Theorem 3. *The generating process in Figure is defined as follows:*

$$[\mathbf{v}_{t-1}, \mathbf{v}_{t+1}] = f^v(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon^v_{t-1}, \epsilon^v_{t+1}), \tag{31}$$

$$\mathbf{v}_{t-1} = f^v_{t-1}(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon^v_{t-1}), \tag{32}$$

$$\mathbf{v}_{t+1} = f^v_{t+1}(\mathbf{c}_t, \mathbf{v}_t, \epsilon^v_{t+1}), \tag{33}$$

where $\mathbf{c}_t \in \mathbb{R}^{n_c}$, $\epsilon^v_{t-1} \in \mathbb{R}^{n_\epsilon}$, and $\epsilon^v_{t+1} \in \mathbb{R}^{n_\epsilon}$. Both f^v_{t-1} and f^v_{t+1} are smooth and have non-singular Jacobian matrices almost anywhere, and f is invertible.

If $\hat{f}^v_{t-1}$ and $\hat{f}^v_{t+1}$ assume the generating process of the true model (f^v_{t-1}, f^v_{t+1}) and match the joint distribution $p_{\mathbf{v}_{t-1},\mathbf{v}_{t+1}}$, then $\mathbf{c}_t$ are block-wise identifiable.

Proof. To simplify the notation, we use f, f_{t-1}, f_{t+1}, ϵ_{t-1} and ϵ_{t+1} represent f^v, f^v_{t-1}, f^v_{t+1}, ϵ^v_{t-1} and ϵ^v_{t+1}, respectively.

For $(\mathbf{v}_{t-1}, \mathbf{v}_{t+1}) \sim p_{\mathbf{v}_{t-1},\mathbf{v}_{t+1}}$, because of the matched joint distribution, we have the following relations between the true variables $(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1})$ and the estimated ones $(\hat{\mathbf{c}}_t, \hat{\mathbf{v}}_{t-2}, \hat{\mathbf{v}}_t, \hat{\epsilon}_{t-1}, \hat{\epsilon}_{t+1})$:

$$\mathbf{v}_{t-1} = f_{t-1}(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}) = \hat{f}_{t-1}(\hat{\mathbf{c}}_t, \hat{\mathbf{v}}_{t-2}, \hat{\epsilon}_{t-1}), \tag{34}$$

$$\mathbf{v}_{t+1} = f_{t+1}(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}) = \hat{f}_{t+1}(\hat{\mathbf{c}}_t, \hat{\mathbf{v}}_t, \hat{\epsilon}_{t+1}), \tag{35}$$

$$\begin{aligned} &(\hat{\mathbf{c}}_t, \hat{\mathbf{v}}_{t-2}, \hat{\mathbf{v}}_t, \hat{\epsilon}_{t-1}, \hat{\epsilon}_{t+1}) \\ &= \hat{f}^{-1}(\mathbf{v}_{t-1}, \mathbf{v}_{t+1}) \\ &= \hat{f}^{-1}(f(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1})) \\ &= h(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1}), \end{aligned} \tag{36}$$

where $\hat{f}_{t-1}$, $\hat{f}_{t+1}$ are the estimated invertible transition function and $h := \hat{f}^{-1} \circ f$ denotes a smooth and invertible function that transforms the true variables $\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1}$ to the estimated ones $\hat{\mathbf{c}}_t, \hat{\mathbf{v}}_{t-2}, \hat{\mathbf{v}}_t, \hat{\epsilon}_{t-1}, \hat{\epsilon}_{t+1}$.

By combining Equation (36) into Equation (34) yields the following:

$$
\begin{aligned}
&f_{t-1}(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}) \\
&= \hat{f}_{t-1}(h_{\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1})).
\end{aligned} \tag{37}
$$

For $i \in \{1, \ldots, n_v\}$ and $j \in \{1, \ldots, n_\epsilon\}$, we take partial derivative of the i-th dimension of $\mathbf{v}_t$ on both sides of Equation (37) w.r.t. $\epsilon_{t+1,j}$ and have:

$$
\begin{aligned}
0 &= \frac{\partial f_{t-1,i}(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1})}{\partial \epsilon_{t+1,j}} \\
&= \frac{\partial \hat{f}_{t-1,i}(h_{\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1}))}{\partial \epsilon_{t+1,j}}.
\end{aligned} \tag{38}
$$

The equation equals zero because there is no $\epsilon_{t+1,j}$ in the left-hand side of the equation. Expanding the derivative on the right-hand side gives:

$$
\sum_{l \in \{1, \ldots, n_c + n_v + n_\epsilon\}} \frac{\partial \hat{f}_{t-1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), l}} \cdot \frac{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), l}}{\partial \epsilon_{t+1,j}} = 0. \tag{39}
$$

Since $\hat{f}_{t-1}$ is invertible, the determinant of $\mathbf{J}_{\hat{f}_{t-1}}$ does not equal to 0, meaning that for $n_c + n_v + n_\epsilon$ different values of $\hat{f}_{t-1,i}$, each vector $[\frac{\partial \hat{f}_{t-1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), 1}}, \ldots, \frac{\partial \hat{f}_{t-1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), n_c + n_v + n_\epsilon}}]$ are linearly independent. Therefore, the $(n_c + n_v + n_\epsilon) \times (n_c + n_v + n_\epsilon)$ linear system is invertible and has the unique solution as follows:

$$
\frac{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), l}}{\partial \epsilon_{t+1,j}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1}) = 0. \tag{40}
$$

According to Equation (40), for any $l \in \{1, \ldots, n_c + n_v + n_\epsilon\}$ and $j \in \{1, \ldots, n_\epsilon\}$, $h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), l}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1})$ does not depend on $\epsilon_{t+1,j}$. In other word, $(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1})$, does not depend on ϵ_{t+1}.

For $i \in \{1, \ldots, n_v\}$ and $k \in \{1, \ldots, n_v\}$, we take partial derivative of the i-th dimension of $\mathbf{v}_t$ on both sides of Equation (37) w.r.t. $\mathbf{v}_{t,k}$ and have:

$$
\begin{aligned}
0 &= \frac{\partial f_{t-1,i}(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1})}{\partial \mathbf{v}_{t,k}} \\
&= \frac{\partial \hat{f}_{t-1,i}(h_{\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1}))}{\partial \mathbf{v}_{t,k}} \\
&= \sum_{l \in \{1, \ldots, n_c + n_v + n_\epsilon\}} \frac{\partial \hat{f}_{t-1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), l}} \cdot \frac{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), l}}{\partial \mathbf{v}_{t,k}} = 0.
\end{aligned} \tag{41}
$$

That means that for $n_c + n_v + n_\epsilon$ different values of $\hat{f}_{t-1,i}$, each vector $[\frac{\partial \hat{f}_{t-1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), 1}}, \ldots, \frac{\partial \hat{f}_{t-1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1}), n_c + n_v + n_\epsilon}}]$ are linearly independent. Therefore, the $(n_c + n_v + n_\epsilon) \times (n_c + n_v + n_\epsilon)$ linear system is invertible. In other word, $(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1})$, does not depend on $\mathbf{v}_t$.

Similarly, by combining Equation (36) and (35), we have:

$$
\begin{aligned}
&f_{t+1}(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}) \\
&= \hat{f}_{t+1}(h_{\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1})).
\end{aligned} \tag{42}
$$

For $i \in \{1, \ldots, n_c\}$ and $j \in \{1, \ldots, n_\epsilon\}$, we take partial derivative of the i-th dimension of $\mathbf{c}_t$ on both sides of Equation (42) w.r.t. $\epsilon_{t-1,j}$ and have:

$$
\begin{aligned}
0 &= \frac{\partial f_{t+1,i}(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1})}{\partial \epsilon_{t-1,j}} \\
&= \frac{\partial \hat{f}_{t+1,i}(h_{\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1}))}{\partial \epsilon_{t-1,j}} \\
&= \sum_{k \in \{1, \ldots, n_c + n_v + n_\epsilon\}} \frac{\partial \hat{f}_{t+1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}), k}} \cdot \frac{\partial h_{(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}), k}}{\partial \epsilon_{t-1,j}}.
\end{aligned} \tag{43}
$$

Since $\hat{f}_{t+1}$ is invertible, for $n_c + n_v + n_\epsilon$ different values of $\hat{f}_{t+1,i}$, each vector $[\frac{\partial \hat{f}_{t+1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}), 1}}, \ldots, \frac{\partial \hat{f}_{t+1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}), n_c + n_v + n_\epsilon}}]$ are linearly independent. Therefore, the $(n_c + n_v + n_\epsilon) \times (n_c + n_v + n_\epsilon)$ linear system is invertible and has the unique solution as follows:

$$
\frac{\partial h_{(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}), k}}{\partial \epsilon_{t-1,j}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1}) = 0, \tag{44}
$$

meaning that $(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1})$ does not depend on ϵ_{t-1}.

For $i \in \{1, \ldots, n_c\}$ and $k \in \{1, \ldots, n_v\}$, we take partial derivative of the i-th dimension of $\mathbf{v}_t$ on both sides of Equation (42) w.r.t. $\mathbf{v}_{t-2,k}$ and have:

$$
\begin{aligned}
0 &= \frac{\partial f_{t+1,i}(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1})}{\partial \mathbf{v}_{t-2,k}} \\
&= \frac{\partial \hat{f}_{t+1,i}(h_{\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1}))}{\partial \mathbf{v}_{t-2,k}} \\
&= \sum_{l \in \{1, \ldots, n_c + n_v + n_\epsilon\}} \frac{\partial \hat{f}_{t+1,i}}{\partial h_{(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}), l}} \cdot \frac{\partial h_{(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1}), l}}{\partial \mathbf{v}_{t-2,k}},
\end{aligned} \tag{45}
$$

meaning that $(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1})$ does not depend on $\mathbf{v}_{t-2}$. According to Equation (36), we have $(\hat{\mathbf{c}}_t, \hat{\mathbf{v}}_{t-2}, \hat{\mathbf{v}}_t, \hat{\epsilon}_{t-1}, \hat{\epsilon}_{t+1}) = h_{\mathbf{c}}(\mathbf{c}_t, \mathbf{v}_{t-2}, \mathbf{v}_t, \epsilon_{t-1}, \epsilon_{t+1})$.

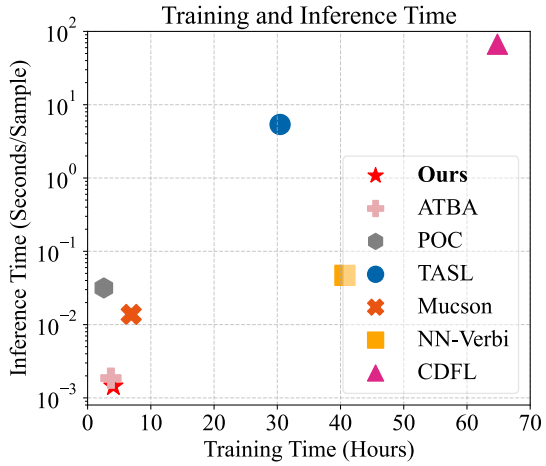


Figure 6. Comparison of training and inference time between the proposed framework and baseline methods. The framework universally outperforms baselines in terms of faster inference speed and shorter training time across all experimental settings.

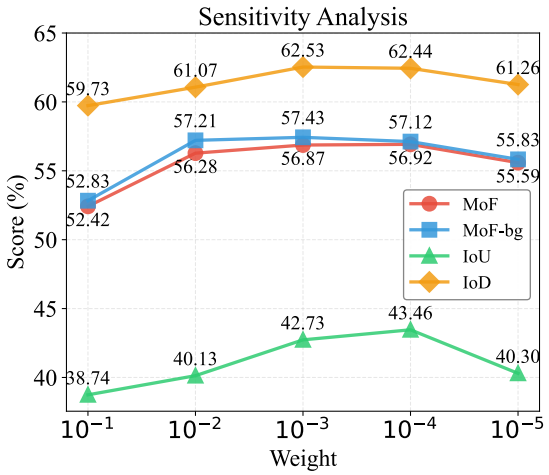


Figure 7. Sensitivity analysis of the hyperparameter for loss term $\mathcal{L}_s$ across datasets. Performance rises notably within a specific hyperparameter range and stabilizes thereafter, verifying the existence of a robust optimal hyperparameter range.

By using the fact that $(\mathbf{c}_t, \mathbf{v}_{t-2}, \epsilon_{t-1})$ does not depend on ϵ_{t+1} and $\mathbf{v}_t$, $(\mathbf{c}_t, \mathbf{v}_t, \epsilon_{t+1})$ does not depend on ϵ_{t-1} and $\mathbf{v}_{t-2}$, we have $\hat{\mathbf{c}}_t = h_{\mathbf{c}}(\mathbf{c}_t)$, i.e., the latent action variables are block-wise identifiable. □

Table 6. Architecture details. L:length of input feature, $|\mathbf{x}_t|$:the dimension of $\mathbf{x}_t$, $|y|$: the numbers of action class, ReLU:Rectified Linear Unit.

Configuration	Description	Output
$1.\phi$	Transformer	
Input $\mathbf{x}_{1:T}$ Dense	Input Feature 256 neurons	$BatchSize \times T \times \lvert\mathbf{x}_t\rvert$ $BatchSize \times T \times 256$
$2.\psi$	Visual Encoder	
Input $\mathbf{z}_{1:T}$ Dense	Feature Latent Variables 256 neurons; ReLU	$BatchSize \times T \times 256$ $BatchSize \times T \times 256$
$3.\eta$	Action Encoder	
Input $\mathbf{v}_{1:T}$ Dense	Visual Latent Variables 256 neurons; ReLU	$BatchSize \times T \times 256$ $BatchSize \times T \times 256$
$4.\kappa$	Visual Decoder	
Input $\mathbf{v}_{1:T}$ Dense	Visual Latent Variables 512 neurons; ReLU	$BatchSize \times T \times 256$ $BatchSize \times T \times 512$
$5.\xi$	Action Decoder	
Input $\mathbf{c}_{1:T}$ Dense	Visual Latent Variables 512 neurons; ReLU	$BatchSize \times T \times 256$ $BatchSize \times T \times 512$
$6.\Gamma$	Classifier	
Input $\mathbf{c}_{1:T}$ Dense	Action Latent Variables $\lvert y\rvert$ neurons; ReLU	$BatchSize \times T \times 256$ $BatchSize \times T \times \lvert y\rvert$

11. Implementation Details

11.1. Model Details

We choose ATBA[78] as the backbone and Transformer as the encoder and decoder, while the classifier is implemented using linear layers. Architecture details of the proposed method are shown in Table 6.

11.2. Experiment Details

We use ADAMW optimizer in all experiments and report the Mean-over-Frames(MoF), Mean-over-Frames without Background(MoF-bg), Intersection-over-Union(IoU) and Intersection-over-Detection(IoD) as evaluation metrics. The experiments are implemented by Pytorch on a single NVIDIA RTX 3090 24GB GPU.

11.3. Basic Training Settings

All model training adopts the AdamW optimizer with hyperparameter configurations set as 1e-4 for weight decay and 5e-4 for initial learning rate; the model is trained for 400/300/200/300 epochs for Breakfast [37], Hollywood [3], Crosstask [88], GTEA [15], respectively.

11.4. Computational Efficiency Analysis

This section compares the computational efficiency of the proposed framework against baseline methods, focusing on training and inference time. As shown in Fig. 6, the framework achieves faster inference speed and shorter training

Table 7. Experiment results on the GTEA dataset.

EXP	$\mathcal{L}_r$	$\mathcal{L}_s$	$\mathcal{L}_{KL}$	δ	MoF	IoU	IoD
1					53.3	40.9	58.7
2	✓				54.3	38.4	61.6
3		✓			54.6	40.3	61.6
4			✓		54.5	42.0	61.0
5				✓	53.9	41.1	59.4
6		✓	✓		54.7	39.3	61.4
7	✓		✓		55.2	42.4	60.7
8	✓	✓			55.4	41.0	60.9
9		✓	✓	✓	56.4	41.9	61.6
10	✓	✓		✓	55.5	40.8	60.9
11	✓	✓	✓		56.4	41.6	**63.3**
12	✓	✓	✓	✓	**56.6**	**42.6**	62.1

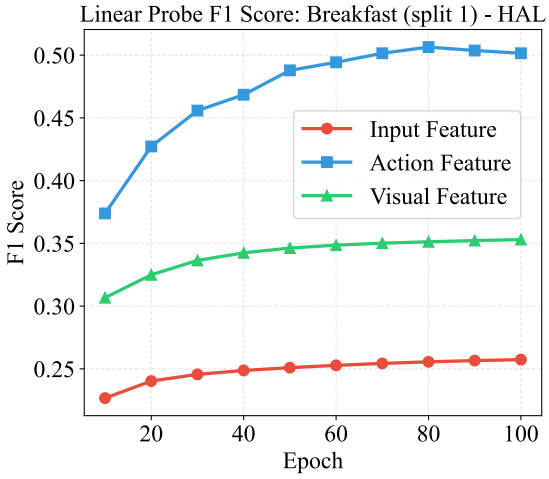


Figure 8. The F1 score with linear probing on Breakfast.

time compared to existing baselines across all experimental settings.

11.5. Hyperparameter Sensitivity Analysis

This section analyzes the sensitivity of the hyperparameter for loss term $\mathcal{L}_s$ across different datasets. As shown in Fig. 7, the performance of the proposed framework improves significantly as the hyperparameter value increases within a specific range, and then tends to stabilize with further increases in the hyperparameter value.

12. Additional Experimental Results

12.1. Ablation Study Extensions

The minor performance gaps across distinct loss function configurations reflect the inherent design principle of our framework: rather than relying on a single loss term to drive performance gains, the model leverages the complementary nature of multiple loss objectives to achieve incremental improvements. An analysis of the extended ablation study results (Table 7) reveals that no individual loss term or simple pairwise combination (e.g., $\mathcal{L}_r + \mathcal{L}_{KL}$) can match the performance of multi-loss combinations, which demonstrates that each loss term targets a unique aspect of the learning process—such as reconstruction fidelity or latent space regularization—and their integration addresses the limitations of single-objective optimization.

12.2. Linear Probing Results

Linear probing results (Fig. 8) demonstrate that action latent variables yield higher F1 scores than raw visual features, with this performance divergence rooted in the distinct semantic encoding characteristics of the two feature forms. Visual features tend to capture transient, task-irrelevant visual fluctuations, whereas action latent variables, regularized by the hierarchical causal generative process, encode stable, temporally consistent semantic information of action states. The superior linear probing performance of action latent variables validates that the learned representations capture action-specific semantic structures instead of superficial visual patterns, which confirms the effective disentanglement of hierarchical latent variables and the efficacy of temporal smoothness regularization on the hierarchical causal data generation process.

12.3. Empirical Motivation

This section empirically validates the core motivation of the proposed method, as shown in Fig. 9. Frame-to-frame visual feature distances exhibit large and frequent fluctuations, while ground-truth action boundaries typically occur after prolonged visual fluctuations instead of coinciding with such variations.

13. Discussion of the Identification Results

13.1. Bounded, Continues and Positive Density

The assumptions used in our theoretical analysis are also common in the field of identification, such as [22] and [36]. For example, the bounded and continuous density assumption requires that both observed and latent variables are continuous and lie within bounded ranges. In video data, such conditions are typically satisfied because pixel values and latent transitions change smoothly over time and usually remain within reasonable bounds, which ensures the corresponding probability densities are continuous and bounded. The positive density assumption is also achievable with sufficiently rich observational data. Moreover, injective linear operators imply that the information in the latent variables is equivalent to that contained in continuous multi-frame

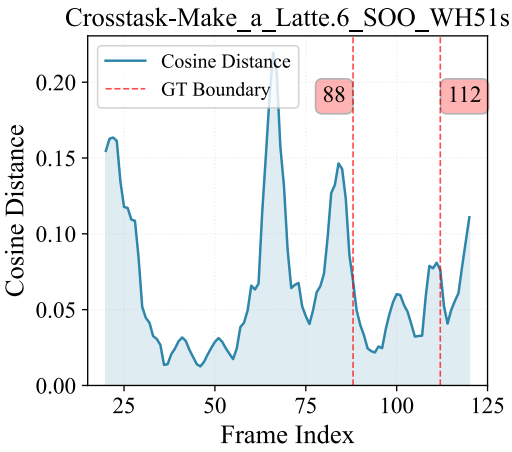


Figure 9. Frame-to-frame visual feature distance and ground-truth action boundary distribution. Visual features show large frequent fluctuations, and action boundaries arise after prolonged visual variations rather than coinciding with them.

video observations. Please refer to Appendix B for the detailed explanations of the assumptions, how they relate to real-world scenarios, and under which conditions they are satisfied.

13.2. Injective Linear Operators

A common practice in nonparametric identification involves leveraging the injectivity property of linear operators [6, 22]. At its core, this idea signifies that distinct input distributions fed into a linear operator will produce distinct output distributions from that same operator. To foster a clearer grasp of this premise, we present a series of illustrations depicting the distributional mapping $p_{\mathbf{a}} \to p_{\mathbf{b}}$, where $\mathbf{a}$ and $\mathbf{b}$ represent random variables.

Example 1 (Inverse Transformation). *$b = g(a)$, where g is an invertible function.*

Example 2 (Additive Transformation). *$b = a + \epsilon$, where $p(\epsilon)$ must not vanish everywhere after the Fourier transform (Theorem 2.1 in [59]).*

Example 3. *$b = g(a) + \epsilon$, where the same conditions from Examples 1 and 2 are required.*

Example 4 (Post-linear Transformation). *$b = g_1(g_2(a) + \epsilon)$, a post-nonlinear model with invertible nonlinear functions g_1, g_2, combining the assumptions in **Examples 1-3**.*

Example 5 (Nonlinear Transformation with Exponential Family). *$b = g(a, \epsilon)$, where the joint distribution $p(a, b)$ follows an exponential family.*

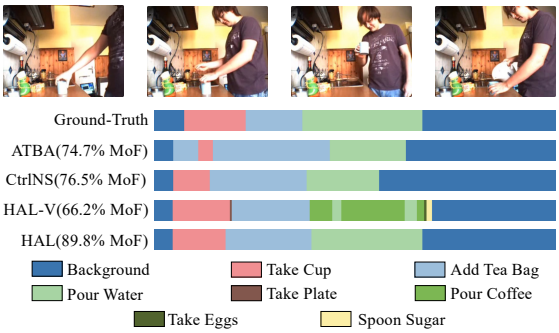
Example 6 (General Nonlinear Transformation). *$b = g(a, \epsilon)$, a general nonlinear formulation. Certain deviations from the nonlinear additive model (**Example 3**), e.g., polynomial perturbations, can still be tractable.*

13.3. Non-singular Jacobia

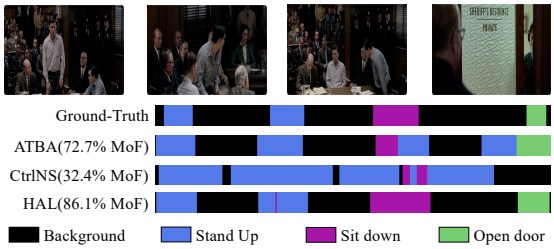
This assumption also finds frequent application in the work of [36]. From a mathematical perspective, it indicates that the Jacobian matrix mapping latent variables to observed variables is of full rank. In practical situations, this translates to each latent variable having a corresponding observation. To meet this requirement, those independent latent variables that exert no impact on observations can be readily excluded.

14. More Qualitative Results

We provide more qualitative results on Breakfast [37], Hollywood [3] and GTEA [15] datasets in figure 10.

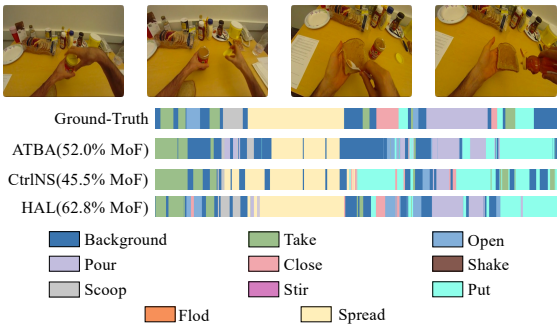


(a) *P07-stereo01-P07-tea* on Breakfast.

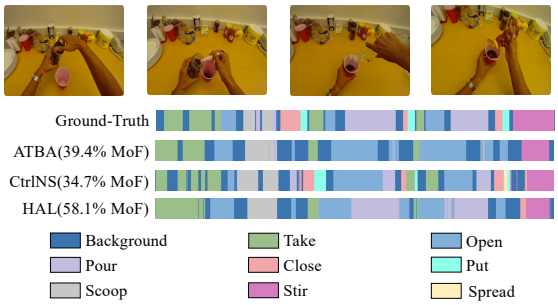


(b) *0146* on Hollywood.

(c) *0782* on Hollywood.



(d) *S1-Peanut-C1* on GTEA.



(e) *S4-CofHoney-C1* on GTEA.

Figure 10. More qualitative results.