Computer Vision and Image Understanding 270 (2026) 104842

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Vision-based mistake analysis in procedural activities: A review of advances and challenges

Konstantinos Bacharidis *, Antonis Argyros

Institute of Computer Science, Foundation for Research and Technology-Hellas, Greece

ARTICLE INFO

Communicated by Hazel Doughty

Keywords:
Video-based mistake analysis
Procedural activity understanding
Error detection

ABSTRACT

Mistake analysis in procedural activities is a critical area of research with applications spanning industrial automation, physical rehabilitation, education and human–robot collaboration. This paper reviews vision-based methods for detecting and predicting mistakes in structured tasks, focusing on procedural and executional errors. By leveraging advancements in computer vision, including action recognition, anticipation and activity understanding, vision-based systems can identify deviations in task execution, such as incorrect sequencing, use of improper techniques, or timing errors. We provide a comprehensive overview of existing datasets, evaluation metrics and state-of-the-art methods, categorizing approaches based on their use of procedural structure, supervision levels and learning strategies. Open challenges, such as distinguishing permissible variations from true mistakes and modeling error propagation are discussed alongside future directions, including neuro-symbolic reasoning and counterfactual state modeling. This work aims to establish a unified perspective on vision-based mistake analysis in procedural activities, highlighting its potential across diverse domains and aspects.

1. Introduction

Making mistakes is intrinsic to human nature. While they can act as critical drivers for learning and skill refinement (Reason, 1990), they may also lead to costly or dangerous consequences. The ability to detect, anticipate, and respond to mistakes, ideally before they escalate into failures, is essential in contexts where safety, efficiency, or performance are paramount. In such scenarios, real-time mistake analysis can support timely interventions and corrective actions, ultimately improving both human and system-level outcomes.

Mistake analysis can take on multiple, complementary forms. At times, the challenge lies in recognizing that a mistake has occurred (*mistake recognition*); in others, it is about identifying precisely when it happens during the task (*mistake detection*); and in critical situations, the key lies in anticipating mistakes early enough to prevent them altogether (*early mistake recognition*). These perspectives naturally arise from the way humans reason about everyday errors and offer an intuitive framework for understanding how intelligent systems might tackle similar challenges.

The impact of robust mistake analysis methods spans multiple domains. In industrial automation, they can enhance productivity by minimizing human error, ensuring proper task execution, and reducing the risk of injuries (see Fig. 1). In human–robot collaboration, anticipating potential mistakes facilitates smoother cooperation, reduces failure rates, and improves task handover efficiency. Similarly, in rehabilitation, education, or assisted living, such systems can guide users through complex tasks, offering feedback and corrections that enhance skill acquisition or daily independence. In embodied robotics, mistake analysis plays a dual role: supporting human–robot interaction (Nikolaidis et al., 2017) and enabling robots to recognize and recover from abnormal or unexpected situations (Garrabé et al., 2024), a critical step toward robust autonomy.

A unifying aspect across these diverse scenarios and domains is that they all involve procedural activities, i.e. structured tasks composed of ordered, goal-driven action sequences. Whether in manufacturing, cooking, or caregiving, such procedures typically follow established protocols and depend on the correct execution of each step. This structure not only makes deviations easier to define and detect, but also means that even small mistakes, such as step skipping, executing actions in the wrong order, or using an incorrect tool, can significantly affect outcomes. Consequently, procedural activities provide a natural and impactful setting for mistake analysis: their regularity enables modeling of expected behaviors, while their error sensitivity underscores the value of timely detection and intervention. Understanding where and how procedures go off track supports more effective training, safer automation, and helps users maintain task success in real-world environments.

* Corresponding author.
E-mail address: kbach@ics.forth.gr (K. Bacharidis).

https://doi.org/10.1016/j.cviu.2026.104842
Received 15 December 2025; Received in revised form 9 May 2026; Accepted 5 June 2026
Available online 9 June 2026

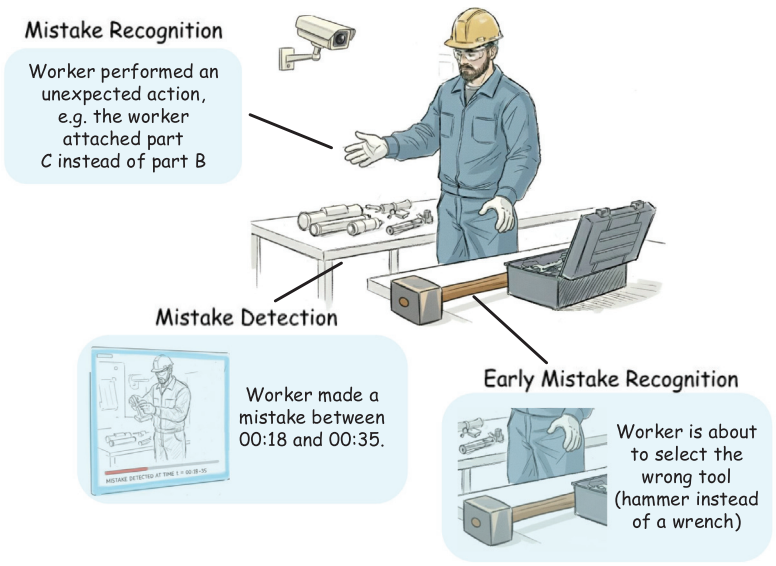

Fig. 1. Illustration of mistake analysis in an industrial assembly task: (1) recognize an unexpected or omitted action given an activity protocol (*mistake recognition*), e.g., the worker "attached part C" instead of "grabbing part B", (2) temporally detect an error within the execution of an action (*mistake detection*) and (3) predict a mistake before it fully occurs (*early mistake recognition*), e.g., warn worker about selecting the wrong tool.

Recent perspectives on egocentric vision highlight the growing importance of video-based systems for understanding and supporting human activities in real-world environments (Plizzari et al., 2024; Yang et al., 2025). Computer vision has emerged as a powerful modality for error detection in procedural tasks, offering a non-invasive, cost-effective, and scalable alternative to physical sensors. Advances in action and multimodal understanding (Stergiou and Poppe, 2025) enable the detection of subtle deviations, such as incorrect motion trajectories, missing steps, or tool misuse, by grounding visual observations within the specific constraints of a procedural protocol.

Despite recent progress, mistake analysis in procedural activities remains highly challenging (Aggarwal and Ryoo, 2011; Herath et al., 2017). A major difficulty lies in the variability of task execution: even well-defined procedures often allow multiple valid action sequences rather than a single canonical order. This complicates the distinction between genuine errors and acceptable alternatives. Further ambiguity arises at the action level, where steps vary in motion dynamics, timing, or appearance, making the boundary between atypical yet correct and truly erroneous behavior non-trivial. Additional challenges stem from viewpoint changes (e.g., ego- vs. exocentric), object multifunctionality, and activity compositionality, which demand fine-grained spatiotemporal reasoning. Crucially, mistake detection requires judging whether an action is appropriate in context, making it a higher-order reasoning task beyond static recognition.

The increasing significance of this problem has driven recent progress in vision-based methods and benchmark datasets for mistake understanding. Yet, a unified survey remains lacking. To the best of our knowledge, this work is the first systematic review of video-based mistake analysis in procedural activities. Beyond consolidating datasets, error taxonomies, evaluation protocols, and state-of-the-art methods, we also introduce a unified conceptual framework grounded in the notion of a *procedural protocol* and its role as the underlying structure connecting procedural activity modeling and mistake analysis. This perspective enables a consistent categorization of existing approaches and highlights key challenges and open questions in the field.

2. Mistake analysis: Problem statement

As mentioned in Section 1, video-based mistake analysis in procedural activities relies on understanding both the temporal structure of actions and the constraints governing their valid execution. In this section, we first formalize the notion of procedural activities and their associated execution protocols, and then define mistake analysis tasks relative to this formulation.

2.1. Procedural activities and protocols

Procedural activities consist of temporally ordered sequences of actions performed to achieve a specific goal. Their correct execution is governed by a *procedural protocol*, which defines the set of admissible actions and the constraints on how they can be performed and ordered.

A *procedural protocol* $\mathcal{P}_y$ associated with an activity y is defined as a structured specification of valid action sequences and execution constraints. A general abstraction of $\mathcal{P}_y$ is a directed graph:

$$\mathcal{P}_y = \mathcal{G}_y = (\mathcal{V}_y, \mathcal{E}_y),$$

where $\mathcal{V}_y$ denotes the set of admissible actions (or states), and $\mathcal{E}_y \subseteq \mathcal{V}_y \times \mathcal{V}_y$ encodes valid transitions between them. This formulation is agnostic to the underlying modality and can capture procedural knowledge derived from various sources, including symbolic descriptions, textual instructions, or visual observations.

More generally, $\mathcal{P}_y$ induces a set of valid action sequences:

$$\mathcal{L}(\mathcal{P}_y) = \left\{ (a_1, \ldots, a_N) \mid (a_i, a_{i+1}) \in \mathcal{E}_y \right\},$$

which defines the space of protocol-compliant executions.

At the lowest level, the protocol may be instantiated without explicit relational structure, considering only the set of admissible actions $\mathcal{V}_y$ without modeling transitions. This corresponds to structure-free formulations, where procedural reasoning is performed without an explicit model of action dependencies. A more structured variant introduces local transition constraints, where each action is associated with a restricted set of admissible successors, enabling incremental modeling of procedural progression without requiring a global view of the full transition structure.

At the most general level, the protocol is represented as a directed graph $\mathcal{G}_y = (\mathcal{V}_y, \mathcal{E}_y)$, capturing the full set of valid transitions. A common special case is the directed acyclic graph, which models procedures with fixed or partially ordered steps. However, real-world procedures often involve repetitions, optional actions, or recovery behaviors, which introduce cycles. As such, $\mathcal{P}_y$ is not restricted to acyclic structures, and different approaches instantiate or relax these structural assumptions in various ways (see Section 7).

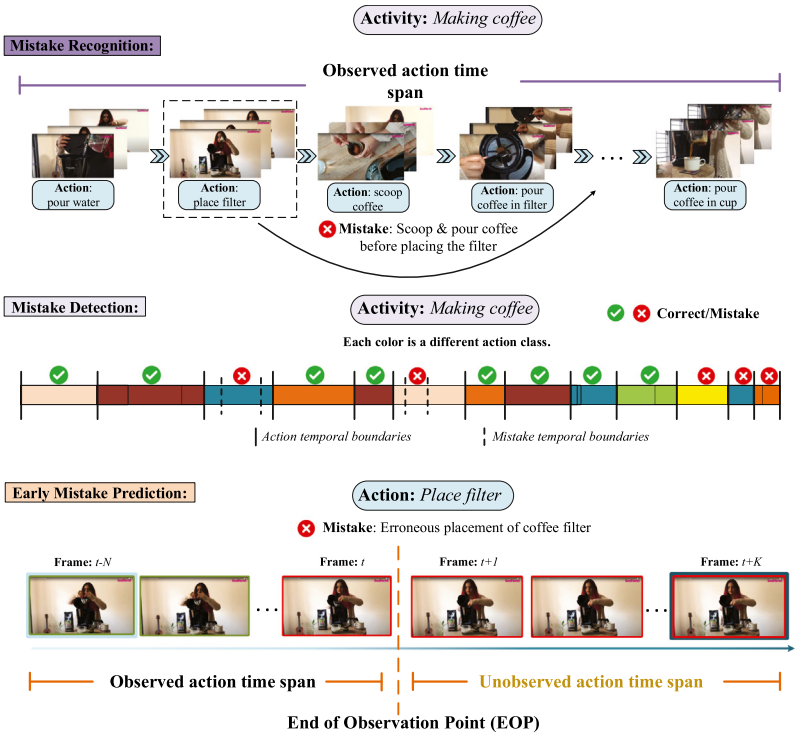

Fig. 2. Illustration of the distinction between mistake recognition (MR), early mistake recognition (EMR), and mistake detection (MD) in a filter-coffee example. MR identifies an error only after the action is completed (e.g., omitting coffee grounds). EMR anticipates the error during the ongoing action (e.g., detecting the intent before the filter is added). MD temporally localizes the exact segment where the error occurs, with action and error intervals potentially overlapping.

Video-based instantiation: In the context of video understanding, the procedural protocol may either be explicitly provided (e.g., as a predefined sequence, graph, or textual instruction) or implicitly inferred from data.

Given an untrimmed video of length T frames, $\mathbf{X} = \mathbf{x}_{1:T} = \langle x_1, \dots, x_T \rangle$, capturing the execution of an activity, a temporal grounding process produces a sequence of clips $V = \{v_1, v_2, \dots, v_N\}$. Each clip v_n is associated with an action label $a_n \in \mathcal{A}$, and an activity recognition step assigns a high-level activity label $y \in \mathcal{Y}$. The resulting structured representation is:

$$S = \{(v_n, a_n, y)\}_{n=1}^{N},$$

which captures the observed execution of the procedure. Under this formulation, the procedural protocol $\mathcal{P}_y$ defines the space of valid executions, and the observed sequence S corresponds to a particular instantiation whose compliance can be assessed with respect to $\mathcal{L}(\mathcal{P}_y)$.

The process of the clip-centered temporal action grounding may be instantiated either as *temporal action detection* (B. Wang et al., 2023), which identifies sparse temporal intervals corresponding to action occurrences, or as *temporal action segmentation (TAS)* (Ding et al., 2024), which assigns an action label to each temporal unit of the video, resulting in a dense temporal decomposition of the execution. Both paradigms serve as alternative mechanisms for constructing a temporally grounded representation of procedural activities.

2.2. Mistake analysis: Definition and tasks

Building on the notion of procedural protocols, mistakes can be defined as deviations from the constraints imposed by $\mathcal{P}_y$. Given the structured sequence S, with associated action sequence $a_{1:N}$, the goal of mistake analysis is to identify and characterize such deviations.

In its entirety, this process naturally follows a two-stage formulation, where a temporal grounding method, $g(\cdot)$, first maps the input video to a structured representation S, followed by a protocol-conditioned reasoning mechanism, $f(\cdot)$, that reason about the presence of a deviation, M:

$$\mathbf{X} \xrightarrow{g} S \xrightarrow{f} M,$$

This formulation highlights that temporal grounding serves as a perception-level module, while mistake analysis operates at a higher semantic level by evaluating whether the grounded actions conform to $\mathcal{P}_y$.

A deviation, referred to as a *mistake* or *error*, occurs when the observed sequence violates the protocol constraints. We define a per-segment violation indicator:

$$m_n = \mathbb{I}\big[(a_{n-1}, a_n) \notin \mathcal{E}_y \;\vee\; a_n \notin \mathcal{V}_y \;\vee\; \phi(v_n, a_n, y) = 0\big].$$

where $\phi(\cdot)$ evaluates whether the execution of action a_n satisfies expected constraints (e.g., motion, posture, object interaction). These violations can be broadly grouped into two categories: **procedural mistakes**, corresponding to violations of the protocol structure (i.e., $(a_{n-1}, a_n) \notin \mathcal{E}_y$ or $a_n \notin \mathcal{V}_y$), and **executional mistakes**, corresponding to failures in satisfying execution constraints (i.e., $\phi(v_n, a_n, y) = 0$). We expand upon this taxonomy in Section 3.

As depicted in Fig. 2, mistake analysis can take the form of: (a) **mistake recognition**, (b) **mistake detection**, and (c) **early mistake recognition**. Among these, similar to action understanding (Stergiou and Poppe, 2025), mistake recognition constitutes the foundational task, as it provides the basis upon which detection and early prediction are built.

Mistake recognition: The task of mistake recognition is to learn a mapping:

$$f : (S, \mathcal{P}_y) \to M,$$

where $M = \{m_n\}_{n=1}^{N}$ is a binary sequence indicating the presence of mistakes.

A more informative formulation additionally predicts mistake types and subtypes:

$$f : (S, \mathcal{P}_y) \to (M, C, E),$$

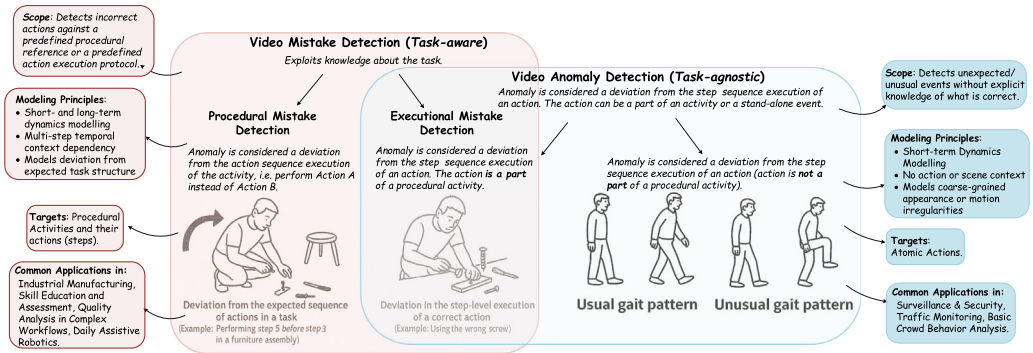

Fig. 3. Illustration of the association between video anomaly and mistake detection in procedural activities. Human sketches in the figure generated with OpenAI (2025).

enabling fine-grained and explainable feedback.

Mistake detection: Mistake detection extends recognition by jointly localizing and classifying erroneous segments:

$$f : (S, \mathcal{P}_y) \mapsto L, \quad L = \{([t_{\text{start}}^{(j)}, t_{\text{end}}^{(j)}], c_j)\}_{j=1}^{m}.$$

Unlike recognition, detection does not assume pre-segmented clips and instead identifies mistake boundaries directly.

Early mistake recognition: Early recognition addresses the anticipatory setting, where mistakes must be predicted before full observation. Given a partially observed segment $v^{\text{partial}} = v_{1:t}$, together with the sequence of previously observed and fully grounded clips $S_{1:n-1}$, the goal is to anticipate future deviations with respect to the procedural protocol $\mathcal{P}_y$:

$$f : (v^{\text{partial}}, S_{1:n-1}, \mathcal{P}_y) \to (M, C, E).$$

This formulation reflects that predictions must be made under incomplete information, requiring the model to jointly reason over the fully observed past $S_{1:n-1}$, the partially observed current segment v^{partial}, and the expected evolution of the procedure as constrained by $\mathcal{P}_y$. Such anticipatory reasoning enables proactive intervention before errors fully materialize.

A graphical illustration of the three subproblems is provided in Fig. 2.

2.3. Distinguishing mistake and video anomaly detection

Within the broader scope of mistake analysis, which encompasses mistake recognition, detection, and early prediction, we concentrate here on the task of mistake detection, as it can be viewed as a more generalized form of recognition, capturing not only whether a mistake occurs but also when and where it occurs within the activity. This spatio-temporal specificity makes detection a natural point of comparison to video anomaly detection (VAD), which similarly aims to localize unexpected events in continuous video streams.

In principle, mistake detection is related to VAD (Nayak et al., 2021; Abdalla et al., 2024), however, they differ fundamentally in the contextual grounding of what constitutes an error. Mistake detection requires task-specific procedural knowledge to distinguish permissible variations from true deviations (Fig. 3), whereas VAD typically identifies departures from normal appearance or motion patterns without understanding task goals. Such anomalies exhibit distinct visual or semantic properties, making them recognizable as unexpected events. Procedural mistake detection differs markedly since mistake in procedural activities, unlike generic anomalies, are defined relative to a structured sequence of actions required to complete a task and involve missing, adding, modifying, or incorrectly executing steps, i.e. errors that depend on long-term temporal and goal-conditioned reasoning rather than local visual irregularities. By contrast, VAD flags unusual events in a goal-agnostic and unstructured manner based solely on appearance or motion cues.

Both notions intersect in executional errors, where the correct action is visually plausible but performed with incorrect technique or reduced quality. This aligns with VAD because: (a) deviations are analyzed at the atomic action level, (b) errors appear as local visual irregularities (e.g., shaky motion), and (c) both rely on modeling normal behavior to detect subtle deviations that do not change the global sequence but degrade execution quality.

Finally, while most VAD methods rely on static or short-term temporal cues, mistake detection requires long-range reasoning to determine whether an action sequence follows a prescribed procedure. This difficulty is amplified in egocentric video (Haneji et al., 2024; Peddi et al., 2023; Sener et al., 2022), where viewpoint shifts, occlusions, and object-scale changes further challenge conventional VAD approaches.

3. A taxonomy of mistakes

A fundamental step in mistake analysis for procedural activities is establishing an error-type taxonomy. As stated in Section 2.2, broadly mistakes fall into two *coarse-grained* categories: **procedural mistakes**, arising from deviations in the sequence relative to the activity protocol, and **executional mistakes**, occurring when an action is performed incorrectly in terms of motion quality, temporal fidelity, or object manipulation. Each category can be further decomposed into *fine-grained* subcategories.

This organization naturally extends the binary problem of mistake recognition into a multi-class classification setting, where each detected mistake is assigned a coarse-grained label $c_i \in C$ (procedural or executional) and a fine-grained subtype $e_i \in \mathcal{E}_{c_i}$. This layered formulation, referred to as *error category recognition*, enables more detailed analyses of deviations, providing actionable insights into the causes and nature of mistakes in structured activities. Due to the emerging nature of the problem, however, existing works adopt varied definitions of mistake/error types, particularly for executional mistakes. For instance, Peddi et al. (2023), define the categories {*Order Error, Omission Error, Technique Error, Timing Error, Temperature Error*}, with the first two being procedural and the remainder executional. In contrast, Lee et al. (2024) propose a different but partially overlapping scheme, distinguishing *Step Omission*, *Step Addition* and *Step Correction* as procedural, while identifying *Step Modification* and *Step Slip* as executional. Although both taxonomies include omissions and technique-based deviations, their definitions differ in granularity and interpretation.

Following the distinctions introduced in prior work (Ding et al., 2023; Peddi et al., 2023), and grounded in the mistake formulation presented in Section 2, in this survey we adopt the taxonomy illustrated in Fig. 4. Under this taxonomy, *procedural mistakes* can be further divided into the following cases: (a) **Omitted actions:** an expected step

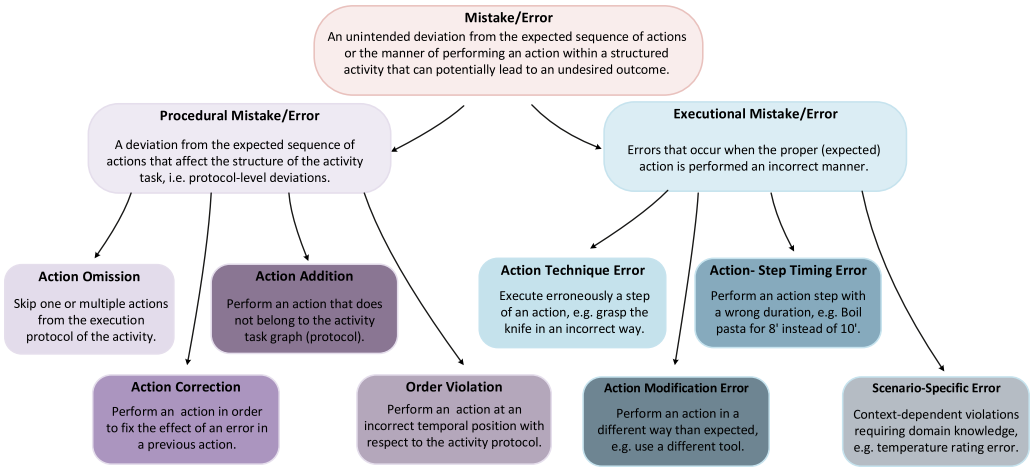

Fig. 4. Hierarchical taxonomy of error types. The taxonomy distinguishes between *procedural* and *executional* errors. Procedural errors include omission, unnecessary (insertion), corrective actions, and order violations. Executional errors arise from incorrect action realization, such as improper technique, timing, or object interaction. Scenario-specific errors are included to account for domain-dependent cases.

is skipped, manifesting as a sequence-level deviation from the protocol, i.e. there exists a valid sequence $\tilde{a}_{1:M} \in \mathcal{L}(\mathcal{P}_y)$ such that $\exists i$ s.t. $\tilde{a}_i \notin \{a_1, \dots, a_N\}$, which may induce a transition violation $(a_{n-1}, a_n) \notin \mathcal{E}_y$ when intermediate steps are not admissible; (b) **Unnecessary actions:** an extra step outside the protocol is performed, typically corresponding to $a_n \notin \mathcal{V}_y$; (c) **Corrective actions:** an unexpected step is introduced to compensate for a prior mistake, often resulting in transitions not defined by the protocol, i.e., $(a_{n-1}, a_n) \notin \mathcal{E}_y \quad \wedge \quad \exists k < n$ s.t. $m_k = 1$, (d) **Order violations:** an action is executed at an incorrect temporal position, directly violating the transition constraints, i.e., $(a_{n-1}, a_n) \notin \mathcal{E}_y$.

In contrast, the categorization of *executional mistakes* is inherently more nuanced and task-dependent, as it depends on the characteristics of the actions and constraints of the operational domain. These deviations are captured by violations of the execution consistency function, $\phi(v_n, a_n, y)$:

$$\phi_k(v_n, a_n, y) = \mathbb{I}\big[\psi_k(v_n, a_n, y) \in \Omega_k(a_n, y)\big],$$

where $\psi_k(\cdot)$ extracts action-relevant features (pose, timing, objects, etc.) and $\Omega_k(a_n, y)$ denotes the set of admissible feature set configurations for action a_n under activity y along dimension k.

These deviations of the executional consistency of an action, a_n, can be broadly grouped into the following categories:

Action technique errors: refer to discrepancies in how an action is performed. The procedural structure is preserved, actions occur in the correct order with the appropriate objects, but execution deviates from the expected technique, e.g., improper grip, incorrect alignment.

Temporal execution errors: capture deviations related to the *timing or rhythm* of task execution. These are premature or delayed step initiation, violations of step duration, or incorrect pacing that compromises task performance. For instance, removing an object from a heat source too soon may yield an incorrect result even if all other steps are correct.

Action modification errors: occur when an action is performed via an alternative approach deviating from the predefined protocol, such as using a different tool, skipping a sub-step, or combining multiple steps. Even if the intended outcome is achieved, the deviation introduces inconsistencies or risks relative to the standard procedure.

Scenario-specific errors: are context-dependent, tied to the semantics and requirements of actions within an activity, occurring when execution violates domain-specific constraints not captured by categories like technique or timing. E.g. in surgery, placing a suture in the wrong anatomical region is an error, even if technique and timing are correct.

4. Mistake analysis datasets

A foundational step in mistake analysis is the robust understanding of procedural activities, a task supported by a diverse range of video benchmarks. Several large-scale datasets focus primarily on correct task execution, such as IKEA ASM (Ben-Shabat et al., 2021), ATTACH (Aganian et al., 2023), HA4M (Cicirelli et al., 2022), and MECCANO (Ragusa et al., 2023). These datasets provide valuable resources for modeling complex human–object interactions in manufacturing and assembly settings. Although they do not explicitly target mistake analysis in their current form, they serve as important precursors for procedural understanding, and could become highly relevant for mistake-centric research through future extensions or synthetic augmentation strategies.

In this survey, we therefore concentrate on the subset of datasets introduced between 2018 and 2025 that explicitly include mistake annotations, highlighting the relatively recent emergence of this research direction. Fig. 5 illustrates their temporal distribution and relative scale.

4.1. Datasets: overview & key attributes

Table 1 provides an overview of the datasets. Only tasks relevant to procedural action and activity understanding are included; tasks a dataset may support but not directly related to mistake analysis are omitted.

Epic-Tent (Jang et al., 2019): was the first dataset to offer annotated task-level errors in a real-world procedural activity setting. It comprises over 5.4 h of ego-centric video recorded from 29 participants, each equipped with two head-mounted cameras, tasked with assembling a tent in an outdoor setting. It includes rich multi-modal annotations, such as frame-level action and error labels, 2D gaze positions, and self-reported uncertainty levels, enabling a range of research applications in egocentric action and activity recognition, human-object interaction, and mistake detection. It supports coarse-to-fine procedural understanding, with the tent assembly task decomposed into 12 sub-activities, each comprising of low-level steps from a total of 38 actions.

For mistake analysis, Epic-Tent provides coarse, segment-level annotations indicating whether the overall task or its sub-components were completed correctly or incorrectly, without distinguishing between procedural (e.g., skipping a step, incorrect order) and executional (e.g., misplacing a pole, using incorrect force) errors, though both may occur. Unlike other datasets targeting mistake analysis, Epic-Tent did

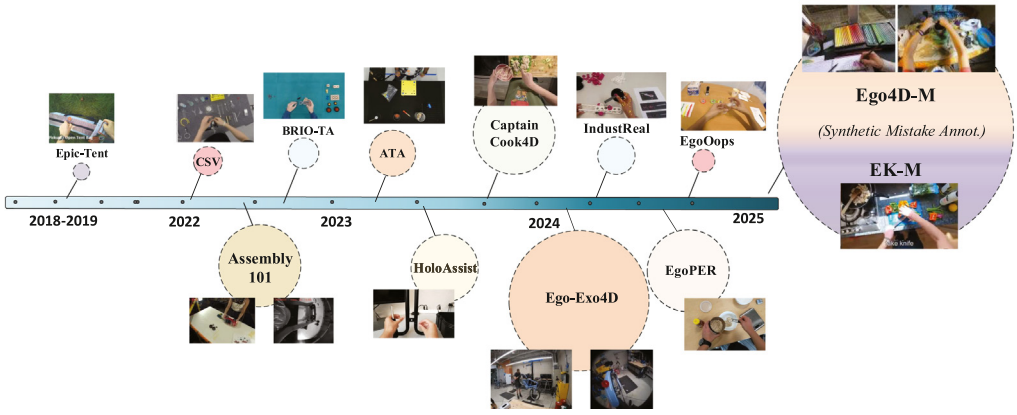

Fig. 5. Timeline of datasets supporting mistake analysis (2018–2025). Circle sizes refer to dataset scale. Datasets: Epic-Tent (Jang et al., 2019), CSV (Qian et al., 2022), Assembly-101 (Sener et al., 2022), BRIO-TA (Moriwaki et al., 2022), ATA (Ghoddoosian et al., 2023), HoloAssist (X. Wang et al., 2023), CaptainCook4D (Peddi et al., 2023), Ego-Exo4D (Grauman et al., 2022), IndustReal (Schoonbeek et al., 2024), EgoPER (Lee et al., 2024) and EgoOops (Haneji et al., 2024). Ego4D-M and EK-M (Li et al., 2025) are synthetic extensions of Ego4D (Grauman et al., 2022) and EK-100 (Damen et al., 2022), generated to incorporate explicit mistake annotations.

not define evaluation protocols for this task, as nearly every sample included at least one procedural error, preventing binary success/failure splits. To address these limitations and explicitly enable online mistake detection, Flaborea et al. (2024) introduced *Epic-Tent-O*, proposing an uncertainty-based dataset partitioning strategy: videos from high-confidence participants were used for training, while those from less confident performers, more error-prone, were reserved for testing.

Chemical Sequence Verification (CSV) (Qian et al., 2022): addresses the challenges of procedural activity understanding and mistake analysis in the context of chemical procedures. It encompasses 14 activities, each representing a chemical experiment conducted by 82 volunteers following predefined scripts. The dataset comprises a collection of 70 video recordings, partitioned into train–test subsets under a 80–20 split. The recorded sequences include deviations from the predefined execution protocols, incorporating step-level transformations such as additions, deletions and reordering of procedural steps. Annotations follow a weak supervision scheme, providing video-level activity and action labels without temporal boundaries for the actions. Notably, it does not include explicit annotations specifying the procedural mistake types. The dataset supports multiple tasks, including activity, action and mistake recognition. Additionally, it introduces the task of early mistake recognition to predict errors before they fully manifest, enabling proactive intervention.

Assembly101 (Sener et al., 2022): is a large-scale benchmark designed for action and activity understanding in assembly activities. It contains over 362 videos which amount for 6000 video clips of individuals assembling 101 toy car models, captured from multiple cameras (exo-and ego-centric) and viewing angles in a constrained laboratory environment. It supports a wide range of tasks, including fine-grained action recognition, action localization, action anticipation, activity recognition and mistake or error detection. Each video is annotated with frame-level action annotations under two granularity levels (*1380 fine-* (single-task) and *202 coarse*-grained (multiple fine-grained)) and mistake annotations, making it suitable for both step-level and task-level reasoning. The dataset provides multi-modal data, including synchronized RGB video from multiple viewpoints, depth information and 3D hand pose.

For mistake analysis, annotations exist for 1.01K of the 6K benchmark clips, provided at a coarse action-segment level, focusing on procedural errors, such as missing, or misordered steps. While executional errors (e.g., incorrect orientation or grasp) may occur in the data, they are relatively rare and not annotated. The annotation process does not differentiate between procedural and executional errors and no formal taxonomy of the mistake types is provided to support fine-grained mistake recognition. Subsequent works by Ding et al. (2023) and Flaborea et al. (2024) extended Assembly101 to better support the mistake detection task with additional annotations and better restructuring. Specifically, Ding et al. (2023) enriched it with mistake-type information and explicit part-to-part connection details to facilitate procedural and mistake reasoning, whereas (Flaborea et al., 2024) enabled the support of *online detection*, producing the variant *Assembly101-O*. In this adaptation, all correct sequences were moved to the training set, while sequences with errors were reserved for validation/testing. Procedure lengths were adjusted to better suit real-time detection requirements.

BRIO Toy Assembly (BRIO-TA) (Moriwaki et al., 2022): The dataset comprises 75 video recordings capturing both normal and anomalous assembly processes of a toy car model. The dataset includes three distinct error types that may arise during the assembly process: (a) step ordering variation, (b) step omission and (c) abnormal duration. Each erroneous sequence exhibits only one of these three anomaly types. The dataset provides segment-wise temporal annotations of action occurrences along with video-level mistake annotations. It supports the tasks of action recognition and segmentation, as well as mistake recognition. RGB videos is the only supported modality. Performance is evaluated using standard action segmentation metrics, including Accuracy, Intersection over Union (IoU) and Edit Distance, alongside mistake recognition metrics such as Accuracy. However, the dataset does not explicitly define an evaluation protocol or metrics tailored to each specific error type.

Anomalous Toy Assembly (ATA) (Ghoddoosian et al., 2023): The design of the ATA dataset was motivated by the challenge of identifying mistakes in procedural activities, where the absence of fine-grained temporal annotations poses significant difficulties. The dataset comprises of 141 untrimmed videos of 32 participants assembling 3 different toy models, recorded from 4 distinct viewpoints. It includes both correct and erroneous executions, where errors can be seen or unseen during training, making it suitable for evaluating generalization in mistake detection. ATA is annotated with weak supervision in the form of video transcripts describing action sequences, while errors are not explicitly labeled in the transcripts, simulating real-world conditions where mistake annotations are unavailable. The captured errors encompass execution mistakes (e.g., misplacement of parts) and procedural mistakes (e.g., step omissions or order deviations), making it a comprehensive benchmark for evaluating mistake detection methods for both supervised and zero-shot settings. The dataset provides multiple modalities, including RGB videos, ASR (automatic speech recognition) transcripts and gaze data, enabling multimodal learning.

Table 1
Datasets targeting mistake analysis.

Dataset	Year	Domain/Env.	View	#Activ	#Actions	Tasks	Error type	Error Gran.	#Objs	3DM	#Seqs.	Dur.	#Prps.	Modalities & Annots
EpicTent (Jang et al., 2019)	2018, 2023	Assembly/Real	Ego	1 (12)	38	MR	PEs**	Fine	NA	✗	24	>5.4 h	29	RGB, Gaze
CSV (Qian et al., 2022)	2022	Chemistry/Lab	Ego	14	106	MR, SR, EMR	PEs	Coarse	NA	✗	70	11.1 h	82	RGB
Assembly101 (Sener et al., 2022)	2022	Toy Assembly /Lab	Ego, Exo	101	202	MR, ASD, SR	PEs, EEs*	Coarse	15	✗	362	167.0 h	53	RGB, 3D Hand Pose
BRIO-TA (Moriwaki et al., 2022)	2022	Assembly	Exo	1	23	MR, SR	PEs	Fine	10	✗	75	2.9 h	15	RGB
ATA (Ghoddoosian et al., 2023)	2023	Toy Assembly /Lab	Exo	3	15	MR, SR	PEs	Fine	11	✗	141	24.8 h	32	RGB, Audio, Text, Pose, Obj-BB
HoloAssist (X. Wang et al., 2023)	2023	Assembly/Lab	Ego	20	414	MR, SR, ASD, SA, EMR	PEs, EEs	Coarse	16	✗	350	166 h	222	RGB, Depth, Pose (Hand & Head), Gaze, Text, IMU
Captain- Cook4D (Peddi et al., 2023)	2023	Cooking/Real	Ego	24	352	MR, SR, EMR, TAML	PEs, EEs	Fine	NA	✗	384	94.5 h	8	RGB, Depth, Text, TGs
Ego-Exo4D (Grauman et al., 2024)	2024	Physical Exercise, Assembly/Real	Exo, Ego	6	186	MR, SR, SA	PEs	Fine (Derived)	5.6K***	✗	628	30 h	740	RGB, Audio, Gaze, Depth, IMU, Text, TGs, Bar, Mag
IndustReal (Schoonbeek et al., 2024)	2024	Toy Assembly /Lab	Ego	2	75	MR, SR, ASD, TAML	PEs, EEs	Coarse	36	✓	84	5.8 h	27	RGB, Depth, Gaze, Pose (Hand and Body), State-BB
EgoPER (Lee et al., 2024)	2024	Cooking/Real	Ego	5	70	MR, TAML	PEs, EEs	Fine	35	✗	386	28 h	11	RGB, Depth, Audio, Gaze, Hand-BB, Obj-BB, TGs
EgoOops (Haneji et al., 2024)	2024	Diverse/Real	Ego	5	46	MR, TAML	PEs, EEs	Fine	58	✗	40	6.8 h	4	RGB, Text, Object (label-only)
EK-M (Li et al., 2025)	2025	Cooking /Real (synth.)	Ego	Inh.	12.3K	MR, TAML	PEs, EEs	Fine	Inh.	✗	>221K	UA	37	RGB, Text
Ego4D-M (Li et al., 2025)	2025	Diverse /Real (synth.)	Ego	Inh.	16K	MR, STAML	PEs, EEs	Fine	Inh.	✗	>257K	UA	248	RGB, Text, PNR, Region-BB

Tasks: MR: mistake recognition, ASD: activity state detection, SR: step recognition, TAML: temporal action and mistake localization, STAML: spatiotemporal action and mistake localization, SA: skill assessment, EMR: early mistake recognition. Error Granularity: Coarse: (correct, mistake, or correction), Fine: fine-grained mistake classes and/or semantic mistake explanation via textual descriptions, Fine (Derived): fine-grained mistake signals inferred from procedural structure or task constraints, not manually annotated.
Note: TAML, STAML correspond to Mistake Detection.
Abbreviations: Prps: Participants. Text: transcript availability, PEs: procedural errors, EEs: execution errors, 3DM: 3D models publicly available, OF: Optical Flow, Hand-BB: hand bounding boxes, Obj-BB: generic object bounding boxes, Region-BB: spatial bounding boxes localizing the error region, State-BB: Spatial bounding boxes localizing components that signify assembly state progression, Bar: Barometer, Mag: Magnetometer, TG: Task Graph. * sparse annotations, ** refined in subsequent works, *** only a subset of the total annotated objects is used for mistake analysis (exact number not provided), NA: not annotated, UA: unavailable. Inh.: synthetic dataset inherits the specific attribute from original.

HoloAssist (X. Wang et al., 2023): was designed to advance interactive AI assistants in real-world settings. It captures 20 collaborative object-centric manipulation tasks between two individuals: a *task performer*, who executes the task while wearing a mixed-reality headset that records seven synchronized data streams, and a *task instructor*, who observes the performer's first-person feed in real time and provides verbal guidance, including interventions in case of mistakes. The dataset includes 166 h of interaction data from 350 instructor–performer pairs drawn from 222 participants. Annotations include text summaries, intervention types, mistake labels, and segment-level action annotations at two abstraction levels: (a) coarse-grained, describing high-level task steps, and (b) fine-grained, referring to atomic actions composing each step. There are 414 coarse-grained and 1887 fine-grained action classes.

For mistake analysis, HoloAssist provides segment-level binary labels denoting correct or erroneous action execution. It contains both procedural and executional errors, though without distinguishing between types. Notably, it introduces intervention annotations, a novel direction for AI assistive scenarios, covering three intervention types: correcting errors, following up with more instructions, and previous action confirmation. These are temporally aligned with the conversation transcripts, allowing for explainability.

CaptainCook4D (Peddi et al., 2023): This is one of the largest and most well structured datasets targeting the understanding of mistakes/errors in complex procedural activities within the domain of cooking. It comprises 384 recordings totaling over 94.5 h of real-world kitchen interactions, capturing both standard and intentionally erroneous executions of 24 recipes. The activities are structured from a pool of 352 coarse-grained actions which are, in turn, defined from a set of fine-grained actions (steps). A standout feature of CaptainCook4D is its multimodal design, which includes RGB video, depth data, 3D object annotations, gaze data and audio. Unlike previous datasets that either lacked error annotations or only considered a binary notion of correctness, CaptainCook4D offers a rich and fine-grained taxonomy of 7 mistake types that capture both procedural and executional deviations. Mistakes are annotated at both coarse- and fine-grained levels, allowing models to reason not only about what went wrong but also when and how this occurred.

Ego-Exo4D (Grauman et al., 2024): was designed to support research in ego-exo video learning and multi-modal perception, focusing on skilled human activities. It contains time-synchronized ego- and exocentric video data, collected from 740 participants across 123 scenes in 13 cities worldwide. Participants performed 43 skilled physical and procedural activities unscripted within natural settings. In addition to RGB video, the dataset provides multi-modal sensory data, including audio, IMU, eye gaze, RGB and grayscale SLAM recordings, and 3D environment point clouds. It also includes time-indexed video–language resources, such as participant narrations during task execution and spoken expert commentaries evaluating performance. The dataset supports tasks related to activity understanding, including modeling instructor–learner relationships, recognizing fine-grained steps and task structures, assessing skill proficiency, and recovering 3D body and hand movements from egocentric video. Apart from action(step) and activity annotations, it provides 2D/3D pose annotations and object annotations, with an average of 5.5 objects annotated with correspondences between the two views in each take. Mistake analysis is introduced as a downstream problem within the task of procedure understanding. For this purpose, only 6 activities are considered, covering 186 step annotations. In total 628 sequences are recorded. The focus is on detecting procedural mistakes by identifying missing steps and recognizing steps that were not intended for the given activity.

IndustReal (Schoonbeek et al., 2024): was created to support the task of procedure step recognition in the context of industrial settings, with a particular focus on handling execution errors. It contains 84 egocentric videos, from 27 participants, captured during the assembly of a toy car, where the primary goal is to recognize and understand individual steps of the assembly process and detect execution errors within these steps. IndustReal focuses on two activities (a) the activity of assembling a toy car and (b) the activity of maintenance. The first is the most challenging, with the task being divided into 5 sub-activities, each containing multiple actions (steps). The total set of distinct actions in the dataset is 75. The structured breakdown of the assembly activity enables the tracking of task progression by identifying which sub-activity (or sub-task) has been completed. The dataset is enriched with multimodal sensor data, including video recordings and depth information (from stereo or depth cameras), providing a richer understanding of spatial and temporal context.

For mistake analysis, apart from annotations for procedural deviations (e.g., omissions), IndustReal also includes executional error cases. For this error type, it provides frame-level annotations for errors occurring during assembly at the action level, such as incorrect object interaction. In total, it contains 38 errors, 14 of which are exclusive to the validation and test sets. It was the first to relate mistake analysis and assembly state progression, using sub-task completion as a means for fine-grained procedural error detection.

EgoPER (Lee et al., 2024): EgoPER is a recent dataset that introduces a multi-modal benchmark for the study of mistake/error detection in egocentric procedural tasks. The dataset features 5 diverse cooking activities {*preparation of pinwheels, coffee, quesadilla, tea and oatmeal*} performed by 11 participants. It contains over 120 h of video recordings captured with a head-mounted camera in unconstrained environments. Unlike prior datasets focusing solely on successful task completions and generalized mistake recognition, EgoPER explicitly incorporates a rich taxonomy of errors, including step omission, addition, modification, slip, and correction. Although all errors are scripted, their execution was left natural, meaning participants had flexibility in how they introduced the error, leading to realistic and diverse instantiations of each error type. Each video is annotated with segment-wise action labels, object bounding boxes, hand-object interactions, gaze tracking, and audio, offering a comprehensive view of correct and erroneous executions.

EgoOops (Haneji et al., 2024): The dataset consists of 40 video recordings from 5 diverse task domains, namely *electrical circuit assembly, color mixture experiments, ionic reactions, toy block construction and cardboard crafts*, captured in authentic but constrained (in terms of environment), educational and workshop settings. A key feature of EgoOops, is procedural text integration, with texts tightly aligned with video segments to facilitate step-level grounding. EgoOops offers segment-wise annotations, including video–text alignments, detailed mistake labels (covering order and execution mistakes), object label annotations (microQR) and natural language descriptions of why specific actions deviate from the intended procedure, allowing for mistake explainability. Procedural mistakes in EgoOops follow the common ordering error scenarios found in related datasets, such as skipped, swapped, or repeated steps. In addition, the dataset introduces a detailed taxonomy of 6 executional mistakes, which occur when participants fail to correctly follow instructions in terms of executing the steps of an action (e.g., grasp wrong object). Finally, participants were instructed to follow scripted task and error scenarios, ensuring controlled coverage of error types; however, unintentional errors also occurred naturally during execution and were also annotated.

Ego4D-M & EPIC-KITCHENS-M: Addressing the scarcity of large-scale mistake data, Li et al. (2025) introduced EK-M and Ego4D-M, which were synthetically created by repurposing existing egocentric corpora via a data engine. Unlike datasets relying on staged or naturally occurring errors, these benchmarks are generated by systematically cross-matching instruction texts with video segments of disparate actions (e.g., pairing a "pick up hammer" instruction with a "pick up bolt" video) based on Semantic Role Labeling (SRL) mismatches. This automated misalignment process allows for the inheritance of rich annotations from the source datasets (Ego4D (Grauman et al., 2022) and EPIC-KITCHENS(EK) (Damen et al., 2022)), resulting in benchmarks

that are orders of magnitude larger than prior works, with approximately 257K and 221K samples respectively, covering 12,283 and 16,099 actions respectively, with supervision for semantic, temporal (frame-level Point-of-No -Return that signifies the reference frame for the mistake), and spatial (bounding box) mistake attribution.

4.2. Datasets: Comparison & discussion

The recent surge in datasets targeting procedural activity understanding and mistake analysis has considerably enriched the field, each offering distinct perspectives in terms of domain coverage, annotation granularity, sensing modalities and supported tasks. Datasets such as Assembly101, EpicTent and CaptainCook4D focus on structured multi-step activities (e.g., assembly or cooking), while others like EgoOops and CSV capture the variability of everyday tasks. Industrial and instructional contexts are represented by datasets such as EgoPER, IndustReal and BRIO-TA, while Holo-Assist and Ego-Exo4D emphasize AR guidance and ego-exocentric coordination. However, no single dataset fully captures the multifaceted nature of real-world mistake understanding. A deeper look shows that while individual datasets address certain challenges effectively, crucial aspects of the problem space remain underexplored. In what follows, we discuss several such aspects.

Size, modality diversity and domain generality: Datasets like Assembly101 and EpicTent are, to this date, the largest in size, providing thousands of annotated sequences for tasks such as action recognition, anticipation and mistake detection. However, they tend to focus on a narrower set of tasks or domains (e.g., assembly or camping), which limits their domain generality. Ego-Exo4D, while smaller in comparison, offers valuable insights from both first-person and third-person perspectives, being the first dataset to incorporate synchronized egocentric and exocentric perspectives, enabling new opportunities for cross-perspective activity analysis, but lacks explicit mistake annotation. In contrast, datasets like CaptainCook4D and EgoOops are relatively smaller, but they offer a richer multi-modal information (e.g., depth, RGB, audio, gaze tracking), which makes them valuable for research in real-time and multi-modal systems. Nonetheless, large-scale datasets such as Assembly101 address industrial use cases and are focused on specific tasks, which may limit their generalization across other procedural activities.

Hierarchical task structuring & mistake taxonomies: Although several datasets provide hierarchical task and action-step annotations, their alignment with mistake labels is often limited. EpicTent, CaptainCook4D, and IndustReal offer the most coherent hierarchical structures, enabling detailed analysis of sub-activities within broader tasks. Assembly101, despite its focus on assembly procedures, is comparatively coarse, typically marking transitions between high-level steps rather than fine-grained actions. Ego-Exo4D provides synchronized egocentric and exocentric views but lacks explicit hierarchical task decomposition. Industrial datasets such as BRIO-TA, ATA, and CSV include useful annotations for task verification but generally omit detailed hierarchical structure. Holo-Assist offers partial task segmentation but not the layered breakdown available in EpicTent and CaptainCook4D. EgoOops, while centered on mistake detection, does not emphasize task structuring, limiting analysis of how errors relate to task decomposition.

Mistake annotation specificity and the presence of systematic taxonomies are also inconsistent. Datasets such as IndustReal, CaptainCook4D, and EgoOops provide rich, context-sensitive annotations of execution and procedural errors, including incorrect tool use, missing steps, or mis-ordered actions, particularly in egocentric settings. Contrary, EpicTent, BRIO-TA, and ATA mainly annotate procedural deviations such as skipped or misordered steps, resulting in task-specific but not context-sensitive error definitions. Assembly101, HoloAssist, and Ego-Exo4D were designed primarily for procedural understanding, making error recognition a secondary task, and thus provide only sparse or implicit error annotations, without error taxonomies.

Leveraging Industrial Priors and 3D Models: A distinctive feature of industrial datasets like IndustReal, often absent in more general procedural benchmarks, and the vast majority of existing mistake analysis datasets, is the inclusion of deterministic priors such as 3D technical drawings, 3D part CAD models (Schoonbeek et al., 2025). While most datasets facilitate data-driven learning of error patterns, the availability of 3D assets enables a paradigm shift toward model-based verification (Stanescu et al., 2024). Furthermore, the presence of CAD files in procedural understanding benchmarks that currently lack mistake labels, such as HA4M (Cicirelli et al., 2022), HA-ViD (H. Zheng et al., 2023), and, IKEA Video Manuals (Liu et al., 2024), offers a unique opportunity for dataset expansion. By utilizing existing 3D-printed components or digital twins, researchers can systematically acquire new video sequences that explicitly capture mistake scenarios. This enables the transformation of foundational activity datasets into robust mistake analysis benchmarks, where the 3D reference provides a geometric "gold standard" for identifying discrepancies in execution or assembly state.

Mistake/error accumulation: Most of the reviewed datasets, including Assembly101 (original), EpicTent, CaptainCook4D, BRIO-TA, ATA, CSV, EgoOops and Ego-Exo4D, do not explicitly model or annotate error accumulation. For instance, in furniture manufacturing, a missing screw may lead to the incorrect placement of another component, which in turn could compromise the structural integrity of the final assembly, potentially resulting in collapse. In such cases, an erroneous prior state can propagate additional errors, exacerbating the overall failure. In many of these datasets, errors are either isolated events or are followed by corrections that resolve the problem within the same sequence segment. For example, Assembly101 (original) and EpicTent include procedural errors but lack temporal linking that would allow tracking their downstream impact. Similarly, EgoOops emphasizes error detection but not how one error might lead to others. CaptainCook4D offers multimodal recordings with rich supervision but treats errors at a segment level without modeling cumulative consequences.

Exceptions are the Assembly101 variant in Ding et al. (2023) and, more recently, the IndustReal dataset. The first incorporates the notion of error accumulation using a rule-based action transition analysis scheme, where an action transition rule is classified as transitive or intransitive, depending on how mistakes propagate through the action sequence. Transitive rules imply that any subsequent dependent action after an incorrect anchor action is also a mistake, while intransitive rules consider only specific dependent actions as mistakes. Although very generalizable, this approach is accompanied by physical annotations of specific cases in the dataset. Contrary, IndustReal emphasizes long-horizon task monitoring in real industrial environments, where an initial error can propagate through subsequent actions. Its design supports tracing such error chains, offering a realistic depiction of procedural degradation over time, crucial for systems assisting task execution.

5. Evaluation metrics in mistake analysis

The evaluation of models for mistake analysis in procedural activities is tightly coupled with the task formulation, which includes identifying a mistake (mistake recognition), anticipating it before it fully unfolds (early mistake recognition), or temporally localizing it within a video sequence (mistake detection). In this section, we categorize evaluation protocols in existing works by target task and further divide them into two groups: (a) *generalized metrics*, assessing overall system performance using standard classification or detection scores, and (b) *error type-specific metrics*, evaluating performance over specific mistake categories or structural aspects.

5.1. Generalized mistake recognition metrics

The predominant evaluation strategy adopted in prior work (Lee et al., 2024) involves formulating mistake recognition either as a *binary classification task*, wherein each action is labeled as correct or mistaken, or as a *multi-class classification task* that further distinguishes between specific error types. In these settings, standard classification metrics such as precision, recall, F1 score and the area under the receiver operating characteristic curve (AUC) are commonly used to quantify model performance.

While these general-purpose metrics provide a foundational assessment, several works have proposed adaptations that better reflect the temporal and procedural nature of the mistake detection task. One such task-specific metric is the *Error Detection Accuracy (EDA)* (Lee et al., 2024; Kung et al., 2025), originally introduced in Lee et al. (2024). EDA is defined as the ratio of correctly detected erroneous segments over the total number of ground-truth erroneous segments across all test videos. segments. Specifically, given the test set, V_t, consisting of K videos, $V_t = \{V_1, \ldots, V_K\}$, where each video V_k is decomposed into a set of segments $V_k = \{v_i^{(k)}\}_{i=1}^{N_k}$, with corresponding ground-truth mistake labels $m_i^{(k)} \in \{0, 1\}$ and predicted mistake labels $\hat{m}_i^{(k)} \in \{0, 1\}$, the EDA is defined as:

$$\text{EDA} = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathbf{1}\{m_i^{(k)} = 1 \wedge \hat{m}_i^{(k)} = 1\}}{\sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathbf{1}\{m_i^{(k)} = 1\}}, \tag{1}$$

where $\mathbf{1}\{\cdot\}$ denotes the indicator function, returning 1 if its argument is true and 0 otherwise. A segment $v_i^{(k)}$ is considered erroneous ($m_i^{(k)} = 1$) if at least a subset of its frames is labeled as a mistake, following a frame-centered evaluation scheme. This formulation aggregates detections across all test videos and captures the model's ability to correctly localize error-prone intervals while tolerating limited temporal imprecision at segment boundaries. Essentially, EDA is a *recall-type* metric that quantifies the proportion of ground-truth erroneous segments successfully recognized across the test set.

A related metric, the *Anomaly Segment Accuracy (ASA)*, was introduced in the BRIO-TA dataset (Moriwaki et al., 2022) to quantify segment-level performance in mistake recognition and detection tasks. ASA is a segment-level analogue of the standard accuracy metric, adapted to mistake analysis. The ASA metric measures the proportion of action segments that are correctly classified as either normal or erroneous, and is defined as

$$\text{ASA} = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathbf{1}\{\hat{m}_i^{(k)} = m_i^{(k)}\}}{\sum_{k=1}^{K} N_k}, \tag{2}$$

where $\mathbf{1}\{\cdot\}$ denotes the indicator function.

By construction, ASA treats both correct and erroneous segments symmetrically, in contrast to EDA which focuses exclusively on erroneous segments. Consequently, ASA provides a balanced segment-level measure of classification performance, assessing correctness across the entire procedural execution rather than solely focusing on erroneous segment localization.

5.2. Generalized early mistake recognition metrics

This task resembles early action recognition (EAR), where the goal is to anticipate an event before it fully unfolds. Therefore, similar protocols and evaluation metrics have been partially adopted. In EAR (Zhao and Wildes, 2021), common strategies include measuring classification accuracy at fixed observation percentages (e.g., 10%, 20%, …, 100% of the action duration, termed *anticipation windows*) or computing metrics like AUC to capture performance over time. Similarly, recent works in early mistake recognition Peddi et al. (2023), Qian et al. (2022), X. Wang et al. (2023) evaluated model performance at different anticipation windows, gauging predictive accuracy before the full error occurred. As an example, CaptainCook4D (Peddi et al., 2023) introduces an early error detection task, where models are evaluated at fixed intervals prior to the annotated mistake point (e.g., 2s before). Performance is measured using Top-1 accuracy and F1-score within these anticipation windows. This allows for assessing how well the model can detect an impending errors with limited temporal context. In a related but slightly different fashion, SVIP (Qian et al., 2022) evaluates sequence verification models on partially observed video snippets to assess the ability to recognize incomplete or incorrect procedures early on. While SVIP focuses on binary classification accuracy of a completed vs. incomplete (or incorrect) procedure, its use for early segment evaluation reflects similar motivations to those found in EAR literature.

5.3. Generalized mistake detection metrics

As outlined in Section 6, mistake detection can be formulated as a temporal action localization (TAL) problem, where the objective is to identify the type of mistake and its temporal boundaries within the video. Under this format, several recent works (Haneji et al., 2024; Peddi et al., 2023; Moriwaki et al., 2022) adopt evaluation protocols derived from established TAL benchmarks (Lai et al., 2024; Zhong et al., 2023), employing metrics such as mean Average Precision (mAP) and Recall at X (R@X). These metrics are computed at varying temporal Intersection-over-Union (tIoU) thresholds, typically ranging from 0.1 to 0.5 in increments of 0.1 or 0.2.

Under this scheme, mAP is calculated across all mistake action classes, explicitly excluding the "correct" class to isolate error detection performance. Furthermore, a predicted mistake label is valid only if it corresponds to the same procedural step as the ground-truth segment, thereby enforcing alignment in both temporal and semantic dimensions. This evaluation strategy ensures that models are assessed not only for their ability to detect mistakes but also for their step-level precision.

Beyond conventional TAL metrics, some recent works have introduced task-specific measures that capture higher-order aspects of procedural consistency. Notably, Schoonbeek et al. (2024) propose the *Procedure Order Similarity (POS)* metric to assess how well the predicted sequence of completed steps adheres to the canonical step order. POS is defined as:

$$\text{POS} = 1 - \min\left(\frac{\text{DamLev}(\mathcal{P}, \hat{\mathcal{P}})}{|\mathcal{P}|},\ 1\right), \tag{3}$$

where $\mathcal{P} = \{a_i\}_{i=1}^{N}$ is the ordered ground-truth step sequence, $\hat{\mathcal{P}} = \{\hat{a}_j\}_{j=1}^{M}$ is the predicted sequence, and DamLev denotes the Damerau–Levenshtein distance (Damerau, 1964) without substitutions. This metric accounts for missing, repeated, or misordered steps providing a holistic measure of procedural correctness.

5.4. Mistake/error type-specific metrics

Omission errors using action frequencies: Ghoddoosian et al. (2023) proposed a metric to evaluate the performance of mistake detection methods by expressing the discrepancies between expected and predicted action frequencies for specific action pairs. The metric relies on defining error-specific functions $\{F_e\}$ that operate over these frequencies. Each function maps the observed frequency f_a of actions in the test video to the number of instances n_e in which an error e occurs. For example, the error "*Loose Assembly*", when a component is positioned but not secured, is expressed as:

$$F_{\text{Loose Assembly}} = \max(f_{\text{place component}} - f_{\text{tighten bolt}}, 0), \tag{4}$$

where f_a denotes the frequency of action $a \in \{\text{place component}, \text{tighten bolt}\}$ within the video sequence.

To evaluate mistake detection, a baseline method is typically employed, which involves generating two distinct segmentation results: S_0 (a constrained offline step segmentation using a reference transcript

from the training set) and S_τ (a predicted step segmentation with $\tau > 0$). The corresponding action frequency distributions, f_0 and f_τ, are computed for these segmentations. Using these distributions, each error e occurrence is determined as:

$$n_e = F_e(f_\tau) \times (1 - \min(F_e(f_0), 1)). \tag{5}$$

This formulation ensures that errors are detected based on deviations from the reference segmentation while mitigating false positives. Specifically, the term $(1 - \min(F_e(f_0), 1))$ conditions error detection on discrepancies between the predicted and reference transcripts. The computed error occurrences n_e are subsequently used to quantify the overall error detection performance by computing the F1-score.

Omission errors using edit distance: Lee et al. (2024) introduced the *Omission Intersection over Union (O-IoU)* metric to evaluate the performance of mistake detection methods, particularly in scenarios involving omissions, defined as:

$$\text{O-IoU} = \frac{|GT_o \cap D_o|}{|GT_o \cup D_o|}, \tag{6}$$

where $\{GT_o, D_o\}$ refer to the set of ground-truth, and detected omission errors respectively. D_o is determined by identifying the closest sequence from the training videos to the predicted steps in the test video, based on the Edit distance. Specifically, given $\hat{\mathcal{P}}$, the set of predicted steps derived from an action segmentation method, and $\mathcal{P}$ the set of steps corresponding to the best-matched training sequence, then, the set of omitted steps, D_o, is estimated as the ratio $D_o = \mathcal{P}/\hat{\mathcal{P}}$, i.e. all steps in $\mathcal{P}$ missing from $\hat{\mathcal{P}}$.

Procedural error assessment using Weighted Distance Ratio (WDR): Qian et al. (2022) introduced WDR as an evaluation metric for procedural mistake detection, which quantifies the performance of a method by assessing the similarity between action sequences from correct and incorrect procedural executions. WDR is defined as the ratio of the average distance between negative video pairs to the average distance between positive video pairs. A *positive pair* consists of two videos in which the procedural steps are executed correctly and in the correct order, whereas a *negative pair* contains at least one video with a procedural error. The metric is defined as:

$$WDR = \frac{\frac{1}{N}\sum_{i=1}^{N} wd_i}{\frac{1}{P}\sum_{j=1}^{P} d_j}, \tag{7}$$

where P and N denote the number of positive and negative pairs, respectively. Here, d_j represents the Euclidean distance between the learned video embeddings in a negative video pair $(V_{j,1}, V_{j,2})$. In contrast, wd_i is the *normalized distance* for a positive video pair $(V_{i,1}, V_{i,2})$, defined as $wd_i = d_i/ed_i$, where d_i is the Euclidean distance between the learned embeddings of the videos, and ed_i is the Levenshtein distance between the corresponding text sequences of pair i. The text sequences symbolically represent the procedural steps performed in the video (e.g., "pick part → attach part → tighten screw"), enabling the metric to account for semantic similarity at the step level in addition to visual similarity.

A higher WDR indicates better performance, as it reflects a model's ability to clearly separate incorrect sequences (larger distances) from correct ones (smaller distances). High WDR suggests that a model effectively distinguishes between valid and invalid sequences. Conversely, a low WDR indicates poor differentiation between correct and incorrect sequences.

5.5. Mistake explanability metrics

With the emergence of methods that provide natural language justifications for detected errors (Patsch et al., 2025), evaluating the semantic quality and diagnostic accuracy of these explanations has become a critical sub-task. Current approaches repurpose standard image and video captioning metrics to assess the similarity between the generated explanation and a ground-truth reference.

The most common evaluation protocols utilize N-gram overlap metrics, including BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and CIDEr (Vedantam et al., 2015). These metrics quantify performance by calculating the precision and recall of word sequences between the predicted and reference explanations. For instance, Patsch et al. (2025) employs CIDEr as a primary metric to capture the consensus of generated explanations with human annotations. While these metrics effectively measure linguistic fluency and textual similarity, they can fail to capture *diagnostic correctness* in the context of mistake analysis. A generated explanation could achieve a high BLEU score by matching the majority of a sentence (e.g., "*The user failed to pick up the object*") while missing or hallucinating the critical causal detail (e.g., specifying the wrong object).

6. Related work on mistake analysis tasks

In this section, we provide a comprehensive overview of mistake analysis tasks and review the existing literature. A broader categorization and presentation of methods, organized according to their methodological characteristics rather than the specific tasks they target, is presented in Section 7. It is important to note that, for mistake recognition, the majority of existing studies treat it as part of the broader mistake detection problem. Accordingly, our analysis focuses on recognition within the context of detection frameworks. Additionally, in these works the classification of error types is typically formulated as a downstream classification task.

6.1. Mistake detection and recognition

As stated in Section 2, mistake detection operates directly on raw video, requiring both error localization and classification, though in existing approaches this localization is inherently constrained to the action level, aligning with predefined boundaries. Consequently, methods can only determine whether an entire action instance contains an error, overlapping with mistake recognition, which assumes access to semantically labeled clips. This limitation stems from the lack of fine-grained temporal localization, pinpointing the precise moment of error occurrence within action boundaries,in most datasets. We discuss these challenges further in Section 8.1.

Several works (Peddi et al., 2023; Haneji et al., 2024; Ghoddoosian et al., 2023) tackled mistake detection as a temporal action localization problem, assuming that deviations from expected procedural flow, such as missing steps, incorrect orderings, or executional anomalies, manifest as misaligned or inconsistent temporal patterns. Most approaches build upon self- or fully supervised action segmentation models fine-tuned on task-specific action pools to segment untrimmed videos into procedural steps. Mistake presence is then inferred through misalignment with expected sequences, unexpected transitions, or out-of-distribution behavior. Methods such as Peddi et al. (2023), Lee et al. (2024) employ spatiotemporal backbones (I3D Carreira and Zisserman, 2017, SlowFast Feichtenhofer et al., 2019) with segmentation or detection modules (e.g., ActionFormer C.-L. Zhang et al., 2022, MS-TCN++ Farha and Gall, 2019, MiniRoad An et al., 2023) followed by mistake recognition processes like segment-level classification (Peddi et al., 2023), misalignment scoring (Lee et al., 2024), or temporal inconsistency detection (Haneji et al., 2024). Some also exploit text-based procedural priors alongside RGB to identify temporal inconsistencies, for example, EgoOops (Haneji et al., 2024) aligns predicted actions with task descriptions to highlight erroneous segments. In weakly supervised settings (Ghoddoosian et al., 2023), models rely solely on ordered transcripts and constrained alignment objectives to localize unseen mistakes without direct annotations. Despite architectural differences, these methods share a common goal: to detect *when* mistakes occur in untrimmed video and to *segment* the corresponding intervals.

Recently (Li et al., 2025) pushed the boundaries of offline mistake detection beyond simple temporal intervals by introducing Mistake

Attribution (MATT). Unlike standard detection schemes followed by existing methods, which flag the entire duration of a deviation (entire action segment (procedural error) or a portion of the action segment (executional error)), MATT aims to spatiotemporally localize the mistake occurrence. To achieve this, they introduce a mechanism to detect the *Point-of-No-Return (PNR)* frame, which corresponds to the exact instant the mistake becomes irreversible. PNR serves as a temporal anchor for spatial reasoning, enabling models to localize the errors's visual manifestation (via bounding boxes).

Finally, recent works attempt to reformulate this offline setting into an online pipeline, making predictions incrementally as new frames arrive, using only past context. Flaborea et al. (2024) introduced PREGO, combining an *online action recognition* module with a *symbolic reasoning* branch driven by a pre-trained LLM that predicts the next expected action and flags discrepancies as mistakes. Plini et al. (2024) extend this by adding chain-of-thought reasoning and few-shot in-context examples, improving adaptability and temporal precision.

6.2. Early mistake recognition

Early mistake recognition methods aim to identify errors in video segments when only an initial portion of a step is observed. Unlike early action recognition (Kong and Fu, 2022), which seeks to infer the action being performed from a predefined action set, early mistake recognition focuses on detecting deviations from the intended action based on partial observations. This is challenging as it requires attending subtle cues in the evolving action dynamics, while errors may manifest in diverse and often previously unseen forms.

Despite this strong relationship between the two tasks and the extensive research dedicated to the former, the latter has received comparatively less attention, with only a few works addressing it (Peddi et al., 2023), (Qian et al., 2022), (X. Wang et al., 2023). In Qian et al. (2022) mistake detection and early recognition is targeted as a video alignment problem, leveraging a Transformer architecture to align pairs of video sequences while jointly optimizing a video-level activity recognition loss. To facilitate early mistake recognition, they propose a baseline method based on the assumption that a reference, correct execution of an activity, P_{ref}, and a candidate execution, P_{cand}, which may contain mistakes, generally span similar time intervals. Under this assumption, the time span of both executions is partitioned into k intervals and the ℓ_2 distance between P_{ref} and P_{cand} is computed in the feature space f. A sudden increase in similarity distance acts as an indicator of an unexpected event, which may correspond to a mistake.

In X. Wang et al. (2023), the authors introduce an indirect form of early mistake recognition through an *intervention forecasting* task, framed as temporal prediction: the model anticipates when and what type of intervention is needed based on ongoing task progression. Expert-issued real-time verbal interventions act as supervisory signals, marking moments where execution begins to deviate from the expected procedure and enabling the model to learn patterns associated with impending failures. Finally, Peddi et al. (2023) cast early mistake detection within classical action anticipation frameworks (Lai et al., 2024). Given only the first half of an action segment, the model predicts whether the execution will end correctly or incorrectly. Using top-performing convolutional (SlowFast Feichtenhofer et al., 2019, X3D Feichtenhofer, 2020) and transformer models (Omnivore Girdhar et al., 2022, VideoMAE Tong et al., 2022), they report a substantial drop relative to full-segment mistake recognition, underscoring the difficulty of forecasting errors under limited temporal context.

7. Methodological & supervision-based categorization

Procedural activities are inherently structured, with each action contributing to a defined sequence toward a specific goal. Beyond this sequential organization, the internal characteristics of each action instance, i.e. its constituent steps, object manipulations, and evolving scene states, also determine execution correctness. These intra-action dynamics often reveal subtle deviations that may not disrupt the overall sequence but still indicate incorrect or suboptimal execution, particularly in execution-level error analysis. Mistake detection therefore requires reasoning about correctness at both the macro sequence level and the micro patterns within actions. This typically follows a multi-stage pipeline: low-level visual perception modules (e.g., object detection Zou et al., 2023, pose C. Zheng et al., 2023, gaze estimation Cazzato et al., 2020) extract cues; mid-level modules track interactions, model temporal dynamics, or recognize actions; and high-level reasoning components compare extracted information against task models or learned activity graphs to assess alignment with expected execution patterns, flagging deviations at either level as potential errors.

Existing methods for mistake analysis often share core design principles, i.e. video inputs, temporal modeling, and action understanding, but differ in problem formulation, supervision use, and procedural knowledge integration. The broadest categorization follows the overarching design pipeline, shaped by the targeted error class: procedural, executional, or both. Procedural-focused methods emphasize step dependencies, action ordering (Seminara et al., 2024; Plini et al., 2024), or task progression models (Qian et al., 2022), whereas executional ones (Mazzamuto et al., 2025) focus on fine-grained motion, posture, or object manipulations, often ignoring broader context. Hybrid approaches (Lee et al., 2024) attempt to unify both perspectives, typically requiring more complex architectures capable of jointly modeling high-level task structure and low-level signals. The mistake family a method targets influences its learning objective, supervision level, and integrated priors. We group existing methods along four key pillars: (a) the specific mistake task; (b) supervision regime; (c) learning paradigm; and (d) degree of procedural structure integration, from structure-agnostic to hierarchy-aware models. Table 2 summarizes representative methods along these dimensions.

7.1. Procedural structure utilization

Modeling procedural activities from video requires capturing temporal dependencies, structural constraints, and permissible variations in execution. Central to this problem is the notion of a *procedural protocol* $\mathcal{P}_y$, which defines the set of valid action sequences associated with an activity y. The effectiveness of existing methods is therefore closely tied to how $\mathcal{P}_y$ is represented and operationalized.

Under the formulation introduced in Section 2, a natural abstraction of $\mathcal{P}_y$ is to expressed it as a directed graph $\mathcal{G}_y = (\mathcal{V}_y, \mathcal{E}_y)$, where nodes correspond to admissible actions (or states), and edges encode valid temporal transitions. This abstraction provides a unifying perspective that subsumes a broad range of procedural representations proposed in the literature and is explicitly adopted by graph-based approaches discussed later in this section.

Other modeling paradigms can be interpreted as imposing different structural constraints on this general graph formulation:

- **Linear protocols:** A linear protocol corresponds to a directed path graph with a single valid execution trajectory, $\mathcal{P}_y = \langle a_1, a_2, \dots, a_K \rangle$, where transitions are restricted to $(a_k, a_{k+1}) \in \mathcal{E}_y$. This representation assumes a fixed ordering of actions and is commonly adopted in template-driven approaches, where mistake analysis is formulated as an alignment problem between the observed sequence S and a canonical template.
- **State-transition protocols:** An alternative formulation defines the protocol over a state space $\mathcal{S}$, where nodes represent system states and edges encode valid transitions between them. Actions are interpreted as operators that induce state transformations, $a : S_t \rightarrow S_{t+1}$. Under this view, correctness is determined by the validity of state evolution rather than explicit action ordering.

Table 2
Method taxonomy across key design factors.

Method	Task	Error type	Error granularity	Error metric	Dataset	View	Learning strategy	Modality/Input	Superv.	Procedural structure	Online	Video trim.	Action & Activity learning method
Moriwaki et al. (2022) *ICIP'22*	MD	PEs, EEs	Coarse	TALM, ASA	BRIO-TA (Moriwaki et al., 2022)	Ego	OC	RGB	Fully	STM	✗	✓	I3D (Carreira and Zisserman, 2017) + MS-TCN (Farha and Gall, 2019)
Ding et al. (2023) *arXiv'23*	MD	PEs	Coarse, Fine	CPM	Assembly101 (Sener et al., 2022)	NV	TC	Text (AL)	Full	STM, SKG	✗	✓	TempAgg (Sener et al., 2020)
Narasimhan et al. (2023) *arXiv'23*	MD	PEs	Coarse	CPM	COIN (Tang et al., 2019)	Ego/ Exo	TC	RGB	Weak	TDM	✗	✓	VideoTaskformer (Narasimhan et al., 2023)
X. Wang et al. (2023) *ICCV'23*	MD EMP	PEs, EEs	Coarse	CPM	HoloAssist (X. Wang et al., 2023)	Ego	OC	RGB, HP, EG	Full	STM	✗	✓	TimeSformer (Bertasius et al., 2021)
Ghoddoosian et al. (2023) *ICCV'23*	MD	PEs	Coarse	CPM, TALM	ATA (Ghoddoosian et al., 2023), CSV (Qian et al., 2022)	Exo	OC, OCC, VTA	RGB	Weak	TDM	✗	✓	I3D (Carreira and Zisserman, 2017) + OadTR (Wang et al., 2021) + Viterbi variant (Ghoddoosian et al., 2023)
Schoonbeek et al. (2024) *WACV'24*	MR	PEs	Coarse	POS CPM (F1)	IndustReal (Schoonbeek et al., 2024)	Ego	TC, OCC	RGB, VL, Stereo	Weak	TDM	✓	Both	MViTv2 (Li et al., 2022)
Peddi et al. (2023) *NeurIPS'24*	MTR, EMP, MD	PEs, EEs	Coarse, Fine	CPM	CaptainCook4D (Peddi et al., 2023)	Ego	CC	RGB	Full	SFM (*MTR*, *EMP*), STM (*D*)	✗	✓	Omnivore (Girdhar et al., 2022) + ActionFormer (C.-L. Zhang et al., 2022)
Lee et al. (2024) *CVPR'24*	MD	PEs	Coarse	EDA, CPM (AUC), O-IoU, TALM	EgoPER (Lee et al., 2024), ATA (Ghoddoosian et al., 2023) HoloAssist (X. Wang et al., 2023)	Ego	OCC	RGB	Full	SFM	✗	✗	(I3D (Carreira and Zisserman, 2017)+FasterRCNN (Ren et al., 2016) +GCN) + ActionFormer (C.-L. Zhang et al., 2022)
Flaborea et al. (2024) *CVPR'24*	MD	PEs	Coarse	CPM	Assembly101-O (Flaborea et al., 2024), Epic-Tent-O (Flaborea et al., 2024)	Ego	TC, OCC	RGB	Weak	TDM	✓	Both	OadTR (Wang et al., 2021) + Llama 2.0
Seminara et al. (2024) *NeurIPS'24*	MD	PEs	Coarse	CPM	Assembly101-O (Flaborea et al., 2024), Epic-Tent-O (Flaborea et al., 2024) + Task	Ego	TC, OCC	RGB	Weak	PTG	✓	Both	MinoRoad (An et al., 2023) + Graph Transformer (Seminara et al., 2024)
Haneji et al. (2024) *arXiv'24*	MD	PEs	Coarse	TALM	EgoOops (Haneji et al., 2024)	Ego	VTA, TC	RGB	Fully	TDM, STM	✗	✓	StepFormer (Dvornik et al., 2023) + Drop-DTW (Dvornik et al., 2021)
Plini et al. (2024) *arXiv'24*	MD	PEs	Coarse	CPM	Assembly101-O (Flaborea et al., 2024), Epic-Tent-O (Flaborea et al., 2024)	Ego	TC, OCC	RGB	Weak	TDM	✓	Both	MiniRoad (An et al., 2023) + Llama 3.1 (CoT)
Mazzamuto et al. (2025) *CVPR'25*	MD	EEs	Coarse	CPM	HoloAssist (X. Wang et al., 2023), Epic-Tent (Jang et al., 2019)	Ego	OCC	RGB, EG	✗	SFM	✓	Both	Gaze-centered Transformer Autoenc. + MoCoDAD (Flaborea et al., 2023)
Huang et al. (2025) *CVPR'25*	MD	PEs	Coarse	EDA, CPM (Prec, AUC)	EgoPER (Lee et al., 2024), HoloAssist (X. Wang et al., 2023), Captain-Cook4D (Peddi et al., 2023)	Ego	OCC	RGB, TG	Weak	PTG	✗	Both	I3D (Carreira and Zisserman, 2017) + ActionFormer (C.-L. Zhang et al., 2022) +Graph Structure Operations
Hou et al. (2025) *ICME'25*	MD	EEs PEs**	Coarse	EDA CPM (AUC)	HoloAssist (X. Wang et al., 2023), EgoPER (Lee et al., 2024)	Ego	OCC	RGB	Weak	STM	✗	Both	(ActionFormer (C.-L. Zhang et al., 2022)+ CDC)+ GMM+Gaussian Smoothing Module
Kung et al. (2025) *arXiv'25*	MD	PEs, EEs	Coarse	EDA, CPM (AUC)	EgoPER (Lee et al., 2024)	Ego	VTA, TC	RGB, Text	Weak	PTG	✗	✗	Llama 3 + ASFormer (Yi et al., 2021) + HierVL (Ashutosh et al., 2023)
Lee and Elhamifar (2025) *ICCV'25*	MD	PEs EEs	Coarse Fine	TALM, CPM	EgoPER (Lee et al., 2024), CaptainCook4D*(Peddi et al., 2023)	Ego	TC	RGB, TG	Weak	PTG	✗	Both	DiffAct (Liu et al., 2023) + G2Vid (Dvornik et al., 2022) + (VideoCLIP (Xu et al., 2021) + GPT4o mini)

(*continued on next page*)

Table 2 (*continued*).

Method	Task	Error type	Error granularity	Error metric	Dataset	View	Learning strategy	Modality/Input	Superv.	Procedural structure	Online	Video trim.	Action & Activity learning method
Patsch et al. (2025)	MD-e	PEs,	Coarse,	CPM (F1)	Epic-Tent-O (Flaborea et al., 2024),	Ego	OC,	RGB,	Fully	STM	✓	Both	(ViT (Dosovitskiy et al., 2020) + Q-Former (Li et al., 2023)) + (HardFormer (Shamil et al., 2024) + Q-Former (Li et al., 2023)) + Llama 2.0
ICCV'25		EEs	Fine	NGM	HoloAssist (X. Wang et al., 2023)		VTA	HP					
Li et al. (2025)	(ST) MD	EEs	Coarse	CPM, TALM,	EK-M (Li et al., 2025),	Ego	VTA	RGB	Fully	STM	✗	✓	SRL (Gardner et al., 2018) + InternVideo2 (Wang et al., 2024) + Transformer Heads
arXiv'25		PEs**	Fine	OBJM	Ego4D-M (Li et al., 2025)			Text					

This table summarizes procedural video understanding methods across multiple dimensions. Tasks: {MR—mistake recognition, MTR—mistake type recognition, EMP—early mistake recognition, MD—mistake detection (temporal), (ST)MD—spatio-temporal MD, -e: explainability support}. Modalities: {AL—action labels, HP—hand pose, EG—eye gaze, VL—visible light, TG—task graph}. Proc. structure: {STM—stepwise model, TDM—template-driven model, SKG—semantic knowledge graph, PTG—procedural task graph, SF—structure-free}. Metrics: {CPM—classification (Acc, Prec, F1, AUC), TALM—temporal localization (Edit Distance, IoU, F1@0.5), EDA—error detection accuracy, POS—procedure order similarity, ASA—anomaly section accuracy, O-IoU–omission IoU, NGM–n-gram metrics, OBJM—object detection}. Misc.: {NV—non-vision, GCN—graph convolutional networks, ACoT—Automatic Chain of Thought(Z. Zhang et al., 2022), TAS—Temporal Action Segmentation, CDC—Causal Dilated Convolution, GMM—Gaussian Mixture Model}. *Note:* Only top-performing model configuration listed. *evaluated on subset, **does not explicitly model procedural structure.

This formulation is closely related to finite-state models and is particularly suitable for object-centric or physically grounded tasks.

While the above formulations assume that the procedural protocol $\mathcal{P}_y$ is explicitly represented, either as a graph, sequence, or state-transition system, a significant class of approaches does not construct such an explicit structural model. Instead, these methods rely on implicit temporal reasoning learned directly from data, without explicitly modeling the transition structure $\mathcal{E}_y$. From the perspective of the protocol abstraction introduced in Section 2, such approaches can be interpreted as operating on a degenerate or implicit form of $\mathcal{P}_y$, where structural dependencies between actions are not explicitly encoded. As a result, procedural consistency is not enforced through membership in $\mathcal{L}(\mathcal{P}_y)$, but is instead inferred indirectly from learned representations.

Overall, based on how (and whether) a method incorporates this procedural knowledge, existing mistake analysis approaches can be categorized into: *structure-free*, *step-wise*, *graph-based*, and *template-driven* models. A visual overview is shown in Fig. 6.

The primary technical challenge shared across these tasks is the mapping of continuous, noisy video data to these discrete symbolic protocols while remaining robust to permissible execution variations. To address execution variability, recent approaches move beyond rigid protocol definitions and introduce more flexible modeling strategies. These include the use of *soft constraints*, where transitions between actions are modeled probabilistically rather than deterministically, allowing atypical but valid sequences to be accommodated. Other approaches adopt a *query-based perspective*, where the protocol dynamically defines the set of permissible next actions conditioned on the current execution context. Additionally, *hierarchical formulations* decompose procedures into high-level goals and low-level steps, enabling stricter enforcement at the semantic level while allowing flexibility in the execution.

7.1.1. Structure-free methods

Structure-free methods (SFM) approach procedural video understanding without imposing any prior assumptions about the underlying task structure. These models treat the input video as a continuous or disjoint sequence of frames or clips, learning to predict action labels directly. While simple and often scalable, they lack an explicit notion of step boundaries or temporal dependencies, which can limit their ability to capture procedural coherence or detect structural anomalies such as missing or disordered steps. In CaptainCook4D, Peddi et al. (2023) evaluated several baseline methods (VideoMAE Tong et al., 2022, X3D Feichtenhofer, 2020, Omnivore Girdhar et al., 2022, SlowFast Feichtenhofer et al., 2019), focusing on structure-free approaches for supervised error recognition. These frameworks classify short clips as correct/mistake using only visual features from fixed-length segments, without leveraging procedural context, sequential dependencies, graph-based reasoning, or alignment with predefined task templates.

Under a different setting, Lee et al. (2024) utilize a two-stage, structure-free framework for procedural error detection. A temporal convolutional network first segments the input into per-frame step assignments, augmented with relational hand/object graphs to generate rich feature representations. For each step, a set of *prototypes* of correct executions is learned via contrastive loss, pulling frame features toward nearest prototypes and pushing them from others. At test time, each frame is scored by its cosine similarity to the closest prototype of the predicted step; low similarity indicates a procedural error.

A distinct line of research within the structure-free paradigm is proposed by Mazzamuto et al. (2025), who approach mistake detection through unsupervised gaze modeling. Rather than relying on annotated action steps, temporal boundaries, or an explicit procedural structure, their method treats gaze behavior as an implicit signal of task regularity. They introduce a gaze completion task, training a model to predict future gaze distributions from past egocentric video and eye movements during correct task executions. Mistakes are detected by comparing predicted and observed gaze, with significant deviations indicating potential attentional inconsistencies in execution.

7.1.2. Step-wise models

Step-wise models (STM) explicitly incorporate procedural structure by segmenting a task into discrete steps or actions, without the use of an explicit task graph and no template alignment against a master protocol.

Methods in this category typically adopt a supervised two-stage pipeline: segment the action, then classify it as correct or incorrect. X. Wang et al. (2023) encodes video clips of annotated action steps using a pre-trained ViT (Dosovitskiy et al., 2020), feeding per-step embeddings into a TimeSformer (Bertasius et al., 2021) with a classification head that jointly predicts the step label and a binary correctness flag. The BRIO-TA baseline (Moriwaki et al., 2022) similarly uses spatiotemporal backbones such as I3D (Carreira and Zisserman, 2017) or X3D (Feichtenhofer, 2020) followed by temporal segmentation models (e.g., MS-TCN Farha and Gall, 2019) for frame-wise mistake prediction. Extending this paradigm, Patsch et al. (2025) propose MistSense, which fuses explicit hand-pose features with RGB streams, each modeled via a dedicated temporal encoder and aligned with Q-Formers, to address fine-grained *executional errors*. While these approaches treat a step as a monolithic unit for binary classification, Li et al. (2025) advances the STM paradigm via *Mistake Attribution*. Instead of a simple binary flag, their method conditions the step-wise analysis on Semantic Role Labeling (Gardner et al., 2018), decomposing the textual description of an action into specific components (Predicate, Object). By cross-attending these role tokens with the video step features, the model can explicitly attribute the error to a specific semantic violation (e.g., recognizing an incorrect *object* manipulation despite the correct *action*). This moves modeling to fine-grained semantic and spatiotemporal localization.

A notable departure from these supervised paradigms is the approach in Lee et al. (2024), which despite following a similar action localization pipeline as previous methods, it introduces a Contrastive Step Prototype Learning (CSPL) mechanism enabling step-wise mistake detection in an unsupervised manner. Instead of learning directly from both correct and erroneous executions, CSPL exploits only the first. The model learns discriminative step-specific prototypes in a contrastive embedding space (via InfoNCE loss Oord et al., 2018), encouraging the model to produce similar embeddings for instances of the same step while pushing away different ones. At inference, each step in a test sequence is projected into this space and classified based on proximity to its corresponding prototype. Deviations from the prototype are flagged as errors.

Finally, a recent step-wise approach is the PECC framework (Hou et al., 2025), which extends the two-stage paradigm with a probabilistic treatment of step embeddings. As in prior STM models, PECC first applies a TAS backbone to obtain per-frame step labels, enforcing an explicit action segmentation. A Causal Dilated Convolution (CDC) module refines TAS features to preserve temporal dependencies and ensure causal consistency across boundaries. For each segmented step, PECC fits a Gaussian Mixture Model (GMM) over normal-execution features, yielding a probabilistic representation of step-specific appearance and motion. During inference, frames are scored by per-step log-likelihood under the corresponding GMM and flagged as errors when their likelihood deviates from the learned distribution. While it reliably identifies execution errors, it captures procedural ones only indirectly, typically when misordering or omissions cause inconsistent TAS predictions.

7.1.3. Graph-based models

Graph-based approaches represent procedural knowledge as structured graphs, where nodes correspond to actions and edges encode transitions or dependencies. These models leverage the graph structure to reason over the procedural space, enabling more robust inference of task progression and detection of deviations. This method family is the most prominent in mistake analysis, with existing works being grouped into four clusters based on the type of knowledge embedded in the graph and the way structure is utilized.

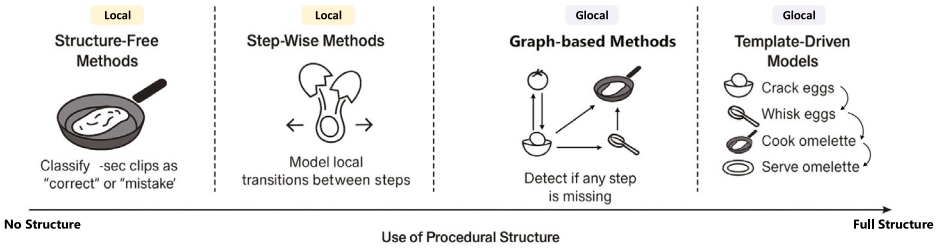

Fig. 6. Method taxonomy based on procedural structure usage. Horizontal axis reflects the *degree of structure imposed*. Vertical axis captures the *scope of structure*. Local models learn short-range temporal dependencies. Global models reason over the full process.

Procedural Task Graph methods (PTG): utilize acyclic graph structures to explicitly encode the temporal and logical structure of a procedure, where nodes represent discrete action steps or intermediate scene states, and, edges define valid transitions or causal dependencies. Such graphs may be manually annotated or automatically induced from video data (Seminara et al., 2024). They are particularly effective in modeling normal task flow and detecting anomalies, as mistakes often correspond to deviations from expected graph transitions.

In a seminal work, Seminara et al. (2024) investigated both approaches and their impact on the action understanding problem and the downstream task of mistake detection. For the more challenging variant of automatic graph construction, they proposed a differential learning scheme, where a neural encoder extracts frame-level features, while a probabilistic adjacency matrix, jointly optimized with step recognition, models step dependencies. To enable differentiable graph learning they proposed the Task Graph Maximum Likelihood (TGML) loss, that allows end-to-end optimization of the task graph's adjacency matrix. It comprises two core components: (a) *a positive reinforcement term*, maximizing edge probabilities in the adjacency matrix for observed action transitions and (b) *a contrastive penalization term*, penalizing edges corresponding to action transitions absent in training sequences.

Building on the use of explicit procedural structures, Lee and Elhamifar (2025) propose *GTG2Vid*, a method that formalizes the alignment of an observed video sequence to a predefined task graph as a dynamic programming problem. The method defines novel frame and node skipping dropping policies via a cumulative cost function over frame-node assignments, combining: (i) *matching costs* between each frame and candidate graph node (action), (ii) *frame-drop costs* that penalize advancing in time without transitioning to a new graph node, allowing the model to ignore ambiguous or visually uninformative frames (background actions) and (iii) *node-drop costs* that penalize skipping graph nodes, enabling the representation of omitted, reordered, or prematurely bypassed procedural steps. On top of this alignment, the *Error Recognition Module* evaluates the visual consistency of each frame with the expected embedding of its assigned graph node, identifying off-graph behavior as errors.

While previous methods optimize for alignment to a single path, Huang et al. (2025) leverages the task graph to explicitly handle execution variability. In their AMNAR framework, the graph functions as a query engine that retrieves *all* permissible transitions given the current state, rather than enforcing a unique trajectory. This allows the model to generate adaptive "normal" representations for multiple valid branches, effectively distinguishing between graph-compliant variations and true off-graph mistakes.

Semantic Knowledge Graph methods (SKG): utilize graphs to represent semantic relationships among objects, actions and their attributes within the task domain. These graphs are constructed from textual knowledge bases, external corpora, or structured ontologies. Rather than focusing on temporal ordering, semantic graphs provide contextual grounding for actions by representing affordances, object-action interactions and high-level concepts. When combined with vision models, they support reasoning about action plausibility given the current scene configuration, offering indirect support for error detection via semantic inconsistency analysis. Despite the potential of this strategy to capture richer affordance-level reasoning and to enforce semantic validity of action–object interactions, only a few of existing methods (Ding et al., 2023) have adopted it due to practical challenges: fine-grained objects in procedural tasks are difficult to detect and track reliably, generic detectors rarely cover task-specific components, and grounding semantic relations in visual representations requires substantial task-specific annotation.

Rule-based Graph methods (RG): These methods incorporate domain knowledge in the form of explicit procedural rules or logic, often defined by experts or extracted from formal manuals and usually are combined with the former two classes. The rules govern permissible action transitions or constraints on execution order, giving rise to graphs that represent normative task flows. Such representations are well-suited to high-precision domains such as industrial assembly or medical workflows, where deviations from the rule-defined paths are critical indicators of mistakes. Although each cluster reflects a different design philosophy for incorporating prior knowledge and modeling task regularities, a number of existing methods can belong to more than one class. An exemplar hybrid approach is (Ding et al., 2023), which builds a knowledge base of spatial (object-to-object relations) and temporal (orderings of spatial relations) graph structures and defines a set of logical rules in the temporal structure to detect ordering mistakes in the toy assembly and disassembly procedures in the Assembly101 (Sener et al., 2022) dataset.

Template-Driven Model (TDM): TDMs incorporate *external procedural templates*, such as transcripts, instructional scripts, or curated step sequences, as priors to guide the understanding of complex multi-step activities. These templates define the *canonical action ordering* and serve as reference structures against which observed video sequences are aligned. Essentially, given a procedural template represented as an ordered action sequence and the predicted sequence, TDMs seek the optimal alignment between them under temporal consistency or monotonicity constraints, interpreting deviations as procedural or executional errors. These models leverage strong task priors encoding *what should be done and in what order*, enabling both action segmentation and mistake detection under weak supervision. Ghoddoosian et al. (2023) employs a method based on action transcripts, using the *Constrained Discriminative Forward Loss (CDFL)* (Li et al., 2019) to align frame-level predictions with expected transcripts while maximizing the decision margin between valid-invalid labels. Mistake detection is set by measuring deviations from the canonical template without error examples during training. Narasimhan et al. (2023) introduces *VideoTaskformer*, which learns procedural structure by predicting randomly masked steps in instructional videos. Instead of aligning predictions to transcripts, it applies a masked modeling objective (Devlin et al., 2019) over ordered step descriptions, using a transformer to reason over global context and identify missing or out-of-order steps disrupting procedural flow. Schoonbeek et al. (2024) employs a two-stage template-driven pipeline: an Assembly State Detection (ASD) module based on YOLOv8-m (Jocher et al., 2023) recognizes current assembly states from video, followed by a Procedure Step Recognition (PSR) stage mapping detected states to step sequences via non-parametric heuristics. Deviations from canonical templates, defined by task instructions, indicate procedural error presence.

7.2. Learning strategies and objectives

7.2.1. Ordinary classification (OC)

Methods (Moriwaki et al., 2022; X. Wang et al., 2023; Peddi et al., 2023; Patsch et al., 2025) following this learning strategy treat both action and mistake recognition/detection as standard supervised classification tasks. A backbone network, such as a 3D convolutional model (e.g., I3D Carreira and Zisserman, 2017) or a transformer architecture (e.g., TimeSformer Bertasius et al., 2021), is trained end-to-end using a multi-class cross-entropy loss for classifying action steps and a binary cross-entropy loss for labeling each step as correct or erroneous. Crucially, this approach requires dense, segment-level annotations, including both ground-truth action class and associated correctness label for each step. As such, it necessitates the availability of annotated sequences containing both correct and erroneous executions during training, which can be costly and time-intensive to curate, particularly in long-horizon tasks.

At inference, each segmented clip is classified independently, typically without access to broader task context. To mitigate prediction noise and improve temporal coherence, simple post-processing strategies such as majority voting or sliding-window smoothing are often applied across adjacent segments. Such learning objectives, inherent in structure-free methods, do not explicitly account for procedural dependencies between steps. This limits their ability to detect subtle or high-level error types, such as improper ordering of steps, repetitions, or omissions, which may only be apparent when reasoning over the entire task trajectory. Moreover, they may struggle to generalize to novel mistake types or execution variants unseen during training, given their reliance on direct supervision.

7.2.2. One-class classification (OCC)

In this context, models for mistake recognition or detection are trained solely on samples of correct behavior, aiming to identify deviations from the learned norm during inference. This approach suits mistake and broader anomaly detection tasks, where anomalous events, such as procedural mistakes, are rare and lack explicit error examples for training. Existing works follow different strategies within this framework. Some adopt *prototype learning*, clustering normal executions as reference points at test time. For instance, Lee et al. (2024) learns multiple contrastive prototypes per procedural step and score incoming frames by their distance to the nearest prototype, while Seminara et al. (2024) extend this idea using a differentiable task graph that models step transitions and flags off-graph deviations as errors. However, static representations struggle to capture the diversity of branching tasks. Addressing this, Huang et al. (2025) introduce the AMNAR framework, which leverages a task graph to model valid transitions. It uses the graph to dynamically reconstruct *multiple* valid action representations conditioned on the current context, ensuring that permissible variations in execution order are not misclassified as mistakes.

Other approaches explore OCC with *predictive and step-aware reasoning* strategies, where the model anticipates future procedural steps and evaluates the consistency of observed behavior against these predictions. PREGO (Flaborea et al., 2024) and TI-PREGO (Plini et al., 2024) utilize an OCC framework that anticipates the next step embedding based on symbolic task progression predicted by an LLM, while grounding the actual observation through a visual encoder. Discrepancies between predicted and actual embeddings are used to detect errors as they occur in real time, without ever observing mistakes during training.

7.2.3. Task Completion (TC)

The *Task Completion* learning objective encompasses methods that evaluate model performance based on the accurate realization of an entire task, rather than relying solely on localized predictions at the frame or segment level. In this paradigm, models aim to capture the temporal dependencies and structural coherence of sub-events that collectively define a procedural sequence. The notion of a "task" may vary in granularity: at the *activity level*, it involves a sequence of high-level action steps (e.g., "crack egg", "whisk", "pour" to complete "make an omelet"); whereas at the *action level*, it may refer to the unfolding of a single action via a trajectory of intermediate states, such as human pose configurations, hand-object interactions, or gaze fixations. Deviations between the predicted and canonical sequence, such as missing, reordered, or inconsistent steps, are interpreted as mistakes.

Training objectives in this category often include masked step prediction, sequence reconstruction, or permutation-based reasoning, encouraging the model to infer and enforce task-level consistency. Unlike conventional classification frameworks, TC approaches do not require explicit error labels; correctness is inferred from whether the predicted trajectory conforms to a learned procedural schema. For instance, Narasimhan et al. (2023) learn structured task representations by predicting randomly masked steps in instructional videos, allowing the model to identify steps that break the expected temporal or logical flow. In a complementary formulation, a subset of methods approach this objective via anticipation-based reasoning, where correctness is evaluated by comparing the model's recognized step with the expected next step, as inferred from a structured procedure. PREGO (Flaborea et al., 2024) and TI-PREGO (Plini et al., 2024) formulate the task as a joint problem of *step recognition* and *step anticipation*, where a mismatch between anticipated and recognized steps signals a deviation from the canonical trajectory of the action-steps of the activity.

7.2.4. Video–text alignment as a supervisory signal (VTA)

VTA-based approaches utilize procedural text (e.g., step descriptions from manuals, task guides, or LLM-generated scripts) as a supervisory signal to guide action detection, anticipation, and activity understanding (Kahatapitiya et al., 2024; Chen et al., 2024; Li et al., 2024). These methods typically segment untrimmed videos into candidate action clips by aligning them to textual templates using techniques such as dynamic time warping, cross-modal attention, or contrastive learning between video and language embeddings, then pass the aligned segments to classification or matching modules to predict the activity or next action.

In mistake analysis, VTA enables assessing step correctness by evaluating alignment strength or semantic consistency between predicted visual content and the textual description of correct behavior. Its adoption remains limited due to scarce datasets like EgoOops (Haneji et al., 2024) and CaptainCook4D (Peddi et al., 2023), which provide paired instructional text and annotated mistakes. Haneji et al. (2024) align egocentric video segments with procedural transcripts. StepFormer++ predicts the current step and localizes mistakes by detecting misalignments between visual input and expected procedural steps, using cross-attention to learn temporal and semantic alignment between video tokens and step embeddings. To address data limitations, Kung et al. (2025) propose a counterfactual VTA framework for mistake analysis using Ego4D (Grauman et al., 2022). While Ego4D lacks error annotations, it provides rich activity- and action-level captions. LLaMA 3.2 8B is used to generate counterfactual descriptions at action and activity levels, describing missing, misordered, or incorrectly executed steps. These synthetic error-augmented captions are paired with correct video clips for contrastive training of HierVL (Ashutosh et al., 2023), a hierarchical video–language model.

Moving beyond sentence-level alignment, *MisFormer* (Li et al., 2025) utilizes Semantic Role Labeling (SRL) to decompose procedural instructions into fine-grained components (e.g., Predicate (verb), Object). Unlike previous VTA approaches that treat the text as a single embedding, MisFormer employs a dual-branch transformer that cross-attends these specific role tokens against video patches. This enables the model to determine specifically *which* part of the instruction was violated (e.g., correct action (verb) but wrong object). Complementing these attribution-focused methods, MistSense (Patsch et al., 2025) leverage

VTA for post-hoc explanation rather than just detection. It employs a gating mechanism that, upon detecting an error, projects video-based features into the embedding space of a LLM. This allows the system to generate natural language descriptions of the error (e.g., "*User is wrongly mounting a screw*"), bridging the gap between binary detection and interpretable user feedback.

7.3. Mistake/error types and modeling choices

The selection of a modeling paradigm is inherently linked to the target error category within the mistake taxonomy. Procedural errors, such as step omissions, incorrect ordering, or invalid transitions, require models capable of capturing global task structure and long-range dependencies. As illustrated in Fig. 6, these errors are most effectively addressed by structure-aware approaches, including graph-based and template-driven models, which explicitly encode the procedural protocol and enforce consistency with valid execution paths.

Contrary, executional errors arise from incorrect action performance, such as improper object usage, abnormal motion patterns, or imprecise manipulation. Detecting such errors requires fine-grained modeling of visual appearance, motion dynamics, and human-object interactions. These are typically handled by structure-free or step-wise models (see Fig. 6) that focus on local transitions and micro-patterns rather than global task-level dependencies.

Certain mistake categories, such as early-stage deviations or context-dependent errors, require both procedural reasoning and detailed visual understanding. These cases motivate hybrid approaches that jointly model global task structure and local action execution, highlighting the inherently multi-scale nature of mistake analysis. This mapping reveals a fundamental trade-off depicted along the horizontal axis of Fig. 6: models that impose stronger structural constraints are better suited for identifying procedural inconsistencies, while more flexible, structure-free approaches excel at capturing execution-level variability. Bridging this gap remains an open challenge, motivating future research toward unified frameworks that integrate structured reasoning with fine-grained perception.

7.4. Supervision levels in mistake analysis

7.4.1. Fully supervised approaches

Fully supervised approaches rely on datasets with explicit error annotations at a fine-grained level. In this setting, mistakes are annotated with precise spatial and temporal localization, typically at the frame or segment level, enabling models to learn the error type and also its exact temporal occurrence in the video. Most methods following this paradigm operate in an offline setting (Peddi et al., 2023; Sener et al., 2022; X. Wang et al., 2023), where detailed annotations are used to supervise both step segmentation and mistake classification. Ding et al. (2023) leverage fine-grained labels to construct spatiotemporal knowledge graphs guiding mistake reasoning. Similarly, in BRIO-TA, Moriwaki et al. (2022) provide segment-level annotations indicating correct or erroneous execution, allowing models to be trained using standard supervised classification objectives at the action segment level.

7.4.2. Weakly-supervised methods

Weak supervision is characterized by two complementary perspectives: (1) the methodological design leveraging auxiliary supervision signals and (2) the granularity of annotations available during training.

In mistake analysis, the first perspective concerns the absence of explicit mistake annotations, compensated by exploiting the procedural structure (actions and their transitions) of the activity. This is exemplified by methods leveraging global step order to detect execution inconsistencies (Grauman et al., 2024; Ghoddoosian et al., 2023; Schoonbeek et al., 2024; Narasimhan et al., 2023; Lee and Elhamifar, 2025). Although these methods do not require annotated mistake labels for training, they still rely on step-level annotations, i.e. knowledge of actions and their order. The second perspective relates to annotation granularity, where weak supervision labels mistakes only at the video level (e.g., indicating a mistake's presence) without specifying when it occurs. In such cases, models must implicitly localize deviations based on high-level error cues.

In Grauman et al. (2024), a weak supervision strategy is explored with two supervision levels: (1) *instance-level*, where video segments and key-step labels are given, and (2) *procedure-level*, where segments are unlabeled and key-step names are provided; in this second variant, a pre-trained video–language model supplies pseudo-labels, exemplifying self-supervised weak supervision. Similarly, Ghoddoosian et al. (2023) use only activity transcripts, without frame labels or error examples, aligning model predictions to valid sequences via a transcript-consistency loss. In contrast, Narasimhan et al. (2023) learn procedural structure by predicting masked steps using only step-level annotations, enabling detection of missing or out-of-order steps.

More recent structure-driven methods (Lee and Elhamifar, 2025; Huang et al., 2025) employ predefined task graphs as indirect supervision, jointly performing step alignment and error recognition through a graph-constrained dynamic programming objective. These frameworks allow both frame and node skipping, enabling the detection of deviations without dense annotations. In a complement form of weak supervision, PECC (Hou et al., 2025), requires only step-level annotations of correct executions and no error labels, avoiding reliance on external procedural structure, textual cues, or predefined transitions. PECC trains a temporal action segmentation backbone on correct sequences and fits Gaussian Mixture Models in a one-class manner. By modeling steps independently, it flags low-likelihood feature deviations as executional errors, though this independence limits its ability to detect procedural errors.

Finally, a recent sub-line of weakly-supervised approaches (Flaborea et al., 2024; Plini et al., 2024) departs from previous formulations and defines mistake detection as a combination of *step recognition* and *step anticipation*. These methods exploit the structured nature of task transcripts as supervision signals, without requiring any frame-level error annotations. Notably, they leverage pre-trained LLMs as reasoning agents to infer expected next steps based on procedural understanding.

7.4.3. Unsupervised methods

Unsupervised methods for mistake recognition or detection aim to identify deviations in action execution without labeled mistakes, annotated boundaries, or knowledge of an action's procedural role. They assume only correct executions are available during training, with mistakes manifesting as deviations, statistically or structurally distinct from valid behavior. Accordingly, such methods learn compact representations or generative models of correct task performance, flagging as outliers those instances that cannot be well explained, reconstructed, or predicted by the model.

Existing unsupervised methods follow strategies centered on *predictive modeling*, where states of modalities (e.g., gaze, human 2D/3D pose) are predicted and mistakes inferred from large prediction deviations. They rely on latent modeling of correct trajectories, often as feature embeddings, where deviations from learned distributions are flagged as mistakes/anomalies. This category of mistake recognition/detection methods is related to VAD methods (see Section 2.3). However, even though they do not explicitly model task semantics, action steps, or procedural knowledge, unsupervised mistake analysis methods are applied to procedural activities (e.g., cooking, object assembly), where mistakes are subtle and often arise from missing, improperly ordered, or poorly executed steps. Mazzamuto et al. (2025) leverage egocentric gaze prediction as an unsupervised proxy for task regularity. It defines a novel gaze completion task and measures deviations between predicted and observed gaze to detect mistakes online, without using any labels or

manual annotations. To capture this, they formulate the gaze completion task with a transformer-based model that learns to predict future gaze trajectories based on past egocentric video and gaze history. In inference, discrepancies between predicted and actual observed gaze are interpreted as anomalies, with large deviations signaling possible mistakes or disruptions in task execution.

8. Challenges and future directions

Despite significant progress in procedural activity understanding, a number of challenges and open problems remain. Addressing them is critical for advancing mistake analysis, particularly in developing methods that are robust, interpretable and generalizable. We outline the major open problems in the field and highlight promising research directions.

8.1. Challenges in mistake analysis

Datasets, modality & evaluation bottlenecks: Despite growing interest in mistake analysis for procedural activities, the limited number and scale of comprehensive datasets remain major bottlenecks. Most existing video datasets target action or activity recognition, detection, or anticipation, typically assuming correct task execution. As a result, mistake annotations (when available) are often coarse, restricted to binary video-level labels (mistake vs. correct), and omit key aspects such as temporal boundaries, causal origins, or semantic types (e.g., omission, misordering). The scarcity of temporally localized and semantically disambiguated annotations hampers model development and evaluation, while annotation protocols remain labor-intensive and domain-specific, particularly for subtle or ambiguous errors. Beyond annotation gaps, datasets exhibit limited modality coverage and biased recording perspectives. Modalities such as gaze, object state, 2D/3D pose, or synchronized audio can enhance error analysis but are often absent due to sensor or collection constraints. Most datasets adopt egocentric viewpoints, especially for household or instructional tasks, limiting applicability to industrial contexts where exocentric or hybrid ego–exo views are more suitable. The lack of standardized evaluation metrics further complicates comparison: most works rely on general-purpose classification or segmentation metrics (e.g., Acc., F1-score, IoU) that poorly capture the challenges of mistake analysis tasks, hindering cross-method benchmarking.

Ambiguity between permissible variations and true errors. A persistent challenge in procedural mistake analysis is distinguishing valid execution variability from actual errors. Many tasks permit multiple correct action orders, interaction styles, or tool-use strategies that achieve the same goal without affecting outcome quality. This flexibility complicates annotation: annotators often lack a shared definition of what constitutes an error, leading to high labeling costs and inter-annotator disagreement, especially when correctness criteria are implicit or task goals under-specified. As a result, existing datasets tend to simplify error taxonomies and overlook real-world procedural nuance. Most modeling approaches rely on distributional patterns learned from data, lacking explicit representations of task goals, constraints, or causal dependencies. This causes rare but valid behaviors to be misclassified as errors, while context-dependent violations remain undetected without deeper reasoning about action affordances, object roles, or causal effects (e.g., omissions, order violations). Recent video–language models (Tang et al., 2025) jointly encode temporal and semantic information and can be prompted with structured task goals or contextual cues. However, fine-grained error attribution remains difficult, particularly for task-specific correctness rules. Vision–language frameworks such as PREGO (Flaborea et al., 2024) and TI-PREGO (Plini et al., 2024) show promise in detecting procedural anomalies, though execution-level errors remain largely underexplored. Moreover, their sensitivity to prompt phrasing and contextual formulation (Zhou et al., 2022) raises robustness concerns.

Disjoint embedding spaces between LLMs & video encoders: A key limitation in current mistake analysis frameworks lies in the lack of task-level reasoning, where models fail to encode procedural goals, causal dependencies, and object affordances, resulting in systems that primarily detect low-level visual deviations without understanding whether an observed deviation truly constitutes a mistake (Mazzamuto et al., 2025). Video–Language Models (Video LLMs) (Flaborea et al., 2024; Plini et al., 2024) have emerged as a promising direction, integrating the semantic richness of LLMs with the temporal and spatial modeling capacity of video encoders. However, representational misalignment between the two modalities (vision, language) remains a core challenge. Current designs rely on dual-stream architectures with shallow fusion, handcrafted prompts, or minimal supervision of cross-modal correspondences. As a result, they struggle to ground complex procedural logic or contextual correctness in visual data (Rawal et al., 2023) and remain limited to detecting procedural mistakes, which require alignment with predefined step sequences, rather than executional mistakes, which require fine-grained perception of action quality and context-dependent deviations.

8.2. Future directions in mistake analysis

Lifelong learning & open-vocabulary recognition: A major limitation of current methods is the reliance on closed-world training setups, where action–object classes, and task flows are assumed fixed and fully known during training. However, real-world procedures are dynamic, evolving across domains, involving diverse object-tool configurations, and introducing novel behaviors or error patterns unseen during training. This motivates a shift toward lifelong learning and open-vocabulary recognition frameworks for more robust, scalable, and generalizable analysis. Lifelong learning paradigms, particularly for video understanding (e.g., Gowda et al., 2024), enable models to incrementally learn from streaming video while retaining performance on prior tasks. Such adaptability is crucial for vision-based mistake analysis, especially for executional mistakes, where new action styles may arise due to tool substitutions. Complementing continual learning, open-vocabulary recognition allows models to describe and detect mistakes beyond closed taxonomies, enabling zero-shot recognition of novel mistake classes. Future methods could draw inspiration from action understanding works that generalize beyond fixed label sets using vision–language alignment and compositional prompts in multimodal large language models (e.g., Gupta et al., 2024; Chatterjee et al., 2023).

Explainability via symbolic reasoning integration: A major challenge in vision-based mistake analysis is the opacity of end-to-end neural models, which provide little insight into why an action is deemed incorrect—an issue made critical by the flexibility of human procedures and the need to distinguish permissible variations from true violations. A promising remedy is neuro-symbolic integration, combining video perception with symbolic procedural constraints, causal rules, or temporal logic. Such systems could (a) parse video into symbolic step and object-state representations (Flaborea et al., 2024; Plini et al., 2024; Gouidis et al., 2024), (b) enforce procedural logic using symbolic reasoners or temporal logic automata (Choi et al., 2024), and (c) ground detected discrepancies in explicit task rules or semantic-role violations (Li et al., 2025). This would improve both detection accuracy and interpretability while enabling zero-shot generalization across new procedures. LLM-based natural-language explanations (Patsch et al., 2025) illustrate the potential for intuitive feedback, but ensuring factual grounding and verifiable structure remains difficult without explicit symbolic constraints. Moreover, standard captioning metrics (e.g., BLEU, CIDEr) are insufficient for evaluating mistake explanations (Patsch et al., 2025), as they capture surface textual overlap rather than causal correctness, highlighting the need for logic-based evaluation protocols.

Understanding error dependencies & propagation: Building on interpretable representations for procedural understanding, the next step is to move beyond detection and examine how errors influence downstream actions within task sequences. In real-world industrial or medical settings, errors rarely occur in isolation; initial deviations can propagate, altering execution and causing more severe failures. Understanding these inter-error dependencies is essential for systems that not only detect mistakes but anticipate and mitigate cascading effects. While early works like AMNAR (Huang et al., 2025) use graph structures to predict valid next-steps based on current history, future methods must extend this to explicitly learn the likelihood of specific error transitions (e.g., Error A increases the probability of Error C) to enable early interventions preventing cascading failures. Integrating predictive capabilities into real-time mistake detection could enhance automated quality control and prevention. Recent synthetic-domain works like (Hong et al., 2021), can model explicit visual state transitions, providing a foundation for inferring causal chains of mistakes via object state transitions in real-world procedural tasks.

Enriching datasets via hypothetical transitions & synthetic errors: Procedure-aware video mistake understanding require modeling both actual and hypothetical scene state transitions, capturing action-induced transformations and plausible deviations. Several works highlighted the link between action/activity recognition and object/scene state changes (Saini et al., 2023; Wang et al., 2016; Bacharidis and Argyros, 2023; Souček et al., 2022; Niu et al., 2024). A promising direction is augmenting datasets with synthetic state-change sequences and counterfactual exemplars (Kung et al., 2025; Souček et al., 2024; Chatzi et al., 2024; Souček et al., 2025), enabling models to reason about causal and semantic effects of actions by grounding visual representations in expected and unexpected outcomes. This fosters richer temporal representations that distinguish correct progressions from deviations/failures. Recent works (Lee and Elhamifar, 2025; Kung et al., 2025) showed that, given prior knowledge of an activity's protocol, structured procedural knowledge bases (e.g., WikiHow Koupaee and Wang, 2018) or LLM-generated textual descriptions can define expected pre-/post-action states and hypothetical post-conditions for potential failures. These textual representations can then be aligned with visual features to reinforce plausible state transitions and decouple counterfactual ones via contrastive learning. Complementing these generative approaches, Li et al. (2025) proposed a retrieval-based augmentation strategy that constructs semantically attributed mistake video samples by cross-matching instruction texts with video segments of disparate actions (e.g., pair *pick up hammer* with *pick up bolt*), though visual consistency and scene semantics are not fully preserved.

9. Conclusions

Vision-based mistake analysis in procedural activities is a rapidly evolving field with potential to improve task performance, safety, and efficiency across domains. This paper reviews existing methods, datasets, and evaluation metrics, outlining challenges and opportunities in detecting and predicting procedural and executional errors. By categorizing approaches by procedural structure, supervision, and learning strategies, we offer a structured view of the state of the art. Key challenges remain, such as the ambiguity between permissible variations and true mistakes, limited comprehensive datasets, and the need for models that reason about error dependencies and propagation. Addressing these requires innovations such as neuro-symbolic reasoning, leveraging large language models for procedural understanding, and enriching datasets with counterfactual transitions and synthetic mistakes. Future research should emphasize explainable mistake analysis, stronger cross-task generalization, and real-time predictive detection, unlocking new possibilities for human–robot collaboration, automation, and assistive technologies, ultimately transforming how mistakes are understood and mitigated.

CRediT authorship contribution statement

Konstantinos Bacharidis: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Antonis Argyros:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded and supported by the European Union (EU - HE Magician – Grant Agreement 101120731).

Data availability

No data was used for the research described in the article.

References

Abdalla, M., Javed, S., Radi, M.A., Ulhaq, A., Werghi, N., 2024. Video anomaly detection in 10 years: A survey and outlook. arXiv:2405.19387.

Aganian, D., Stephan, B., Eisenbach, M., Stretz, C., Gross, H.-M., 2023. Attach dataset: Annotated two-handed assembly actions for human action understanding. In: 2023 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 11367–11373.

Aggarwal, J.K., Ryoo, M.S., 2011. Human activity analysis: A review. Acm Comput. Surv. (Csur) 43 (3), 1–43.

An, J., Kang, H., Han, S.H., Yang, M.-H., Kim, S.J., 2023. Miniroad: Minimal rnn framework for online action detection. In: ICCV. pp. 10341–10350.

Ashutosh, K., Girdhar, R., Torresani, L., Grauman, K., 2023. Hiervl: Learning hierarchical video-language embeddings. In: CVPR. pp. 23066–23078.

Bacharidis, K., Argyros, A., 2023. Repetition-aware image sequence sampling for recognizing repetitive human actions. In: ICCV (ICCV) Workshops. pp. 1878–1887.

Ben-Shabat, Y., Yu, X., Saleh, F., Campbell, D., Rodriguez-Opazo, C., Li, H., Gould, S., 2021. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 847–859.

Bertasius, G., Wang, H., Torresani, L., 2021. Is space–time attention all you need for video understanding?. In: ICML, Vol. 2, p. 4.

Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 6299–6308.

Cazzato, D., Leo, M., Distante, C., Voos, H., 2020. When i look into your eyes: A survey on computer vision contributions for human gaze estimation and tracking. Sensors 20 (13), 3739.

Chatterjee, D., Sener, F., Ma, S., Yao, A., 2023. Opening the vocabulary of egocentric actions. NeurIPS 36, 33174–33187.

Chatzi, I., Benz, N.C., Straitouri, E., Tsirtsis, S., Gomez-Rodriguez, M., 2024. Counterfactual token generation in large language models. arXiv:2409.17027.

Chen, Y., Chen, D., Liu, R., Zhou, S., Xue, W., Peng, W., 2024. Align before adapt: Leveraging entity-to-region alignments for generalizable video action recognition. In: CVPR. pp. 18688–18698.

Choi, M., Goel, H., Omama, M., Yang, Y., Shah, S., Chinchali, S., 2024. Towards neuro-symbolic video understanding. In: ECCV. Springer, pp. 220–236.

Cicirelli, G., Marani, R., Romeo, L., Domínguez, M.G., Heras, J., Perri, A.G., D'Orazio, T., 2022. The ha4m dataset: Multi-modal monitoring of an assembly task for human action recognition in manufacturing. Sci. Data 9 (1), 745.

Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al., 2022. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. Int. J. Comput. Vis. 130 (1), 33–55.

Damerau, F.J., 1964. A technique for computer detection and correction of spelling errors. Commun. ACM 7, 171–176.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In: Conference of the North American Chapter of the Association for Computational Linguistics. pp. 4171–4186.

Ding, G., Sener, F., Ma, S., Yao, A., 2023. Every mistake counts in assembly. https://arxiv.org/abs/2307.16453, arXiv:2307.16453.

Ding, G., Sener, F., Yao, A., 2024. Temporal action segmentation: An analysis of modern techniques. IEEE Trans. Pattern Anal. Mach. Intell. 46 (2), 1011–1030. http://dx.doi.org/10.1109/TPAMI.2023.3327284.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929.

Dvornik, M., Hadji, I., Derpanis, K.G., Garg, A., Jepson, A., 2021. Drop-dtw: Aligning common signal between sequences while dropping outliers. NeurIPS 34, 13782–13793.

Dvornik, N., Hadji, I., Pham, H., Bhatt, D., Martinez, B., Fazly, A., Jepson, A.D., 2022. Flow graph to video grounding for weakly-supervised multi-step localization. In: European Conference on Computer Vision. Springer, pp. 319–335.

Dvornik, N., Hadji, I., Zhang, R., Derpanis, K.G., Wildes, R.P., Jepson, A.D., 2023. Stepformer: Self-supervised step discovery and localization in instructional videos. In: CVPR. pp. 18952–18961.

Farha, Y.A., Gall, J., 2019. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: CVPR. pp. 3575–3584.

Feichtenhofer, C., 2020. X3d: Expanding architectures for efficient video recognition. In: CVPR. pp. 203–213.

Feichtenhofer, C., Fan, H., Malik, J., He, K., 2019. Slowfast networks for video recognition. In: ICCV. pp. 6202–6211.

Flaborea, A., Collorone, L., Di Melendugno, G.M.D., D'Arrigo, S., Prenkaj, B., Galasso, F., 2023. Multimodal motion conditioned diffusion model for skeleton-based video anomaly detection. In: ICCV. pp. 10318–10329.

Flaborea, A., Di Melendugno, G.M.D., Plini, L., Scofano, L., De Matteis, E., Furnari, A., Farinella, G.M., Galasso, F., 2024. Prego: online mistake detection in procedural egocentric videos. In: CVPR. pp. 18483–18492.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N.F., Peters, M.E., Schmitz, M., Zettlemoyer, L., 2018. Allennlp: A deep semantic natural language processing platform. In: Proceedings of Workshop for NLP Open Source Software. NLP-OSS, pp. 1–6.

Garrabé, É., Teixeira, P., Khoramshahi, M., Doncieux, S., 2024. Enhancing robustness in language-driven robotics: A modular approach to failure reduction. arXiv:2411.05474.

Ghoddoosian, R., Dwivedi, I., Agarwal, N., Dariush, B., 2023. Weakly-supervised action segmentation and unseen error detection in anomalous instructional videos. In: ICCV. pp. 10128–10138.

Girdhar, R., Singh, M., Ravi, N., Maaten, L. Van Der, Joulin, A., Misra, I., 2022. Omnivore: A single model for many visual modalities. In: CVPR. pp. 16102–16112.

Gouidis, F., Papoutsakis, K., Patkos, T., Argyros, A., Plexousakis, D., 2024. Enabling visual intelligence by leveraging visual object states in a neurosymbolic framework: A position paper. In: Australasian Joint Conference on Artificial Intelligence. Springer, pp. 312–320.

Gowda, S.N., Moltisanti, D., Sevilla-Lara, L., 2024. Continual learning improves zero-shot action recognition. In: ACCV. pp. 3239–3256.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al., 2022. Ego4d: Around the world in 3000 h of egocentric video. In: CVPR. pp. 18995–19012.

Grauman, K., Westbury, A., Torresani, L., Kitani, K., Malik, J., Afouras, T., Ashutosh, K., Baiyya, V., Bansal, S., Boote, B., et al., 2024. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In: CVPR. pp. 19383–19400.

Gupta, R., Rizve, M.N., Unnikrishnan, J., Tawari, A., Tran, S., Shah, M., Yao, B., Chilimbi, T., 2024. Open vocabulary multi-label video classification. In: ECCV. Springer, pp. 276–293.

Haneji, Y., Nishimura, T., Kameko, H., Shirai, K., Yoshida, T., Kajimura, K., Yamamoto, K., Cui, T., Nishimoto, T., Mori, S., 2024. Egooops: A dataset for mistake action detection from egocentric videos with procedural texts. arXiv:2410.05343.

Herath, S., Harandi, M., Porikli, F., 2017. Going deeper into action recognition: A survey. IVC 60, 4–21.

Hong, X., Lan, Y., Pang, L., Guo, J., Cheng, X., 2021. Transformation driven visual reasoning. In: CVPR. pp. 6903–6912.

Hou, T., Li, S., Jiang, X., Wang, Z., Shen, F., Xu, X., 2025. Probabilistic embeddings with causal constraint for error detection in egocentric procedural videos. In: 2025 IEEE International Conference on Multimedia and Expo. ICME, pp. 1–6. http://dx.doi.org/10.1109/ICME59968.2025.11209124.

Huang, W.-J., Li, Y.-M., Xia, Z.-W., Tang, Y.-M., Lin, K.-Y., Hu, J.-F., Zheng, W.-S., 2025. Modeling multiple normal action representations for error detection in procedural tasks. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27794–27804.

Jang, Y., Sullivan, B., Ludwig, C., Gilchrist, I., Damen, D., Mayol-Cuevas, W., 2019. Epic-tent: An egocentric video dataset for camping tent assembly. In: ICCV Workshops.

Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics yolov8. https://github.com/ultralytics/ultralytics.

Kahatapitiya, K., Arnab, A., Nagrani, A., Ryoo, M.S., 2024. Victr: Video-conditioned text representations for activity recognition. In: CVPR. pp. 18547–18558.

Kong, Y., Fu, Y., 2022. Human action recognition and prediction: A survey. IJCV 130 (5), 1366–1401.

Koupaee, M., Wang, W.Y., 2018. Wikihow: A large scale text summarization dataset. arXiv:1810.09305.

Kung, C.-H., Ramirez, F., Ha, J., Chen, Y.-T., Crandall, D., Tsai, Y.-H., 2025. What changed and what could have changed? state-change counterfactuals for procedure-aware video representation learning. arXiv:2503.21055.

Lai, B., Toyer, S., Nagarajan, T., Girdhar, R., Zha, S., Rehg, J.M., Kitani, K., Grauman, K., Desai, R., Liu, M., 2024. Human action anticipation: A survey. arXiv:2410.14045.

Lee, S.-P., Elhamifar, E., 2025. Error recognition in procedural videos using generalized task graph. In: Proceedings of the International Conference on Computer Vision. pp. 10009–10021.

Lee, S.-P., Lu, Z., Zhang, Z., Hoai, M., Elhamifar, E., 2024. Error detection in egocentric procedural task videos. In: CVPR. pp. 18655–18666.

Li, W., Fan, H., Wong, Y., Kankanhalli, M., Yang, Y., 2024. Topa: Extending large language models for video understanding via text-only pre-alignment. NeurIPS 37, 5697–5738.

Li, Y., Jain, A., Bellos, F., Corso, J.J., 2025. Mistake attribution: Fine-grained mistake understanding in egocentric videos. arXiv preprint arXiv:2511.20525.

Li, J., Lei, P., Todorovic, S., 2019. Weakly supervised energy-based learning for action segmentation. In: ICCV. pp. 6243–6251.

Li, J., Li, D., Savarese, S., Hoi, S., 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International Conference on Machine Learning. PMLR, pp. 19730–19742.

Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C., 2022. Mvitv2: Improved multiscale vision transformers for classification and detection. In: CVPR. pp. 4804–4814.

Lin, C.-Y., 2004. ROUGE : A package for automatic evaluation of summaries. In: Text Summarization Branches Out. Association for Computational Linguistics, Barcelona, Spain, pp. 74–81, https://aclanthology.org/W04-1013/.

Liu, Y., Eyzaguirre, C., Li, M., Khanna, S., Niebles, J.C., Ravi, V., Mishra, S., Liu, W., Wu, J., 2024. Ikea manuals at work: 4d grounding of assembly instructions on internet videos.

Liu, D., Li, Q., Dinh, A.-D., Jiang, T., Shah, M., Xu, C., 2023. Diffusion action segmentation. In: Proceedings of the International Conference on Computer Vision. pp. 10139–10149.

Mazzamuto, M., Furnari, A., Sato, Y., Farinella, G.M., 2025. Gazing into missteps: Leveraging eye-gaze for unsupervised mistake detection in egocentric videos of skilled human activities. In: CVPR. pp. 8310–8320.

Moriwaki, K., Nakano, G., Inoshita, T., 2022. The brio-ta dataset: Understanding anomalous assembly process in manufacturing. In: IEEE ICIP. pp. 1991–1995.

Narasimhan, M., Yu, L., Bell, S., Zhang, N., Darrell, T., 2023. Learning and verification of task structure in instructional videos. arXiv:2303.13519.

Nayak, R., Pati, U.C., Das, S.K., 2021. A comprehensive review on deep learning-based methods for video anomaly detection. Image Vis. Comput. 106, 104078.

Nikolaidis, Stefanos, Zhu, Yu Xiang, Hsu, David, Srinivasa, Siddhartha, 2017. Human-robot mutual adaptation in shared autonomy. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. pp. 294–302.

Niu, Y., Guo, W., Chen, L., Lin, X., Chang, S.-F., 2024. Schema: State changes matter for procedure planning in instructional videos. arXiv:2403.01599.

Oord, A.v.d., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748.

OpenAI, 2025. Chatgpt and large language model. https://chat.openai.com/.

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318.

Patsch, C., Wu, Y., Zakour, M., Salihu, D., Steinbach, E., 2025. Mistsense: Versatile online detection of procedural and execution mistakes. In: Proceedings of the International Conference on Computer Vision. pp. 14528–14537.

Peddi, R., Arya, S., Challa, B., Pallapothula, L., Vyas, A., Wang, J., Zhang, Q., Komaragiri, V., Ragan, E., Ruozzi, N., et al., 2023. Captaincook4d: A dataset for understanding errors in procedural activities. arXiv:2312.14556.

Plini, L., Scofano, L., Matteis, E. De, di Melendugno, G.M.D., Flaborea, A., Sanchietti, A., Farinella, G.M., Galasso, F., Furnari, A., 2024. Ti-prego: Chain of thought and in-context learning for online mistake detection in procedural egocentric videos. arXiv:2411.02570.

Plizzari, C., Goletto, G., Furnari, A., Bansal, S., Ragusa, F., Farinella, G.M., Damen, D., Tommasi, T., 2024. An outlook into the future of egocentric vision: C. Plizzari et al. Int. J. Comput. Vis. 132 (11), 4880–4936.

Qian, Y., Luo, W., Lian, D., Tang, X., Zhao, P., Gao, S., 2022. Svip: Sequence verification for procedures in videos. In: CVPR. pp. 19890–19902.

Ragusa, F., Furnari, A., Farinella, G.M., 2023. Meccano: A multimodal egocentric dataset for humans behavior understanding in the industrial-like domain. Comput. Vis. Image Underst. 235, 103764.

Rawal, I.S., Matyasko, A., Jaiswal, S., Fernando, B., Tan, C., 2023. Dissecting multimodality in videoqa transformer models by impairing modality fusion. arXiv:2306.08889.

Reason, J., 1990. Human Error. Cambridge University Press.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans. PAMI 39, 1137–1149.

Saini, N., Wang, H., Swaminathan, A., Jayasundara, V., He, B., Gupta, K., Shrivastava, A., 2023. Chop & learn: Recognizing and generating object-state compositions. In: ICCV. pp. 20247–20258.

Schoonbeek, T.J., Houben, T., Onvlee, H., Van der Sommen, F., et al., 2024. Industreal: A dataset for procedure step recognition handling execution errors in egocentric videos in an industrial-like setting. In: WACV. pp. 4365–4374.

Schoonbeek, T.J., Hung, S.-H., Lehman, D., Onvlee, H., Kustra, J., de With, P.H., Van der Sommen, F., 2025. Learning to recognize correctly completed procedure steps in egocentric assembly videos through spatio-temporal modeling. Comput. Vis. Image Underst. 104528.

Seminara, L., Farinella, G.M., Furnari, A., 2024. Differentiable task graph learning: Procedural activity representation and online mistake detection from egocentric videos. arXiv:2406.01486.

Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhania, D., Wang, R., Yao, A., 2022. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In: CVPR. pp. 21096–21106.

Sener, F., Singhania, D., Yao, A., 2020. Temporal aggregate representations for long-range video understanding. In: ECCV. pp. 154–171.

Shamil, M.S., Chatterjee, D., Sener, F., Ma, S., Yao, A., 2024. On the utility of 3d hand poses for action recognition. In: European Conference on Computer Vision. Springer, pp. 436–454.

Souček, T., Alayrac, J.-B., Miech, A., Laptev, I., Sivic, J., 2022. Look for the change: Learning object states and state-modifying actions from untrimmed web videos. In: CVPR. pp. 13956–13966.

Souček, T., Damen, D., Wray, M., Laptev, I., Sivic, J., 2024. Genhowto: Learning to generate actions and state transformations from instructional videos. In: 2024 Conference on Computer Vision and Pattern Recognition. CVPR, pp. 6561–6571. http://dx.doi.org/10.1109/CVPR52733.2024.00627.

Souček, T., Gatti, P., Wray, M., Laptev, I., Damen, D., Sivic, J., 2025. Showhowto: Generating scene-conditioned step-by-step visual instructions. In: CVPR. pp. 27435–27445.

Stanescu, A., Mohr, P., Thaler, F., Kozinski, M., Skreinig, L.R., Schmalstieg, D., Kalkofen, D., 2024. Error management for augmented reality assembly instructions. In: 2024 IEEE International Symposium on Mixed and Augmented Reality. ISMAR, pp. 690–699. http://dx.doi.org/10.1109/ISMAR62088.2024.00084.

Stergiou, A., Poppe, R., 2025. About time: Advances, challenges, and outlooks of action understanding. IJCV 1–65.

Tang, Y., Bi, J., Xu, S., Song, L., Liang, S., Wang, T., Zhang, D., An, J., Lin, J., Zhu, R., et al., 2025. Video understanding with large language models: A survey. IEEE Trans. CSVT.

Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J., 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In: CVPR. pp. 1207–1216.

Tong, Z., Song, Y., Wang, J., Wang, L., 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. NeurIPS 35, 10078–10093.

Vedantam, R., Lawrence Zitnick, C., Parikh, D., 2015. Cider: Consensus-based image description evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4566–4575.

Wang, X., Farhadi, A., Gupta, A., 2016. Actions˜ transformations. In: CVPR. pp. 2658–2667.

Wang, X., Kwon, T., Rad, M., Pan, B., Chakraborty, I., Andrist, S., Bohus, D., Feniello, A., Tekin, B., Frujeri, F.V., et al., 2023. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In: ICCV. pp. 20270–20281.

Wang, Y., Li, K., Li, X., Yu, J., He, Y., Chen, G., Pei, B., Zheng, R., Wang, Z., Shi, Y., et al., 2024. Internvideo2: Scaling foundation models for multimodal video understanding. In: European Conference on Computer Vision. Springer, pp. 396–416.

Wang, X., Zhang, S., Qing, Z., Shao, Y., Zuo, Z., Gao, C., Sang, N., 2021. Oadtr: Online action detection with transformers. In: ICCV. pp. 7565–7575.

Wang, B., Zhao, Y., Yang, L., Long, T., Li, X., 2023. Temporal action localization in the deep learning era: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 46 (4), 2171–2190.

Xu, H., Ghosh, G., Huang, P.-Y.B., Okhonko, D., Aghajanyan, A., Feichtenhofer, F.M.L.Z.C., 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In: Conference on Empirical Methods in Natural Language Processing. https://api.semanticscholar.org/CorpusID:238215257.

Yang, J., Liu, S., Guo, H., Dong, Y., Zhang, X., Zhang, S., Wang, P., Zhou, Z., Xie, B., Wang, Z., et al., 2025. Egolife: Towards egocentric life assistant. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 28885–28900.

Yi, F., Wen, H., Jiang, T., 2021. Asformer: Transformer for action segmentation. arXiv:2110.08568.

Zhang, C.-L., Wu, J., Li, Y., 2022. Actionformer: Localizing moments of actions with transformers. In: ECCV. Springer, pp. 492–510.

Zhang, Z., Zhang, A., Li, M., Smola, A., 2022. Automatic chain of thought prompting in large language models. arXiv:2210.03493.

Zhao, H., Wildes, R.P., 2021. Review of video predictive understanding: Early action recognition and future action prediction. arXiv:2107.05140.

Zheng, H., Lee, R., Lu, Y., 2023. Ha-vid: A human assembly video dataset for comprehensive assembly knowledge understanding. Adv. Neural Inf. Process. Syst. 36, 67069–67081.

Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., Kehtarnavaz, N., Shah, M., 2023. Deep learning-based human pose estimation: A survey. ACM Comput. Surv. 56 (1), http://dx.doi.org/10.1145/3603618.

Zhong, Z., Martin, M., Voit, M., Gall, J., Beyerer, J., 2023. A survey on deep learning techniques for action anticipation. arXiv:2309.17257.

Zhou, K., Yang, J., Loy, C.C., Liu, Z., 2022. Learning to prompt for vision-language models. IJCV 130 (9), 2337–2348.

Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J., 2023. Object detection in 20 years: A survey. Proc. IEEE 111 (3), 257–276. http://dx.doi.org/10.1109/JPROC.2023.3238524.