applied sciences

MDPI

Article

A Neuro-Symbolic Spatiotemporal Vision Transformer for Zero-Shot Safety Behavior Recognition

Rui Song [1], Yi Zhang [1], Fangzhou Hu [1], Ankai Zhang [1,*], Zhengwei Chang [2] and Linghao Zhang [2]

[1] China Electric Power Research Institute, Beijing 100192, China; songrui@epri.sgcc.com.cn (R.S.); zhangyi2@epri.sgcc.com.cn (Y.Z.); hufangzhou@epri.sgcc.com.cn (F.H.)
[2] Power System Security and Operation Key Laboratory of Sichuan Province, State Grid Sichuan Electric Power Research Institute, Chengdu 610041, China; changzw@126.com (Z.C.); zlh.nju@gmail.com (L.Z.)
* Correspondence: zhangankai@epri.sgcc.com.cn

Abstract

This study evaluates video-level behavior recognition on the seven-class Construction Meta Action (CMA) benchmark using a neuro-symbolic spatiotemporal Vision Transformer. An MViTv2-S backbone feeds a neural classification head and eight weakly supervised predicate variables, which are scored against fixed class prototypes; a temporal extension uses six log-domain operators over four ordered segments. On 540 official test videos, the fused model achieves 93.35% Macro-AP, compared with 93.23% for the neural branch. Zero-shot recognition is evaluated as leave-one-class-out transfer within CMA: the target class is excluded from task-specific head training and model selection, while Kinetics pretraining and target-class prototypes remain available. Mean held-out AP is 47.66% for the rule branch and 44.69% for the neural branch, without a statistically established difference. Log-domain computation maintains finite outputs and gradients under the reported stress conditions. The measured offline cost is 199.36 ms/video for single-clip file processing and approximately 1.17 s/video for four-segment extraction on an RTX 3080 Ti laptop GPU. The findings support inspection of prototype-based score construction and numerical behavior within the evaluated benchmark. Predicate semantics are not independently annotated, predictions are not attributed to individual workers, and cross-site generalization and live-stream performance have not been validated.

Keywords: safety behavior recognition; neuro-symbolic AI; vision transformer; differentiable temporal logic; numerical stability; event ordering; zero-shot recognition; edge deployment

Check for updates

Academic Editors: Min Yang, Ziyu Lyu and Chengming Li

Received: 10 August 2026
Revised: 26 October 2026
Accepted: 30 October 2026
Published: 12 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Construction and electric power operations motivate research on video recognition of worker behaviors associated with safety risks [1]. A behavior label alone, however, does not establish a regulatory violation, which also depends on work authorization, protective measures, equipment state, and site context. This study examines single-label, video-level recognition on the Construction Meta Action (CMA) benchmark. Its focus is the relationship between learned video features, prototype-based class scores, and numerical evaluation of temporal evidence. The seven CMA categories define the recognition task; they do not constitute a complete taxonomy of workplace hazards. Throughout this paper, zero-shot recognition refers to target-class-held-out transfer within CMA, with the target class excluded from task-specific head training and model selection. Supervision from the remaining CMA classes, Kinetics pretraining, and predefined target-class prototypes are

still used. The experiments therefore address a bounded benchmark setting rather than the recognition of arbitrary violations at unseen sites.

Existing visual safety applications range from object detection for protective equipment [2–4] to spatiotemporal recognition of worker actions [5–8], yet these tasks differ fundamentally. Object detectors identify visible entities in individual frames, whereas action recognition integrates temporal information across sequences. Models trained on fixed category sets typically require additional supervision to accommodate new labels, and while open-vocabulary detection and vision-language learning offer alternative semantic interfaces [9,10], translating explicit action sequences or site rules into actionable decisions still necessitates a defined reasoning mechanism and appropriate evaluation protocol. Neuro-symbolic methods have sought to bridge learned representations with structured concepts and reasoning [11–16], with capabilities ranging from executing symbolic programs and constraining learning via logic to performing direct logical inference; stable log-space fuzzy reasoning has also been explored in logLTN [17]. Our contribution is therefore not the first use of symbolic reasoning or logarithmic computation, but rather a compact, task-specific integration of class-prototype scoring, an independently inspectable predicate pathway, and segment-level temporal operators, assessed through branch comparisons, held-out-class tests, and numerical stress tests.

Specifically, this study makes three contributions. First, we evaluate a video-level architecture that combines a neural classification head with an eight-predicate prototype-scoring head, both sharing an MViTv2-S backbone and fused through a learned scalar gate. Second, we implement prototype compatibility and six temporal scoring operators in the float32 log domain and examine their numerical behavior over the reported test ranges. Third, we characterize closed-set recognition, leave-one-class-out transfer, branch outputs, internal predicate interventions, and offline computational cost on CMA. Inspection is limited to the defined scoring variables and their contributions; the experiments do not establish frame-level visual grounding or independent semantic validity of the predicates.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 defines the architecture, predicate supervision, scoring functions, and temporal operators. Section 4 presents the evaluation protocols and results. Section 5 discusses the interpretation and limits of the evidence, and Section 6 concludes the study.

2. Related Works

2.1. Safety Behavior Recognition at Work Sites

Video behavior recognition has progressed from engineered motion features to learned spatiotemporal representations. I3D and SlowFast model video dynamics with convolutional architectures [5,18], whereas MViTv2 uses multiscale Transformer representations [6]. Construction-safety applications include protective-equipment detection [2,4], the Construction Meta Action benchmark and STR-Transformer [7], and power-personnel behavior recognition [8]. The benchmark in [7] is particularly relevant because it provides seven worker-action classes and public evaluation material. These studies motivate behavior recognition, but their task definitions do not by themselves establish compliance with a complete safety procedure. Recent adapted vision-language work addresses fall-hazard identification and captioning [19]; its task and outputs differ from seven-class video classification and should not be treated as a directly interchangeable benchmark.

2.2. Neuro-Symbolic AI and Differentiable Logic

Logic Tensor Networks provide differentiable first-order logic for learning and reasoning [11]. Logical Neural Networks combine logical semantics with learned parameters and support inference under incomplete information [13]. Differentiable rule learning has also been studied using noisy examples [14], while NS-VQA and the Neuro-Symbolic Concept Learner connect visual representations with programmatic or compositional reasoning [12,16]. These approaches demonstrate that symbolic components can perform more than training-time regularization. Our rule branch is a simpler, fixed-prototype scoring mechanism designed for the present video task; it is not a general first-order theorem prover.

Numerical behavior depends on the chosen operators and their implementation. Probability products can underflow, and taking a logarithm after rounded sigmoid outputs can produce non-finite values. LogLTN explicitly develops fuzzy reasoning in logarithm space [17], so log-space stability is an established methodological direction. Differentiable Temporal Logic applies temporal constraints to action segmentation [15], demonstrating that temporal reasoning is also not unique to this work. Our contribution is the implementation and evaluation of a particular combination of prototype compatibility, a neural–rule scoring interface, and a small temporal rule set. The stress tests compare the specified implementations, rather than all neuro-symbolic systems.

2.3. Zero-Shot and Open-Set Action Recognition

Zero-shot recognition transfers semantic information to classes without target-class training examples; careful separation of training categories and evaluation conditions is essential [20]. CLIP demonstrates language-based transfer through large-scale image–text pretraining [10]. VideoMAE improves video representations through self-supervised pretraining [21], but pretraining quality alone does not establish zero-shot semantic composition. Here, "zero-shot" refers to withholding a CMA target class during task-specific head training and model selection. It does not mean that related actions were absent from Kinetics pretraining. We distinguish this setting from a fully CMA-training-free Kinetics label-transfer baseline and from evaluation on a different site or dataset.

2.4. Lightweight Video Backbones and Edge Deployment

MViTv2 provides a multiscale representation for image and video tasks [6]. TimeSformer and ViViT explore alternative video-attention architectures [22,23], building on the Transformer architecture [24], while Temporal Segment Networks illustrate segment-based video modeling [25]. MobileNetV2 studies efficient image representations [26], but image-model efficiency cannot determine the cost of a multi-pass video pipeline. In this study, parameter count, single-clip model inference, file processing, and four-segment extraction are reported separately. These measurements characterize the tested offline implementation on a laptop GPU; no embedded-device optimization or live-stream evaluation is performed.

2.5. Positioning of This Work

Table 1 compares the purpose and reasoning interface of representative approaches. It is a qualitative positioning table, not an empirical ranking or a claim that an entire method family lacks a capability. The proposed system provides class-prototype compatibility and temporal evidence alongside the neural prediction. Its value is assessed through the separately reported branch outputs, numerical behavior, and target-class-held-out evaluation, rather than a claim of large closed-set gains or universal superiority.

Table 1. Qualitative positioning of representative methods. Capabilities and costs depend on implementation; no cross-method numerical-stability or deployment benchmark is implied.

Representative Work	Primary Interface	Relation to This Study
I3D/SlowFast/MViTv2 [5,6,18]	Learned video features and class scores	Provide visual representation; no claim of explicit prototype logic in the cited classifiers.
CLIP/Grounding DINO [9,10]	Text-conditioned visual recognition	Semantic transfer or localization; task and temporal outputs differ.
LTN/LNN/logLTN [11,13,17]	Differentiable logic and logical inference	Establish prior logic and log-space reasoning; no blanket claim that these lack inference or stability.
NS-VQA/concept learner [12,16]	Compositional visual reasoning	Structured visual concepts and execution in different tasks.
Differentiable Temporal Logic [15]	Temporal constraints for segmentation	Prior temporal reasoning; different task and reasoning interface.
This study	Video-level class prototypes plus temporal scores	Separate neural/rule outputs, task-specific tests, and explicitly bounded claims.

3. Materials and Methods

3.1. Overall Architecture

The model comprises two prediction heads sharing an MViTv2-S backbone pretrained on Kinetics-400 [6,27]. For the static configuration, 16 uniformly sampled frames at 224 × 224 resolution produce a 768-dimensional video feature. The backbone contains approximately 34.46 M parameters, and the complete model contains 34.76 M. The neural head predicts seven class logits; the predicate head predicts eight predicate logits. Prototype matching produces a separate rule-score vector, which can classify independently of the neural classification head while still depending on the shared visual features. Figure 1 distinguishes this pathway from the four-pass temporal extension. All rules are soft compatibility terms, not guaranteed safety constraints.

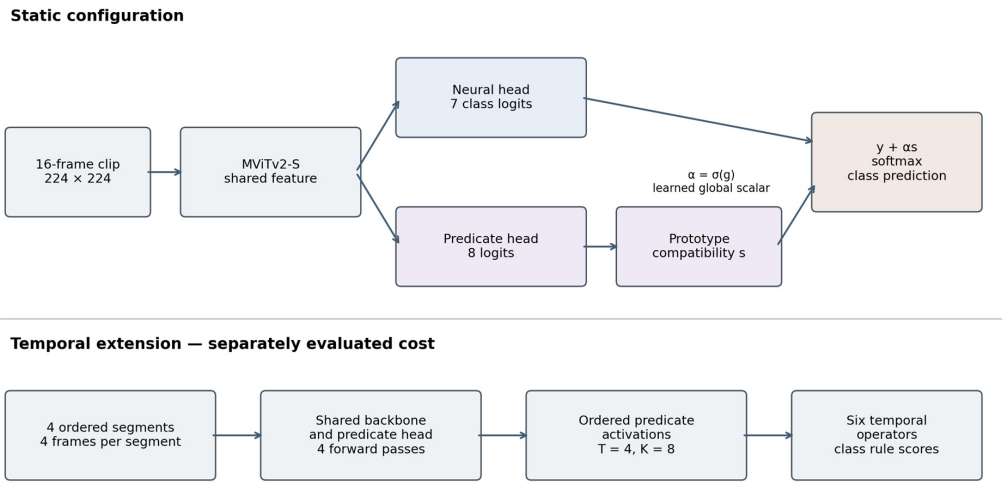


Figure 1. Neural and rule scoring pathways. The learned scalar gate weights the static rule scores before addition to the neural logits. The temporal extension uses four ordered segment-level backbone passes and fixed temporal operators; its computational cost is measured separately. Color coding: blue indicates the neural heads; purple indicates the predicate heads and prototype compatibility; orange indicates the fused prediction; gray indicates the backbone and pipeline modules.

Each head uses layer normalization, dropout, and two fully connected layers. The predicate logits are mapped to class scores through predefined binary prototypes using float32 computations. The fused scores are converted to class probabilities using softmax. Predicate activations and rule scores are available for inspection, but their names do not make them independently validated measurements of physical quantities. The evaluated task assigns one class label to each video clip in CMA. Neither the static model nor the four-segment extension produces person-indexed action labels, spatial localization outputs, or multi-label predictions.

3.2. Predicate Atomization Formalization of Security Rules

The authors defined eight predicates by decomposing the seven target behavior classes into motion, posture, interaction, and context cues. These prototypes are task-specific semantic priors, rather than rules extracted automatically from statutory safety procedures. The temporal rules were likewise specified by the authors using the intended behavior descriptions. No external expert-panel validation or independent frame-level predicate annotation was performed. Table 2 states the intended meaning of each latent variable; Table 3 gives the temporal rule combinations and their assumptions. The rules are held fixed for scoring, but this design does not demonstrate independence from all dataset-specific knowledge.

Table 2. Weakly supervised semantic definition of semantic atomic predicates.

Predicate Name	Weakly Supervised Semantic Definition
climbing_motion	The underlying visual semantics of continuing to climb upwards or extending limbs upwards
rapid_downward_motion	Potential visual semantics of rapid downward movement or out-of-control gestures of the human body
hand_to_mouth	Potential visual semantics of hands close to the mouth and nose area
upright_or_static	The underlying visual semantics of a human body remaining upright or relatively still
social_interaction	Potential visual semantics of multiple people talking or oriented towards each other in close proximity
locomotion	The underlying visual semantics of the human body's continuous positional movement
helmet_interaction	Potential visual semantics of the head region related to hard hats
unsafe_context	Potential visual semantics of high altitude, edge or other dangerous working scenarios

The Construction Meta Action (CMA) dataset [7] defines seven target behaviors, yet the semantic boundaries between predicates are inherently overlapping. An upright person may be moving, social interaction can occur during locomotion, and an unsafe context describes a condition rather than a discrete action. Crucially, the model predicts these variables from learned video features; it does not explicitly measure physical quantities such as displacement, velocity, body-keypoint distances, gaze direction, or regulatory compliance. Consequently, the semantic validity of these predicates requires evidence beyond mere agreement with class prototypes.

Table 3. Fixed temporal scoring templates. All terms use predicates in Table 2; the final column identifies modeling assumptions rather than independently validated safety requirements. The "post" term is Equation (10).

Class	Three Temporal Terms	Assumption and Limitation
Climbing	eventual(climbing_motion); always(unsafe_context); weak_precedence(climbing_motion, unsafe_context)	A context-dependent prior; persistent context makes the ordering term less informative.
Falling	consecutive(rapid_downward_motion); weak_precedence(rapid_downward_motion, upright_or_static); post(rapid_downward_motion, upright_or_static)	Post-event stillness is not proof of safety; missing views or missing subsequent state can suppress the score.
Smoking	eventual(hand_to_mouth); consecutive(hand_to_mouth); always(unsafe_context)	Hand-to-mouth motion is not unique to smoking; persistent unsafe context is a restrictive prior.
Standing	always(upright_or_static); never(locomotion); never(social_interaction)	An exclusive benchmark template; standing and interaction can coexist.
Talking	always(social_interaction); always(upright_or_static); never(rapid_downward_motion)	Coarse interaction cues do not establish speech or a safety violation.
Walking	consecutive(locomotion); never(upright_or_static); never(rapid_downward_motion)	Upright walking is physically possible; the negative posture term is a discriminative assumption.
Helmet wearing	always(helmet_interaction); always(upright_or_static); never(locomotion)	Static helmet interaction is a restricted template, not a general helmet-compliance test.

Each behavior class is represented by an eight-dimensional binary prototype vector, which serves as the basis for generating category-level weakly supervised predicate labels. Because these labels are automatically derived from class prototypes rather than independently annotated by humans, they should be interpreted at this stage as latent semantic variables that are consistent with the category prototype, not as ground-truth annotations of physical concepts.

Binary prototype encoding uses $r(c,k) = 1$ for an expected-positive predicate and $r(c,k) = 0$ for an expected-negative predicate. Both values contribute to the score, so zero is not a "don't care" state. This compact encoding creates a discriminative code for the single-label benchmark, but an expected-negative entry is a modeling assumption rather than proof that the physical concept is absent. Safety regulations do not generally require all predicates omitted from a rule to be false.

This limitation is illustrated by the case of walking while smoking: the act activates the locomotion predicate, which the smoking prototype treats as negative, thereby potentially lowering the smoking score even though the additional motion does not negate the smoking

behavior itself. Although a ternary mask could exclude irrelevant predicates and multi-label outputs could represent simultaneous actions, neither modification was evaluated in this study. The binary formulation is retained to maintain consistency with the reported experiments, and its limitations are analyzed in detail in Section 5.3.

3.3. Compatibility Calculation of Differentiable Rules

A direct product of predicate probabilities becomes increasingly susceptible to numerical underflow as the number of factors grows, and finite-precision rounding can produce exact zeros before a logarithm is ever applied. The severity of this issue depends on the specific predicate values, floating-point precision, and the exact computation path, meaning there is no universal predicate-count threshold at which gradients are guaranteed to vanish.

To address this, we evaluate explicit product/log and sigmoid-then-log baselines against a float32 log-sigmoid implementation. However, these comparisons are specific to our experimental setup; we do not generalize the observed numerical behavior to all t-norm implementations or to neuro-symbolic methods more broadly.

Let z be the predicate-logit vector, σ the sigmoid function, and $K = 8$. We define the log compatibility with the binary prototype of class c as

$$l_c(z) = \sum_{k=1}^{K} \left[r_{c,k} \, log \, \sigma(z_k) + \left(1 - r_{c,k}\right) log \, \sigma(-z_k) \right], \quad (1)$$

This is a factorized Bernoulli compatibility score. The learned predicates need not be statistically independent, so it is not an empirically calibrated joint probability of physical events. Stable log-sigmoid evaluation avoids forming a small probability product before taking its logarithm. It does not guarantee informative gradients in every parameter regime, and the numerical claims are restricted to the tests in Section 4.8.

The rule logit is normalized by the predicate count and a positive temperature:

$$s_c = \frac{1}{K\tau} l_c(z) \quad (2)$$

The temperature is $\tau = 0.75$, selected on the validation split. Division by K averages the predicate contributions; division by $\tau < 1$ sharpens their scale rather than smoothing it. Scores are non-positive but have no finite lower bound for unrestricted logits. Consequently, temperature scaling does not bound the effect of a strongly conflicting predicate. Both direct product aggregation and log-domain summation have O(K) arithmetic complexity; the contribution is numerical implementation and its measured behavior, not an asymptotic complexity improvement.

3.4. Additive Gated Fusion

This article combines neural classification scores and rule compatibility scores in the logits space:

$$y_{fused} = y_{neural} + \alpha s, \alpha = \sigma(g) \quad (3)$$

Here y_{neural} and s are C-dimensional score vectors, $C = 7$, and g is a learned scalar initialized to -1.1, giving α approximately 0.25. The same scalar applies across samples; it is not an input-dependent mechanism that detects faulty rules. Although α lies in $(0,1)$, this does not prevent rule scores from dominating when their magnitude is large. Adding scores is a product-of-experts-style heuristic after exponentiation, not a calibrated Bayesian posterior. Calibration and the empirical contribution of fusion are evaluated separately in Sections 4.4 and 4.9.

3.5. Multi-Task Balanced Loss Function

Model training uses a multi-task joint loss function. The total loss is composed of fused classification loss, neural auxiliary loss, predicate loss, rule loss and consistency regularization, weighted as follows:

$$L = L_{cls} + 0.15L_{neural} + 0.25L_{pred} + 0.15L_{rule} + 0.05L_{cons}, \quad (4)$$

The fused classification loss L_{cls} is class-weighted cross-entropy with label smoothing $\varepsilon = 0.05$. MixUp uses a mixing parameter of 0.2; class weights are proportional to the inverse square root of training frequency. L_{neural} is the auxiliary neural-head cross-entropy. L_{pred} is binary cross-entropy against the prototype-derived weak labels. L_{rule} is cross-entropy on the rule scores, enabling standalone rule-head classification. L_{cons} is the symmetric KL divergence between the neural and rule output distributions. These terms encourage classification, prototype consistency, and agreement between branches; they do not supply independent predicate supervision.

Training uses AdamW with a head learning rate of 2×10^{-4}, a backbone learning rate one tenth of that value, and weight decay 0.05. The learning rate warms up over five epochs; the backbone is frozen during the first two epochs. Exponential moving averaging uses a decay of 0.999. Batch size is one with eight-step gradient accumulation. The development schedule allows up to 60 epochs and early stopping after 15 epochs without validation Macro-AP improvement. The main closed-set experiment uses seed 17; the frozen-feature and leave-one-class-out experiments use three seeds, 17, 29, and 41. Final training on the combined development data is described in Section 4.3.

3.6. Differentiable Temporal Scoring

Static compatibility aggregates evidence over a clip and does not explicitly distinguish event order or adjacent persistence. The temporal extension predicts ordered segment-level predicates and applies six differentiable operators: eventual occurrence, always, never, weak precedence, consecutive duration, and post-hazard state evidence. The operators supply soft scores, not proofs that a physical safety procedure has been satisfied.

For the temporal configuration, each video is divided into $T = 4$ non-overlapping intervals. Each interval is processed separately by the shared backbone and predicate head to obtain ordered predicate logits. This requires four backbone evaluations and differs from the single-pass static configuration. The choice of $T = 4$ fixes the temporal discretization and computational budget for the reported evaluation; it has not been selected through a recognition-accuracy sensitivity study. Events within one segment cannot be temporally ordered by these operators, and a two-segment persistence term does not correspond to a fixed physical duration across clips. Stress tests at larger T assess numerical behavior, not the adequacy of temporal resolution for recognizing actions.

Let $p(t,k) = \sigma(z(t,k))$, and let ℓ denote a log-domain score. For readability, the equations below use probabilities; implementation evaluates the corresponding logarithmic expressions in float32. These are differentiable fuzzy aggregations. In particular, overlapping candidate events are not independent, so the noisy-OR aggregations should not be read as calibrated event probabilities.

Eventual occurrence aggregates evidence for predicate p in at least one segment:

$$l(\Diamond p) = log\left[1 - \prod_{t=1}^{T}\left(1 - p_{t,p}\right)\right] \quad (5)$$

For $a = \Sigma_t \log(1 - p(t,p)) \leq 0$, evaluate $\log(1 - \exp(a))$ as $\text{log1p}(-\exp(a))$ when $a \leq -\log 2$ and as $\log(-\text{expm1}(a))$ otherwise. The reported implementation uses a probability floor of 10^{-7} for complement terms. Such safeguards avoid undefined logarithms in the tested range; they are numerical approximations and do not establish universal stability.

Always aggregates positive evidence across all segments:

$$l(\Box p) = \sum_{t=1}^{T} log\, p_{t,p} \tag{6}$$

Never aggregates negative evidence across all segments:

$$l(\neg \Diamond p) = \sum_{t=1}^{T} log\left(1 - p_{t,p}\right) \tag{7}$$

Weak precedence aggregates evidence for p at time t and q at time t or later. Define $Q(t)$ as the noisy-OR aggregation of q over the suffix beginning at t. Because the suffix includes t, the operator admits same-segment co-occurrence; it does not enforce strictly earlier onset:

$$l(p \preccurlyeq q) = log\left[1 - \prod_{t=1}^{T}\left(1 - p_{t,p}Q_t\right)\right], Q_t = 1 - \prod_{u=t}^{T}\left(1 - p_{u,q}\right) \tag{8}$$

Each factor is close to one when either $p(t,p)$ or $Q(t)$ is small. The outer complement aggregates evidence from candidate start positions. When q is persistent, $Q(t)$ approaches one, and this expression approaches eventual occurrence of p. The operator then supplies no additional ordering information. We retain this formulation to describe the reported method accurately, and avoid interpreting it as strict temporal precedence.

Consecutive duration aggregates evidence from adjacent pairs. For two consecutive segments:

$$l(\triangle p) = log\left[1 - \prod_{t=1}^{T-1}\left(1 - p_{t,p}p_{t+1,p}\right)\right] \tag{9}$$

Post-hazard state evidence aggregates all strictly ordered pairs of hazard h and subsequent state s:

$$l(h \rightsquigarrow s) = log \sum_{t=1}^{T-1} \sum_{u=t+1}^{T} p_{t,h}p_{u,s} \tag{10}$$

The nested logsumexp computes the logarithm of the sum of pairwise products over $t < u$, functioning as an unnormalized evidence score rather than a log probability. This value can be positive; for instance, when all predicates equal one, the evidence mass is $T(T-1)/2$ and the score is its logarithm. The scale depends on T, which remains fixed in the reported configuration. Furthermore, the label "recovery" denotes only the intended post-event pattern; the predicate upright_or_static does not establish that a person is uninjured or has returned to safety.

Analytical examples clarify these distinctions, representing evaluations of displayed formulas for specified inputs rather than additional video experiments. For h = (0.10, 0.55, 0.90, 0.60) and s = (0.05, 0.10, 0.30, 0.85), Equation (10) sums six pairwise products to 1.5225, yielding a log score of approximately 0.4204. The largest single pair contributes 0.765 but does not represent the aggregate probability. If subsequent-state activations approach zero, the unfloored score approaches negative infinity. While a static posture after a fall may support a post-event pattern, neither its presence nor absence proves recovery or rules out a fall. Similarly, for sequences p and q, exchanging them changes the exponentiated weak-precedence score from approximately 0.9796 to 0.3700. Although the operator differentiates these cases analytically, this contrast does not establish a recognition-accuracy gain on actual videos.

Persistent inputs expose further boundaries of the temporal operators. If q is one in every segment, weak precedence reduces to the eventual occurrence of p. If both h and s persist, the post-hazard score increases with the number of admissible pairs even though no transition has been isolated. These properties characterize the operators as soft evidence aggregators rather than complete temporal specifications. Adopting an onset-based or normalized alternative would fundamentally change the method and require separate evaluation.

The consecutive operator used here is consecutive (p, 2): it aggregates evidence for p in at least one adjacent pair of segments, as defined in Equation (9). The second argument denotes two adjacent segments, not two frames or a duration in seconds. No longer-duration consecutive operator is evaluated in this study.

The templates in Table 3 use occurrence or persistence of the class-associated cue, supplemented by context, posture, or exclusion terms; the falling template additionally includes subsequent-state evidence. Each template contains three equally weighted terms, giving the same nominal term count and weighting scheme across classes. The terms describe the authors' fixed modeling choices, not empirically optimized or independently validated temporal definitions of the behaviors. Equal term counts do not equalize their numerical scales, particularly because the post-hazard term is unnormalized.

The eventual, always, never, and consecutive operators have O(T) cost. The explicit pair-based implementation of weak precedence and post-hazard evidence has O(T^2) cost. The fixed $T = 4$ configuration has small rule overhead, but requires four backbone passes. The numerical stress results in Section 4.8 concern the specified implementations and tested ranges.

4. Experiment

4.1. Dataset and Leak-Free Partitioning

We utilize the public Construction Meta Action (CMA) dataset [7], which comprises 1595 video clips across seven behavior classes. For subgroup reporting, climbing, falling, and smoking are designated as risk-related classes, while standing, talking, walking, and helmet wearing serve as comparison classes. It is important to note that these benchmark labels do not necessarily establish a contextual safety violation in every real-world scene. To prevent data leakage from derived clips originating from the same source video, data splits are organized by source group: the training set contains 835 videos from 167 groups, validation includes 220 videos from 44 groups, and the official test split consists of 540 videos from 108 groups. These source groups are strictly disjoint across splits, addressing source-video leakage, though not all possible sources of distributional dependence.

Model selection, early stopping, and hyperparameter tuning (including temperature, threshold, and fusion weights) were conducted exclusively on development data; official test labels were never used for these purposes. Test predictions were reserved solely for final metrics, uncertainty estimates, and diagnostic analyses after the configuration was fixed. The leave-one-class-out experiments follow a separate development-set protocol where model selection excludes the held-out class, and the 220-video evaluation pool is used only for final class-wise AP. These two evaluation pools are distinct and not interchangeable.

For video-level performance intervals, the resampling unit is the source group to retain dependence among its derived clips. Development analyses employ 500 bootstrap draws over 44 groups, while official test analyses use 2000 draws over 108 groups. Held-out-class comparisons summarize results across seven classes, reporting class-resampling intervals and paired tests. These methods quantify different sources of uncertainty: a source-group bootstrap does not capture variation across independently trained models, and the limited number of classes restricts precision for transfer comparisons.

4.2. Experimental Results

Accuracy is the fraction of correctly classified clips. Macro-AP is the unweighted mean of one-versus-rest average precision over the seven classes; Macro-F1 is the mean of per-class F1. The class-prior AP baseline uses each class prevalence in the evaluation pool. ECE summarizes the difference between binned confidence and empirical accuracy; the Brier score and negative log-likelihood additionally assess probabilistic prediction [28].

Percentage-point differences are distinguished from relative percentage changes. In the method, $C = 7$, $K = 8$, $T = 4$, σ is the sigmoid, $\tau = 0.75$ is the rule temperature, α is the learned scalar static gate, and β denotes the separately selected temporal-combination weight. Table 4 reports the validation set classification performance.

Table 4. Validation set classification performance.

Category	Precision/%	Recall/%	F1/%	AP/%	Number of Samples
Climb	100.00	100.00	100.00	100.00	20
Fall	100.00	88.89	94.12	97.84	45
Smoking	100.00	88.89	94.12	98.78	45
Stand	75.00	53.33	62.34	75.25	45
Chat	63.89	92.00	75.41	91.44	25
Walk	52.50	70.00	60.00	64.48	30
Wear a safety helmet	83.33	100.00	90.91	100.00	10
Overall	-	-	82.41	89.68	220
Three types of difficulty	-	-	96.08	98.87	110

On the 540 official CMA test videos [7], the fused model achieves 93.35% video-level Macro-AP. The corresponding neural branch achieves 93.23%, a descriptive difference of 0.12 percentage points. Published results from STR-Transformer [7] provide benchmark context, while the adapted vision-language study in [19] addresses fall-hazard identification and captioning. Because these external methods were not reimplemented under the present evaluation protocol, their published metrics are not used to establish superiority, broader monitoring coverage, or a model-efficiency advantage. The contribution assessed here is the behavior of an explicit prototype-scoring pathway, its numerical implementation, and target-class-held-out transfer within CMA.

4.3. Ablation Experiment and Zero-Sample Verification

After development-stage model selection, the epoch-25 checkpoint is trained further to a fixed epoch 30 using all 1055 development videos. No official test label is used to select this schedule. Single-clip inference on the 540 test videos gives 92.96% accuracy, 90.94% Macro-F1, and 93.35% Macro-AP; the three risk-related classes average 98.91% AP. The source-group bootstrap 95% confidence interval for Macro-AP is [87.77%, 98.38%]. Aggregating predictions by source group yields 95.56% Macro-AP, a different evaluation unit; the main result remains video-level Macro-AP. Table 5 reports the class-level results on the official test set, and a comparison with published methods is given in Table 6.

The quantitative comparisons in this study use the internally evaluated backbones and branch outputs under the same protocol (Table 7). STR-Transformer [7] provides the source benchmark and relevant methodological context, while the adapted vision-language model in [19] addresses fall-hazard identification and captioning. Neither external method was reimplemented under our preprocessing and evaluation procedure. We therefore do not use their published results to claim a numerical advantage on CMA. The controlled comparison within this study is the S3D/MViTv2-S evaluation in Table 8.

Table 5. Per-class performance on the official CMA test set (540 videos).

Category	Precision/%	Recall/%	F1/%	AP/%	Number of Samples
climb	95.56	95.56	95.56	96.73	45
fall	100.00	99.00	99.50	100.00	100
smoking	98.68	100.00	99.34	99.99	150
stand	83.02	88.00	85.44	90.21	100
chat	83.33	66.67	74.07	74.24	45
walk	85.25	94.55	89.66	93.21	55
Wear a safety helmet	97.56	88.89	93.02	99.08	45
Macro	91.91	90.38	90.94	93.35	540

Table 6. Comparison with public methods.

Method	Years	Index Caliber	Macro-AP/%
STR-Transformer [7]	2023	Public CMA Test AP	88.70
DA-VLF [19]	2026	Public CMA Test AP	95.44
This article MViTv2-S fusion	2026	Official test Macro-AP	93.35

Table 7. Published benchmark context. Metric labels are retained from their sources; these results do not establish same-protocol superiority.

Method	Evaluation/Reported Metric	Value (%)
STR-Transformer [7]	Published AP; not reimplemented here	88.70
This study, MViTv2-S fusion	Video-level Macro-AP; 540 test videos	93.35

Table 8. Comparison of backbones and prediction branches under identical split, sampling, preprocessing, and metric conditions.

Backbone	Branch	Accuracy/%	Macro-F1/%	Macro-AP/%	Difficulty AP/%
MViTv2-S	neural	92.96	90.99	93.23	98.93
MViTv2-S	rule	92.41	90.21	93.33	98.72
MViTv2-S	fused	92.96	90.94	93.35	98.91
S3D	neural	77.22	73.85	77.19	87.95
S3D	rule	77.41	73.77	79.00	89.61
S3D	fused	77.04	73.57	77.68	88.39

4.4. Calibration and Deployment Performance

On the 220-video validation set, the rule branch achieves a Macro-AP of 91.45%, outperforming both the neural branch (89.01%) and the fused model (89.68%). The deployment measurements of the complete model are summarized in Table 9. However, fusion attains the highest accuracy and Macro-F1, demonstrating that ranking quality and top-class decision performance do not necessarily favor the same branch. It is important to clarify

that "rule-only" refers to standalone classification by the rule head and does not imply independence from learned visual features or class-derived supervision.

Table 9. Deployment performance (NVIDIA GeForce RTX 3080 Ti Laptop GPU, NVIDIA Corporation, Santa Clara, CA, USA).

Index	Result
Total parameters	34.76 M
Deployment model file size	132.61 MB
Pure model median calculation latency	31.78 ms
End-to-end file latency	199.36 ms/video
Pure model throughput	31.47 videos/second
End-to-end throughput	5.02 videos/second
Reserved video memory for peak inference	0.523 GB

Table 8 compares backbones and branch outputs under identical conditions, including split, single-clip sampling, preprocessing, metrics, and source-group bootstrap. The fused MViTv2-S model exceeds the fused S3D model by 15.67 percentage points in Macro-AP, indicating that the backbone accounts for most closed-set performance gains. Within the MViTv2-S architecture, fusion surpasses the neural branch by only 0.12 points, whereas on S3D, the rule branch exceeds fusion by 1.32 points. These results suggest that the learned gate does not consistently select the optimal branch for Macro-AP. All reported within-backbone differences are descriptive; no statistically established gain is claimed.

4.5. Error Analysis and Interpretability

Frozen 768-dimensional features from the epoch-25 main model are used to retrain the prediction heads and gate with seeds 17, 29, and 41 (Table 10). This measures head-level sensitivity conditional on a fixed learned representation, rather than the contribution of each component to full end-to-end training. Rule-only Macro-AP is 91.54% ± 0.09%, neural-only is 90.38% ± 0.31%, and full fusion is 90.51% ± 0.40%. A separately examined probability mixture with rule weight 0.75 reaches 90.90% ± 0.71%; it is a development-set diagnostic, not an additional held-out test improvement. Figure 2 displays the existing ablation results.

Table 10. Multi-task loss ablation under frozen features (3 random seeds).

Set Up	Macro-AP/%	Macro-F1/%	Difficulty AP/%	ECE/%
neural_only	90.38 ± 0.31	79.90	98.59	7.19
rule_only	91.54 ± 0.09	81.08	98.63	13.58
classification_only	90.41 ± 0.50	79.89	98.52	7.11
full	90.51 ± 0.40	79.16	98.58	7.50
without_neural_aux	90.07 ± 1.02	78.47	98.39	7.46
without_predicate	90.15 ± 1.05	78.33	98.42	8.00
without_rule	90.48 ± 0.37	79.51	98.57	7.27
without_consistency	90.08 ± 1.01	78.47	98.41	7.87

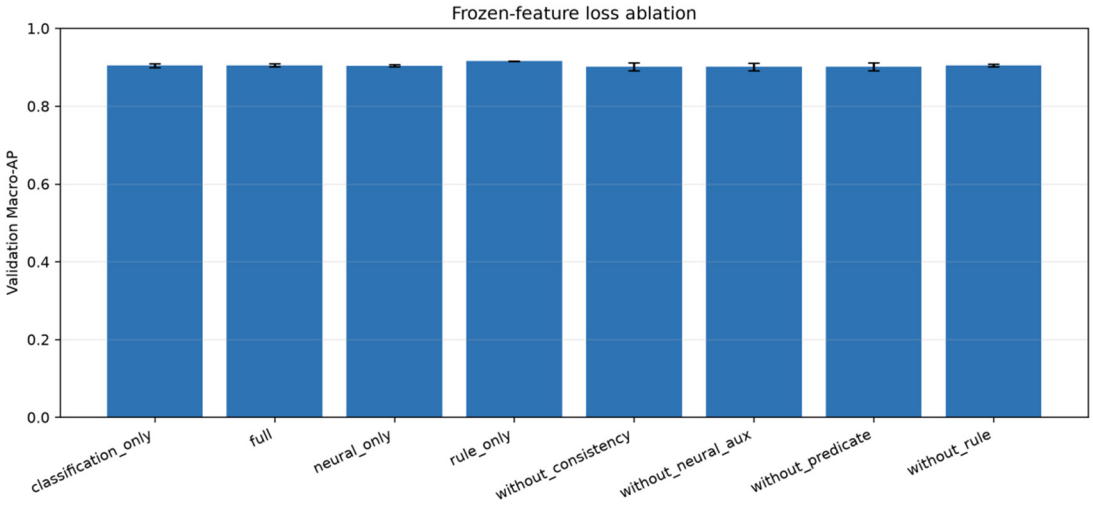


Figure 2. Freeze feature multi-task loss ablation results.

The ablations do not establish that every auxiliary term improves every metric. For example, omitting the rule loss gives 90.48% Macro-AP versus 90.51% for full fusion, while Macro-F1 rises from 79.16% to 79.51% and ECE decreases from 7.50% to 7.27%. Several differences are small relative to the reported seed variation. The evidence supports inspecting the branches separately and further studying fusion, rather than claiming that the complete loss is uniformly optimal.

4.6. Innovation Point Effect Verification

4.6.1. Numerical Stability and Complexity of Regular Operators

On an NVIDIA GeForce RTX 3080 Ti Laptop GPU (NVIDIA Corporation, Santa Clara, CA, USA), 30 independent stress tests were conducted on the predicate number $K \in \{8,16,32,64,128\}$, logit standard deviation 2 to 32, and FP16/FP32 combination. When FP16 and $K = 8$, the traditional product t-norm only obtains limited gradient in 10% of the experiments when the standard deviation is 2, and drops to 0 when the standard deviation is not less than 4; the method of first sigmoid and then taking log also fails from standard deviation 4. In this article, the forward value and gradient finite rate of float32 log-sigmoid are 100% in all combinations. In real training, NaN appeared in the 5th to 6th round of the old implementation. After the repair, new NaNs did not appear in the 7th to 40th round. The delay test also shows that the three operators all grow approximately linearly with K. Therefore, the operator contribution of this work is positioned as a numerically stable O(K) logarithmic domain implementation, rather than changing the asymptotic complexity of the product t-norm itself. The specific results are shown in Table 13 and Figure 8.

4.6.2. Gated Fusion and Multi-Task Loss Ablation

In order to reduce the cost of complete video retraining and ensure that the visual representation is completely consistent, the frozen 768-dimensional features extracted by the epoch 25 main model are used to reinitialize and train only the classification head, predicate head and rule gate. Using 3 random seeds, this protocol can isolate the head and loss effects. The Macro-AP of the pure regular head is 91.54% ± 0.09%, which is higher than the 90.38% ± 0.31% of the pure neural head; the complete fusion is 90.51% ± 0.40%. The best offline strategy is a probability mixture with a rule weight of 0.75, reaching 90.90% ± 0.71%. The specific results are shown in Table 10 and Figure 2.

4.6.3. Predicate Consistency and Counterfactual Faithfulness (Counterfactual Faithfulness)

Against category-prototype weak labels, the eight predicates average 92.90% AP (Table 11; Figure 3). The positive-label counts correspond to the 220-video validation pool. This measures consistency with the imposed prototype labels, not agreement with independently annotated physical events. In particular, high AP may reflect successful category coding without demonstrating a transferable, disentangled visual concept.

Table 11. Predicate-by-predicate performance under category prototype weak label.

Predicate	Positive Sample	Precision/%	Recall/%	F1/%	AP/%
climbing_motion	20	90.91	100.00	95.24	100.00
rapid_downward_motion	45	100.00	91.11	95.35	98.28
hand_to_mouth	45	100.00	88.89	94.12	97.34
upright_or_static	80	87.06	92.50	89.70	96.63
social_interaction	25	57.14	96.00	71.64	91.19
locomotion	30	52.63	66.67	58.82	60.27
helmet_interaction	10	66.67	100.00	80.00	100.00
unsafe_context	110	100.00	87.27	93.20	99.51

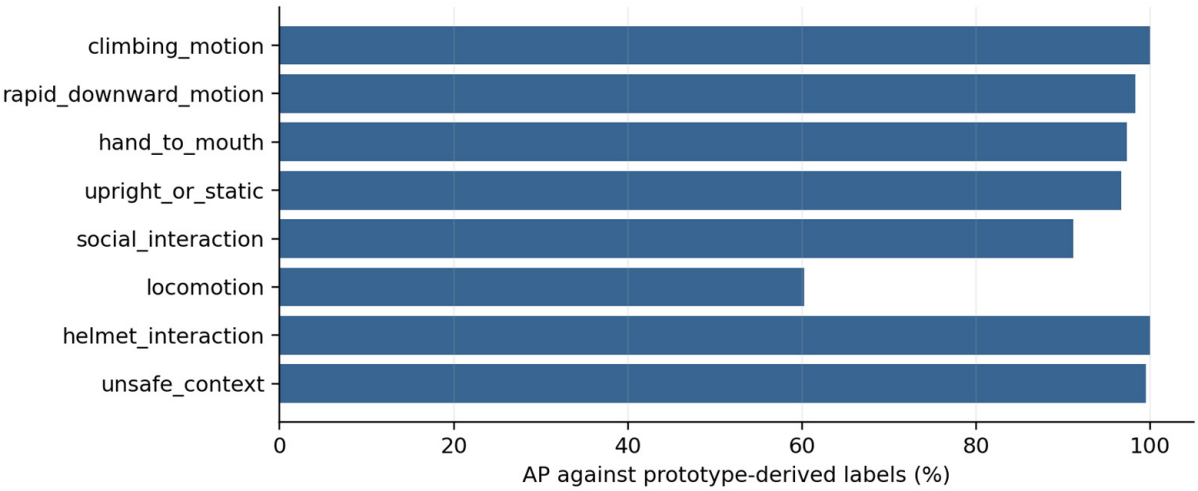


Figure 3. Per-predicate AP against prototype-derived validation labels. The labels are not independent manual annotations.

The recorded counterfactual analysis changes the internal predicate representation while holding the neural logits fixed. For a clip of class c, all predicate logits with $r(c,k) = 1$ are negated, zk → $-z_k$; the fused output is then recomputed. The control negates the same number of randomly selected prototype-zero logits. The recorded mean decrease in true-class probability is 5.99% for the targeted intervention and 1.30% for the control, with a reported prediction-change rate of 4.55% (Figure 4). This diagnostic shows sensitivity to the specified internal variables. It does not intervene on the physical video scene, establish semantic validity, or prove that the same effect would occur under a real-world action change. The modest prediction-change rate is consistent with strong neural-branch influence.

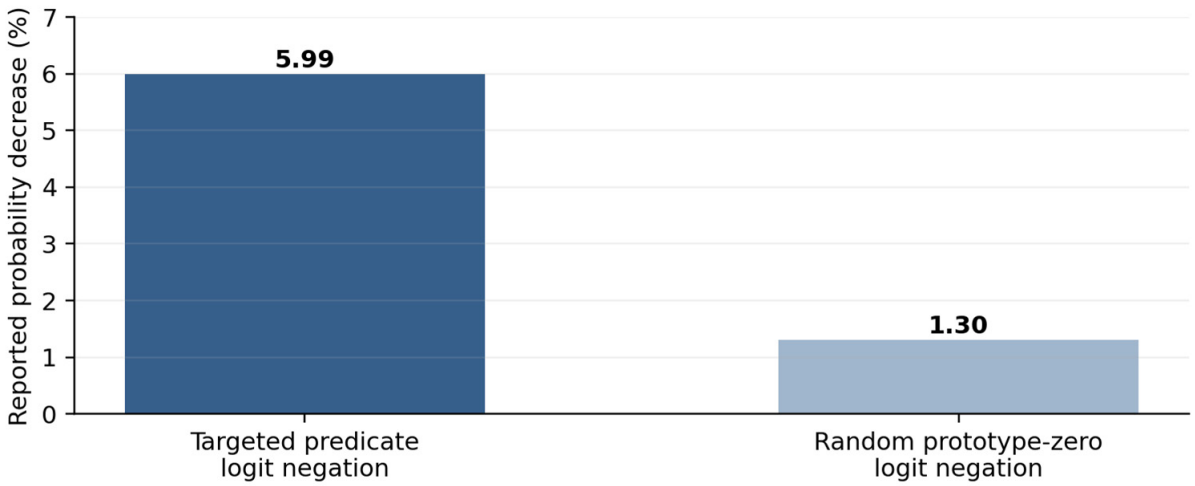


Figure 4. Recorded predicate-logit intervention summary. The targeted and control mean probability decreases are 5.99% and 1.30%, respectively, as reported in the existing experiment. This is an internal intervention, not a physical counterfactual video.

4.6.4. Strict Leave-One-Out Zero-Sample Experiment

In the strict leave-one-out experiment, it is necessary to ensure that the frozen backbone has never been exposed to the CMA training label, and the left-out category does not participate in head training and early stopping. Each CMA class contains at least one exclusive predicate that never takes a positive value for other classes, so the number of positive training samples for this predicate after leaving out the class is 0. Among the average results of the seven categories, the AP reserved for the rule/neural/fusion branches are 47.66%/44.69%/45.89% respectively; the average semantic ranking of the correct category prototype of the rule branch is 1.76. The rule ranks first for correct prototypes on falling, smoking, standing, and hard hats, and has limited effectiveness on climbing, talking, and walking. The specific results are shown in Table 12 and Figure 5.

Table 12. Leave-one-class-out AP and reported prototype rank (three seeds; mean ± standard deviation). Evaluation uses the 220-video pool.

Set Aside Class	A Priori/%	Neural AP/%	Rule AP/%	Fusion AP/%	Rule Prototype Ranking
climb	9.09	16.10 ± 7.34	11.67 ± 0.77	15.48 ± 5.58	2.00
fall	20.45	74.33 ± 3.52	71.26 ± 3.96	77.43 ± 2.41	1.00
smoking	20.45	94.79 ± 0.42	91.25 ± 1.53	95.10 ± 0.39	1.00
stand	20.45	22.61 ± 3.27	49.36 ± 2.58	23.83 ± 3.74	1.00
chat	11.36	10.33 ± 0.20	17.33 ± 2.52	11.05 ± 0.18	3.00
walk	13.64	12.62 ± 0.43	19.27 ± 0.22	13.93 ± 0.44	3.33
Wear a safety helmet	4.55	82.04 ± 19.23	73.48 ± 14.90	84.42 ± 16.60	1.00



Figure 5. Leave-one-out zero-sample AP and category prior baseline.

4.7. Held-Out-Class Transfer and Fixed Rank Fusion

The primary transfer protocol withholds a target CMA class from task-specific head training and model selection. It still permits supervision from the other six CMA classes and uses a Kinetics-pretrained backbone. This is target-class-held-out transfer under a shared dataset distribution, not recognition with no CMA supervision whatsoever. Class descriptions and prototypes are semantic side information. We separately report a Kinetics label-transfer baseline with no CMA fitting, and its fixed rank combination with the held-out rule branch.

For each held-out class, the backbone is frozen at its Kinetics-400 pretrained weights and has not been fine-tuned on CMA. The head is trained on the other six classes; early stopping uses only their validation samples. Held-out labels are used only for final AP evaluation in the 220-video pool. Across three seeds, mean held-out AP is 47.66% for the rule branch, 44.69% for the neural branch, and 45.89% for fusion (Table 12; Figure 6). The 2.97-point rule-versus-neural difference has a paired class-bootstrap 95% interval of $[-1.92\%, 8.30\%]$, so superiority is not established. For smoking, neural AP is 94.79% versus 91.25% for the rule branch, consistent with useful pretrained visual structure. The recorded mean prototype rank is 1.76 out of seven; the correct prototype ranks first for falling, smoking, standing, and helmet wearing. The neural held-out output is an auxiliary diagnostic and should not be equated with an independently specified semantic zero-shot baseline.

For the Kinetics label-transfer baseline, fixed source-label scores are mapped to target class names without fitting to CMA labels. Classes without a mapping, falling and helmet wearing, receive constant scores and hence prevalence-level AP. This baseline achieves 43.67% Macro-AP. Only this component is CMA-training-free; the rule scores used in the fixed rank combination are the held-out-class scores reported in Table 12.

Each class score is converted to its percentile rank across the evaluation pool, and the rule and Kinetics ranks are averaged with a fixed 0.5:0.5 weight. The resulting 58.17% Macro-AP describes a combination of held-out-class rule transfer and a training-free source classifier. It is not evidence that the entire system was trained without CMA data. Percentile normalization uses the unlabeled evaluation pool and is therefore a batch/transductive ranking procedure rather than an independently calibrated per-video predictor. The rule branch exceeds prevalence for all seven classes: its 47.66% mean AP exceeds 14.29% by 33.37 percentage points, with the reported class-bootstrap interval [13.71%, 54.76%] and one-sided sign-test $p = 0.0078$. The reported prototype-rank permutation test gives

$p = 0.0011$. These statistics concern the rule component and its semantic ranking; they are not tests of a 58.17% fully training-free model.

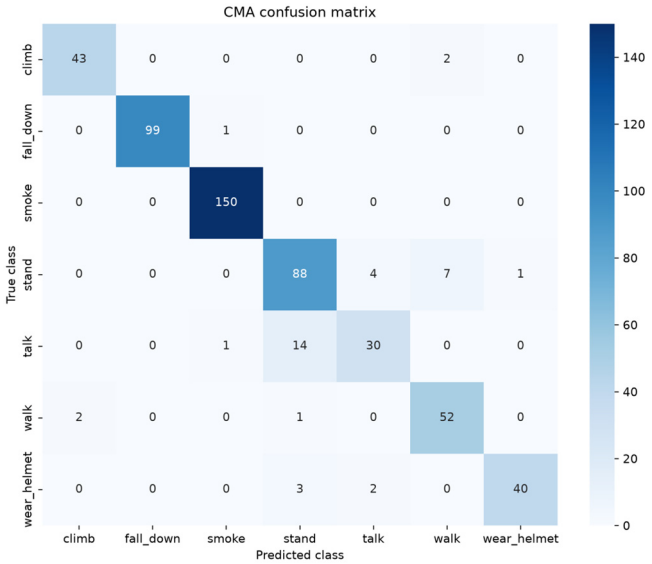


Figure 6. Leave-one-class-out AP and prevalence baseline. Each target class is excluded from head training and model selection; the backbone is frozen at Kinetics pretraining.

The fixed rank combination exceeds the rule component by 10.51 percentage points, but its reported interval $[-2.59\%, 27.81\%]$ includes zero, and the one-sided Wilcoxon p-value is 0.1562. Its comparison with Kinetics transfer gives $p = 0.0781$. Neither comparison establishes an improvement at the 0.05 level. Figure 7 retains the recorded class scores; differences calculated from displayed means are subject to rounding.

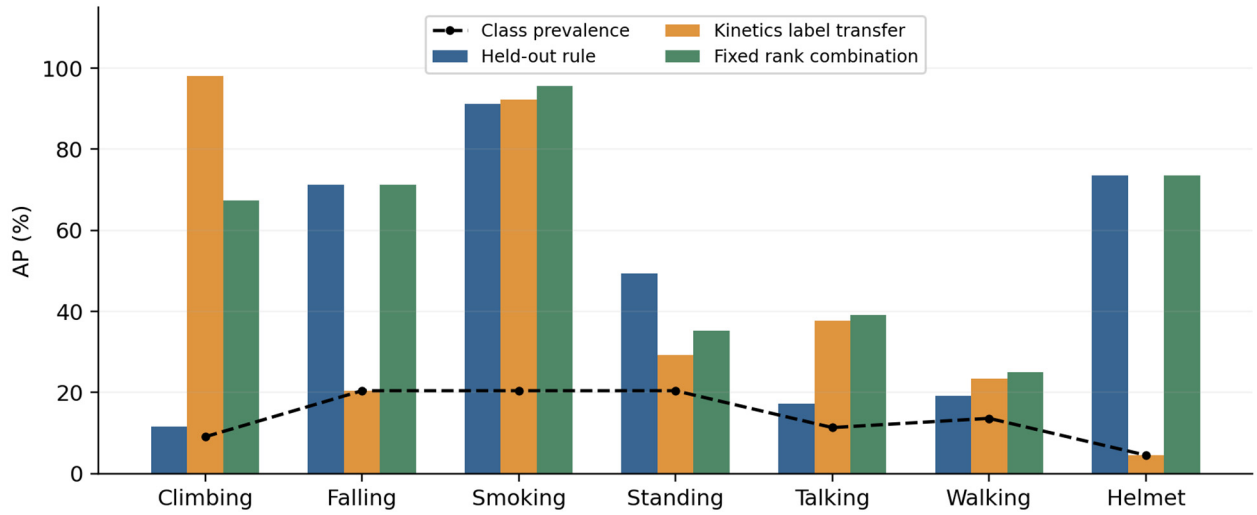


Figure 7. Held-out rule scores, training-free Kinetics label transfer, and their fixed rank combination. "Training-free" applies only to the Kinetics component.

Statistical interpretation. Intervals and tests are reported for the comparisons for which they are available. Other numerical differences are descriptive. Failure to reject a difference is not evidence of equivalence. The class-level tests involve only seven classes and should be read alongside the class-wise results; bootstrap intervals over test source groups do not replace repeated model training.

4.8. Numerically Stable Differentiable Sequential Logic

Static-operator stress tests were conducted on an RTX 3080 Ti laptop GPU using 30 trials per condition, with predicate counts K in {8, 16, 32, 64, 128}, logit standard deviations from 2 to 32, and both FP16 and FP32 inputs. Table 13 highlights the $K = 8$ FP16 comparison, where the existing product-based baseline yields finite gradients in only 10% of trials at standard deviation 2 and fails completely from 4 upward; the sigmoid-then-log method similarly fails from 4 upward. In contrast, float32 log-sigmoid evaluation maintains finite outputs and gradients across the entire tested grid. Training traces confirm this stability: NaNs occurred at epochs 5–6 under the earlier implementation but ceased entirely over epochs 7–40 after adopting the log-domain approach. This serves as an implementation diagnostic rather than a universal claim about all product t-norm systems. Operator timing scales approximately linearly with K, as shown in Figure 8.

Table 13. Finite gradient rate of FP16 rule operator ($K = 8$, 30 trials).

Logit Standard Deviation	Product t-Norm/%	Sigmoid-Log/%	This Article Log-Sigmoid/%
2	10.0	100.0	100.0
4	0.0	0.0	100.0
8	0.0	0.0	100.0
16	0.0	0.0	100.0
32	0.0	0.0	100.0

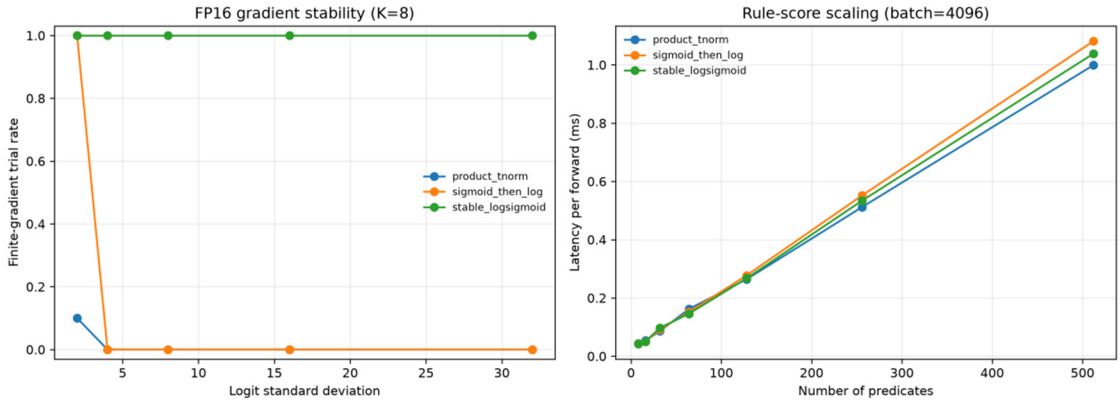


Figure 8. FP16 stability of regular operators and predicate number expansion delay.

Temporal stress tests evaluated sequence lengths T in {4, 16, 64, 256} and logit standard deviations {2, 4, 8, 16, 32} across FP16/FP32 precisions, totaling 40 combinations with ten forward/backward trials each. The float32 log-domain evaluation achieved 100% finite forward values and gradients throughout the grid, whereas the naive probability-space implementation averaged only 6.50% (FP16) and 12.50% (FP32) finite gradients, dropping to zero at T = 256 and scale 32. Table 14 lists the temporal-logic experimental hyperparameters. These stress-grid delays are distinct from per-video deployment timings at $T = 4$. We characterize these findings as a substantial reduction in numerical instability under the tested conditions—covering sequence lengths 4–256, logit scales 2–32, and both precisions—rather than a general resolution of t-norm numerical collapse for all possible operating conditions.

Table 14. Temporal logic experimental hyperparameters.

Project	Set Up	Selection Basis
Number of ordered segments/number of frames per segment	4/16	fixed agreement
Fragments overlap	0	Avoid duplication of evidence
Logic temperature τ	0.75	Inherit static rules
Closed set fusion α	0.45	Validation set 0~1, step size 0.05
Timing loss weight λ_T	0.1/0.25/0.5	See only class validation sample selection
optimizer/learning rate/decay	AdamW/10^{-3}/10^{-4}	Freeze feature head training
Batch size/maximum number of rounds/early stop	64/250/35	fixed agreement
random seed	17, 29, 41	Report means and standard deviations

Category-level activations (Figure 9) provide a descriptive view of temporal head behavior. For falling, rapid_downward_motion rises from 0.582 in segment 1 to 0.848 in segment 4; for climbing, climbing_motion decreases from 0.926 to 0.785. Intermediate values are neither reported nor interpolated. Smoking and standing remain near 1.0 across segments, talking hovers around 0.64, walking ranges from 0.602 to 0.765, and helmet wearing stays between 0.775 and 0.807. These summaries reflect activation patterns only and do not constitute independently labeled temporal events or single-video visual explanations.

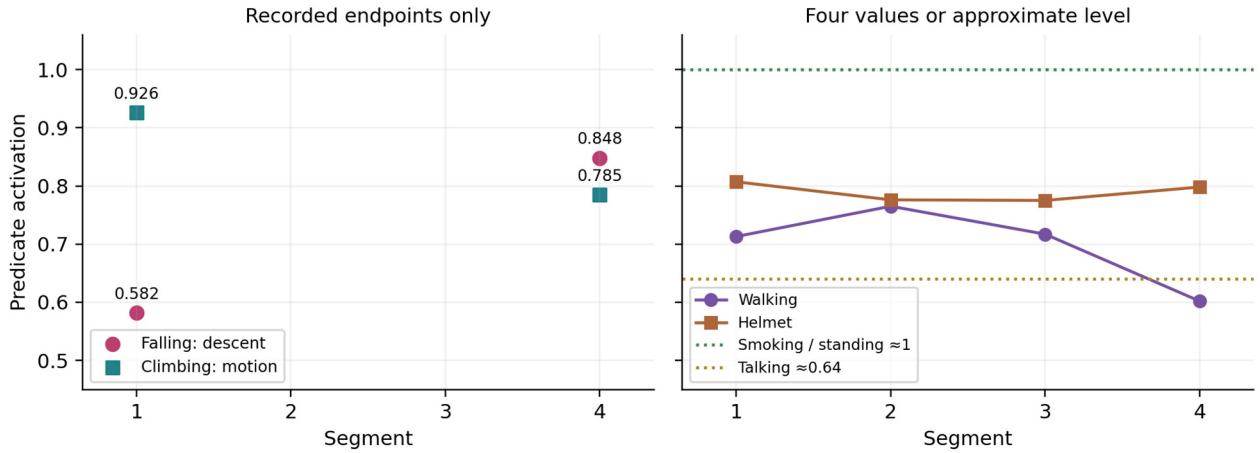


Figure 9. Reported segment-level predicate activations. Falling and climbing show endpoints only. Smoking, standing, and talking are approximate summary levels; walking and helmet wearing use the four recorded values. No unreported intermediate endpoint values are inferred. Both panels share the same vertical axis scale.

At CPU batch size 64 and $T = 4$, median static-rule and temporal-rule evaluation times are 0.163 ms and 1.502 ms per batch, respectively. The increment is approximately 0.021 ms per video and adds no learned parameters. Four-segment extraction for 540 videos takes 633.14 s, approximately 1.17 s per video, using a single process and workers = 0. This

includes decoding, preprocessing, and four backbone passes. It is an offline extraction measurement, not a streaming throughput benchmark. Feature caching and segment batching are possible engineering improvements but were not measured here.

4.9. Calibration, Deployment, and Error Analysis

On the official test set, the model achieves an Expected Calibration Error (ECE) of 3.64%, a multiclass Brier score of 0.1213, and a negative log-likelihood of 0.2985. Source-group bootstrap intervals are [89.07%, 96.67%] for accuracy and [84.96%, 95.73%] for Macro-F1. These metrics describe performance on the tested distribution and checkpoint only; they do not establish reliability under site shift or justify a safety-critical alarm threshold. Class-specific operating thresholds and escalation policies require separate validation for the intended deployment environment.

Table 15 reports computational measurements obtained on an RTX 3080 Ti laptop GPU. At batch size one, single-clip model latency is 31.78 ms (equivalent to 31.47 clips/s), with peak GPU memory usage of approximately 0.523 GB. Single-clip end-to-end file processing requires 199.36 ms/video, yielding a throughput of 5.02 videos/s. The separately measured four-segment extraction cost is approximately 1.17 s/video. These figures represent different configurations and pipeline scopes; notably, 5.02 videos/s is not the temporal-pathway throughput. Furthermore, clip throughput alone cannot determine the number of supported live camera streams without accounting for clip duration, sampling stride, arrival rate, and buffering policy.

Table 15. Computational measurements and their scope (NVIDIA GeForce RTX 3080 Ti Laptop GPU, NVIDIA Corporation, Santa Clara, CA, USA, unless specified).

Measurement	Configuration and Scope	Result
Parameters	Complete model	34.76 M
Model file size	Deployment weights	132.61 MB
Pure model latency	Single 16-frame clip; batch 1	31.78 ms
Pure model throughput	Single-clip model benchmark	31.47 clips/s
End-to-end file latency	Single clip; decode and preprocessing included	199.36 ms/video
End-to-end throughput	Single-clip file pipeline	5.02 videos/s
Peak GPU memory	Single-clip inference	0.523 GB
Four-segment extraction	540 videos; single process; workers = 0	633.14 s total; ≈1.17 s/video
Additional temporal rule cost	CPU; batch 64; $T = 4$	≈0.021 ms/video

5. Results

5.1. What the Existing Results Establish

The evidence supports a benchmark study of video-level classification and prototype-based score construction. On CMA, the fused model achieves 93.35% Macro-AP, compared with 93.23% for its neural branch; this small difference does not establish a substantial accuracy benefit from adding the rule pathway. The rule head nevertheless provides a separately computed class-score vector whose dependence on predefined prototypes and predicate logits can be examined. This is inspection of the scoring computation, not a verified explanation of which person or image region caused a prediction. Numerical stress tests support finite log-domain computation over the tested ranges. These findings complement prior neuro-symbolic and log-space reasoning work [11,13,15,17], while remaining specific to the evaluated architecture, prototype definitions, and CMA protocols.

5.2. Backbone Contribution and Transfer Scope

The 15.67-point Macro-AP difference between fused backbones vastly exceeds the 0.12-point difference between neural-only and fused MViTv2-S outputs. Because all branches share visual features and utilize auxiliary losses, branch-output comparisons do not constitute a complete end-to-end backbone-only ablation. Nevertheless, results support attributing most closed-set performance to the visual representation. The frozen-feature study further indicates that loss components and fusion do not consistently improve every metric. Consequently, the method is best understood as an alternative scoring and inspection interface with measurable transfer behavior, rather than a uniformly superior classifier.

Held-out-class AP demonstrates transfer under the stated exclusion protocol, though with substantial variation across categories. The rule-versus-neural confidence interval includes zero, and fixed-rank combination shows no statistically established advantage over its components. Since Kinetics pretraining may contain related action semantics and six CMA classes still supervise each rule head, these results do not establish recognition without task-domain supervision, arbitrary new safety violations, or out-of-domain generalization. Additionally, because fixed-rank combination uses the unlabeled evaluation pool, its performance reflects batch-ranking capability rather than online calibrated alarm readiness.

5.3. Predicate Meaning and Physical Failure Modes

The predicate pathway makes the score computation inspectable, but the semantics of its eight variables remain weakly supervised. Agreement with class prototypes and sensitivity to logit interventions establish internal consistency, not independent correspondence to physical events. Binary prototypes also impose negative constraints on variables that may coexist with the target behavior. For an expected-negative entry $r(c,k) = 0$, its contribution to the normalized rule score is $\log \sigma(-\mathrm{zk})/(K\tau)$, with derivative $-\sigma(z_k)/(K\tau)$. Increasing that predicate logit therefore decreases the class score monotonically; for a logit increase $\Delta > 0$, the score decrease is at most $\Delta/(K\tau)$. This is a determinate response to a restrictive prototype, not evidence that concurrent-action recognition is reliable. For example, locomotion can suppress a smoking score even when both actions are present. Repeated or large conflicts can still strongly suppress the score, and the global fusion gate provides no sample-specific guarantee of correction. Ternary masks and multi-label prediction would change the tested model and require separate evaluation.

The present model predicts one label for a whole video clip and has no person-indexed output. It cannot assign concurrent actions to individual workers, even if the backbone retains some spatial information internally. Person detection and tracking [3,29] could provide a different input interface, but their integration has not been evaluated here. Camera motion can confound motion-related predicates because the pipeline has no explicit camera-motion compensation. A fall that leaves the field of view may remove the visual evidence needed by the falling template. The predicate upright_or_static can respond to a stationary post-event state; it does not require standing up, and it does not establish recovery or absence of injury. Conversely, low subsequent-state activation may suppress the temporal score without disproving a fall. The talking class is a benchmark behavior label, not a judgment that cooperation or speech is unsafe. The system does not estimate speech content, gaze, or lip motion. The frequencies and severity of these failure modes under distance, illumination changes, occlusion, or camera instability have not been measured.

5.4. Computational Scope and Remaining Validation

Laptop-GPU measurements establish computational costs for reported offline configurations only. They do not verify real-time multi-camera service, embedded-device performance, energy consumption, or field reliability. Four-segment extraction costs are

distinct from single-clip inference and end-to-end processing. Live systems require specified sampling policies and workloads, followed by latency and throughput measurements under those specific conditions. Translating behavior scores into operational safety decisions necessitates site-dependent risk interpretation and human review.

The main closed-set result uses one training seed; source-group bootstrap intervals do not quantify variation across independently trained models. The three-seed experiments concern only the stated ablation and held-out-class protocols. Independent predicate annotations, systematic rule-perturbation tests, temporal-resolution sensitivity experiments, and matched-protocol external baselines are not available. Figures 3, 4 and 9 summarize predicate agreement, internal interventions, and segment-level activations. They do not link a particular prediction to visible evidence in a particular frame. Consequently, the study establishes inspectability of the scoring computation, but does not verify visual grounding or person-level attribution. That distinction limits the interpretation of the model's explanations. No frame-level examples, localization results, or attention maps [30] are claimed as part of the present evidence.

6. Conclusions

This study evaluates a neuro-symbolic spatiotemporal Vision Transformer for video-level recognition of seven CMA behavior classes. Its prototype-based rule branch exposes a defined score-construction pathway, while the strong neural branch accounts for most closed-set performance. Leave-one-class-out experiments provide evidence of target-class-held-out transfer within CMA, but do not establish a statistically significant rule-versus-neural advantage or generalization to unseen sites. The log-domain implementation maintains finite computation within the reported stress-test ranges. Single-clip file processing and four-segment extraction have distinct offline costs, and the latter requires approximately 1.17 s/video on the tested laptop GPU. The demonstrated contribution is therefore a benchmark evaluation of classification, score inspection, and numerical behavior. Person-level action attribution, concurrent-action recognition, independent visual grounding, and live-stream evaluation require separate methodological development and validation.

Author Contributions: Conceptualization, A.Z. and Y.Z.; methodology, A.Z.; software, A.Z.; validation, A.Z., F.H. and R.S.; writing—original draft preparation, A.Z.; writing—review and editing, Y.Z.; resources, Z.C.; supervision, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the State Grid Corporation of China technology project "Research and Application of Key Technologies for Multi-Objective Interpretable On-Site Operation Violation Recognition Based on Vision-Language Models", grant number 5700-202426249A-1-1-ZN.

Institutional Review Board Statement: Not applicable. This study used only the publicly available Construction Meta Action benchmark dataset and did not involve any new human or animal subjects.

Informed Consent Statement: Not applicable. This study used only the publicly available Construction Meta Action benchmark dataset, and no new personal data were collected from participants.

Data Availability Statement: The Construction Meta Action (CMA) dataset is provided by the authors of [7] through their ConstructionActionRecognition repository: https://github.com/S1mpleyang/ConstructionActionRecognition (accessed on 15 October 2026). The repository identifies the dataset as research-only. No proprietary dataset is used in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jiang, Y.; Li, Y.; Li, Z.; Sun, L.; Zhou, A. Statistics and law analysis of personal safety accidents of power grid enterprises in China from 2014 to 2024. *Sci. Rep.* **2026**, *16*, 932. [CrossRef] [PubMed]
2. Wu, B.; Wang, C.; Sun, Z. Image detection algorithm of workers' unsafe behaviors based on ME-YOLO11. *Comput. Eng. Appl.* **2026**, *62*, 103–115. [CrossRef]
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
4. Nath, N.D.; Behzadan, A.H.; Paal, S.G. Deep Learning for Site Safety: Real-Time Detection of Personal Protective Equipment. *Autom. Constr.* **2020**, *112*, 103085. [CrossRef]
5. Feichtenhofer, C.; Yin, H.; Fan, Y.; Malik, J. SlowFast Networks for Video Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6202–6211.
6. Li, Y.; Wu, C.Y.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; Feichtenhofer, C. Improved Multiscale Vision Transformers for Classification and Detection. *arXiv* **2021**, arXiv:2112.01526.
7. Yang, M.; Wu, C.; Guo, Y.; Jiang, R.; Zhou, F.; Zhang, J.; Yang, Z. Transformer-based deep learning model and video dataset for unsafe action identification in construction projects. *Autom. Constr.* **2023**, *145*, 104703. [CrossRef]
8. Jin, H.; Yan, Y.; Zhang, C.; Qi, D. Electric power personnel behavior recognition algorithm based on multi-target sequence capture and spatio-temporal feature extraction. *High Volt. Eng.* **2025**, *51*, 5788–5800.
9. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv* **2023**, arXiv:2303.05499.
10. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8748–8763.
11. Badreddine, S.; d'Avila Garcez, A.; Serafini, L.; Spranger, M. Logic Tensor Networks. *Artif. Intell.* **2022**, *303*, 103649. [CrossRef]
12. Yi, K.; Wu, J.; Gan, C.; Torralba, A.; Kohli, P.; Tenenbaum, J.B. Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding. *arXiv* **2018**, arXiv:1810.02338.
13. Riegel, A.; Gray, A.; Luus, F.; Khan, N.; Makondo, N.; Akhalwaya, I.Y.; Qian, H.; Fagin, R.; Barahona, F.; Sharma, U.; et al. Logical Neural Networks. *arXiv* **2020**, arXiv:2006.13155.
14. Evans, R.; Grefenstette, E. Learning Explanatory Rules from Noisy Data. *J. Artif. Intell. Res.* **2018**, *61*, 1–45. [CrossRef]
15. Xu, Z.; Rawat, Y.; Wong, Y.; Kankanhalli, M.; Shah, M. Don't Pour Cereal into Coffee: Differentiable Temporal Logic for Temporal Action Segmentation. *arXiv* **2022**, arXiv:2203.02311.
16. Mao, J.; Gan, C.; Kohli, P.; Tenenbaum, J.B.; Wu, J. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In Proceedings of the Seventh International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
17. Badreddine, S.; Serafini, L.; Spranger, M. logLTN: Differentiable Fuzzy Logic in the Logarithm Space. *arXiv* **2023**, arXiv:2306.14546. [CrossRef]
18. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6299–6308.
19. Li, Y.; Xu, F.; Zhang, Z.; Mei, X.; Huang, H. Construction site fall hazard identification and automated captioning using adapted vision-language models. *Autom. Constr.* **2026**, *183*, 106790. [CrossRef]
20. Xian, Y.; Lampert, C.H.; Schiele, B.; Akata, Z. Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2251–2265. [CrossRef] [PubMed]
21. Tong, Z.; Song, Y.; Wang, J.; Wang, L. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. In Proceedings of the Advances in Neural Information Processing Systems 35, New Orleans, LA, USA, 28 November 2022.
22. Bertasius, G.; Wang, H.; Torresani, L. Is Space-Time Attention All You Need for Video Understanding? In Proceedings of the Thirty-Eighth International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 813–824.
23. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; Schmid, C. ViViT: A Video Vision Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 6836–6846.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Thirty-First Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
25. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 20–36.

26. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
27. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The Kinetics Human Action Video Dataset. *arXiv* **2017**, arXiv:1705.06950.
28. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On Calibration of Modern Neural Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1321–1330.
29. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
30. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–6226.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.