SE-JEPA: LEARNING REPRESENTATIONS OF STRUCTURED EVENT SEQUENCES VIA LATENT PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured event data contains within-event field dependencies and cross-event contextual relationships that can provide complementary information for representation learning. A key challenge is to integrate information from related events into field-level predictive learning. We propose the **S**tructured-**E**vent **J**oint-**E**mbedding **P**redictive **A**rchitecture (**SE-JEPA**), a self-supervised framework that combines local field modeling with cross-event context through masked-field latent prediction. SE-JEPA constructs within-event representations using field-specific encodings and retrieves field-aligned context through same-field cross-event attention. An event-level residual auxiliary objective further uses pooled event representations to learn context-induced corrections to local predictions. The two mechanisms optimize a shared encoder and predictor using latent targets supplied by an exponential moving-average teacher. Experiments on four public structured-event benchmarks covering diverse downstream tasks demonstrate the effectiveness of SE-JEPA under frozen linear evaluation and full fine-tuning.

1 INTRODUCTION

Financial transactions, retail purchases, and business processes generate large collections of structured event records (Babaev et al., 2022b; Osin et al., 2025b; Tax et al., 2017). Each event is described by numerical and categorical fields, while related events associated with the same entity or process provide a broader context for interpreting individual records. For example, the amount, merchant, and channel characterize a transaction, whereas other transactions from the same customer may reveal spending preferences and behavioral patterns. Learning transferable representations from unlabeled event records provides an important foundation for downstream prediction (Babaev et al., 2022b; Li et al., 2025). Effective structured-event representations should capture information at two levels: dependencies among fields within an event and contextual relationships across related events.

Self-supervised methods derive learning signals from structured records through different objectives. Generative and reconstructive approaches learn from attribute prediction or masked-content recovery (Yoon et al., 2020; Padhi et al., 2021a), while contrastive approaches align related views of records or sequences (Babaev et al., 2022b; Bahri et al., 2022; Yue et al., 2022). These objectives entail different design choices. Observation-space prediction requires parameterizations and losses suited to numerical and categorical fields, with high-cardinality attributes placing additional demands on the prediction output space. Contrastive learning depends on the relationships preserved by subsequence sampling or feature perturbations. The combination of heterogeneous fields and cross-event relationships makes the organization of learning targets and predictive context an important design consideration for structured-event representation learning.

Latent prediction offers an alternative paradigm in which learning targets are defined in representation space. This principle have been explored across diverse domains, including images (Assran et al., 2023), videos (Bardes et al., 2024), tabular data (Thimonier et al., 2025), and time series (Lee & Sim, 2026a; Petersen et al., 2026). In particular, predicting masked-field representations from visible fields provides a direct learning signal for within-record dependencies (Thimonier et al., 2025). Beyond the prediction target, the information supporting prediction also deserves attention. In structured event data, visible fields provide local evidence, while related events can offer additional attribute-level references and broader behavioral context. This motivates a central question:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

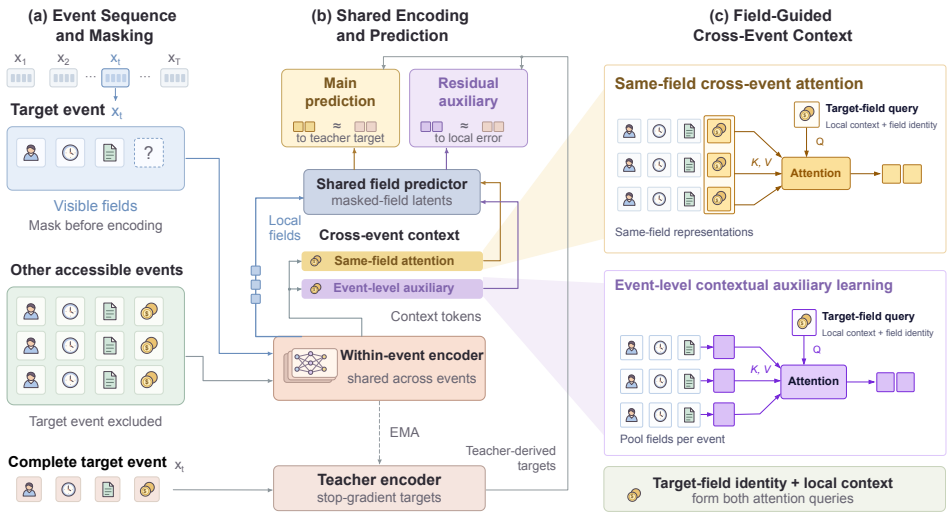

Figure 1: **Overview of SE-JEPA. (a) Event sequence and masking.** Visible fields form the online input, and other available events in the same sequence provide cross-event context. **(b) Shared encoding and prediction.** A shared within-event encoder produces field representations. An EMA teacher encodes the target event with its original missingness pattern to provide stop-gradient targets. **(c) Field-guided cross-event context.** Two parallel branches share the field predictor. Same-field attention retrieves corresponding field representations to augment the main prediction. Event-level auxiliary learning uses pooled event representations to match the context-induced prediction change to the stop-gradient local prediction error.

how can related-event context enrich masked-field latent prediction while preserving within-event field modeling?

We propose **SE-JEPA**, the *Structured-Event Joint-Embedding Predictive Architecture*, a self-supervised framework that combines local field modeling with cross-event context. Our central idea is to retain field-level latent prediction targets while enriching their predictive context at two granularities: aligned fields and whole events. As illustrated in Figure 1, *same-field cross-event attention* retrieves corresponding field representations from related events to provide aligned evidence for the main prediction. These representations have undergone within-event interaction, preserving field identity while incorporating information from other attributes. *Event-level residual auxiliary learning* uses pooled event representations to learn context-induced corrections to local predictions through a paired residual objective. Both mechanisms use a shared within-event encoder and field predictor, with field-level latent targets supplied by an exponential-moving-average teacher. Together, they incorporate local field dependencies and cross-event information into representation learning. Experiments on four public structured-event benchmarks show that SE-JEPA achieves the best performance on three of the four benchmarks under both frozen linear evaluation and full fine-tuning. Under frozen evaluation, its maximum gains over the strongest reported baselines reach 5.01 percentage points in AUROC and 1.63 percentage points in accuracy. These results support the effectiveness of context-enriched field-level latent prediction for structured-event representation learning.

The main contributions can be summarized as follows:

- We propose SE-JEPA, a self-supervised framework for structured-event representation learning that integrates within-event field dependencies and cross-event context through masked-field latent prediction.
- We develop two complementary mechanisms for incorporating related-event information into latent prediction: same-field cross-event attention provides field-aligned context for the main prediction and downstream encoding, while event-level residual auxiliary learning trains shared representations through context-induced corrections to local predictions.

- Experiments on four public structured-event benchmarks demonstrate the effectiveness of SE-JEPA across diverse downstream tasks. SE-JEPA achieves the best performance on three of the four benchmarks under both frozen linear evaluation and full fine-tuning, with controlled ablations examining the contributions of the two context mechanisms.

2 Related Work

Self-Supervised Learning for Structured Data and Event Sequences. Self-supervised approaches to structured records typically employ generative and reconstructive, contrastive, or hybrid objectives. *Generative and reconstructive methods* derive supervision by predicting event attributes or recovering corrupted content. VIME (Yoon et al., 2020) combines value reconstruction with mask estimation; TabBERT (Padhi et al., 2021a) integrates field- and sequence-level Transformers for masked prediction; and PRAGMA (Ostroukhov et al., 2026) applies masked modeling to multi-source banking event records. Generative sequence models also predict event attributes through autoregressive factorization. For example, FlexTPP (Draxler et al., 2025) models mixed-type events using discrete and continuous prediction heads. These objectives require attribute representations, predictive distributions, and losses suited to the value domains of individual fields. *Contrastive methods* learn representations by aligning related views and distinguishing unrelated observations. CoLES (Babaev et al., 2022b) constructs positive pairs from subsequences of the same entity; SCARF (Bahri et al., 2022) generates tabular views through random feature corruption; and, in related time-series research, TS2Vec (Yue et al., 2022) performs hierarchical contrastive learning over augmented context views. Their learning signals depend on the relationships preserved by view construction. *Hybrid methods* combine different sources of supervision: SAINT (Somepalli et al., 2021) jointly optimizes contrastive and denoising objectives, while MLEM (Moskvoretskii et al., 2024) aligns generative and contrastive representations of event sequences. Together, these studies highlight two interconnected design questions: how to construct suitable learning targets for heterogeneous fields, and how to organize within-event and cross-event information to support representation transfer.

Joint-Embedding Predictive Architectures. Joint-embedding predictive architectures derive supervision by predicting learned target representations and have been explored in vision, tabular learning, time series, and reinforcement learning. These approaches can be understood through two perspectives on the relationships underlying their prediction tasks: component prediction and cross-observation prediction. *Component prediction* captures dependencies among parts of an observation. I-JEPA (Assran et al., 2023) and V-JEPA (Bardes et al., 2024) predict representations of masked spatial and spatiotemporal regions, respectively. T-JEPA (Thimonier et al., 2025) predicts the latent representations of a target feature subset from visible features within the same tabular record. *Cross-observation prediction* organizes learning around sequential or dynamical relationships. CF-JEPA (Lee & Sim, 2026a) predicts future representations at multiple horizons, while HEPA (Petersen et al., 2026) conditions future-representation prediction on the horizon. seq-JEPA (Ghaemi et al., 2025) studies action-conditioned sequential prediction, and TD-JEPA (Bagatella et al., 2026) uses temporal-difference learning to capture policy-conditioned, long-term latent dynamics. These methods illustrate how prediction targets, context, and conditioning information adapt latent prediction to different relationships in the data.

SE-JEPA builds on within-record field-level latent prediction and investigates how related events can provide complementary information for target-field prediction. It organizes cross-event context into two complementary forms: same-field cross-event attention retrieves field-aligned representations from related events to support the main prediction pathway and downstream encoding, while event-level residual auxiliary learning uses pooled event representations to learn context-induced corrections to local predictions. The two mechanisms share a within-event encoder and field predictor, with field-level latent targets supplied by an exponential-moving-average teacher. Together, they integrate within-event field dependencies and cross-event context into masked-field latent learning.

3 Structured-Event Joint-Embedding Predictive Architecture

Structured-**E**vent **J**oint-**E**mbedding **P**redictive **A**rchitecture **(SE-JEPA)** learns representations from structured-event sequences by combining within-event field modeling with cross-event context. Fig-

ure 1 presents the overall framework. A shared within-event encoder first captures dependencies among visible fields, and a field predictor estimates masked-field representations supplied by an EMA teacher (Section 3.1).

Two independently enabled branches incorporate context from other events in the same sequence (Section 3.2). **Same-field cross-event attention** retrieves corresponding field representations to augment the main prediction path. **Event-level contextual auxiliary learning** uses pooled event representations to learn a context-induced correction to the local prediction. The two branches share the within-event encoder and field predictor and are trained through the main prediction objective and the auxiliary objective. For downstream tasks, SE-JEPA retains the online encoder and, when enabled, same-field attention; the auxiliary branch contributes through pretraining updates to the shared encoder (Section 3.3).

3.1 Within-Event Field Modeling

Structured events and masking. Let $X = (x_1, \ldots, x_T)$ denote a structured-event input sequence, where each event $x_t = (x_{t,1}, \ldots, x_{t,F})$ contains F fields with fixed semantic identities. The index t denotes an event's position in the input sequence. During pretraining, cross-event context is drawn from the other valid events in the same sequence, excluding the target event. From the observed fields $\mathcal{O}_t$, we sample nonempty visible and target subsets $\mathcal{S}_t, \mathcal{T}_t \subseteq \mathcal{O}_t$ with $\mathcal{S}_t \cap \mathcal{T}_t = \varnothing$. The local student encodes the sampled visible fields in $\mathcal{S}_t$. The teacher, context-event encoder, and downstream encoder process complete events with their original missingness patterns. Known aliases are assigned together to prevent equivalent fields from appearing on both sides of a prediction task. Further masking and tokenization details are given in Appendix A.2.

Local field representations. We model within-event field dependencies through masked-field latent prediction. Field-specific value encodings are combined with field-identity, field-type, and observation-state embeddings to form input tokens. The shared encoder E_θ applies self-attention within each event, yielding contextualized field-level representations. The visible-field representations serve as local context for predicting the teacher representations of the masked fields. For the visible input, define

$$H_t^{\mathcal{S}} = E_\theta(x_t|_{\mathcal{S}_t}), \qquad \bar{h}_t^{\mathcal{S}} = \frac{1}{|\mathcal{S}_t|} \sum_{f \in \mathcal{S}_t} h_{t,f}^{\mathcal{S}}, \tag{1}$$

where each field representation lies in $\mathbb{R}^d$. An exponential-moving-average (EMA) teacher encodes the complete target event before artificial masking, preserving its original missingness pattern:

$$z_{t,f} = \mathrm{sg}([E_{\bar{\theta}}(x_t)]_f)\,. \tag{2}$$

Here sg denotes stop-gradient. The teacher encodes fields within the target event and does not access other events.

A shared field predictor. Let $P_\psi(H, \mathcal{T}; r)$ jointly process visible-field tokens and target-field queries, where r_f denotes an optional context offset for target field f. Each offset is linearly projected to the predictor's hidden dimension and added only to the corresponding target-field query before predictor self-attention. The two cross-event branches introduced below supply this optional context input. With zero context offsets, the local prediction is

$$p_{t,f} = \left[P_\psi(H_t^{\mathcal{S}}, \mathcal{T}_t; 0)\right]_f, \qquad f \in \mathcal{T}_t. \tag{3}$$

Both cross-event branches use the same visible-field input and contextualized teacher targets, while supplying context through different memory structures and learning objectives.

3.2 Leveraging Cross-Event Context

Within-event modeling captures dependencies among fields, while other events in the same sequence can provide additional context for predicting masked-field representations. SE-JEPA incorporates this context through two independently switchable mechanisms: *same-field cross-event attention* augments the main prediction path, whereas *event-level contextual auxiliary learning* provides an additional pretraining signal from pooled event representations.

During pretraining, let $\mathcal{C}_t$ denote the indices of the other valid events in the same input sequence, excluding the target event x_t. The shared encoder independently encodes each context event as $H_s = E_\theta(x_s)$ for $s \in \mathcal{C}_t$, where $h_{s,f}$ denotes its representation at field position f. We write $\bar{h}_t^{\mathcal{S}}$ for the mean of the visible-field representations of the target event.

Same-field cross-event attention. The relevance of a context event may depend on the field being predicted. To preserve field correspondence across events, this branch organizes context representations by field identity. For target field f, it constructs the memory $\mathcal{M}_{t,f} = \{h_{s,f} : s \in \mathcal{C}_t\}$ and retrieves context using a query formed from the local visible-field representations and the field-identity embedding:

$$\begin{aligned} c_{t,f} &= \mathrm{Attn}_{\mathrm{sf}}\left(\bar{h}_t^{\mathcal{S}} + e_f, \mathcal{M}_{t,f}\right), \\ u_{t,f} &= a \tanh(\gamma_{\mathrm{sf}}) c_{t,f}. \end{aligned} \tag{4}$$

Here, e_f is a learnable field-identity embedding shared across events, $a \in \{0, 1\}$ is a configuration switch that enables or disables this branch, and $\mathrm{Attn}(q, \mathcal{M})$ denotes multi-head attention with keys and values derived from $\mathcal{M}$. Attention parameters are shared across fields, but each query accesses only the corresponding field positions in other events. Because these representations have already been contextualized by within-event attention, they also carry information about other fields in their respective events.

The retrieved offsets $u_t = \{u_{t,f} : f \in \mathcal{T}_t\}$ condition the shared predictor, yielding the main masked-field prediction objective:

$$\begin{aligned} \hat{z}_{t,f} &= \left[P_\psi\left(H_t^{\mathcal{S}}, \mathcal{T}_t; u_t\right)\right]_f, \\ \mathcal{L}_{\mathrm{pred}}(a) &= \frac{1}{d}\mathbb{E}_{\mathcal{I}}\left[\|\hat{z}_{t,f} - z_{t,f}\|_2^2\right]. \end{aligned} \tag{5}$$

Here, $z_{t,f}$ is the teacher representation, d is the latent dimension, and $\mathbb{E}_{\mathcal{I}}$ denotes the empirical mean over the collection $\mathcal{I}$ of valid target-field occurrences in a minibatch. Occurrences under different target masks are counted separately; sample and mask indices are left implicit. The learnable scale γ_{sf} is initialized to zero, so the branch initially leaves local predictions unchanged. Setting $a = 0$ recovers local latent prediction; enabling the branch does not introduce a separate local-only loss.

Event-level contextual auxiliary learning. This branch uses event-level context to provide an auxiliary training signal for the shared encoder. This branch summarizes each context event by mean-pooling its field representations, $m_s = F^{-1}\sum_{j=1}^{F} h_{s,j}$, and constructs the shared event memory $\mathcal{N}_t = \{m_s : s \in \mathcal{C}_t\}$. Each target field queries this memory using its local context, field identity, and field type:

$$\begin{aligned} q_{t,f} &= \bar{h}_t^{\mathcal{S}} + \widetilde{e}_f + \widetilde{e}_{\tau_f}, \\ v_{t,f} &= \tanh(\gamma_{\mathrm{ev}})\,\mathrm{Attn}_{\mathrm{ev}}\left(q_{t,f}, \mathcal{N}_t\right). \end{aligned} \tag{6}$$

Here, τ_f denotes the field type, the embeddings are specific to this branch, and γ_{ev} is initialized to zero. For a given target event, all target fields access the same pooled event representations, while their field-dependent queries allow them to assign different attention weights to the context events.

To train this branch, we compare the contextual prediction $\tilde{p}_{t,f} = [P_\psi(H_t^{\mathcal{S}}, \mathcal{T}_t; v_t)]_f$, where $v_t = \{v_{t,f} : f \in \mathcal{T}_t\}$, with a paired local prediction $p_{t,f} = [P_\psi(H_t^{\mathcal{S}}, \mathcal{T}_t; 0)]_f$ produced by the same predictor without a cross-event offset. Their difference is trained to predict the remaining error of the local prediction:

$$\mathcal{L}_{\mathrm{aux}} = \frac{1}{d}\mathbb{E}_{\mathcal{I}}\Big[\Big\| \underbrace{\left(\tilde{p}_{t,f} - p_{t,f}\right)}_{\text{context-induced change}} - \underbrace{\mathrm{sg}\left(z_{t,f} - p_{t,f}\right)}_{\text{local residual target}}\Big\|_2^2\Big]. \tag{7}$$

The paired predictions use the same local-context activations and matched predictor dropout. Stop-gradient applies only to the residual target; $p_{t,f}$ remains differentiable in the prediction difference. Although the forward value equals contextual prediction MSE, the resulting gradient depends on the difference between the contextual and local prediction Jacobians. Appendix D provides the derivation.

The two branches operate in parallel, sharing the field encoder and predictor while using separate attention parameters and memory organizations. The auxiliary branch pairs its contextual output with

the zero-offset local prediction $p_{t,f}$. Same-field cross-event attention is retained for downstream encoding; the event-level auxiliary modules are used only during pretraining and influence transfer through updates to the shared encoder. Memory construction details are provided in Appendix A.3.

3.3 Learning and Transfer

Pretraining. We optimize the main prediction objective together with the event-level auxiliary objective:

$$\mathcal{L}_{\text{SE-JEPA}} = \mathcal{L}_{\text{pred}}(a) + \lambda \mathcal{L}_{\text{aux}}, \qquad a \in \{0, 1\}, \quad \lambda \geq 0. \tag{8}$$

The switch a controls same-field cross-event attention, while λ weights the auxiliary objective; setting $\lambda = 0$ disables auxiliary learning. We optimize the online encoder, predictor, and enabled context modules. After optimization step k, the teacher is updated by

$$\bar{\theta} \leftarrow \mu_k \bar{\theta} + (1 - \mu_k)\theta, \tag{9}$$

where μ_k is the EMA decay coefficient. Only the within-event encoder is tracked by EMA. Initialization and configuration conventions are specified in Appendix A.4.

Downstream representation. After pretraining, we discard the teacher, field predictor, and event-level auxiliary modules. We retain the online encoder and, when enabled, the same-field attention parameters and learned scale. Each complete input event is encoded as $H_t = E_\theta(x_t)$, preserving its original missingness pattern. Let $\mathcal{V}$ denote the indices of all valid events in the input sequence. For complete-sequence classification, downstream field and event representations are computed as

$$\begin{aligned} \widetilde{h}_{t,f} &= h_{t,f} + a \tanh(\gamma_{\text{sf}}) \, \text{Attn}_{\text{sf}} \left(h_{t,f}, \{ h_{s,f} : s \in \mathcal{V} \} \right), \\ g_t &= \frac{1}{F} \sum_{f=1}^{F} \widetilde{h}_{t,f}. \end{aligned} \tag{10}$$

Downstream encoding reuses the learned same-field attention parameters and scale. Each complete-event field token serves as a query over the corresponding field representations of all valid events, including its own event. The retrieved context is added directly to the query field representation.

Frozen evaluation and full fine-tuning. For frozen evaluation, we fix the entire representation map, average the complete-event representations g_t, and train only a linear task head. Full fine-tuning jointly updates the retained encoder, enabled same-field attention and scale, trainable event readout, and task head. The readout can incorporate event position and time-gap attributes before attention pooling; The trainable readout applies event-wise transformations and can incorporate event positions and time gaps before attention pooling. Appendix A.5 specifies both readouts and their trainable parameters. Configuration selection and readout tuning are described in Appendix **??**.

4 Experiments

We evaluate SE-JEPA through downstream prediction, frozen representation evaluation, and label-efficiency analysis, and examine the contributions of its two cross-event context mechanisms through controlled ablations.

4.1 Experimental Setup

Datasets and tasks. We consider four public structured-event benchmarks covering financial behavior, retail purchases, and business processes. **MBD-mini** (Mollaev et al., 2025) involves predicting the purchase of four banking products in the following month. **AlphaBattle** (Boosters, n.d.) concerns credit-default prediction from transaction histories. **Retail/X5** uses purchase sequences for four-class age-group prediction, following the task formulation in EBES (Osin et al., 2025a). **BPI17** (van Dongen, 2017) predicts offer acceptance from loan-application event logs. Appendix B.1 describes the data sources, input fields, sequence construction, and data splits.

Table 1: Downstream performance after supervised training or full fine-tuning. Aggregation models and supervised sequence models are trained from scratch; pretrained encoders are fully fine-tuned. Best and second-best results are shown in **bold** and underlined, respectively.

Method	MBD-mini mAUC ↑	AlphaBattle AUC ↑	Retail/X5 Acc. ↑	BPI17 AUC ↑
Aggregation-based tabular models				
XGBoost-Agg	76.45	74.16	46.25	74.78
CatBoost-Agg	77.22	75.07	47.81	**76.64**
MLP-Agg	74.59	73.89	46.33	71.96
Supervised sequence models (from scratch)				
GRU	74.46	73.80	54.79	71.05
Transformer	78.02	75.54	53.16	72.07
Contrastive, reconstructive, and generative pretraining				
CoLES	75.40	74.23	54.34	68.46
MLEM	—	73.87	—	72.08
TabBERT	77.98	76.05	55.14	75.28
JEPA-based pretraining				
T-JEPA	74.75	73.17	53.34	72.02
CF-JEPA	74.33	75.72	50.64	72.76
HEPA	78.05	75.17	53.36	71.91
SE-JEPA (ours)	**79.09**	**76.95**	**55.54**	75.32

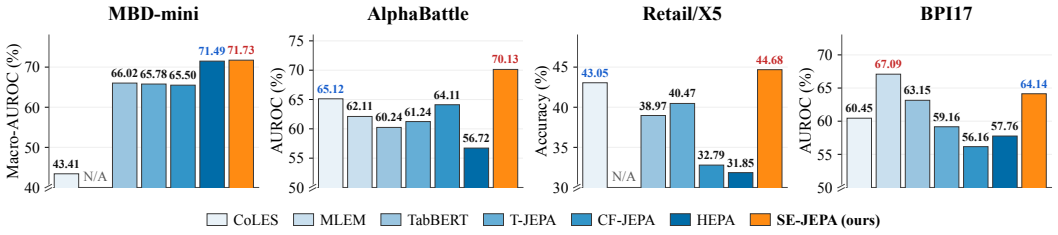

Figure 2: **Frozen representation evaluation on four public structured-event benchmarks.** Only a linear task head is trained for each frozen encoder. The best and second-best results are highlighted in red and blue, respectively.

Baselines. We compare SE-JEPA with eleven baselines in four groups: (i) *aggregation-based tabular models*: XGBoost-Agg (Chen & Guestrin, 2016), CatBoost-Agg (Prokhorenkova et al., 2018), and MLP-Agg; (ii) *supervised sequence models*: GRU (Cho et al., 2014) and Transformer (Vaswani et al., 2023); (iii) *contrastive, reconstructive, and generative pretraining*: CoLES (Babaev et al., 2022a), MLEM (Moskvoretskii et al., 2024), and TabBERT (Padhi et al., 2021b); and (iv) *JEPA-based pretraining*: T-JEPA (Thimonier et al., 2025), CF-JEPA (Lee & Sim, 2026b), and HEPA (Petersen et al., 2026). The suffix "-Agg" indicates that a model operates on sequence-level aggregated features. Appendix B.2 specifies the implementations, structured-event input adaptations, and method-specific training and readout settings.

Evaluation settings and metrics. For downstream prediction, aggregation-based models and supervised sequence models are trained from scratch, while pretrained encoders are fully fine-tuned with their downstream readouts and task heads. Frozen evaluation trains only a linear task head on fixed sequence representations. We report macro-averaged AUROC over the four labels for MBD-mini, accuracy for Retail/X5, and AUROC for AlphaBattle and BPI17. All scores are expressed as percentages, with higher values indicating better performance. Dataset-specific SE-JEPA configurations, optimization settings, training budgets, and checkpoint-selection rules are provided in Appendix B.3.

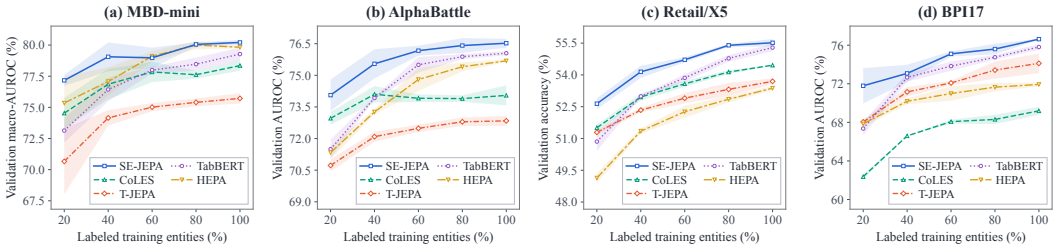

Figure 3: **Label efficiency on four structured-event benchmarks.** Curves show mean HPO-validation performance over three downstream random seeds, and shaded regions indicate ± one sample standard deviation. Each method uses a fixed pretrained checkpoint per dataset. SE-JEPA achieves the highest mean performance in different dataset–label-fraction settings.

4.2 Main Results

Strong performance across diverse downstream tasks. Table 1 compares supervised training and full fine-tuning across the four benchmarks. SE-JEPA obtains 79.09 macro-AUROC on MBD-mini, 76.95 AUROC on AlphaBattle, and 55.54% accuracy on Retail/X5, improving over the strongest baseline in each column by 1.04, 0.90, and 0.40 percentage points, respectively. On BPI17, SE-JEPA reaches 75.32 AUROC, ranking second to CatBoost-Agg at 76.64. These results establish SE-JEPA as a competitive approach across product-purchase, credit-risk, demographic, and process-outcome prediction tasks.

Leading performance in frozen linear evaluation. Figure 2 evaluates the task information accessible through a linear head fitted to fixed pretrained representations. SE-JEPA achieves 71.73 on MBD-mini, 70.13 on AlphaBattle, and 44.68 on Retail/X5, exceeding the strongest reported baseline by 0.24, 5.01, and 1.63 percentage points, respectively. On BPI17, its score of 64.14 ranks second to MLEM at 67.09. The particularly large gain on AlphaBattle highlights the linear accessibility of task-relevant information in SE-JEPA's pretrained representations, complementing its performance after supervised adaptation.

Sustained advantages with limited labeled data. Figure 3 compares SE-JEPA with CoLES, T-JEPA, TabBERT, and HEPA under full fine-tuning with 20–100% of labeled training entities. Each method uses a fixed pretrained checkpoint for each dataset. Curves report mean performance on the hyperparameter-selection validation split (HPO-Val) over three downstream random seeds; shaded regions show one sample standard deviation. SE-JEPA achieves the highest mean score in 19 of the 20 dataset–label-fraction settings. At 20% labeled data, its gains over the strongest competing baseline are 1.82, 1.10, 1.13, and 3.75 percentage points on MBD-mini, AlphaBattle, Retail/X5, and BPI17, respectively. These results show that SE-JEPA remains effective when supervised adaptation uses a smaller labeled sample.

4.3 Contributions of Cross-Event Context

Consistent gains from same-field cross-event attention. Table 2 compares local-only prediction, the two context mechanisms enabled individually, and their joint configuration under the same data split, downstream readout, and training budget. Same-field attention improves performance across all four benchmarks, with gains of 0.39, 0.48, 1.07, and 0.23 percentage points on MBD-mini, AlphaBattle, Retail/X5, and BPI17, respectively. The largest improvement occurs on Retail/X5, where accuracy increases from 54.47% to 55.54%. These consistent gains support the value of preserving field correspondence when incorporating cross-event context.

Complementary benefits of event-level auxiliary learning and same-field attention. Combining the two mechanisms further improves performance over same-field attention alone on AlphaBattle and BPI17, yielding gains of 0.37 and 1.14 percentage points, respectively. On MBD-mini and Retail/X5, same-field attention alone achieves the highest scores among the evaluated configurations.

Table 2: Component ablation under full fine-tuning. I: field-aligned cross-event interaction retained downstream; R: residual auxiliary prediction used only during pretraining. Scores are percentages; parentheses indicate changes in percentage points relative to Local only. Best results are shown in **bold**.

Configuration	I	R	MBD-mini	AlphaBattle	Retail/X5	BPI17
Local only	×	×	78.29	75.93	54.47	73.95
Residual auxiliary	×	✓	78.39 (+0.10)	75.93 (0.00)	54.45 (-0.02)	73.43 (-0.52)
Field-aligned interaction	✓	×	**78.68** (+0.39)	76.42 (+0.49)	**55.54** (+1.07)	74.18 (+0.23)
Joint SE-JEPA	✓	✓	78.52 (+0.23)	**76.79** (+0.86)	55.53 (+1.06)	**75.32** (+1.37)

Although auxiliary learning alone yields limited gains and reduces performance on Retail/X5 and BPI17, the joint configuration achieves the highest ablation scores on AlphaBattle and BPI17, improving over same-field attention alone by 0.37 and 1.14 percentage points. BPI17 provides a particularly clear example: auxiliary learning alone scores 73.43, while the joint model reaches 75.32, exceeding the local-only model by 1.37 points. Overall, the joint configuration improves upon local-only prediction on all four benchmarks. These results suggest that event-level auxiliary learning can complement the retained same-field pathway, with the magnitude of this benefit depending on the downstream task.

4.4 Representation Visualization

Fine-tuned representations exhibit class-related clustering. Figure 4 presents three-dimensional t-SNE visualizations of SE-JEPA's pre-classifier representations after full fine-tuning, using 200 validation examples per class. On both MBD-mini and Retail/X5, samples from different classes exhibit varying degrees of local clustering. Although some overlap remains, the class-related concentrations suggest that the fine-tuned representations capture information relevant to the downstream labels. This qualitative pattern is consistent with SE-JEPA's strong downstream classification performance.

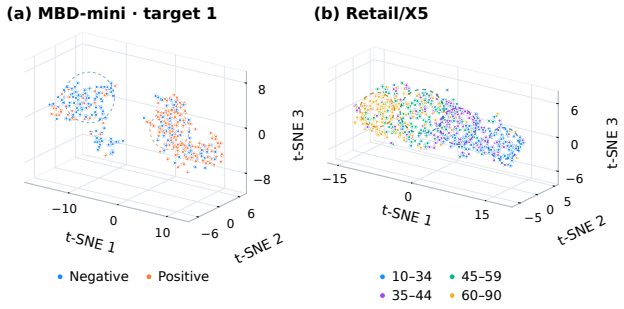

Figure 4: **Fine-tuned representation geometry.** Three-dimensional t-SNE of SE-JEPA's pre-classifier representations after supervised fine-tuning on (a) MBD-mini (target 1) and (b) Retail/X5. Dashed circles denote post-hoc, label-informed local cores rather than decision boundaries.

5 Conclusion

We introduced SE-JEPA, a joint-embedding predictive framework for representation learning from structured-event sequences. SE-JEPA retains masked-field latent prediction while enriching its predictive context through two field-guided cross-event mechanisms. Same-field attention provides field-aligned evidence for the main prediction pathway and downstream encoding, whereas event-level contextual auxiliary learning trains the shared representation through context-induced prediction corrections. Across four public benchmarks, SE-JEPA ranks first on three of the four benchmarks under both frozen linear evaluation and full fine-tuning. It also maintains strong performance under reduced label budgets. Controlled ablations show that same-field attention improves downstream performance across all four benchmarks, while event-level auxiliary learning provides additional gains when combined with same-field attention on AlphaBattle and BPI17. Together, these results support field-guided cross-event context as a useful design for predictive representation learning on structured-event sequences. Our evaluation focuses on sequence-level classification under fixed field schemas. Extending SE-JEPA to schema variation, streaming prediction, and adaptive context selection provides natural directions for future work.

AI USE STATEMENT

We used generative AI tools to polish the writing, assist in finding relevant literature and reference materials, and carry out experiments according to author-specified instructions, including which tests to run and how to run them. We reviewed the AI-assisted text, consulted the cited sources, and checked the experimental outputs. We take responsibility for the final content of this work.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, June 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Assran_Self-Supervised_Learning_From_Images_With_a_Joint-Embedding_Predictive_Architecture_CVPR_2023_paper.html.

Dmitrii Babaev, Nikita Ovsov, Ivan Kireev, Maria Ivanova, Gleb Gusev, Ivan Nazarov, and Alexander Tuzhilin. Coles: Contrastive learning for event sequences with self-supervision. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD/PODS '22, pp. 1190–1199. ACM, 2022a. doi: 10.1145/3514221.3526129. URL http://dx.doi.org/10.1145/3514221.3526129.

Dmitrii Babaev, Nikita Ovsov, Ivan Kireev, Maria Ivanova, Gleb Gusev, Ivan Nazarov, and Alexander Tuzhilin. CoLES: Contrastive learning for event sequences with self-supervision. In *Proceedings of the 2022 International Conference on Management of Data*, pp. 1190–1199. Association for Computing Machinery, 2022b. doi: 10.1145/3514221.3526129. URL https://doi.org/10.1145/3514221.3526129.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. In *International Conference on Learning Representations*, pp. 36211–36244, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3d158f054ff0cb83397367234899db07-Paper-Conference.pdf.

Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. SCARF: Self-supervised contrastive learning using random feature corruption. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=CuV_qYkmKb3.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Boosters. Alfa Battle 2.0 Competition. https://boosters.pro/championship/alfabattle2/overview, n.d. Accessed: 2026-09-24.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Felix Draxler, Yang Meng, Kai Nelson, Lukas Laskowski, Yibo Yang, Theofanis Karaletsos, and Stephan Mandt. Transformers for mixed-type event sequences. In

Advances in Neural Information Processing Systems, volume 38, pp. 114948–114980. Curran Associates, Inc., 2025. doi: 10.52202/085713-3833. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a6c7515ac435277dc92b75a07bb2257c-Abstract-Conference.html.

Hafez Ghaemi, Eilif B. Muller, and Shahab Bakhtiari. seq-JEPA: Autoregressive predictive learning of invariant-equivariant world models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 32943–32973. Curran Associates, Inc., 2025. doi: 10.52202/085713-1104. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2f63d2963526bdd9ff1b8bcc2dc9905a-Paper-Conference.pdf.

Jaehoon Lee and Sunghyun Sim. CF-JEPA: Mask-free forward prediction with asymmetric encoder utilization for time-series representation learning, 2026a. URL https://arxiv.org/abs/2606.07031.

Jaehoon Lee and Sunghyun Sim. Cf-jepa: Mask-free forward prediction with asymmetric encoder utilization for time-series representation learning, 2026b. URL https://arxiv.org/abs/2606.07031.

Guilin Li, Yun Zhang, Xiuyuan Chen, Chengqi Li, Bo Wang, Linghe Kong, Wenjia Wang, Weiran Huang, and Matthias Tan. PANTHER: Generative pretraining beyond language for sequential user behavior modeling. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 163622–163643. Curran Associates, Inc., 2025. doi: 10.52202/085713-5462. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/ef7a500bc90d45e362c82d63a0eff22d-Abstract-Conference.html.

Dzhambulat Mollaev, Ivan Kireev, Mikhail Orlov, Alexander Kostin, Ivan Karpukhin, Maria Postnova, Gleb Gusev, and Andrey Savchenko. Multimodal banking dataset: Understanding client needs through event sequences. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, pp. 6476–6480. ACM, November 2025. doi: 10.1145/3746252.3761635. URL http://dx.doi.org/10.1145/3746252.3761635.

Viktor Moskvoretskii, Dmitry Osin, Egor Shvetsov, Igor Udovichenko, Maxim Zhelnin, Andrey Dukhovny, Anna Zhimerikina, and Evgeny Burnaev. Mlem: Generative and contrastive learning as distinct modalities for event sequences, 2024. URL https://arxiv.org/abs/2401.15935.

Dmitry Osin, Igor Udovichenko, Egor Shvetsov, Viktor Moskvoretskii, and Evgeny Burnaev. Ebes: Easy benchmarking for event sequences. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 5730–5741. ACM, August 2025a. doi: 10.1145/3711896.3737428. URL http://dx.doi.org/10.1145/3711896.3737428.

Dmitry Osin, Igor Udovichenko, Egor Shvetsov, Viktor Moskvoretskii, and Evgeny Burnaev. EBES: Easy benchmarking for event sequences. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5730–5741. Association for Computing Machinery, 2025b. doi: 10.1145/3711896.3737428. URL https://doi.org/10.1145/3711896.3737428.

Maxim Ostroukhov, Ruslan Mikhailov, Vladimir Iashin, Artem Sokolov, Andrei Akshonov, Vitaly Protasov, Andrey Goncharov, Dmitrii Beloborodov, Vince Mullin, Roman Yokunda Enzmann, Georgios Kolovos, Jason Renders, Pavel Nesterov, and Anton Repushko. PRAGMA: Revolut foundation model, 2026. URL https://arxiv.org/abs/2604.08649. Version 2, August 24, 2026.

Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre Dognin, Jerret Ross, Ravi Nair, and Erik Altman. Tabular transformers for modeling multivariate time series. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3565–3569. IEEE, June 2021a. doi: 10.1109/ICASSP39728.2021.9414142. URL https://arxiv.org/abs/2011.01843.

Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre Dognin, Jerret Ross, Ravi Nair, and Erik Altman. Tabular transformers for modeling multivariate time series, 2021b. URL https://arxiv.org/abs/2011.01843.

Jonas Petersen, Gian-Alessandro Lombardi, Riccardo Maggioni, Camilla Mazzoleni, Federico Martelli, and Philipp Petersen. Hepa: A self-supervised horizon-conditioned event predictive architecture for time series, 2026. URL https://arxiv.org/abs/2605.11130.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 6639–6649, Red Hook, NY, USA, 2018. Curran Associates Inc.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training, 2021. URL https://arxiv.org/abs/2106.01342.

Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. Predictive business process monitoring with LSTM neural networks. In *Advanced Information Systems Engineering*, volume 10253 of *Lecture Notes in Computer Science*, pp. 477–492. Springer International Publishing, 2017. doi: 10.1007/978-3-319-59536-8_30. URL https://doi.org/10.1007/978-3-319-59536-8_30.

Hugo Thimonier, José Lucas De Melo Costa, Fabrice Popineau, Arpad Rimmel, and Bich-Liên Doan. T-JEPA: Augmentation-free self-supervised learning for tabular data. In *International Conference on Learning Representations*, pp. 58743–58768, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/93f01c8d9b355d7bbe3f353b44ccde66-Paper-Conference.pdf.

Boudewijn van Dongen. BPI Challenge 2017. 4TU.Centre for Research Data, 2017. URL https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b. Dataset.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. VIME: Extending the success of self- and semi-supervised learning to tabular domain. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11033–11043. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/7d97667a3e056acab9aaf653807b4a03-Abstract.html.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2Vec: Towards universal representation of time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8980–8987, 2022. doi: 10.1609/aaai.v36i8.20881. URL https://ojs.aaai.org/index.php/AAAI/article/view/20881.

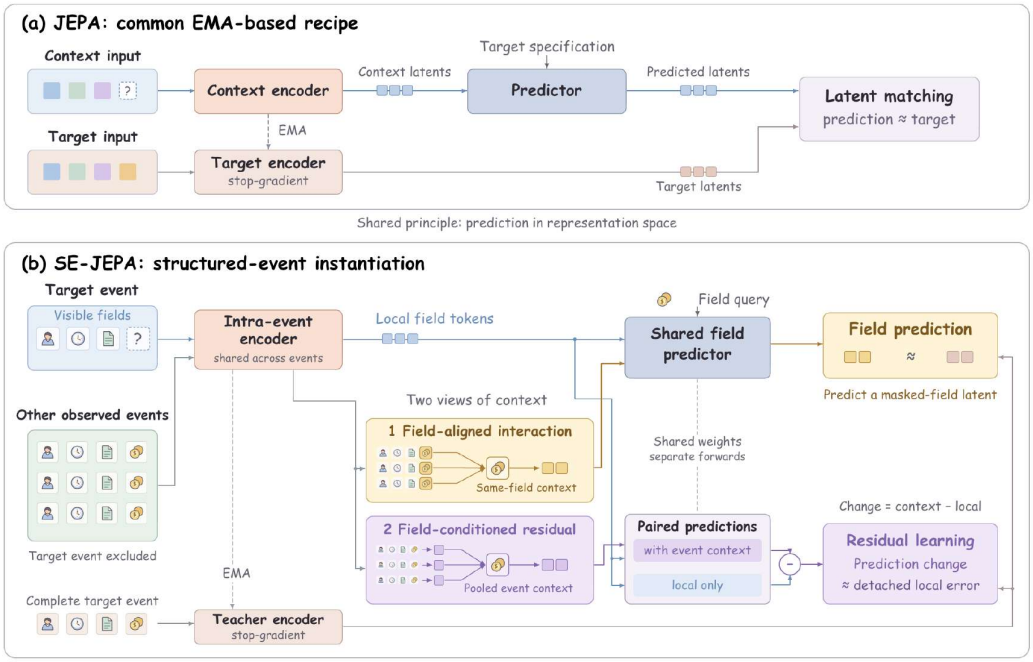

Figure 5: **SE-JEPA as a structured-event instantiation of JEPA. (a)** An EMA-based JEPA formulation learns to predict target representations from context representations. **(b)** SE-JEPA combines visible within-event fields with two granularities of cross-event context. Same-field attention supports the main latent prediction, while pooled event context supports residual auxiliary learning. The auxiliary prediction difference $\tilde{p} - p$ is matched to $\mathrm{sg}(z - p)$, where p is the local prediction and z is the teacher target. Both context memories exclude the target event. The paired predictions share predictor parameters, and the teacher encodes the complete target event. Both context mechanisms are shown enabled; gated context injection is omitted for clarity.

A Further Details of SE-JEPA

This section provides the implementation details of SE-JEPA, covering field encoding and masking, cross-event context construction, pretraining, and downstream adaptation. We use t to index a target event, s to index a context event, and f to index a field. The latent dimension is denoted by d.

A.1 Relationship to JEPA

SE-JEPA follows the joint-embedding predictive principle of learning representations by predicting target embeddings from available context (Assran et al., 2023; Thimonier et al., 2025). Figure 5(a) illustrates an EMA-based formulation: an online encoder produces context representations, a predictor estimates the specified target representations, and a target encoder provides stop-gradient learning targets. The target encoder follows the online encoder through exponential moving-average updates.

Figure 5(b) illustrates this principle for structured events. Building on within-record latent prediction (Thimonier et al., 2025), SE-JEPA combines visible target-event fields with representations of related events. Same-field cross-event attention supplies field-aligned context for the main prediction, while event-level contextual auxiliary learning supplies pooled event context for a paired residual objective. Both context reads are conditioned on the target field and the visible target-event context. The teacher encodes the complete target event to provide field-level latent targets.

A.2 Field Encoding and Mask Sampling

Each event is represented by a schema-indexed collection of numerical and categorical fields. Field identity, type, and observation status determine its token representation, while observed-field masks define the local prediction task.

Field tokens and natural missingness. Numerical values are encoded by field-specific affine maps, and categorical values use field-specific ranges of an embedding table. Each value encoding is combined with learned field-identity, field-type, and observation-state embeddings. Naturally missing values use learned type-specific representations and their corresponding observation states.

The local online encoder receives the sampled visible fields in $\mathcal{S}_t$, all of which are observed. The teacher, context encoder, and downstream encoder process complete events, retaining all field positions and their original missingness patterns. Padding events are excluded from encoding and aggregation.

The within-event encoder includes a learned register token that participates in self-attention. Field summaries and prediction targets are formed from the field tokens. Local summaries average the visible field tokens; complete-event summaries average all F field positions, including naturally missing positions. Teacher targets are taken from the field outputs after the encoder's final layer normalization. This layer normalization is the final normalization applied to the targets.

Mask sampling and field aliases. Known aliases of the same underlying quantity form a shared sampling unit, and other fields form singleton units. A unit is active when at least one of its members is observed. Units are assigned jointly to the visible or target side, and their observed members determine the resulting field masks.

An eligible target event contains at least two active units. For each eligible sequence, we sample one target event and select a nonempty visible subset, leaving at least one active unit for prediction. Each of the M target masks is sampled from the remaining units:

$$\mathcal{S}_t \cap \mathcal{T}_t^{(m)} = \varnothing, \qquad m = 1, \ldots, M. \tag{11}$$

Different target masks may overlap. The visible raw fields are gathered before value encoding. Eligibility determines which sequences contribute a pretraining prediction instance; downstream evaluation uses the benchmark-defined sample population.

Predictor inputs and context offsets. The predictor projects visible-field representations to its hidden dimension. A target query combines a learned mask token with a fixed sinusoidal encoding of its schema-field index. The projected visible tokens receive the corresponding field-index encodings. The register token is passed to the predictor with zero field-index encoding.

In $P_\psi(H, \mathcal{T}; r)$, r collects the context offsets injected into the target queries. The same-field and event-level readers supply offsets u_t and v_t, respectively. Each offset is mapped to the predictor dimension using the weight matrix of the visible-token input projection and added to its corresponding target query. The predictor jointly processes the context and target tokens and maps target outputs back to dimension d.

Loss reduction. Let $\mathcal{I}$ contain all valid target-field occurrences in a minibatch:

$$\mathcal{I} = \left\{(i, m, f) : f \in \mathcal{T}_i^{(m)}\right\}, \tag{12}$$

where i indexes sampled target events and m indexes target masks. For errors $\epsilon_{i,m,f} \in \mathbb{R}^d$, the reduction is

$$\frac{1}{d|\mathcal{I}|} \sum_{(i,m,f)\in\mathcal{I}} \|\epsilon_{i,m,f}\|_2^2. \tag{13}$$

A field appearing under multiple masks contributes once per occurrence. The main and auxiliary objectives use the same target support and normalization.

A.3 Cross-Event Context Construction

Cross-event context is constructed from accessible events within the same input sequence. Their encoded representations form field-aligned and event-level memories.

Context membership. For target event t, let $\mathcal{W}_t(X)$ denote the task-permitted observation window. The pretraining context is

$$\mathcal{C}_t = \{s \in \mathcal{W}_t(X) : s \text{ is valid}\} \setminus \{t\}. \tag{14}$$

The target event is excluded before context encoding. Each context event is encoded independently by the shared online encoder, preserving its natural missingness pattern. Padding is excluded from attention.

In the complete-sequence setting, the context may contain accessible events on both sides of the target event. For prefix-based tasks, the input sequence specifies the available observation window. Both memories are sample-local. An empty memory produces a zero context offset, leaving the local visible fields as the prediction input.

Memory construction. Let $h_{s,f}$ denote field f of context event s after within-event encoding. Same-field memory contains

$$\mathcal{M}^{\mathrm{sf}}_{t,f} = \{h_{s,f} : s \in \mathcal{C}_t\}. \tag{15}$$

Each token retains its field identity and incorporates information from within-event field interactions.

The event-level reader uses pooled event representations:

$$m_s = \frac{1}{F}\sum_{j=1}^{F} h_{s,j}, \qquad \mathcal{M}^{\mathrm{ev}}_t = \{m_s : s \in \mathcal{C}_t\}. \tag{16}$$

Its target-field query combines the visible local summary with learned field-identity and field-type embeddings. The same-field query combines the visible local summary with the target-field identity embedding.

Gated context injection. The readers have separate attention parameters, η_{sf} and η_{ev}, shared across fields within each reader. Their outputs are scaled by learned signed gates:

$$\begin{aligned} u_{t,f} &= a\tanh(\gamma_{\mathrm{sf}})c^{\mathrm{sf}}_{t,f}, \\ v_{t,f} &= \tanh(\gamma_{\mathrm{ev}})c^{\mathrm{ev}}_{t,f}, \end{aligned} \tag{17}$$

where $a \in \{0, 1\}$ is fixed by the model configuration.

The main prediction uses u_t. The auxiliary objective pairs a prediction using v_t with the zero-offset local prediction. The two context computations update the shared encoder and predictor through their respective objectives.

A.4 Pretraining Procedure

Algorithm 1 summarizes pretraining. Each update samples target events and field masks, constructs teacher targets and context memories, and optimizes the joint prediction objective.

Initialization and teacher updates. Context-augmented pretraining starts from a local checkpoint C_0 containing the online encoder, teacher encoder, and field predictor. Newly introduced context readers use zero-initialized gates. The local pretraining stage and the subsequent context-augmented stage together determine the total pretraining budget.

The teacher remains in evaluation mode, and its outputs are computed without gradients. After each online optimization step, its parameters are updated as

$$\bar{\theta} \leftarrow \mu_k\bar{\theta} + (1 - \mu_k)\theta. \tag{18}$$

EMA tracks the within-event encoder. The teacher constructs targets from the complete target event.

Algorithm 1 SE-JEPA Pretraining

Require: Unlabeled data $\mathcal{D}$; local checkpoint C_0; updates K; target masks M; same-field switch a; auxiliary weight λ
Ensure: Online encoder and enabled same-field reader

Restore $(\theta, \bar{\theta}, \psi)$ from C_0
Initialize enabled readers and set $\gamma_{\mathrm{sf}} = \gamma_{\mathrm{ev}} = 0$
$\Theta \leftarrow$ online encoder, predictor, and enabled reader parameters
Keep $E_{\bar{\theta}}$ in evaluation mode
for $k = 1, \ldots, K$ **do**
  Sample eligible target events $\{(X_i, t_i)\}_{i=1}^{B}$ from $\mathcal{D}$
  for $i = 1, \ldots, B$ **do**
    Sample visible fields $\mathcal{S}_i$ and targets $\{\mathcal{T}_i^{(m)}\}_{m=1}^{M}$ ▷ Group-disjoint from $\mathcal{S}_i$
    $\mathcal{C}_i \leftarrow \{s \in \mathcal{W}_{t_i}(X_i) : s \text{ valid}\} \setminus \{t_i\}$
    $z_{i,f} \leftarrow \mathrm{sg}([E_{\bar{\theta}}(x_{i,t_i})]_f)$
    $H_i^{\mathrm{main}} \leftarrow E_\theta(x_{i,t_i}|_{\mathcal{S}_i})$; $u_i \leftarrow 0$
    if $a = 1$ **then**
      $u_i \leftarrow \tanh(\gamma_{\mathrm{sf}})\, \mathrm{SameFieldRead}_{\theta,\eta_{\mathrm{sf}}}(H_i^{\mathrm{main}}, X_{i,\mathcal{C}_i})$
    end if
    if $\lambda > 0$ **then**
      $H_i^{\mathrm{aux}} \leftarrow E_\theta(x_{i,t_i}|_{\mathcal{S}_i})$
      $v_i \leftarrow \tanh(\gamma_{\mathrm{ev}})\, \mathrm{EventRead}_{\theta,\eta_{\mathrm{ev}}}(H_i^{\mathrm{aux}}, X_{i,\mathcal{C}_i})$
    end if
    for $m = 1, \ldots, M$ **do**
      $\hat{z}_{i,m} \leftarrow P_\psi(H_i^{\mathrm{main}}, \mathcal{T}_i^{(m)}; u_i)$
      if $\lambda > 0$ **then**
        $(p_{i,m}, \tilde{p}_{i,m}) \leftarrow \mathrm{PairedPredict}_\psi(H_i^{\mathrm{aux}}, \mathcal{T}_i^{(m)}; 0, v_i)$ ▷ Matched dropout
      end if
    end for
  end for
  $\mathcal{I} \leftarrow \{(i, m, f) : f \in \mathcal{T}_i^{(m)}\}$
  $\mathcal{L}_{\mathrm{pred}} \leftarrow \dfrac{1}{d|\mathcal{I}|} \displaystyle\sum_{(i,m,f)\in\mathcal{I}} \|\hat{z}_{i,m,f} - z_{i,f}\|_2^2$
  $\mathcal{L}_{\mathrm{res}} \leftarrow 0$
  if $\lambda > 0$ **then**
    $\mathcal{L}_{\mathrm{res}} \leftarrow \dfrac{1}{d|\mathcal{I}|} \displaystyle\sum_{(i,m,f)\in\mathcal{I}} \|(\tilde{p}_{i,m,f} - p_{i,m,f}) - \mathrm{sg}(z_{i,f} - p_{i,m,f})\|_2^2$
  end if
  Update Θ with AdamW on $\mathcal{L}_{\mathrm{pred}} + \lambda\mathcal{L}_{\mathrm{res}}$
  $\bar{\theta} \leftarrow \mu_k\bar{\theta} + (1 - \mu_k)\theta$
end for
return online encoder and enabled same-field reader

Paired residual prediction. The main prediction uses local representations H_t^{main} and the same-field offset. The auxiliary computation uses a separate online forward H_t^{aux} and constructs

$$\begin{aligned} p_t^{(m)} &= P_\psi(H_t^{\mathrm{aux}}, \mathcal{T}_t^{(m)}; 0), \\ \tilde{p}_t^{(m)} &= P_\psi(H_t^{\mathrm{aux}}, \mathcal{T}_t^{(m)}; v_t). \end{aligned} \tag{19}$$

The pair reuses the same local representations. The predictor RNG state is saved before the local call and restored before the contextual call, matching their dropout realizations. The main prediction uses its own online forward.

The residual target is detached:

$$\delta_t^{(m)} = \mathrm{sg}\big(z_t - p_t^{(m)}\big). \tag{20}$$

Both predictions in $\tilde{p}_t^{(m)} - p_t^{(m)}$ remain differentiable. Appendix D derives the resulting gradient and its behavior at initialization.

Algorithm 2 SE-JEPA Downstream Adaptation

Require: Pretrained encoder θ; same-field parameters $(\eta_{\mathrm{sf}}, \gamma_{\mathrm{sf}})$ when enabled; switch a; labeled data $\mathcal{D}_{\mathrm{sup}}$; evaluation protocol
Ensure: Downstream predictor $\widehat{y}(X)$
Retain the online encoder and enabled same-field reader
Initialize readout ρ and affine task head W with $W_R = 0$ and $w = 0$
$\Omega \leftarrow$ retained encoder parameters and ρ
function ENCODE($X; \Omega$)
    $H_t \leftarrow E_\theta(x_t)$ for $t \in \mathcal{V}(X)$
    $\widetilde{H} \leftarrow H$
    if $a = 1$ **then**
        $\widetilde{H} \leftarrow \mathrm{SameFieldInteraction}(H; \eta_{\mathrm{sf}}, \gamma_{\mathrm{sf}})$ ▷ Eq. equation 21
    end if
    $g_t \leftarrow F^{-1} \sum_{f=1}^{F} \widetilde{h}_{t,f}$ for $t \in \mathcal{V}(X)$
    return $\mathrm{Readout}_\rho(\{g_t\}; X)$
end function
if frozen evaluation **then**
    Freeze Ω and set its modules to evaluation mode
    Cache $\mathrm{Encode}(X; \Omega)$ without gradients
    Optimize only W on labeled training representations
else
    Jointly optimize (Ω, W) on $\mathcal{L}_{\mathrm{task}}(\mathrm{Linear}_W(\mathrm{Encode}(X; \Omega)), y)$
end if
Select the checkpoint using early-stopping validation data
return $\widehat{y}(X) = \mathrm{Linear}_W(\mathrm{Encode}(X; \Omega))$

A.5 Downstream Representation and Adaptation

Downstream encoding uses the pretrained within-event encoder and the same-field attention module when enabled. Algorithm 2 specifies frozen evaluation and full fine-tuning.

Transferring the context operator. The learned same-field attention operator uses complete field representations as downstream queries. For the complete observation window, its memory contains all valid events, including the query event:

$$\begin{aligned} \widetilde{h}_{t,f} &= h_{t,f} + a \tanh(\gamma_{\mathrm{sf}}) \, \mathrm{Attn}_{\eta_{\mathrm{sf}}} \left(h_{t,f}, \{ h_{s,f} : s \in \mathcal{V}(X) \} \right), \\ g_t &= \frac{1}{F} \sum_{f=1}^{F} \widetilde{h}_{t,f}, \end{aligned} \tag{21}$$

where $\mathcal{V}(X)$ denotes valid events in the input. The input window follows the benchmark's information availability constraints.

Event readout. The downstream readout applies a shared event-wise residual transformation:

$$\bar{g}_t = g_t + W_R G_\omega \left(g_t + \mathrm{PE}(\pi_t) + W_\Delta \Delta_t \right). \tag{22}$$

Here, π_t is the zero-based rank among valid events, PE is a sinusoidal position encoding, and Δ_t is the time-gap feature supplied by the dataset adapter. G_ω applies the same single-token Transformer layer independently to each event.

Attention pooling gives

$$\begin{aligned} \alpha_t &= \frac{\exp(w^\top \bar{g}_t / \sqrt{d})}{\sum_{s \in \mathcal{V}(X)} \exp(w^\top \bar{g}_s / \sqrt{d})}, \\ g(X) &= \sum_{t \in \mathcal{V}(X)} \alpha_t \bar{g}_t. \end{aligned} \tag{23}$$

A linear task head maps $g(X)$ to task logits. Padding is excluded throughout. Position and time-gap inputs specify temporal attributes at readout, while same-field attention provides content-based cross-event interaction.

Frozen evaluation. For frozen evaluation, the readout projection and pooling vector are initialized as $W_R = 0$ and $w = 0$. The fixed representation therefore reduces to

$$g(X) = \frac{1}{|\mathcal{V}(X)|} \sum_{t \in \mathcal{V}(X)} g_t. \tag{24}$$

The encoder, enabled same-field reader, and readout remain in evaluation mode. Representations are extracted without gradients, and only an affine task head is fitted. With this initialization, the explicit position and gap terms in the readout contribute zero.

Full fine-tuning. Full fine-tuning updates the retained encoder, enabled same-field attention and gate, event readout $(G_\omega, W_R, W_\Delta, w)$, and task head. As W_R is optimized, the position and gap inputs can contribute to the event readout.

B Experimental Details

This section specifies the benchmark construction, baseline adaptations, optimization settings, and evaluation protocols used in our experiments.

B.1 Benchmarks and Data Preparation

We evaluate on four public benchmarks covering banking activity, credit risk, retail purchases, and business processes. Each benchmark associates a structured-event sequence with a sample-level classification target.

MBD-mini. MBD-mini is derived from the Multimodal Banking Dataset (Mollaev et al., 2025). Each example contains a customer's observed event history and four binary labels indicating purchases of banking products in the following month. We use the mini release and its predefined folds: folds 0–2 for training, fold 3 for validation, and fold 4 for testing. A deterministic customer-level hash divides fold 3 into early-stopping and hyperparameter-selection subsets. All examples from the same customer share a validation assignment. The primary metric is the unweighted mean of the four label-specific AUROCs.

AlphaBattle. AlphaBattle (Boosters, n.d.) concerns credit-default prediction from transaction histories. We use applications with available training labels and their associated transactions. Applications are sorted by identifier and divided into contiguous training, early-stopping, hyperparameter-selection, and test partitions with proportions of 70%, 10%, 10%, and 10%, respectively. The input features exclude the `product` field. The primary metric is AUROC.

Retail/X5. Retail/X5 follows the customer age-prediction task used in EBES (Osin et al., 2025a). Each customer is represented by a time-ordered purchase sequence and assigned to one of four age groups: 10–34, 35–44, 45–59, and 60–90. Event inputs contain the available numerical and categorical purchase attributes. We reuse the customer assignments in the prepared benchmark split and report four-class classification accuracy.

BPI17. BPI17 (van Dongen, 2017) contains event logs from a loan-application process. Each example is a sequence prefix ending at an `O_Create Offer` event, and the target indicates whether the offer is accepted. Inputs contain information available through that prefix. The outcome fields `Accepted` and `Selected`, together with `OfferID`, `EventID`, and the case identifier, are excluded from predictive features. The test partition contains the designated offer prefixes dated on or after 20 October 2016. The primary metric is AUROC.

B.2 Baseline Implementations and Adaptations

We evaluate eleven baselines spanning aggregation-based models, supervised sequence encoders, contrastive and reconstructive pretraining, and JEPA-based learning. For each method, we describe its structured-event input representation, learning objective, and downstream readout.

Aggregation-based tabular models. XGBoost-Agg (Chen & Guestrin, 2016), CatBoost-Agg (Prokhorenkova et al., 2018), and MLP-Agg operate on fixed-dimensional sequence summaries. The aggregated features include event counts, numerical summary statistics, categorical occurrence statistics, and event-timing summaries where available. XGBoost and CatBoost fit gradient-boosted decision trees, while MLP-Agg uses a feed-forward neural network trained with the task-specific classification loss. The suffix "-Agg" denotes sequence-level feature aggregation.

Supervised sequence models. GRU (Cho et al., 2014) and Transformer (Vaswani et al., 2023) embed numerical and categorical event fields and encode the resulting event sequence. The Transformer uses causal attention and padding masks. A sequence readout and task head produce sample-level predictions. Both models are trained from initialization using the labeled training partition.

Contrastive and reconstructive pretraining. CoLES (Babaev et al., 2022a) encodes random contiguous subsequences with a shared GRU. Positive pairs originate from the same entity, and negative pairs originate from different entities. Training uses a margin-based contrastive objective with hard-negative selection.

MLEM (Moskvoretskii et al., 2024) combines generative and contrastive learning. We connect its public implementation to the benchmark through an event-sequence adapter that preserves event order and exports numerical and categorical fields. Event time is represented by position in the retained sequence. The reported MLEM results cover AlphaBattle and BPI17.

TabBERT (Padhi et al., 2021b) uses field-level and event-level Transformer encoders and is pretrained by predicting masked field values. Each pretrained encoder is evaluated through frozen linear evaluation and full fine-tuning.

JEPA-based pretraining. T-JEPA (Thimonier et al., 2025) is applied to individual events. Its predictor estimates target-field teacher representations from visible fields, and the teacher follows the online encoder through EMA updates. Event representations are averaged to form the downstream sequence representation.

CF-JEPA (Lee & Sim, 2026b) uses its public encoder and forward-prediction implementation with dense event inputs comprising numerical values, observation indicators, scaled categorical identifiers, temporal features, and event-validity indicators.

HEPA (Petersen et al., 2026) uses its public prediction module with a structured-event tokenizer. It samples context prefixes and future segments, conditions prediction on the horizon, and combines latent alignment with variance and covariance regularization.

B.3 Architecture and Optimization Settings

SE-JEPA uses field-specific input encodings, a within-event encoder, a field predictor, and the enabled context readers. Pretraining optimizes the online modules with AdamW. The teacher follows the update in Eq. equation 18. The context-augmented stage restores the local checkpoint described in Appendix A.4.

For supervised adaptation, binary and multilabel tasks use binary cross-entropy with logits. Retail/X5 uses multiclass cross-entropy. The controlled component ablations use the settings in Table 3.

Model configurations. The same-field switch a and auxiliary weight λ define the configurations in Table 4. They share the within-event encoder and teacher-target definition. Same-field attention changes the main prediction condition, and event-level auxiliary learning adds the paired residual objective.

C Supplementary Evaluation Protocols

This section details the label-efficiency and representation visualization protocols accompanying the experimental results.

Table 3: Optimization settings for the controlled component ablations. The auxiliary weight applies to configurations with event-level contextual auxiliary learning.

Setting	Value
Downstream optimizer	AdamW
Learning rate	3×10^{-4}
Weight decay	10^{-4}
Effective batch size	256
Maximum downstream updates	10,000
Validation interval	500 updates
Early-stopping patience	8 validation checks
Auxiliary loss weight	0.05

Table 4: SE-JEPA configurations. The enabled same-field reader is retained during transfer. The event-level auxiliary reader is used during pretraining.

Configuration	Same-field switch a	Auxiliary weight λ
Local only	0	0
Same-field context	1	0
Auxiliary context	0	> 0
Joint context	1	> 0

C.1 Label Efficiency

Label-efficiency evaluation uses 20%, 40%, 60%, 80%, and 100% of the labeled training entities. Each method starts from a fixed pretrained checkpoint for a given dataset. Downstream training is repeated with three random seeds, and the curves report mean validation performance. Shaded regions indicate one sample standard deviation.

C.2 Representation Visualization

We visualize the embeddings immediately before the classifier after full fine-tuning using three-dimensional t-SNE. The visualization uses a class-balanced validation subset with 200 examples per class: 400 examples for the first MBD-mini target and 800 examples for Retail/X5. These plots describe the geometry of the fine-tuned representations on the selected validation cohort.

D Analytical Properties of Residual Context Learning

This section characterizes the optimization behavior of the paired residual objective. We first derive its gradient and then analyze the prediction and gradient structure at zero-gate initialization. The analysis considers one valid target-field occurrence; the minibatch result follows by averaging.

D.1 Gradient of the Paired Residual Objective

Let $p, \tilde{p} \in \mathbb{R}^d$ denote the local and contextual predictions, respectively, and let $z \in \mathbb{R}^d$ denote the detached teacher target. For online parameters ϑ, write $J_\vartheta p$ and $J_\vartheta \tilde{p}$ for their Jacobians.

Proposition 1: Paired residual gradient. For

$$\ell_{\mathrm{res}} = \frac{1}{d} \left\| (\tilde{p} - p) - \mathrm{sg}(z - p) \right\|_2^2, \tag{25}$$

the online gradient is

$$\nabla_\vartheta \ell_{\mathrm{res}} = \frac{2}{d} \left(J_\vartheta \tilde{p} - J_\vartheta p \right)^\top (\tilde{p} - z). \tag{26}$$

Proof. Define

$$e = (\tilde{p} - p) - \mathrm{sg}(z - p). \tag{27}$$

Its forward value is $e = \tilde{p} - z$. The stop-gradient operator gives

$$J_\vartheta e = J_\vartheta \tilde{p} - J_\vartheta p. \tag{28}$$

Applying the chain rule to $d^{-1} e^\top e$ yields Eq. equation 26. □

The objective's forward value equals $d^{-1}\|\tilde{p} - z\|_2^2$, while its gradient contains the contextual–local Jacobian difference. This difference links the auxiliary update to the effect of context on the paired prediction. Maintaining differentiability through both predictions in $\tilde{p} - p$ implements this gradient.

D.2 Behavior at Zero-Gate Initialization

Consider the auxiliary offset

$$v = \tanh(\gamma) c_{\theta,\eta}, \tag{29}$$

where η denotes the event-level reader parameters and $c_{\theta,\eta}$ may also depend on the online encoder. Write the paired predictions as

$$\begin{aligned} p &= P_\psi(H_\theta, \mathcal{T}; 0), \\ \tilde{p} &= P_\psi(H_\theta, \mathcal{T}; v). \end{aligned} \tag{30}$$

The pair uses identical local activations and predictor randomness. In this subsection, γ denotes γ_{ev}.

Proposition 2: Initial prediction and gradient structure. At $\gamma = 0$, the paired predictions coincide:

$$\tilde{p} = p. \tag{31}$$

For shared parameters $\xi = (\theta, \psi)$ and reader parameters η, the auxiliary-loss gradients satisfy

$$\nabla_\xi \ell_{\mathrm{res}} = 0, \qquad \nabla_\eta \ell_{\mathrm{res}} = 0. \tag{32}$$

The gate derivative is

$$\left.\frac{\partial \ell_{\mathrm{res}}}{\partial \gamma}\right|_{\gamma=0} = \frac{2}{d}(p - z)^\top \left.\frac{\partial P_\psi(H_\theta, \mathcal{T}; v)}{\partial v}\right|_{v=0} c_{\theta,\eta}. \tag{33}$$

Proof. Since $\tanh(0) = 0$, the contextual offset is zero. Matched predictor randomness then gives $\tilde{p} = p$. For any shared parameter in ξ, the derivative through $c_{\theta,\eta}$ is multiplied by $\tanh(0)$. Consequently,

$$J_\xi \tilde{p} = J_\xi p. \tag{34}$$

Equation equation 26 gives the first identity in Eq. equation 32.

Similarly, $\partial v / \partial \eta = 0$ at initialization, and the local prediction is independent of η. This gives the second identity.

For the gate,

$$\left.\frac{\partial v}{\partial \gamma}\right|_{\gamma=0} = c_{\theta,\eta}, \tag{35}$$

because $\tanh'(0) = 1$. Substitution into Eq. equation 26 yields Eq. equation 33. □

Zero-gate initialization therefore preserves the local prediction while allowing the gate to receive a learning signal. The main objective continues to train shared parameters. The identities above describe the auxiliary loss contribution to the gradient; optimizer-level weight decay is applied separately.

The same-field gate also starts at zero, so its initial main prediction equals the local prediction. As the gates are optimized, the context readers contribute through their learned offsets. "'