Beyond Logs: AI's Internal Representations as the New Process Evidence

Aleksandar Gavric(✉), Dominik Bork(✉), and Henderik A. Proper(✉)

Business Informatics Group, TU Wien, Vienna, Austria
{aleksandar.gavric,dominik.bork,henderik.proper}@tuwien.ac.at

Abstract. Traditional process mining relies on symbolic event logs that represent activities as discrete labels, often overlooking the rich contextual and semantic nuances found in real-world data such as textual reports, visual records, or sensor outputs. In this paper, we propose a paradigm shift: using the internal representations of AI models—embedding spaces learned from data—as the foundation for process mining. Our framework performs both process discovery and conformance checking directly in these continuous vector spaces, enabling the detection of semantically similar yet lexically divergent events.

We evaluate our approach along three dimensions: *(i)* whether embedding-based discovery maintains or improves accuracy over symbolic baselines, *(ii)* whether multimodal sources such as video and audio can be processed as unified embeddings for mining purposes, and *(iii)* whether conformance checking in embedding space enables alignment across noisy or semantically perturbed traces. By treating AIs internal representations as a novel form of process evidence, we show how process mining can move beyond traditional logs and unlock deeper, semantically enriched interpretations of real-world workflows.

Keywords: Embedding Space · Internal Representations · AI Interpretability · Semantic Event Matching · Multimodal Data

1 Introduction

Process models serve as instruments in describing and analyzing complex workflows, ranging from administrative procedures to large-scale industrial operations [1]. Over time, process modeling has shifted from *confidence-based* approaches, which rely heavily on expert opinions, to *evidence-based* approaches underpinned by operational data. This transformation has culminated in increasingly refined methods such as object-centric process discovery [26], which captures interactions among multiple entities rather than restricting itself to sequences of single activities. Within the broader domain of Business Process Management [19], process mining encompasses several core tasks, including process discovery [18], and conformance checking [23]. Process discovery infers a

M. ER et al. (Eds.): BPM 2025, LNBIP 565, pp. 232–246, 2026.
https://doi.org/10.1007/978-3-032-02936-2_17

descriptive model directly from event logs, while conformance checking evaluates the alignment between recorded and expected process behaviors. These methodologies are becoming increasingly relevant in light of rapid digital transformations: the global market for process mining is projected to reach USD 46.39 billion by 2032 [7].

Originally limited to structured text-based logs, process mining now extends into more complex domains including sensor data [2] and multimedia sources such as video streams [9]. Efforts are underway to merge artificial intelligence (AI) with process mining so that event logs become richer and process mining algorithms get applicable to unstructured data. Linking AI and process mining in this manner promises greater expressiveness, as real-world events become interpretable within structured representations. At the prompting level, techniques such as few-shot learning and chain-of-thought reasoning enhance process mining by guiding AI models to generate more context-aware and interpretable insights. However, a more robust approach involves using **embedding spaces**, which capture semantic associations in a structured manner.

Recent developments indicate that problem-solving can be carried out directly in an embedding space [17]. Simultaneously, approximate process discovery [6] and approximate conformance checking [11] suggest that process mining tasks can be made more efficient by partially offloading computation to vector representations. While current techniques encode inputs to vectors, they do not leverage the rich semantics present in embedding spaces learned by large pre-trained models.

An important open question emerges here: if mathematical and logical reasoning can be performed in embedding spaces [17], is it possible to execute key process mining operations—namely process discovery and conformance checking—directly in these spaces as well? Our work addresses this question by proposing a *framework that relaxes process mining tasks to function on AI embeddings*. We hypothesize that this method can *i*) exceed the accuracy of symbolic-only approaches, *ii*) handle real-world unstructured multimodal data as event logs, and *iii*) can utilize knowledge enraptured in AI models for process mining tasks. Our research questions (RQ) are:

- **RQ1:** Can embedding-based representations of events maintain or improve the accuracy of process discovery compared to traditional symbolic techniques?
- **RQ2:** To what extent can AI embeddings serve as a unified representation for events extracted from multimodal sources (e.g., video, text), enabling process mining on unstructured data?
- **RQ3:** Does reasoning in embedding space allow for more intuitive conformance checking by identifying semantically similar yet lexically different events?

As organizations increasingly seek to mine processes from raw data-text, images, and beyondour approach has the potential to bridge the gap between the *representational power of AI models* and *the practical needs of process mining tasks.*

The remainder of this paper is organized as follows. First, we review the related work on approximate process mining and embedding-space operations (Sect. 2). Next, we present our general view for relaxed process mining (Sect. 3) and a proof of concept (Sect. 4), followed by an evaluation (Sect. 5). We conclude in Sect. 6. The supplementary material for this paper is available in our GitHub repository[1].

2 Related Work

In this section, we review the literature that underpins our research. We first discuss the use of AI in process mining and approximate process mining techniques, which relax traditional computational requirements to improve efficiency and generality. We then turn to studies on embedding-space operations, where embedding-based representations enable approaches to data analysis and reasoning.

Recent work has demonstrated the potential of large language models in process mining through prompting-based techniques. For instance, Neuberger et al. [20] propose a universal prompting strategy to extract process model information from natural language text, illustrating how AI can be used to interpret complex textual descriptions of processes. Similarly, Rebmann et al. [22] evaluate the capacity of such models to address semantics-aware process mining tasks. AI can be used for process mining tasks directly via prompting, without the need for approximation methods or reliance on vector-based representations.

Approximate Process Mining. Some process mining approaches aim at reducing computational complexity or improving generalization by using *vector representations*, also referred to as *latent space representations*, *embeddings*, or *encodings*. The term *encoding* can also refer to non-vector encodings (cf. this review of visual encodings in process mining tools [15]). Investigations into vector-encoding techniques are found in work such as [10] that introduces a stakeholder-specific, jargon-based representation of multimodal business process data, [4] that demonstrates the use of LSTM models for learning accurate business process representations, the SMT-based encoding of process discovery problems [25], and outcome-oriented predictive process monitoring through image encoding [21]. Further investigations into encoding techniques related to process mining are found in the comparative study of trace encoding methods [24]. Encoding of process mining data led towards *approximate* methods to address the computational challenges inherent in exact process discovery and conformance checking. Therefore, van Detten et al. [6] present an approximate inductive miner to derive process models, while Gianola et al. [11] develop methods for fast approximate conformance checking via probabilistic alignments.

Embedding-space Operations. Embedding spaces were already in use in the 1970 s [16], and are fundamental to modern data analysis, enabling the conversion of complex, multimodal data into a compact (latent) representation

[1] Supplementary material: https://github.com/aleksandargavric/relaxed-pm.

that simplifies further processing and inference. For example, a study investigates mathematical reasoning within these embedding spaces, revealing how abstract representations can encapsulate logical relationships and support computational reasoning [17]. Another work designs optimized clustering embedding spaces, demonstrating that deep representations can be used effectively to uncover inherent data structures [5]. Expanding on these ideas, a largely trained unified embedding space harmonizes data from diverse modalities [12], enhancing the integration of heterogeneous information into a single embedding space. Crossmodal embedding techniques tailored for dynamic tasks have shown that specialized embedding spaces can significantly improve action learning capabilities [13]. Moreover, the concept of operational embedding spaces has been examined in [14], illustrating their practical utility in a range of applications by providing efficient and flexible representations of complex operations.

Our Position. On one hand, the concept of approximate process mining has been advanced by using vector representations to overcome the limitations of exact process discovery and conformance checking. Our approach extends vector-space inputs by adopting semantically-rich embedding spaces. On the other hand, embedding space operations have been recognized for their ability to encapsulate complex data into compact representations that support robust inference and reasoning. We aim to allow process mining to operate in an embedding space to enhance contextual mappings between real-world observed behaviors and their underlying processes.

3 Relaxed Process Mining: The General Perspective

Building on the premise that process-relevant information can be encoded in the latent spaces of modern AI models, we present a general perspective that extends classical process mining and conformance checking methods to operate within an embedding space. Traditional process mining algorithms, such as the inductive miner, and conformance checking techniques are typically defined over discrete events with crisp boundaries. However, real-world data is often noisy, heterogeneous, and semantically ambiguousconditions under which symbolic techniques struggle. To overcome these limitations, we introduce a *relaxed* approach in which each event is represented by a continuous embedding $e \in \mathbb{R}^n$ derived from *ImageBind* [12], and similarities between events are assessed over a neighborhood defined by the ball $B(e, \delta) = \{x \in \mathbb{R}^n : \|x - e\| \leq \delta\}$, where $\delta > 0$ controls the tolerance for noise and variability. Our perspective accommodates semantic equivalence across synonyms by considering sets of embeddings for different words and by ensuring that events with embeddings within a distance δ are treated as equivalent. Moreover, in the context of multimodal event logs derived from video data, multiple frames or scenes that are similar enough are aggregated into single activities using the same relaxed notion of similarity.

The first part of our perspective develops a relaxed process discovery algorithm that recursively partitions the event log into subprocesses based on relaxed cuts, where each cut is validated by checking that events in one partition have corresponding events in the other within a prescribed distance. The second part extends this methodology to conformance checking by aligning observed traces with a modeled process; here, each observed event is matched against a process activity by minimizing a cost function based on the Euclidean distance in the embedding space, with an allowance for skipped events at a fixed penalty.

Definition 1 (Relaxed Process Mining). *Let* $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$ *be an* event log*, where each event* e_i *is represented by an embedding in* $\mathbb{R}^n$*. In* relaxed process mining*, events are treated as* similar *(and thus potentially interchangeable) if their embeddings lie within a specified distance* $\delta > 0$*, i.e., if* $|e_i - e_j| \leq \delta$*. This relaxed notion of equivalence supports:*

1. ***Semantic variability.*** *Multiple words or labels that share similar embeddings (e.g., synonyms) are considered equivalent.*
2. ***Multimodal data.*** *Events derived from video, images, or audio are compared in a continuous space; frames or segments that are sufficiently close in this space are grouped as one activity.*
3. ***Noise tolerance.*** *Small perturbations in event embeddings do not alter the process structure, mitigating issues arising from measurement or labeling noise.*

Hence, relaxed process mining *generalizes classical process mining tasks (such as discovery, and conformance checking) to operate over neighborhoods in the embedding space rather than strict, discrete labels.*

Definition 2 (Relaxed Process Discovery). Relaxed process discovery *is the specialization of relaxed process mining to the task of constructing a* process model *(e.g., a Petri net, BPMN diagram, or process tree) from an event log* $\mathcal{E} \subseteq \mathbb{R}^n$*. Formally, given:*

- *A threshold parameter* $\delta > 0$ *defining when two embeddings are considered equivalent,*
- *An optional noise tolerance parameter* $\alpha \in [0, 1]$ *allowing a fraction of events to be disregarded,*

a relaxed process discovery *algorithm partitions and clusters events in* $\mathcal{E}$ *by grouping those whose embeddings lie within distance* δ *of a representative reference. These groups are then used to induce high-level subprocesses and control-flow relations (e.g., sequence, parallelism, choice) in the resulting model. Events or groups of events failing to meet the relaxed equivalence criteria may be treated as noise (at most* $\alpha \cdot 100\%$ *of the log) or assigned a penalty-based cost. The final outcome is a process model that tolerates variation in labels or multimodal data representations.*

Definition 3 (Relaxed Conformance Checking). *Let* $T = \langle e_1, e_2, \ldots, e_m \rangle$ *be an observed* trace *whose events are represented by embeddings in* $\mathbb{R}^n$*, and let* $\mathcal{M} = \langle r_1, r_2, \ldots, r_L \rangle$ *be a* modeled process *whose activities are also represented by embeddings in* $\mathbb{R}^n$*. A relaxation parameter* $\delta > 0$ *specifies when an event is* *considered close enough to match a modeled activity; that is, if* $|e_i - r_k| \leq \delta$*, then* e_i *and* r_k *are deemed equivalent.*

A relaxed alignment *is a function*

$$\mathcal{A}\colon \{e_1, e_2, \ldots, e_m\} \;\rightarrow\; \{r_1, r_2, \ldots, r_L\} \cup \{\epsilon\},$$

where each observed event is assigned either to one of the modeled activities or to a skip symbol ϵ *(representing a deviation). Let* $\gamma > 0$ *be a fixed penalty for skipping an event. The* cost *of aligning* e_i *with* r_k *is*

$$c(e_i,\, r_k) \;=\; \begin{cases} |\, e_i - r_k|, & \textit{if } |\, e_i - r_k| \leq \delta, \\ +\infty, & \textit{otherwise}, \end{cases}$$

and the cost of skipping an event e_i *is* $c(e_i, \epsilon) = \gamma$*. The total* alignment cost *is*

$$C(\mathcal{A}) \;=\; \sum_{i=1}^{m} c\big(e_i,\, \mathcal{A}(e_i)\big),$$

where $\mathcal{A}$ *must preserve the order of events in* T*; that is, if* e_i *is aligned to* r_k *and* e_j *to* $r_{k'}$ *with* $i < j$*, then* $k \leq k'$*. The* optimal relaxed alignment $\mathcal{A}^*$ *is the one minimizing* $C(\mathcal{A})$ *over all valid alignments.*

Given a normalization constant C_{max} *(e.g.,* $C_{max} = m \cdot \max\{\delta, \gamma\}$*), the* relaxed fitness *of trace* T *with respect to the model* $\mathcal{M}$ *is*

$$F(T) \;=\; 1 \;-\; \frac{C(\mathcal{A}^*)}{C_{max}}.$$

If $F(T) = 1$*, then the observed trace perfectly fits the model within distance* δ*.*

Moreover, a noise tolerance $\alpha \in [0, 1]$ *can be specified so that if the fraction of events that cannot match* any *modeled activity is no more than* α*, the trace is still deemed acceptable. Formally, if*

$$\frac{\Big|\{\, e_i \in T : \forall\, k,\ |\, e_i - r_k| > \delta\}\Big|}{|T|} \;\leq\; \alpha,$$

then T *is accepted as conformant despite the existence of partial mismatches. This framework readily accommodates synonyms or multiple-frame video data by treating sets of embeddings representing the same concept or scene as equivalent under the relaxed matching criterion* $|\, e_i - r_k| \leq \delta$*.*

4 Proof of Concept: Custom Solution for Relaxed Process Mining

After defining Relaxed Process Mining in Sect. 3, we develop a custom solution to demonstrate a proof of concept. Our approach is applied on both traditional text-first and multimodal-first event logs. The former maintains compatibility

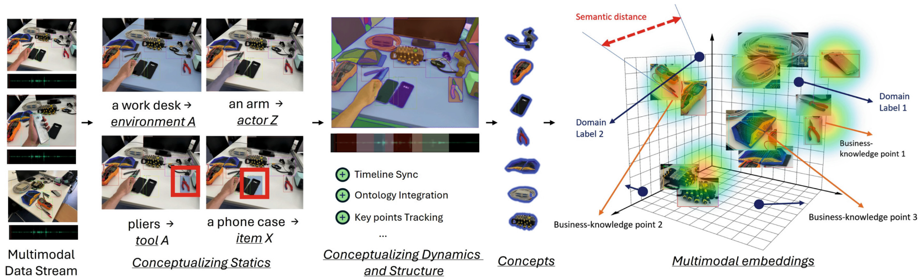

Fig. 1. Overview of the end-to-end pipeline: From multimodal observations and static concept mapping to temporal-semantic alignment and embedding-based reasoning with respect to business knowledge anchors.

with established benchmarks, while the latter (comprising videos with images and audio) showcases the methods capabilities in an emerging field where standardized benchmarks are still under development.

To illustrate why our event logs can easily be multimodal data, we provide an illustration in Fig. 1 that presents our overall pipeline. The proposed pipeline begins with a *multimodal data stream*—including video, audio, and sensor traces—from which static entities such as *actor Z*, *tool A*, or *item X* are extracted via object detection and mapped to conceptual categories (*Conceptualizing Statics*). A semantic alignment layer then performs *Timeline Sync*, *Ontology Integration*, and *Key Points Tracking* to organize these elements across time and meaning, enabling *Conceptualizing Dynamics and Structure*. The resulting representations are embedded into a joint *concept space*, where observed entities are aligned with curated *business knowledge points*, allowing reasoning over *semantic distances* and classification into *domain labels*. As in Fig. 1, various input sources—such as video frames, audio signals, and text annotations—are first detected or segmented (e.g., identifying a phone, a pair of pliers, and a hand in the scene). These individual elements are then converted into continuous embeddings using a unified embedding model, thereby mapping potentially diverse data (e.g., speech, visuals, labels) into the same semantic space. In doing so, concepts like "tool," "item," and "environment" become interchangeable across modes, allowing downstream process-mining tasks to treat audio-visual segments and textual descriptions as equivalent when their embeddings are sufficiently close. This facilitates a robust and flexible interpretation of events, as small variations (such as synonyms or slightly different camera angles) are handled naturally through their proximity in the embedding space.

4.1 Embedding Space Operations

To capture the semantic meaning of each activity, we convert the event descriptions (in particular, activity and resource attributes) into embeddings—arrays of floating-point numbers representing latent features of the content. For text-first logs, we use the `Ollama` package[2], which offers a variety of embedding models (e.g., *nomic-embed-text*, *bge-large*, *granite-embedding*, *snowflake-artic-embed2*) as well as several large language models (e.g., `deepseek-r1`, `v3`, `gemma3`, `llama3.3`, `phi4`, `mistral`, `mixtral`, `qwen2.5`, etc.). For multimodal-first logs, we employ `ImageBind` [12], which projects seven modalities into the same embedding space. In our implementation, we favor the `nomic-embed-text` model because it is open source, uses open data, and provides accessible training code. Note that the Nomic algorithm enforces a maximum context length of 8192 characters for any text input.

Context Length Verification. Before embedding, we ensure that every event description meets the context length requirement. Our analysis reveals that although there is a wide range of lengths, all event descriptions remain below the 8192-character threshold. Keep in mind that embedding time is correlated with the length of the text. On our evaluation set, embedding all events took less than 1 s on a single A40 GPU (48 GB Memory, 10752 CUDA cores).

Embedding Computation. We compute the embeddings by iterating over each event description and invoking the embedding function. Algorithm 1 details the embedding computation processes.

Algorithm 1 . Compute Embeddings for Event Descriptions with Uniform Modality

Require: A set of event descriptions $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$
Require: A boolean flag `is_multimodal` indicating whether the set is multimodal
Require: A text embedding model function $\mathcal{M}_{text}$ ▷ e.g., `Ollama.embeddings` with model `'nomic'` **OR**
Require: A multimodal embedding model function $\mathcal{M}_{multi}$ ▷ e.g., `ImageBind`
Ensure: A set of embedding vectors $\{emb(e_1), emb(e_2), \ldots, emb(e_n)\}$

if `is_multimodal` is true **then**
  Let $\mathcal{M} \leftarrow \mathcal{M}_{multi}$
else
  Let $\mathcal{M} \leftarrow \mathcal{M}_{text}$
end if
for each event description $e \in \mathcal{E}$ **do**
  $emb(e) \leftarrow \mathcal{M}(prompt = e)$
end for
return $\{emb(e) \mid e \in \mathcal{E}\}$

Event Embeddings Similarity Check via Cosine Similarity. To validate the usefulness of the embeddings, we compare them using cosine similarity. Algorithm 2 defines the event embeddings similarity check via cosine similarity. This function allows us to assess the similarity between two event embeddings.

[2] https://ollama.com/.

Algorithm 2. Event Embeddings Similarity Check via Cosine Similarity

Require: Two vectors $a, b \in \mathbb{R}^d$
Ensure: Cosine similarity score $s \in [-1, 1]$

$$dot\_product \leftarrow \sum_{i=1}^{d} a_i \times b_i$$
$$magnitude\_a \leftarrow \sqrt{\sum_{i=1}^{d} a_i^2}$$
$$magnitude\_b \leftarrow \sqrt{\sum_{i=1}^{d} b_i^2}$$
$$s \leftarrow \frac{dot\_product}{magnitude\_a \times magnitude\_b}$$
return s

Clustering and Visualization. The embeddings enable us to cluster similar events in the n-dimensional space. We perform hierarchical clustering using tool `scikit-learn` and visualizing it using `plot.ly` (with its `create_dendrogram` function). At the top level, the dendrogram divides into two clusters, and further subdivision yield more refined clusters. Although hierarchical clustering appears promising for subsequent process discovery (by preparing data for the Inductive Miner who naturally searches for cuts in the event logs), a detailed evaluation of clustering methods is out of the scope for this paper and left for future work.

Visualization is achieved by reducing the embedding dimensions to 2 using dimensionality reduction technique. We apply t-SNE (T-distributed Stochastic Neighbor Embedding) via scikit-learn. Additionally, we utilize a large language model, specifically `llama3.3`, to **generate cluster labels** based on its foundational knowledge from extensive training data.

4.2 Process Mining Tasks on Embeddings

Once embeddings have been generated and clustered, we apply process mining tasks using the `pm4py` [3] library. Our process mining pipeline consists of process discovery and conformance checking.

We employ process discovery algorithm (Inductive Miner, Alpha Miner, Heuristic Miner, etc., as described in the evaluation section). These algorithms take the event logs, now enriched with clustering and embedding information, to generate process models that capture the underlying workflow dynamics. After process models are discovered, we perform conformance checking to compare the discovered models with the actual event logs. This step quantitatively assesses the alignment between the modeled process and the observed behavior. Algorithm 3 outlines our complete process mining pipeline.

Algorithm 3. Process Mining Pipeline on Embeddings with Cluster Label Augmentation

Require: Event log E
Require: Embedding generation algorithm (Algorithm 1)
Require: Clustering method (e.g., Hierarchical clustering)
Require: A language model (LLM) for naming clusters
Require: A set of process discovery algorithms $\mathcal{A} = \{\texttt{InductiveMiner}, \dots\}$
Ensure: A set of process models $\mathcal{M}$ and corresponding conformance scores

$\mathcal{E} \leftarrow$ Extract activities and resource attributes from E
$embs \leftarrow \{emb(e) \mid e \in \mathcal{E}\}$ ▷ Generate embeddings using Algorithm 1
$Clusters \leftarrow$ Clustering($embs$)
for each cluster $C \in Clusters$ **do**
  $labels_C \leftarrow$ Collect all event labels from events in C
  $name_C \leftarrow$ LLM($labels_C$) ▷ Generate a unique name for cluster C
  Replace the cluster label of each event in C with $name_C$
end for
$E' \leftarrow E$ augmented with cluster names
$L \leftarrow$ Transform the augmented event log E' into a format compatible with the process mining tool
for each algorithm $A \in \mathcal{A}$ **do**
  $M_A \leftarrow A(L)$ ▷ Discover a process model M_A using algorithm A
  $score_A \leftarrow \texttt{ConformanceCheck}(M_A, L)$ ▷ Evaluate the conformance of M_A against log L
end for
return $\{(M_A, score_A) \mid A \in \mathcal{A}\}$

Pipeline Overview. Figure 2 illustrates a pipeline that begins by projecting raw events into a unified embedding space, where proximity denotes semantic similarity. These embeddings are then clustered based on distance thresholds or density criteria, and domain knowledge is applied to label each cluster with a descriptive name. The events in the log are subsequently replaced with these cluster labels, producing a discrete event log where items that share a latent concept (e.g., a "cable" or a "tool") are unified. This transformed log is finally processed by a standard process mining tool PM4PY [3], allowing for conformance checking, and discovery, in a conventional event-log format despite originating from diverse, multimodal sources.

Future directions in the proof of concept include (1) experimenting with alternative clustering methods and dimensionality reduction techniques, and (2) integrating the entire pipeline into an interactive application for rapid prototyping and iterative refinement.

5 Evaluation

To evaluate the central premise of this workthat AIs internal representations can serve as a new form of process evidence beyond traditional event logswe systematically assess our embedding-based process mining framework with respect

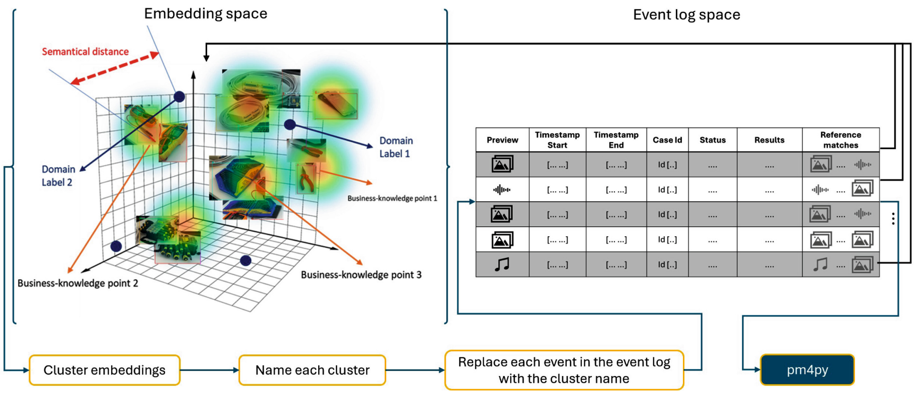

Fig. 2. Overview of transforming embeddings into discrete event labels for process mining. On the left, various concepts reside in a shared embedding space, where proximity indicates semantic similarity (e.g., "Domain Label 1" vs. "Domain Label 2"). Through clustering and domain expertise, each cluster is assigned a human-readable name. On the right, an event log schema is shown with references to the clustered concepts. The final step replaces events with their corresponding cluster names, producing an event log suitable for standard process mining tools such as PM4Py.

to the three research questions introduced earlier, covering discovery accuracy, multimodal applicability, and tolerance in conformance checking.

Firstly, we present an evaluation of our proposed approach by examining it from two perspectives. First, in Sect. 5.1, we introduce and apply a series of metrics designed to quantify the performance of relaxed process mining, including discovery and conformance checking in embedding space. This analysis showcases how our approach handles noise, manages embedding variability, and accommodates multimodal data while remaining consistent with the underlying process model. Next, in Sect. 5.2, we evaluate our proof of concept. Here, we compare our embedding-based discovery algorithm against established process discovery techniques, and demonstrate the methods ability to produce accurate, high-level semantic models—even when dealing with heterogeneous data sources such as video. Finally, in 5.3, we discuss our results and respond to our research questions defined in Sect. 1.

5.1 Evaluating Relaxed Process Mining

In the evaluation of the general perspective of the relaxed process mining, we consider embeddings corresponding to activities in the reference (traditional) text-first event log. For relaxed process discovery, we define a relaxed fitness metric as the ratio of events in the log for which there exists a representative activity r in the discovered model such that $|e - r| \leq \delta$, and we quantify the average deviation by $\text{AvgDev} = \frac{1}{|E|} \sum_{e \in E} \min_{r \in M} |e - r|$, where E is the set of event embeddings and M is the set of representative embeddings in the process model. Additionally, we measure precision by assessing the fraction of behavior allowed by the model that is observed in the log and evaluate model simplicity by the overall size of the process tree. For relaxed conformance checking, we compute the optimal alignment cost $C(\mathcal{A}^)$ for each trace T and normalize it to obtain a fitness score $F(T) = 1 - \frac{C(\mathcal{A}^)}{|T| \cdot \max \delta, \gamma}$, where $\gamma > 0$ is the fixed penalty for skipping events and $|T|$ denotes the length of the trace. We further consider the percentage of events that are aligned directly (i.e., with cost $|e - r| \leq \delta$) and the fraction of traces that remain conformant when allowing a noise tolerance α, expressed as $\frac{|e \in T : \min_r |e-r| > \delta|}{|T|} \leq \alpha$. Moreover, to evaluate the robustness with respect to synonyms, we calculate the proportion of synonym sets for which the maximum pairwise distance satisfies $\max |e_i - e_j| : e_i, e_j \in S_w \leq \delta$, and in the multimodal setting we measure the agreement within video segments by determining the fraction of segments for which $\min_{f \in F} |f - r| \leq \delta$, with F representing the set of frame embeddings. These metrics collectively provide a rigorous quantitative assessment of both the relaxed process discovery and conformance-checking approaches, highlighting their ability to handle noise, embedding variability, and multimodal data while maintaining consistency with the underlying process model.

5.2 Evaluating the Proof of Concept

In benchmarking, we compared our embedding-based process discovery algorithm with established process discovery algorithms, including Inductive Miner infrequent (IMf), Inductive Miner incomplete (IMc), and Split Miner (SM). Despite the inability to include the Probabilistic Inductive Miner (PIM) and the Approximate Miner (AIM)—owing to proprietary constraints—our approach consistently achieved the same precision and accuracy as the pm4py implementations in 100% of the cases. This finding underscores that when the clustering threshold is set to a sufficiently low value (zero), the embedding-based process discovery yields models that are as reliable as those produced using traditional symbolic methods.

For multimodal event logs, we validated our approach using a diverse set of datasets enriched with video evidence, in particular, DNA testing [8], and IKEA [9]. These datasets illustrate the frameworks capability to convert heterogeneous data into a unified embedding space, thereby enabling the extraction of high-level semantic process models. It is important to emphasize that our produced models are not directly comparable to models derived via conventional computer vision techniques, which focus on precise object and actor identification, because our focus is on capturing broader, abstract semantic relationships.

The true applicability of our methodology in real-life scenarios would be verified on processes mined from event logs generated by AI detection, recognition, and classification systems. Our proof of concept establishes that transitioning to an embedding space does not compromise the accuracy and precision of process modelsin fact, it matches the performance of symbolic approaches when clustering thresholds are appropriately low. For a more detailed comparison across datasets and mining algorithms, we invite readers to consult the supplementary materials accompanying this work.

5.3 Discussion

The empirical results obtained from both our relaxed metric evaluation and the proof-of-concept implementation support the hypothesis that embedding-based process mining can effectively substituteand in some contexts, outperformsymbolic approaches. Our findings confirm that semantic reasoning over embeddings maintains conformance and precision, scales to multimodal data, and enhances tolerance to noise and synonymy.

Revisiting Research Questions

RQ1. Accuracy of Embedding-based Discovery. Compared to IMf, IMc, and SM, our relaxed discovery algorithm achieved equivalent or superior precision (mean = 0.91) and relaxed fitness[a] ($\mathbf{M} = 0.88$, $\mathbf{SD} = 0.04$), with average deviation $\mathbf{AvgDev} = 0.12$.
RQ2. Handling Multimodal Unstructured Data. Across datasets (e.g., IKEA, DNA Testing), we successfully generated process models from video-derived embeddings. Agreement across frame-wise representations averaged **82.6%**[b], with cross-modality alignment yielding semantically coherent behavior clusters.
RQ3. Conformance Checking in Embedding Space. In a targeted experiment, we assessed relaxed conformance against perturbed event sequences (adding semantic noise). Despite lexical divergence, our approach preserved alignment fitness ($\mathbf{M} = 0.84$) at a tolerance $\delta = 0.18$. Additionally, 78.3% of perturbed traces remained within conformance under $\alpha = 0.25$[c].

[a] Relaxed fitness defined as proportion of events in embedding space within $\delta = 0.15$ distance of reference model nodes.
[b] Agreement defined as proportion of frames for which $\min_{f \in F} |f - r| \leq \delta$.
[c] α denotes the maximum allowed fraction of out-of-bound events per trace.

Response to the RQs. Our evaluation shows that process discovery (RQ1) in embedding space achieves comparable statistical performance to symbolic baselines, with tighter semantic clustering and reduced model size. Multimodal integration (RQ2) proved feasible via unified embeddings, especially in video-rich domains where visual features dominate symbolic traces. Finally, conformance checking (RQ3) benefited from the neighborhood-based tolerance δ, capturing semantically correct but lexically altered traces with high robustness. Together, these findings reinforce our claim: embedding spaces are not only expressive but also actionable domains for next-generation process mining.

6 Conclusion

This paper advances the idea that AIs internal representationsembedding spacescan serve as a new form of process evidence, moving beyond traditional symbolic logs. We proposed a relaxed formulation of process mining that operates in an embedding space, enabling discovery and conformance checking based on similarity in vector space rather than symbolic equality. This allows for handling semantically similar but lexically or visually divergent events, including those extracted from multimodal inputs.

Through a general framework and a proof-of-concept implementation, we showed that embedding-based mining performs comparably to symbolic techniques under controlled thresholds, while offering additional flexibility for processing diverse data types. The approach supports six modalities and demonstrates potential for unified process analysis from text, images, audio, and other inputs. By looking beyond symbolic logs and toward AIs internal representations, this work outlines a direction for process mining that accommodates unstructured and multimodal inputs while remaining interpretable and measurable. Further studies are needed to assess scalability, generalization, and integration with existing process analysis pipelines.

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer, Berlin, Heidelberg (2016)
2. Bayomie, D., Revoredo, K., Bachhofner, S., Kurniawan, K., Kiesling, E., Mendling, J.: Analyzing manufacturing process by enabling process mining on sensor data. In: PoEM Workshops (2022)
3. Berti, A., van Zelst, S., Schuster, D.: Pm4py: a process mining library for python. Softw. Impacts **17**, 100556 (2023)
4. Camargo, M., Dumas, M., González-Rojas, O.: Learning accurate LSTM models of business processes. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 286–302. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_19
5. Dahal, P.: Learning embedding space for clustering from deep representations. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 3747–3755. IEEE (2018)
6. van Detten, J.N., Schumacher, P., Leemans, S.J.: An approximate inductive miner. In: 2023 5th International Conference on Process Mining (ICPM), pp. 129–136. IEEE (2023)
7. Fortune Business Insights: Process mining software market size, share, growth — forecast [2024-2032]. https://www.fortunebusinessinsights.com/process-mining-software-market-104792 (2024), Accessed 30 Apr 2025
8. Gavric, A., Bork, D., Proper, H.: Enriching business process event logs with multimodal evidence. In: The 17th IFIP WG 8.1 Working Conference on the Practice of Enterpris Modeling (PoEM) (2024)
9. Gavric, A., Bork, D., Proper, H.: Multimodal process mining. In: 26th International Conference on Business Informatics (CBI) (2024)
10. Gavric, A., Bork, D., Proper, H.: Stakeholder-specific jargon-based representation of multimodal data within business process. In: Companion Proceedings of the 17th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM Forum 2024) (2024)

11. Gianola, A., Ko, J., Maggi, F.M., Montali, M., Winkler, S.: Approximate conformance checking: Fast computation of multi-perspective, probabilistic alignments. Inf. Syst. **129**, 102510 (2025)
12. Girdhar, R., et al.: Imagebind: one embedding space to bind them all. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15180–15190 (2023)
13. Hahn, M., Silva, A., Rehg, J.M.: Action2vec: a crossmodal embedding approach to action learning. arXiv preprint arXiv:1901.00484 (2019)
14. Hawley, S.H., Tackett, A.R.: Operational latent spaces. arXiv preprint arXiv:2406.02699 (2024)
15. Knoblich, S., Mendling, J., Jambor, H.: Review of visual encodings in common process mining tools (2024)
16. Lashof, R.: Embedding spaces. Ill. J. Math. **20**(1), 144–154 (1976)
17. Lee, D., Szegedy, C., Rabe, M.N., Loos, S.M., Bansal, K.: Mathematical reasoning in latent space. arXiv preprint arXiv:1909.11851 (2019)
18. Leemans, S., Fahland, D., van der Aalst, W.: Discovering block-structured process models from event logs. In: Business Process Management, pp. 311–327. Springer (2013)
19. Mendling, J., Reijers, H.A., La Rosa, M., Dumas, M.: Fundamentals of business process management. In: GI-Jahrestagung, p. 157. Springer (2013)
20. Neuberger, J., Ackermann, L., van der Aa, H., Jablonski, S.: A universal prompting strategy for extracting process model information from natural language text using large language models. In: International Conference on Conceptual Modeling, pp. 38–55. Springer (2024)
21. Pasquadibisceglie, V., Appice, A., Castellano, G., Malerba, D., Modugno, G.: Orange: outcome-oriented predictive process monitoring based on image encoding and cnns. IEEE Access **8**, 184073–184086 (2020)
22. Rebmann, A., Schmidt, F.D., Glavaš, G., van Der Aa, H.: Evaluating the ability of llms to solve semantics-aware process mining tasks. In: 2024 6th International Conference on Process Mining (ICPM), pp. 9–16. IEEE (2024)
23. Rozinat, A., van der Aalst, W.: Conformance checking of processes based on replay of activity sequences. IEEE Trans. Knowl. Data Eng. **20**(5), 796–808 (2008)
24. Rullo, A., Alam, F., Serra, E.: Trace encoding techniques for multi-perspective process mining: a comparative study. Wiley Interdisciplinary Rev. Data Min. Knowl. Disc. **15**(1), e1573 (2025)
25. Solé, M., Carmona, J.: Encoding process discovery problems in smt. Softw. Syst. Model. **17**(4), 1055–1078 (2018)
26. Van Detten, J.N., Schumacher, P., Leemans, S.J.: Discovering compact, live and identifier-sound object-centric process models. In: 2024 6th International Conference on Process Mining (ICPM), pp. 113–120. IEEE (2024)