Skip Probabilities for Subprocesses

Philipp Bär
RWTH Aachen University
Aachen, Germany
philipp.baer@rwth-aachen.de

Adam T. Burke, Moe T. Wynn
Queensland University of Technology
Brisbane, Australia
{a.burke, m.wynn}@qut.edu.au

Sander J. J. Leemans
RWTH Aachen University & Fraunhofer FIT
Aachen, Germany
s.leemans@bpm.rwth-aachen.de

***Abstract*—Conformance checking techniques compare process models of organizational behavior with observed process executions to reveal their deviations. Traditional alignments concern individual activities and provide a single out of potentially infinitely many explanations for observed deviations. Skip alignments lift insights to subprocesses and provide all possible explanations. Though valuable for analysts and process mining tools, there exist no interpretations *how likely* these deviations are.**

In this paper, we introduce skip probabilities revealing how likely certain subprocesses deviate w.r.t. an event log of observed process executions. We show the formal derivation of this calculation and demonstrate the feasibility of its computation. By analyzing a realistic case, we empirically show that yet hidden process insights can be derived from skip probabilities and how they contribute to targeted process improvement.

***Index Terms*—stochastic process mining, conformance checking, skip alignments, process trees**

I. Introduction

When performing a process, what we do is important, but what we do not do is often more important. *Conformance checking* techniques compare an event log of traces from real-world process executions against a model of desired process executions [1], [2]. This comparison reveals to what extent the recorded and the modeled process correspond, where they deviate, and what these deviations look like. Many process mining techniques such as model repair [3], [4], process comparison [5], and genetic process discovery [6] derive insights from *whether* certain deviations occurred in the log. For instance, in process comparison, cohorts are perceived as dissimilar if they reveal different deviations.

Knowing whether deviations exist is important. Consider a process for issuing road traffic fines. It is relevant to know whether particular 'appeal' activities are performed or not, as revealed by existing techniques. It is even more useful to know how likely it is that some 'appeal' activities or, even more general, the entire 'appeal' subprocess is skipped.

Accordingly, this paper asks *What is the probability of not executing a subprocess?* We term this a *skip probability*. It allows us to detect, stochastically quantify, and interpret process conformance across every subprocess. This can provide analysts with fine-grained process insights and guide targeted process improvement based on which parts of a process are significantly deviating, and so require the most attention. For this calculation we need a source of observations (an event log), a process model, stochastic distribution information for the model, and the possible deviations between log and model.

Part of this research was funded by the Hans Hermann Voss-Stiftung.

For a process model representation, we use *process trees* [1], an established hierarchical representation where subprocesses can be easily associated with subtrees. Existing stochastic process mining tools [7]–[9] are used to obtain information on the distribution in the form of a stochastic language.

For information on deviations, we use *skip alignments* [10]. These are generalizations of *alignments* [11], the state-of-the-art conformance checking technique that synchronizes log traces with a model to reveal their deviations in terms of unexpectedly observed activities (log moves) and missing expected activities (model moves). Skip alignments finitely summarize all alignments, a potentially infinite set in the case of silent loops, by generalizing model moves to skip moves. Hence, they represent conformance on the level of subprocesses, which basic alignments lack [12], simply showing the existence of deviations at the level of individual process activities. In a road fine process, an alignment may show if an appeal was sent, rejected, or upheld, but a skip alignment can show that the entire appeal process was missed.

Neither alignments nor skip alignments provide stochastic conformance, so augmenting skip alignments with skip probabilities allows process mining techniques to derive insights from *whether* and *how likely* deviations occur and how their analysis should be prioritized. Process comparison on the fines process may reveal distinct cohorts that share the same types of deviations but with different likelihoods. Existing process mining tools may not be able to discriminate these cohorts, but distinct skip probabilities reveal hidden differences.

We present the formal derivation, a prototype implementation, and an empirical evaluation on real-world logs, and an analysis on a realistic example model showcasing novel process insights.

In the remainder of this paper, we discuss related work in Section II and introduce basics in Section III. In Section IV, we formally introduce skip probabilities and evaluate them in Section V before concluding in Section VI.

II. Related Work

Many conformance checking techniques have been proposed in literature [2]. For instance, token-based replay provides information on missing and remaining tokens in a Petri net [13] and alignments being the current state-of-the-art in conformance checking, as they result in a detailed identification of deviations in both the log and the model [14]. Several approaches

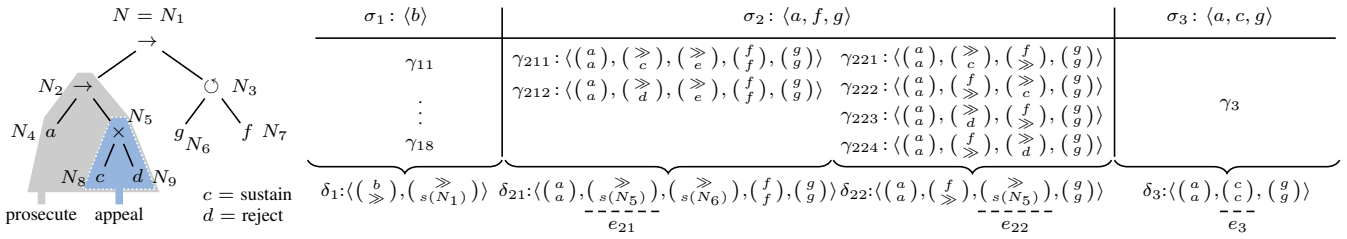


Fig. 1: Running example: Process tree N, alignments, skip alignments, and executions for the traces $\sigma_1, \sigma_2, \sigma_3$.

have been proposed to compute alignments efficiently or approximately [2]. However, for a given trace, infinitely optimal alignments may exist due to (i) silent loops, (ii) multiple equivalent-cost model moves (did we skip transition a or b in a choice between them?), (iii) concurrency in the model, and (iv) concurrency between deviations (a log move and a model move are concurrent by definition). Typical alignment computations arbitrarily choose one of infinitely many optimal alignments, which creates a false sense of certainty if deviations are analyzed, even in aggregated form. (i) is typically avoided by considering silent steps to have a non-zero small cost, however this removes the loop from consideration and thus provides only an approximation [2]. The concurrency challenges (iii) and (iv) could be addressed by computing partially ordered alignments [15], though this would require partially ordered traces. A recent approach addresses (iii) and (iv) by introducing higher-level patterns of deviations, such as swapping or repeating activities [16].

Our approach takes as one of its inputs a separate stochastic path language, provided by a stochastic process model. Stochastic conformance checking techniques such as the Earth Movers' Stochastic Conformance [17] address (iii) by considering all model paths, and (i) by sampling. (ii) is not addressed. The results of this approach can be projected onto process models to indicate the likelihood of an activity to be a synchronous move, which avoids (iv), but have not been leveraged to the conformance of higher-level subprocesses as in this paper.

Our approach in [10] solves (i) and (ii) by lifting model moves to process tree nodes (*skip moves*) by compactly representing *all* optimal alignments, and addresses (iii) and (iv) by reordering rules towards a normal form. While solving (i)-(iv), the approach in [10] does not stochastically quantify conformance and hence lacks actionable process improvement. As such, we build upon [10] to compute skip probabilities by using a second input of a stochastic path language, which allows us to quantify the likelihood of each skip alignment.

[18], [19] use stochastic information in the alignment computation. The first one incorporates stochastic model information into an alignment quality score and based on it, top-k alignments are computed. The second uses stochastic MDP planning to rank and compute optimal alignments. Both approaches do not stochastically quantify model conformance, which is a key contribution of our approach.

In [20], a stochastic Markovian abstraction (fixed length trace infixes) is proposed which compares two stochastic languages based on the expected frequencies of their subtraces. This results in a global measure of similarity and hence can be used to assess conformance for a stochastic log language and a stochastic path language. Unlike our approach, this conformance result remains at the level of the entire model, not subprocesses. Furthermore, by splitting model paths into fixed length subtraces, semantic relations at subprocess level are lost, other than with our skip probabilities.

III. Preliminaries

In this section, we introduce the basics and fundamental concepts of process mining. We denote sequences with $s = \langle a_1, \ldots, a_n \rangle$ and write $x \in s$ for all elements x in s. The set of all finite sequences over a set X is denoted with X^*. A sequence s is a subsequence of a sequence $s' = \langle b_1, \ldots, b_m \rangle$, if s consecutively appears in s', that is, s is an infix of s'. $s_{\downarrow X}$ is s with all elements not in the set X removed, $s_{\downarrow \neq x}$ is s with all elements x removed. The concatenation is defined by $s \cdot s' = \langle a_1, \ldots, a_n, b_1, \ldots, b_m \rangle$ and the interleaving by $s \diamond s' = \{s'' \mid s'' \text{ is a permutation of } s \cdot s' \wedge s''_{\downarrow s} = s \wedge s''_{\downarrow s'} = s'\}$. For example, $\langle a, b \rangle \diamond \langle c \rangle = \{\langle a, b, c \rangle, \langle a, c, b \rangle, \langle c, a, b \rangle\}$. For sets of sequences S, S', it holds that $S \cdot S' = \{s \cdot s' \mid s \in S \wedge s' \in S'\}$ and $S \diamond S' = \bigcup_{s \in S, s' \in S'} s \diamond s'$. A sequence s is a loose subsequence of s', if there exists a sequence s'' and $s \diamond s'' = s'$, i.e., s appears non-consecutively in s'. For a sequence of pairs $s = \langle \binom{e_1}{a_1}, \ldots, \binom{e_n}{a_n} \rangle$, $\pi_1(s)$ resp. $\pi_2(s)$ refer to the sequences of first components $\langle e_1, \ldots, e_n \rangle$ resp. second components $\langle a_1, \ldots, a_n \rangle$ of the pairs in s.

In process mining, the set A is the set of *activities* like *send, reject, etc.* abbreviated to a, b, etc., a *trace* $\sigma \in A^*$ is a sequence of activities. An *event log* $L = [\sigma_1^{m_1}, \ldots, \sigma_k^{m_k}]$ is a multiset of traces on the set of *trace variants* $L_\perp = \{\sigma_1, \ldots, \sigma_k\}$ with multiplicities $m_1, \ldots, m_k$. We use $\sigma \in L$ as a shorthand notation for $\sigma \in L_\perp$. A *stochastic log language* is the probability distribution $\mathcal{P}_L = [\sigma_1^{\frac{m_1}{\sum_i m_i}}, \ldots, \sigma_n^{\frac{m_n}{\sum_i m_i}}]$ on L.

An example event log is $L = [\sigma_1^1 \colon \langle b \rangle, \sigma_2^2 \colon \langle a, f, g \rangle, \sigma_3^7 \colon \langle a, c, g \rangle]$ over the set $L_\perp = \{\sigma_1, \sigma_2, \sigma_3\}$ with $\mathcal{P}_L(\sigma_1) = 0.1$, $\mathcal{P}_L(\sigma_2) = 0.2$, and $\mathcal{P}_L(\sigma_3) = 0.7$; letters abbreviate activities.

A *process tree* is a hierarchical representation of a process. Its leaf nodes are either activities from A or the specific silent activity $\tau \notin A$ representing 'no activity'. Inner nodes are operators and combine the languages of subtrees. The language of a leaf node is the (silent) activity itself. The language of an inner node is obtained by traversing its subtree depending on the operator type: the children's languages are concatenated (sequence), joint (choice), or interleaved (parallel). A loop repeatedly traverses its two children being a *loop iteration*: All but the last iteration traverse the first and then the second child. The last loop iteration traverses only the first child.

Definition 1 (Process Tree). *Over a set of activities A with $\tau \notin A$, a* process tree N *and its* path language $\mathcal{L}$ *are:*

- *labeled activity:* $N = a$, $a \in A$ *with* $\mathcal{L}(N) = \{\langle a \rangle\}$
- *silent activity:* $N = \tau$, $\tau \notin A$ *with* $\mathcal{L}(N) = \{\langle \tau \rangle\}$
- $N = \oplus(N_1, \ldots, N_k)$, *with* $k \geq 2$ *and* $\oplus \in \{\rightarrow, \times, \wedge, \circlearrowleft\}$:
  - *sequence:* $\mathcal{L}(\rightarrow(N_1, \ldots, N_k)) = \mathcal{L}(N_1) \cdot \ldots \cdot \mathcal{L}(N_k)$
  - *choice:* $\mathcal{L}(\times(N_1, \ldots, N_k)) = \mathcal{L}(N_1) \cup \ldots \cup \mathcal{L}(N_k)$
  - *parallel:* $\mathcal{L}(\wedge(N_1, \ldots, N_k)) = \mathcal{L}(N_1) \diamond \ldots \diamond \mathcal{L}(N_k)$
  - *loop:* $\mathcal{L}(\circlearrowleft(N_1, N_2)) = (\mathcal{L}(N_1) \cdot \mathcal{L}(N_2))^* \cdot \mathcal{L}(N_1)$

The set of (silent) activities is $\text{leafs}(N)$. *Every traversal* $\rho \in \mathcal{L}(N)$ *is a* model path. *We write* $N' \in N$ *for the nodes* N' *in* N. *Every node is a* subtree *that starts in* N', *called the* subprocess *of* N'. *A* stochastic path language $\mathcal{P}_N$ *is a probability distribution on* $\mathcal{L}(N)$ *with* $\mathcal{P}_N(\rho) > 0$ *for all* ρ.

Figure 1 shows the running example of a process tree N composed from 9 nodes $N_1, \ldots, N_9$, representing the process of issuing road fines, made from different subprocesses such as *prosecute* (N_2) or *appeal* (N_5, blue). The *appeal* subprocess describes whether an appeal is *sustained* (c) or *rejected* (d). The language $\mathcal{L}(N)$ is infinite, because N contains a loop. Loop iterations of N_3 are $\langle g \rangle, \langle g, f, g \rangle, \ldots$. We consider the paths $\rho_1 \colon \langle a, c, g, f, g \rangle, \rho_2 \colon \langle a, d, g, f, g \rangle, \rho_3 \colon \langle a, c, g \rangle, \rho_4 \colon \langle a, d, g \rangle \in \mathcal{L}(N)$ with probabilities $\{\rho_1^{0.1}, \rho_2^{0.1}, \rho_3^{0.3}, \rho_4^{0.3}\} \subseteq \mathcal{P}_N$.

An alignment [11] synchronizes a trace from the log to a model path revealing where and which deviations between observed and modeled process behavior arise.

Definition 2 (Alignment). *Let* $\sigma \in A^*$ *be a trace, let* $\gg \notin A$ *be the* no move *symbol and let* N *be a process tree. An* alignment γ *for* σ *on* N *is a sequence* $\gamma = \langle \binom{e_1}{a_1}, \ldots, \binom{e_n}{a_n} \rangle \in ((A \cup \{\gg\}) \times (\text{leafs}(N) \cup \{\gg\}))^*$ *such that each element of* γ *is either a* log move $\binom{e_i}{\gg}$ *with* $e_i \in A$, *a* model move $\binom{\gg}{a_i}$ *with* $a_i \in \text{leafs}(N)$, *or a* synchronous move $\binom{e_i}{a_i}$ *with* $a_i \in \text{leafs}(N), e_i \in A, a_i = e_i$. *The sequence of first components matches the trace* $\sigma = \pi_1(\gamma)_{\downarrow_A}$, *the sequence of second components is an element of the path language* $\pi_2(\gamma)_{\downarrow_{\text{leafs}(N)}} \in \mathcal{L}(N)$.

The standard alignment cost function assigns costs of 0 to synchronous moves $\binom{e}{a}$ and to model moves $\binom{\gg}{\tau}$ of silent transitions, while all other moves incur a cost of 1. The sum of movement costs is the alignment cost. An alignment is *optimal* if there is no alignment for σ on N with lower cost. The set of optimal alignments of σ on a fixed model N is $\Gamma(\sigma)$.

In our example, the trace σ_2 has 6 optimal alignments $\gamma_{211}, \gamma_{212}, \gamma_{221}, \ldots, \gamma_{224}$ on the model N, shown in Figure 1.

Next, we define the skip language of N [10]. While $\mathcal{L}(N)$ is the set of all traversals of N, i.e., sequences of activities, the skip language $\mathcal{S}(N)$ *additionally* covers traversals that skip subtrees of N: Instead of deepening into every subtree of N, some subtrees N' might be *skipped* (indicated by a skip $s(N')$), i.e., no activities of N' appear in this *skip traversal* of N'.

Definition 3 (Skip Language). *Let* N *be a process tree, let* $s(N')$ *denote a skip over a subtree* $N' \in N$, *and let* $\text{S}(N) = \{s(N') \mid N' \in N\}$. *Then, the* skip language $\mathcal{S}(N)$ *of* N *is:*

$$\mathcal{S}(a) = \{\langle a \rangle, \langle s(a) \rangle\} \qquad \mathcal{S}(\tau) = \{\langle s(\tau) \rangle\}$$

$$\mathcal{S}(\rightarrow(N_1,\ldots,N_k)) = \{\langle s(\rightarrow(N_1,\ldots,N_k)) \rangle\} \cup \mathcal{S}(N_1) \cdot \ldots \cdot \mathcal{S}(N_k) \setminus \{\langle s(N_1),\ldots,s(N_k) \rangle\}$$

$$\mathcal{S}(\times(N_1,\ldots,N_k)) = \{\langle s(\times(N_1,\ldots,N_k)) \rangle\} \cup \mathcal{S}(N_1) \cup \ldots \cup \mathcal{S}(N_k) \setminus \{\langle s(N_1) \rangle,\ldots,\langle s(N_k) \rangle\}$$

$$\mathcal{S}(\wedge(N_1,\ldots,N_k)) = \{\langle s(\wedge(N_1,\ldots,N_k)) \rangle\} \cup \mathcal{S}(N_1) \diamond \ldots \diamond \mathcal{S}(N_k) \setminus \{\langle s(N_{p_1}),\ldots,s(N_{p_k}) \rangle \mid p \in \langle 1 \rangle \diamond \ldots \diamond \langle k \rangle\}$$

$$\mathcal{S}(\circlearrowleft(N_1,N_2)) = \{\langle s(\circlearrowleft(N_1,N_2)) \rangle\} \cup \{\sigma \mid \sigma \in (\mathcal{S}(N_1) \cdot \mathcal{S}(N_2))^* \cdot \mathcal{S}(N_1) \wedge \langle \ldots, s(N_1), s(N_2), \ldots \rangle \neq \sigma, \langle \ldots, s(N_2), s(N_1), \ldots \rangle \neq \sigma\} \setminus \{\langle s(N_1) \rangle\}$$

For example, the sequence $\langle a, s(N_5), g \rangle$ is in the skip language of N in Figure 1: The subprocess N_5 is skipped, i.e., instead of playing it out to a sequence of activities (either c or d), the execution of the choice is skipped. According to Definition 3, this skip is most general, that is, N_5 is skipped rather than N_8 and N_9. Additionally, the traversal of N is free from superfluous loop iterations (iterations made from skips) prior or after the g like dotted in $\langle a, s(N_5), \underbrace{s(N_6), s(N_7)}, g \rangle$.

Similar to an alignment, a *skip alignment* (equivalently defined for process trees and block-structured Petri nets [10]) synchronizes an observed trace to a model, but using its skip language $\mathcal{S}$ instead of its language $\mathcal{L}$. An alignment γ explains the absence of N' in σ by a sequence of model moves for one possible traversal $\rho \in \mathcal{L}(N')$. A skip alignment δ keeps deviations on the level of subprocesses and instead performs a *skip move* $\binom{\gg}{s(N')}$.

Definition 4 (Skip Alignment). *Let* N *be a process tree, let* $\gg, s(\circ) \notin A$ *for any* $\circ$, *and let* $\sigma \in A^*$. *A* skip alignment *for* σ *on* N *is a* $\delta = \langle \binom{e_1}{a_1}, \ldots, \binom{e_n}{a_n} \rangle$ *with* $e_i \in A \cup \{\gg\}$ *and* $a_i \in \text{leafs}(N) \cup \{\gg\} \cup \text{S}(N)$ *such that each element* $\binom{e_i}{a_i}$ *of* δ *is either a* log move $\binom{e_i}{\gg}$ *with* $e_i \in A$, *a* skip move $\binom{\gg}{s(N')}$ *for some* $N' \in N$, *or a* synchronous move $\binom{e_i}{a_i}$ *with* $a_i \in \text{leafs}(N), e_i \in A, a_i = e_i$. *The sequence of first components matches the trace* $\sigma = \pi_1(\delta)_{\downarrow_A}$, *the sequence of second components is in the skip language* $\pi_2(\delta)_{\downarrow_{\neq \gg}} \in \mathcal{S}(N)$. *A skip alignment is in* normal form *if arbitrarily ordered moves follow the precedence 'log before synchronous before skip moves', i.e., if two moves could be swapped, then their order is determined.*

An alignment γ can be lifted to a skip alignment in normal form δ by replacing sequences of model moves with skip moves, removing loop iterations only consisting of skip moves, and reordering these moves. In that case, we say that γ and δ *coincide*. For example, $\langle \binom{b}{\gg}, \binom{\gg}{s(N_1)} \rangle$ is a skip alignment for the trace $\sigma_1 \colon \langle b \rangle$ on the model N in Figure 1: The execution of the entire tree is skipped and a log move on b is performed. The alignment $\gamma = \langle \binom{\gg}{a}, \binom{\gg}{c}, \binom{\gg}{g}, \binom{b}{\gg} \rangle$ coincides with that skip alignment: First, we lift the three model moves in γ to a skip move on N_1 resulting in $\langle \binom{\gg}{s(N_1)}, \binom{b}{\gg} \rangle$. Then, we enforce the precedence order of the normal form by swapping the skip and the log move resulting in $\delta = \langle \binom{b}{\gg}, \binom{\gg}{s(N_1)} \rangle$. Note that the precedence is required whenever moves can be ordered arbitrarily, e.g., in δ_{21}, the synchronous move $\binom{a}{a}$ precedes the skip move $\binom{\gg}{s(N_5)}$ as their order is not arbitrary: A

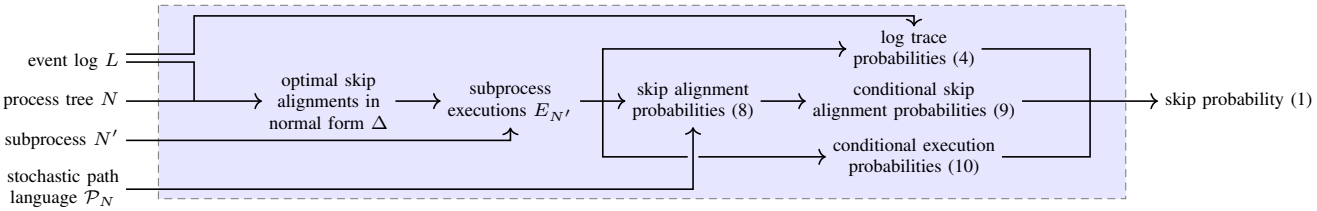


Fig. 2: Overview of the derivation: From the inputs (left), we calculate the skip probability of a subprocess (right).

swap would invalidate the model path (a must come before N_5).

As shown in [10], every γ lifts to a unique δ, but multiple alignments may lift to the same δ, making skip alignments summaries of alignments. The set of coinciding alignments of δ is $\bar{\mathcal{C}}(\delta)$. A skip alignment in normal form is *optimal* if one of its coinciding alignments is optimal. While the set of optimal alignments in potentially infinite, every trace σ has finitely many optimal skip alignments in normal form, which we refer to as $\Delta(\sigma)$.

In our example in Figure 1, it holds that $\Delta(\sigma_1) = \{\delta_1\}$, $\Delta(\sigma_2) = \{\delta_{21}, \delta_{22}\}$, and $\Delta(\sigma_3) = \{\delta_3\}$. The table indicates which alignments lift to which coinciding skip alignments.

IV. Calculating Skip Probabilities

In this section, we derive *skip probabilities* for subprocesses, which quantify how well each subprocess conforms to a log of observed process executions. The calculation is illustrated in Figure 2. We derive the skip probability of a subprocess N' in a process tree N based on an event log L and a stochastic path language of $\mathcal{P}_N$. Intuitively, the skip probability of N' is composed from three factors: The probabilities of the different log traces in L (4) (stochastic log language), the probabilities of the traces' optimal skip alignments in normal form (9), and the probabilities of the executions of N' in these skip alignments (10) (the moves that describe how N' is traversed). As all three probabilities can be formulated on executions, and executions are moves in skip alignments, the derivation of skip probabilities boils down to the computation of all optimal skip alignments in normal form and their stochastic information (8). Since skip alignments can be computed efficiently [10], in the following, we derive the remaining probabilistic information.

A. Executions of Skip Alignments

In this section, we establish executions, which describe how a node in a process tree is traversed within a skip alignment. Intuitively, an execution is a sequence of moves that traverses a node N' in a process tree, either by activities or by skips, i.e., these moves project to a trace in $\mathcal{S}(N')$. Hence, an execution is a stencil that highlights parts of a skip alignment that describe one traversal of a subprocess.

Definition 5 (Execution)**.** *Let N be a process tree, $N' \in N$ a node, δ an optimal skip alignment in normal form, and $\delta' = \langle \delta'_1, \ldots, \delta'_k \rangle$ a loose subsequence of moves in δ. Then, an* execution *is a pair $e = (N', \delta')$ such that:*

(1) δ' traverses N', i.e., $\pi_2(\delta') \in \mathcal{S}(N')$.

(2) δ' does not mix moves of multiple traversals of N', i.e., any move $\delta_i \in \delta$ between δ'_1 and δ'_k that is part of a traversal of N', is in δ'.

(3) If N' is a loop node, then δ' captures all its iterations, i.e., there is no additional loop iteration $\bar{\delta}'$ in δ before or after δ' such that $\bar{\delta}' \cdot \delta'$ or $\delta' \cdot \bar{\delta}'$ would be an execution of N'.

We refer to the skip alignment of e with $\delta(e)$ and to the node executed by e with $N(e)$.

The definition implies that executions do not contain log moves as they do not participate in a subprocess's is traversed.

As a running example, we inspect the appeal subprocess N_5 in Figure 1 (blue) for which we derive a skip probability, i.e., the probability that the appeal decision is missing. An execution of the subprocess N_5 in $\delta_{21} = \langle \binom{a}{a}, \binom{\gg}{s(N_5)}, \binom{\gg}{s(N_6)}, \binom{f}{f}, \binom{g}{g} \rangle$ is $(N_5, \langle \binom{\gg}{s(N_5)} \rangle)$: (1) $\langle s(N_5) \rangle \in \mathcal{S}(N_5)$, (2) N_5 is traversed just once in δ_{21}, and (3) N_5 is no loop hence there cannot be a missing loop iteration. Intuitively, the execution is a description of how N_5 is traversed in δ_{21}. In the skip alignment δ_3, N_5 is executed by firing one of its children (c), hence, the execution of N_5 in δ_3 is $(N_5, \langle \binom{c}{c} \rangle)$. Across the four optimal skip alignments of L, there are three executions of N_5: $\delta_{21}\colon (N_5, \langle \binom{\gg}{s(N_5)} \rangle)$ denoted e_{21}, $\delta_{22}\colon (N_5, \langle \binom{\gg}{s(N_5)} \rangle)$ denoted e_{22}, $\delta_3\colon (N_5, \langle \binom{c}{c} \rangle)$ denoted e_3. Note that there is no execution of N_5 in δ_1 as N_5 is neither traversed nor skipped. The executions of N_5 in the different skip alignments are indicated by dashed lines in Figure 1.

For a fixed process tree N, all optimal skip alignments in normal form are $\Delta = \bigcup_{\sigma \in A^*} \Delta(\sigma)$. We derive all executions:

$$E = \{e \mid e \text{ an execution} \wedge \delta(e) \in \Delta \wedge N(e) \in N\}$$

Nodes $N' \in N$, traces $\sigma \in A^*$, and skip alignments $\delta \in \Delta$ *project* to subsets of E:

$$E_{N'} = \{e \mid e \in E \wedge N(e) = N'\}$$
$$E_\sigma = \{e \mid e \in E \wedge \delta \in \Delta(\sigma) \wedge \delta(e) = \delta\}$$
$$E_\delta = \{e \mid e \in E \wedge \delta(e) = \delta\}$$

We additionally define the set of *skip executions* on E:

$$E_{\text{skip}} = \{e \mid e \in E \wedge e = (N', \langle \binom{\gg}{s(N')} \rangle)\}$$

Note that E and each subset are tightly bound to the process tree N as every execution corresponds to a node in N, but are independent of an event log. Information about the event log is used later when deriving skip probabilities.

For our example, Figure 3 shows the set of executions E_{N_5}. The set is infinite, but e_{21}, e_{22}, and e_3 are the only executions that result from our example log L. Dashed boxes indicate $E_{N_5} \cap E_\delta$ and $E_{N_5} \cap E_\sigma$ for the traces σ_2 and σ_3, and their optimal skip alignments in normal form δ_{21}, δ_{22}, and δ_3. Note that δ_1 and hence σ_1 reveal no executions of N_5, i.e., the box of δ_1 resp. σ_1 would be empty.

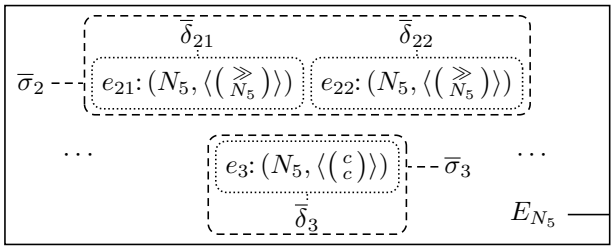


Fig. 3: Drawing from executions (E_{N_5}).

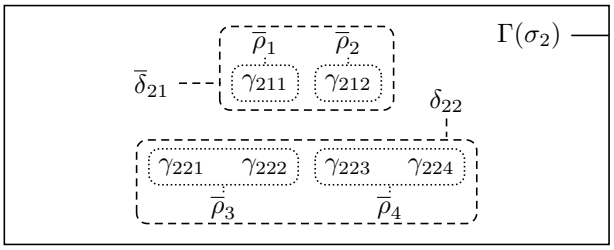


Fig. 4: Drawing from optimal alignments ($\Gamma(\sigma_2)$).

B. Skip Probabilities

In this section, we define skip probabilities for a process tree N and an event log L, which quantify conformance of subprocesses. That is, we annotate each node $N' \in N$ with the probability $P_{N'}(\text{skip})$ that N' is skipped when observing the process N'. We derive skip probabilities from the deviation information captured by all optimal skip alignments in normal form. This guarantees that the probabilities are founded on the full picture of all optimal alignments efficiently summarized by all optimal skip alignments in normal form. Formally, $P_{N'}(\text{skip})$ is the probability that when we draw a random execution, that is, a stencil, from $E_{N'}$, that execution is a skip move:

Definition 6 (Drawing from Executions)**.** *Let N be a process tree and $N' \in N$ a node. Then, the random experiment* drawing from executions of N' *is defined by*

- *the outcomes $E_{N'}$ being the set of executions of N'*
- *the following events:*
  - *$E_{skip} \cap E_{N'}$: skip executions of N', denoted 'skip'*
  - *$E_\delta \cap E_{N'}$ for $\delta \in \Delta$: executions of N' in δ, denoted $\overline{\delta}$*
  - *$E_\sigma \cap E_{N'}$ for $\sigma \in A^*$: executions of N' when synchronizing σ with N, denoted $\overline{\sigma}$*
- *the probability measure $P_{N'}(e)$ for all $e \in E_{N'}$*

Note that we overload the symbols δ and σ with events $\overline{\delta}$ and $\overline{\sigma}$, i.e., with sets of executions, to foster readability. Formally, skip alignments δ and traces σ *project* to subsets $\overline{\delta}$, $\overline{\sigma}$ of $E_{N'}$.

For our example set E_{N_5} in Figure 3, the boxes indicate all non-empty events of the example log L.

The desired skip probability $P_{N'}(\text{skip})$ rewrites with marginalization and the chain rule to individual executions:

$$P_{N'}(\text{skip}) = \sum_{e \in E_{N'}} P_{N'}(\text{skip} \mid e) \cdot P_{N'}(e) \tag{1}$$

The probability $P_{N'}(\text{skip} \mid e)$ is the probability that e is a skip on the node N', i.e., a property we can read off from the execution. Hence, we determine $P_{N'}(\text{skip} \mid e)$ with certainty:

$$P_{N'}(\text{skip} \mid e) = \begin{cases} 1 & \text{if } P_{N'}(e) \neq 0 \wedge e \in E_{\text{skip}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

For our example (Figure 1), $P_{N_5}(\text{skip} \mid e_{21}) = P_{N_5}(\text{skip} \mid e_{22}) = 1$ but $P_{N_5}(\text{skip} \mid e_3) = 0$, i.e., the first two executions are skip moves, the third one is not. To conclude $P_{N'}(\text{skip})$, we still need to derive the execution probability $P_{N'}(e)$, i.e., P_{N_5}.

C. Execution Probabilities

In this section, we derive $P_{N'}(e)$, the probability to draw e from $E_{N'}$. Intuitively, we ask for the probability to observe a certain skip alignment with a certain traversal of N'. First, we rewrite $P_{N'}(e)$ with marginalization and the chain rule:

$$\begin{aligned} P_{N'}(e) &= \sum_{\delta \in \Delta} P_{N'}(e \mid \overline{\delta}) \cdot P_{N'}(\overline{\delta}) \\ &= \sum_{\delta \in \Delta} P_{N'}(e \mid \overline{\delta}) \cdot \left(\sum_{\sigma \in A^*} P_{N'}(\overline{\delta} \mid \overline{\sigma}) \cdot P_{N'}(\overline{\sigma}) \right) \end{aligned} \tag{3}$$

The three factors intuitively describe a level-wise construction of $P_{N'}(e)$: Starting from all executions $E_{N'}$, the traces $\sigma \in A^*$ partition them into subsets $E_\sigma \cap E_{N'}$. Each partition is itself decomposed by optimal skip alignments in normal form $\delta \in \Delta(\sigma)$ into partitions $E_\delta \cap E_\sigma \cap E_{N'}$. In Equation (3), we relax the probability derivation to probabilities for each small partition. The probability $P_{N'}(\overline{\sigma})$ of sets $E_\sigma \cap E_{N'}$ is derived from the stochastic log language. It is then further distributed over the subsets $E_\delta \cap E_\sigma \cap E_{N'}$ resulting in $P_{N'}(\overline{\delta} \mid \overline{\sigma}) \cdot P_{N'}(\overline{\sigma})$. Finally, this probability is distributed to individual executions $e \in E_\delta \cap E_\sigma \cap E_{N'}$ with $P_{N'}(e \mid \overline{\delta}) \cdot P_{N'}(\overline{\delta} \mid \overline{\sigma}) \cdot P_{N'}(\overline{\sigma})$. We discuss how to infer each of these three probabilities using the stochastic path and log languages $\mathcal{P}_N$ and $\mathcal{P}_L$.

1) Trace Probability: For a trace σ, $P_{N'}(\overline{\sigma})$ is the probability that an execution of N' appears in a skip alignment of σ. The probability to observe the trace σ in reality is given by $\mathcal{P}_L(\sigma)$, hence we derive $P_{N'}(\overline{\sigma})$ from it.

Formally, if $\sigma \notin L$, then σ does not contribute to the conformance of L with N', i.e., we set $P_{N'}(\overline{\sigma}) = 0$. Note that not every $\sigma \in L$ may be explained with an execution of N', i.e., possibly $E_\sigma \cap E_{N'} = \emptyset$. Hence, we derive $P_{N'}(\overline{\sigma})$ from the subdistribution of traces that can be explained with N':

$$P_{N'}(\overline{\sigma}) = \begin{cases} \frac{\mathcal{P}_L(\sigma)}{\sum_{\sigma' \in \Omega} \mathcal{P}_L(\sigma')} & \text{if } \sigma \in L \wedge E_\sigma \cap E_{N'} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\Omega = \{\sigma' \mid \sigma' \in L \wedge E(\sigma') \cap E_{N'} \neq \emptyset\}$.

For example, the skip alignment δ_1 projects to the empty set on E_{N_5} in Figure 3. Hence, $P_{N_5}(\sigma_1) = 0$. The probabilities $\mathcal{P}_L(\sigma_2) = 0.2$ and $\mathcal{P}_L(\sigma_3) = 0.7$ rescale to $P_{N_5}(\overline{\sigma}_2) = 0.22$ and $P_{N_5}(\overline{\sigma}_3) = 0.78$ in the subdistribution of $\mathcal{P}_L$ on $\{\sigma_2, \sigma_3\}$.

2) Conditional Skip Alignment Probability: For a fixed trace σ, $P_{N'}(\overline{\delta} \mid \overline{\sigma})$ is the probability that the skip alignment δ contains an execution of N'. We derive this probability from a second random experiment describing the probability to obtain specifically δ when aligning σ to N. First, we introduce this second random experiment. Then, we derive $P_{N'}(\overline{\delta} \mid \overline{\sigma})$.

The experiment is defined on the set of optimal alignments $\Gamma(\sigma)$ of σ. On that set, $P_\sigma(\gamma)$ is the probability to observe the alignment γ for σ. Formally, $P_\sigma(\gamma)$ is the probability that

when drawing from the set of optimal alignments $\Gamma(\sigma)$ of σ, the drawn alignment is γ.

Definition 7 (Drawing from Alignments)**.** *Let N be a process tree, $\sigma \in A^*$ a trace, $\rho \in \mathcal{L}(N)$ a model path, and $\Gamma(\rho) \subseteq \bigcup_{\sigma' \in A^*} \Gamma(\sigma')$ the set of optimal alignments projecting to the model path ρ. Then, the random experiment* drawing from alignments of σ *is defined by*

- *the set of outcomes $\Gamma(\sigma)$ being the set of optimal alignments of σ on N*
- *the following events:*
  - *$\Gamma(\rho) \cap \Gamma(\sigma)$ for $\rho \in \mathcal{L}(N)$: optimal alignments of σ that project to ρ, denoted $\overline{\rho}$*
  - *$\bar{\mathcal{C}}(\delta) \cap \Gamma(\sigma)$ for $\delta \in \Delta$: optimal alignments of σ summarized by δ, denoted $\overline{\delta}$*
- *the probability measure $P_\sigma(\gamma)$ for $\gamma \in \Gamma(\sigma)$*

We overload ρ and δ with events $\overline{\rho}$ and $\overline{\delta}$ to foster readability.

For example, Figure 4 illustrates the outcomes $\Gamma(\sigma_2) = \{\gamma_{211}, \gamma_{212}, \gamma_{221}, \ldots, \gamma_{224}\}$ of σ_2 aligned to the process tree N, see Figure 1. The six alignments project to four model paths (dotted boxes): γ_{211} uses the model path ρ_1, γ_{212} uses ρ_2, γ_{221} and γ_{222} use ρ_3, γ_{223} and γ_{224} use ρ_4. Dashed boxes indicate which alignments summarize to the same skip alignment. The summaries of $\gamma_{211}, \gamma_{212}$ to δ_{21} and $\gamma_{221}, \ldots, \gamma_{224}$ to δ_{22} are the same as in Figure 1, discussed in Section III.

We derive $P_\sigma(\gamma)$ from the stochastic path language $\mathcal{P}_N$. Therefore, we rewrite $P_\sigma(\gamma)$ by marginalization and chain rule:

$$P_\sigma(\gamma) = \sum_{\rho \in \mathcal{L}(N)} P_\sigma(\gamma \mid \overline{\rho}) \cdot P_\sigma(\overline{\rho}) \tag{5}$$

Intuitively, we relax the probability derivation to model paths, i.e., the probability to observe the alignment γ given we see its model components are ρ, and the probability to see ρ itself.

The probability to observe a model path ρ is given by $\mathcal{P}_N(\rho)$, hence we derive $P_\sigma(\overline{\rho})$ from it. All paths $\rho \in \mathcal{L}(N)$, that can optimally be aligned to σ, they form a subdistribution on $\mathcal{L}(N)$ that we derive $P_\sigma(\overline{\rho})$ from:

$$P_\sigma(\overline{\rho}) = \begin{cases} \frac{\mathcal{P}_N(\rho)}{\sum_{\rho' \in \Omega} \mathcal{P}_N(\rho')} & \text{if } \Gamma(\rho) \cap \Gamma(\sigma) \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\Omega = \{\rho' \mid \rho' \in \mathcal{L}(N) \wedge \Gamma(\rho') \cap \Gamma(\sigma) \neq \emptyset\}$.

Next, we derive the conditional probability $P_\sigma(\gamma \mid \overline{\rho})$. If $P_\sigma(\overline{\rho}) = 0$, then no alignment of σ projects to the model path ρ, specifically not γ, i.e., $P_\sigma(\gamma \mid \overline{\rho}) = 0$. Likewise, alignments not using ρ map to $P_\sigma(\gamma \mid \overline{\rho}) = 0$. Because all alignments in $\Gamma(\rho) \cap \Gamma(\sigma)$ are cost-minimal, share the same model path ρ, and align the same trace σ, prior knowledge on ρ does not differentiate these alignments. Hence, we can assume a uniform distribution $P_\sigma(\gamma \mid \overline{\rho})$:

$$P_\sigma(\gamma \mid \overline{\rho}) = \begin{cases} \frac{1}{|\Gamma(\rho)|} & \text{if } P_\sigma(\overline{\rho}) \neq 0 \wedge \gamma \in \Gamma(\rho) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$\Gamma(\rho)$ is finite for every path ρ, hence $\frac{1}{|\Gamma(\rho)|}$ is well-defined.

Since alignments summarize to a unique skip alignment, $P_\sigma(\overline{\delta})$ is the sum of its coinciding alignment probabilities:

$$P_\sigma(\overline{\delta}) = \sum_{\gamma \in \bar{\mathcal{C}}(\delta)} P_\sigma(\gamma) \tag{8}$$

Note that $\bar{\mathcal{C}}(\delta)$ may be infinite, i.e., in practice we may compute $\bar{\mathcal{C}}(\delta)$ up to silent loop iterations (finite). Hence, all deviations are still captured, but their weighting becomes approximate.

For example, the optimal alignments of σ_2 cover the four model paths $\rho_1, \ldots, \rho_4$ with a total probability of $\mathcal{P}_N(\rho_1)+\mathcal{P}_N(\rho_2)+\mathcal{P}_N(\rho_3)+\mathcal{P}_N(\rho_4) = 0.8$, hence $P_{\sigma_2}(\overline{\rho}_3) = {}^{\mathcal{P}_N(\rho_3)}/_{0.8} = 0.375$. The alignment γ_{221} is one of two alignments of σ_2 using ρ_3, hence $P_{\sigma_2}(\gamma_{221} \mid \overline{\rho}_3) = {}^1/_{|\Gamma(\rho_3)|} = 0.5$. This leads to the alignment probability $P_{\sigma_2}(\gamma_{221}) = 0.5 \cdot 0.375 = 0.1875$. Repeating the derivation for $\gamma_{222}, \ldots, \gamma_{224}$ leads to $P_{\sigma_2}(\overline{\delta}_{22}) = P_{\sigma_2}(\gamma_{221}) + \ldots + P_{\sigma_2}(\gamma_{224}) = 0.75$.

Finally, we derive $P_{N'}(\overline{\delta} \mid \overline{\sigma})$ from $P_\sigma(\overline{\delta})$. This is possible, as P_σ is implicitly conditioned by σ. If $P_{N'}(\overline{\sigma}) = 0$, then σ cannot be explained with any execution of N', i.e., also not with an execution from the skip alignment δ, hence $P_{N'}(\overline{\delta} \mid \overline{\sigma}) = 0$. Similarly, if δ contains no execution of N', then it holds that $P_{N'}(\overline{\delta} \mid \overline{\sigma}) = 0$. Otherwise, $P_{N'}(\overline{\delta} \mid \overline{\sigma})$ is $P_\sigma(\overline{\delta})$, but restricted to the subdistribution of skip alignments that execute N':

$$P_{N'}(\overline{\delta} \mid \overline{\sigma}) = \begin{cases} \frac{P_\sigma(\overline{\delta})}{\sum_{\delta' \in \Omega} P_\sigma(\overline{\delta'})} & \text{if } P_{N'}(\overline{\sigma}) \neq 0 \wedge E_\delta \cap E_{N'} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\Omega = \{\delta' \mid \delta' \in \Delta(\sigma) \wedge E(\delta') \cap E_{N'} \neq \emptyset\}$.

In our example, the trace σ_2 has two optimal skip alignments in normal form, δ_{21} and δ_{22}. Hence, for δ_{22} we derive $P_{N'}(\overline{\delta}_{22} \mid \overline{\sigma}_2) = {}^{P_{\sigma_2}(\overline{\delta}_{22})}/_{P_{\sigma_2}(\overline{\delta}_{21}) + P_{\sigma_2}(\overline{\delta}_{22})} = 0.75$. Here, $P_{N'}(\overline{\delta}_{22} \mid \overline{\sigma}_2) = P_{\sigma_2}(\overline{\delta}_{22})$ as both δ_{21} and δ_{22} execute N_5.

3) Conditional Execution Probability: For an execution e and an optimal skip alignment in normal form δ, $P_{N'}(e \mid \overline{\delta})$ is the probability that an execution of N' in δ is e. If δ contains no execution of N' ($P_{N'}(\overline{\delta}) = 0$), then specifically δ contains not the execution e, hence $P_{N'}(e \mid \overline{\delta}) = 0$. Similarly, if e is not part of δ, then $P_{N'}(e \mid \overline{\delta}) = 0$. As we do not consider time, every move in a skip alignment, and thus every execution, appears atomic. Hence, we can assume a uniform distribution across the executions of N' in $\overline{\delta}$:

$$P_{N'}(e \mid \overline{\delta}) = \begin{cases} \frac{1}{|E_\delta \cap E_{N'}|} & \text{if } P_{N'}(\overline{\delta}) \neq 0 \wedge e \in E_\delta \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

This probability is well-defined because $E_\delta \cap E_{N'}$ is finite for every δ. Note that $P_{N'}(\overline{\delta}) = \sum_{\sigma \in A^*} P_{N'}(\overline{\delta} \mid \overline{\sigma}) \cdot P_{N'}(\overline{\sigma})$ is given by Equations (4) and (9).

In our example, the execution e_{22} is the only execution of N_5 in δ_{22}, hence certainty follows, i.e., $P_{N_5}(e_{22} \mid \overline{\delta}_{22}) = 1$. To derive $P_{N_5}(e_{22})$, all three results from Sections IV-C1 to IV-C3 are multiplied, leading to $P_{N_5}(e_{22}) = P_{N_5}(e_{22} \mid \overline{\delta}_{22}) \cdot P_{N_5}(\overline{\delta}_{22} \mid \overline{\sigma}_2) \cdot P_{N_5}(\overline{\sigma}_2) = 0.165$. The same computations for e_{21} and e_3 lead to $P_{N_5}(e_{21}) = 0.055$ and $P_{N_5}(e_3) = 0.78$. Only e_{21} and e_{22} describe a skip of N_5, e_3 does not. Hence, $P_{N_5}(\text{skip}) = P_{N_5}(e_{21}) + P_{N_5}(e_{22}) = 0.22$. We conclude: If N_5 is expected to take place, then with 22% probability, N_5 fails to execute, i.e., the *appeal* was not handled. The other non-zero skip probabilities are 10% for N_1 and 3% for N_6.

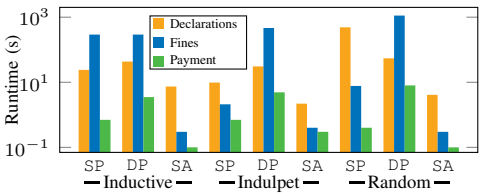


Fig. 5: Runtime overview by log and model.

TABLE I: Logs' Complexities.

Log	Variants
Declarations	753
Fines	231
Payment	89

V. Evaluation

First, we evaluate the runtime on a prototype implementation. Then, we discuss how to exploit skip probabilities for process insights. We use three publicly available event logs (BPIC20 Declarations, Road Fines, BPIC20 Payment, see Table I) and nine process models for our evaluation.

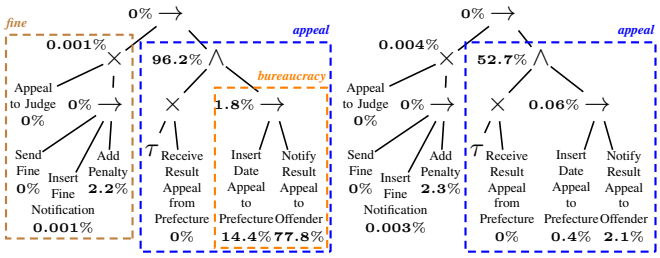
A prototype has been implemented in Python and is publicly available. We apply the following pipeline of five steps, whose instantiation we refer to with SP. First (S1), from each event log, process trees are discovered using the Inductive Miner as it is widely adopted in practice [21], the Indulpet Miner being a combination of miners and often Pareto optimal w.r.t. model quality measures [22], and a random process tree generator as a worst case scenario as it captures no relation to the log, see repository. Second (S2), we compute for each log and model all optimal skip alignments in normal form using [10]. Third (S3), we compute for each skip alignment those coinciding optimal alignments that never traverse both children of a loop with only model moves (no superfluous loop iterations, see (8)). Fourth (S4), we use Ebi [9] to estimate stochastic path languages. The tool uses a Petri net representation of the model, techniques for other model representations exist too [8]. Note that Ebi computes exact model path probabilities independent of the structural silent transitions used in Petri net models. The estimation is based on activity frequencies [7]; different estimators are conceivable, however, we expect similar results, because we use the estimator only to outweigh the (skip) alignments of the *same log trace*. The underlying skip information remains unchanged as *all* skip alignments are considered regardless of the used estimator, causing little variation in the resulting skip probabilities. Fifth (S5), we derive skip probabilities using our approach.

A. Runtime of the Skip Probability Derivation

We measure the runtime of SP aggregated over the steps (S2-S5). We compare the computation time of our approach against the only known other techniques: DP [16] computes deviation patterns from alignments and SA [10] (essentially (S2)) computes all optimal skip alignments in normal form and the relative frequency of skip moves. All computations are reproducible (5.8 GHz i9 CPU, 32 GB RAM). Figure 5 shows results for each of 9 instances, i.e., logs and models.

Across all instances, the average runtime was 91.6 s for SP, 224.6 s for DP, and 1.7 s for SA. SP can never be faster than SA, as the skip alignment computation is a step of SP (S2).

Across all instances, SA is fastest. The difference between SA and SP results from steps (S3-S4): Reversing the summarization and reordering moves takes up to 98% of the runtime of SP



(a) Skip probabilities from SP (b) Frequencies from SA

Fig. 6: Process tree excerpts with annotations.

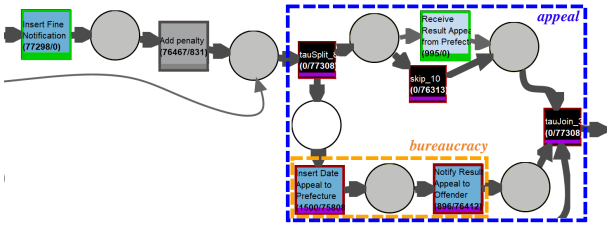


Fig. 7: Frequency annotated Petri net excerpt from RC.

(Inductive Miner, Fines log). It also causes a high runtime of SP for the Declarations log and the random model: 2041 optimal skip alignments for 753 variants summarized 6 930 693 optimal alignments; for each one, a path probability is computed.

On 8/9 instances, DP is slower than SP as an optimization problem needs to be solved. The only converse for the Declarations log and the random model results from the above described blow-up: DP only considers one alignment per trace.

Threats to validity include the different model architectures that the algorithms are designed for (process trees for SP, SA; arbitrary Petri nets for DP). They limit generalizability, but do not invalidate the conclusion that the computation of skip probabilities is feasible runtime-wise.

B. Process Insights from Skip Probabilities

As a case study, we inspect the Road Fines log and its Inductive Miner model, as derivable insights are similar for the other logs and models. We compare the output of our pipeline for skip probabilities SP against three conformance annotations to discuss commonalities and differences in the revealed process insights: Deviation patterns of log and model moves derived by DP [16], relative frequencies of skip moves in skip alignments SA [10], and absolute frequencies of synchronous and model moves from the ProM plugin 'Replay a Log on Petri Net for Conformance Analysis' RC [23]. A comparison against [18] is not possible, as they stochastically compute alignments but do not use them to derive process deviation insights at any level of the process. Figure 6a shows probabilities SP for every subprocess that at least requires to execute one labeled activity:

Using one optimal alignment per trace, DP computes five types of deviation patterns. Out of the 442 patterns found, 140 were skip patterns. Only one skip pattern operates on a process structure higher than activities, being the choice in Figure 6a (fine box). DP provides no statistical information for that skip, while SP detects and quantifies high-level deviations for every level: The probability that the choice 'judge/no judge' was expected but missing out in reality was 0.001%. Given that SP

assess and quantifies high-level deviations for all subprocesses, we conclude that SP complements DP with additional insights.

SA utilizes all optimal skip alignments in normal form for relative frequencies on how often each subprocess was skipped, shown in Figure 6b. The frequencies cannot be interpreted as probabilities, since they reflect the fraction of skip alignments in which a certain subprocess was skipped. This fraction is not corrected w.r.t. the varying number of skip alignments per trace variant, i.e., the traces' importance. That is, the 50.7% at the parallel (appeal box) reflect that half of the computed skip alignments skipped the subprocess, but not half of the observed traces. SP computes interpretable, local probabilities revealing that 96.2% of the intended 'appeal' subprocesses were missing.

Using one optimal alignment per trace, RC annotates each activity with the total number of observed moves (synchronous/model), given in Figure 7. Relying on a single optimal alignment, RC lacks completeness, e.g., it attributes perfect compliance to the activity 'Insert Fine Notification' (no model move), while SP assigns a positive skip probability from inspecting all optimal (skip) alignments. Further, insights from RC remain at the level of individual activities: $4/5$ activities of the 'appeal' subprocess (appeal box) and the entire 'appeal bureaucracy' subprocess (bureaucracy box) are dominated by model moves, but do not reveal whether the observed traces were missing the subprocesses partially or entirely. SP resolves this ambiguity: With a probability of 96.2%, handling an appeal was expected but not observed. Once it was observed, 'appeal bureaucracy' usually takes place (only 1.8% missing). However, the appeal decision tends to not be communicated to the offender (77.8% missing). We conclude that SP is superior to RC regarding finding and quantifying process level deviations.

Overall, the process insights from skip probabilities go beyond the ones of existing techniques, are interpretably quantified, and hence actionable for process improvement.

VI. Conclusion

This paper presented skip probabilities that stochastically quantify model conformance of subprocesses with event logs. This allows us to reason about not just whether deviations exist, or that particular activities may miss out, but how likely deviations at the level of entire subprocesses are.

We defined subprocesses in process trees and their executions, which compare modeled and observed process behavior for groups of related activities. We extracted executions from skip alignments, providing complete deviation information. From executions we derived skip probabilities for any log and process tree with stochastic languages. An investigation of a real world case then showed how this can translate to process insights.

The technique presented has the following limitations: It considers only all optimal (skip) alignments, and the implementation approximates them once this set becomes infinite. This is a direct cause of the yet open challenge of summarizing subprocess probabilities for loops. Additionally, skip probabilities rely on skip alignments, currently only defined for hierarchical process models. Finally, work with domain experts would further validate the technique. Future work may overcome these limitations by investigating deviations with non-minimal costs, non-hierarchical models (e.g., for semi-structured models), exact loop probabilities, and domain expert involvement. Further research may investigate skip probabilities for outcome prediction or process comparison.

References

[1] W. M. P. van der Aalst, *Process Mining Handbook*, 2022.

[2] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking*, 2018.

[3] D. Fahland and W. M. P. van der Aalst, "Model repair - aligning process models to reality," *Inf. Syst.*, vol. 47, 2015.

[4] M. L. van Eck, "Alignment-based process model repair and its application to the evolutionary tree miner," Master's thesis, TU/e, 2013.

[5] T. Brockhoff, M. N. Gose, M. S. Uysal, and W. M. P. van der Aalst, "Process comparison using petri net decomposition," in *Petri Nets*, 2024.

[6] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "A genetic algorithm for discovering process trees," in *IEEE Congress on Evolutionary Computation*. IEEE, 2012.

[7] A. T. Burke, S. J. J. Leemans, and M. T. Wynn, "Stochastic process discovery by weight estimation," in *Process Mining Workshops - ICPM 2020*, S. J. J. Leemans and H. Leopold, Eds., 2020.

[8] ——, "Discovering stochastic process models by reduction and abstraction," in *Petri Nets*, D. Buchs and J. Carmona, Eds., 2021.

[9] S. J. J. Leemans, T. Li, and J. N. van Detten, "Ebi - a stochastic process mining framework," in *ICPM Doctoral Consortium and Demo Track. CEUR Workshop Proceedings, vol. to appear. CEUR-WS. org*, 2024.

[10] P. Bär, M. T. Wynn, and S. J. J. Leemans, "A full picture in conformance checking: Efficiently summarizing all optimal alignments," 2025, BPM in press. [Online]. Available: https://leemans.ch/philipp_paper.pdf

[11] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2012.

[12] A. Adriansyah, "Aligning observed and modeled behavior," Ph.D. dissertation, Mathematics and Computer Science, 2014.

[13] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Inf. Syst.*, vol. 33, no. 1, 2008.

[14] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *WIREs Data Mining Knowl. Discov.*, vol. 2, no. 2, 2012.

[15] S. J. J. Leemans, S. J. van Zelst, and X. Lu, "Partial-order-based process mining: a survey and outlook," *Knowl. Inf. Syst.*, vol. 65, no. 1, 2023.

[16] M. Grohs, H. van der Aa, and J. R. Rehse, "Beyond log and model moves in conformance checking: Discovering process-level deviation patterns," in *BPM*, 2024.

[17] S. J. J. Leemans, W. M. P. van der Aalst, T. Brockhoff, and A. Polyvyanyy, "Stochastic process mining: Earth movers' stochastic conformance," *Inf. Syst.*, vol. 102, 2021.

[18] G. Bergami, F. M. Maggi, M. Montali, and R. Peñaloza, "Probabilistic trace alignment," in *ICPM*. IEEE, 2021.

[19] M. P. de Almeida, K. V. Delgado, S. M. Peres, and M. Fantinato, "Alignment-based conformance checking for stochastic petri nets," *Künstl. Intell.*, 2025.

[20] E. G. Rocha, S. J. J. Leemans, and W. M. P. van der Aalst, "Stochastic conformance checking based on expected subtrace frequency," in *ICPM*, 2024.

[21] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering block-structured process models from event logs - a constructive approach," in *Petri Nets*, 2013.

[22] S. J. J. Leemans, N. Tax, and A. H. M. ter Hofstede, "Indulpet miner: Combining discovery algorithms," in *On the Move to Meaningful Internet Systems. OTM*, 2018.

[23] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, "The prom framework: A new era in process mining tool support," in *Petri Nets*, 2005.