1 uTraVaS: The Utility-Aware TraVaS

The following supplementary material contains the theoretic foundations and experimental results of *uTraVaS*, an optimization framework for *TraVaS* based on a data utility cost function. We refer to mathematical definitions and the standard *TraVaS* algorithm of the main paper *TraVaS: Differentially Private Trace Variant Selection for Process Mining* (M. Rafiei et al.).

1.1 Algorithm Design

As an extension of *TraVaS*, the optimizer framework *uTraVaS* serves as a wrapper around the standard *TraVaS* algorithm and exploits the *composition theorem* and *post-processing immunity* properties of (ϵ, δ)-DP to spend the privacy budgets, ϵ and δ, w.r.t. an optimal data utility preservation. Subsequently, we first introduce these properties and then explain the algorithm.

Theorem 1 ((ϵ, δ)-DP Mechanism Composition [1]) *Let L be a simple event log, and $\mathcal{M}_i, 1 \leq i \leq n$ be (ϵ_i, δ_i)-DP mechanisms. The sequential application of these mechanisms on arbitrary sublogs of L leads to an overall worst-case privacy level parameterized by $(\sum_{1 \leq i \leq n} \epsilon_i, \sum_{1 \leq i \leq n} \delta_i)$. If each $\mathcal{M}_i$ operates on strictly disjoint sublogs of L, the worst-case privacy level is $(max_{1 \leq i \leq n}\ \epsilon_i, max_{1 \leq i \leq n}\ \delta_i)$, so-called parallel composition.*

Post-processing immunity states that (ϵ, δ)-DP is immune to post-processing activities on mechanism outputs [1]. Hence, parties without additional knowledge about an event log L cannot transform the output in a way that makes the information less differentially private.

According to Theorem 1, a privacy budget of (ϵ, δ) may be consumed by either one *travas* query or n subqueries with parameters $(\epsilon/n, \delta/n)$. Algorithm 1 receives the inputs of the *TraVaS* algorithm and the maximum number of subqueries n and returns a merged anonymized event log that preserves more data utility while spending the same privacy budgets. Since the relation between the number of subqueries and the data utility of the resulting anonymized event log is non-linear, *uTraVaS* searches the whole space of possible number of subqueries from 1 to n to find the decomposition structure that makes the best use of the privacy budgets for *TraVaS*.

In line 6, i different anonymized sublogs are compared, and all the unique variants from different sublogs are added to a potential final anonymized log pL'. The rationale behind this is that the partition selection mechanism never generates fake variants, and it may remove some infrequent trace variants. Thus, all the variants that appear in different sublogs are among the original trace variants. Note that the privacy degradation due to the knowledge that one can obtain by comparing i different sublogs are already considered by reducing the privacy parameters from ϵ and δ to ϵ/i and δ/i (see Theorem 1). The frequency of the variants are calculated as we explain in the following. Note that calculating frequency values for the trace variants in pL' is a post-processing step where the noisified frequencies are altered without using original frequency values. Thus, because of the post-processing immunity the provided privacy guarantees are not degraded.

```
Algorithm 1: Utility-Aware TraVaS (uTraVaS)
Input: Event log $L$, DP-Parameters $(\epsilon, \delta)$, Subquery bound $n$
Output: $(\epsilon, \delta)$-DP log $L'$
function optimize $(L, \epsilon, \delta, n)$
    forall $i \in \{1, 2, \ldots, n\}$ do
        create $i$ empty sublogs: $\{sL_1, sL_2, \ldots, sL_i\}$             // initialize
        forall $j \in \{1, \ldots, i-1, i\}$ do
            add travas($L$, $\epsilon/i$, $\delta/i$) to $sL_j$              // run TraVaS
        add all unique variants $\sigma$ from sublogs to a simple log $pL'$
        forall $\sigma \in pL'$ do
            forall $sL \in$ sublogs do
                if $\sigma \in sL$ then
                    $f_\sigma = sL(\sigma)$                 // get frequency from sublog
                else
                    $f_\sigma$ = freqEstimate $(\epsilon, \delta)$          // estimate frequency
                update $pL'(\sigma)$ to $pL'(\sigma)$ + $f_\sigma$ in $pL'$
        update $pL'(\sigma)$ to $pL'(\sigma)$ / $i$ in $pL'$                  // compute mean
        add $pL'$ to the set pool.                           // optimization domain
    $L' = \underset{pL' \in pool}{\mathrm{argmax}}\,$ utility$(L, pL')$          // optimize data utility
    return $L'$
```

For variants contained in all i sublogs, we calculate the mean of the associated noisified frequencies as the best unbiased true frequency estimator due to the symmetric zero-mean of *k-TSGD*. However, there may be some variants that get truncated in some sublogs and do not appear within all sublogs. For such variants, the frequencies must first be estimated before computing the mean. Since we know that the true frequencies lie in the range $[1, 2k]$ due to the k-TSGD interval of $[-k, k]$ and the threshold at k, the maximal removed true frequency is $2k$. Such estimations represent conditional expectation values of the form $E[L(\sigma) \mid \sigma{\in}L{\wedge}L(\sigma){>}0{\wedge}L(\sigma){+}x_\sigma \leq k]$. Consequently, all possible frequency realizations have to be combined with their likelihoods of producing perturbed outputs under *k-TSGD*. Using the set $T(j) = \{l \mid l{\in}Z{\wedge}j + l{\leq}k{\wedge}|l|{\leq}k\}$, we formally define the following estimator.

$$\text{freqEstimate}(\epsilon, \delta) = \frac{\sum\limits_{j=1}^{2k} \sum\limits_{l \in T(j)} j \cdot \text{k-TSGD}[l \mid p, k]}{\sum\limits_{j=1}^{2k} \sum\limits_{l \in T(j)} \text{k-TSGD}[l \mid p, k]} \tag{1}$$

After all n anonymized candidate logs are assembled, their respective data utility is measured by means of a suitable utility evaluation function and w.r.t. the original event log. We employ the *earth mover's distance* as a generic data utility function to quantify the similarity between two distributions of trace variants [2].[1] In a final step, the optimizer

[1] Note that the utility function can be adjusted w.r.t. any specific utility demand.

then releases the best performing candidate to the public.

We particularly note that in line 16, *uTraVaS* uses the utility score to select the best performing anonymized event log based on the number of *travas* subqueries. This is a selection step that utilizes the private original event log to select one of the possible query decomposition options but does not interfere with the random mechanism itself. The private information is never used during the noise injection process or in the post-processing step. Thus, as long as no internal state information such as the actual number of queries or the utility scores is published the final output of the algorithm is (ϵ, δ) differentially private.

The log construction and noise generation routines in Algorithm 1 are parallelizable and only require one data query. As a result, our approach faces a considerably lower complexity compared to traditional DP query methods where iterations need to run sequentially. A more precise computational complexity analysis has been provided as supplementary material in our GitHub repository.[2]

1.2 Experiments

For a direct performance comparison between *TraVaS* and *uTraVaS*, we supplement our experimental results of the main paper with a data utility and result utility evaluation on *uTraVaS*. The event data used (Sepsis, BPIC2013) as well as the setup parameters remain unchanged. Figure 1.1 and Figure 1.2 illustrate side-by-side heatmaps of *relative log similarity* and *absolute log difference* on *uTraVaS*, *TraVaS* and our benchmark for BPIC2013 and Sepsis respectively. Both logs demonstrate superior performance of *uTraVaS*, in particular for $\delta > 0.001$. This observation can be explained by the merging process and frequency estimation of Algorithm 1 that supports infrequent variants of largely unstructured event logs.

With Figure 1.3 and Figure 1.4, we present the corresponding *fitness* and *precision* results for BPIC2013 and Sepsis. Compared to Sepsis event data that again show similarly increased scores, BPIC2013 hardly demonstrates any noticeable effect. Since this event log is considerably larger and possesses a lower trace uniqueness, the noise filter of the discovery algorithm suppresses the relatively smaller frequency deviations caused by *uTraVaS*.

We conclude that *uTraVas* provides significant performance increase for most privacy settings, particularly on unstructured, small or highly trace-unique event data.

[2] `https://github.com/wangelik/TraVaS`

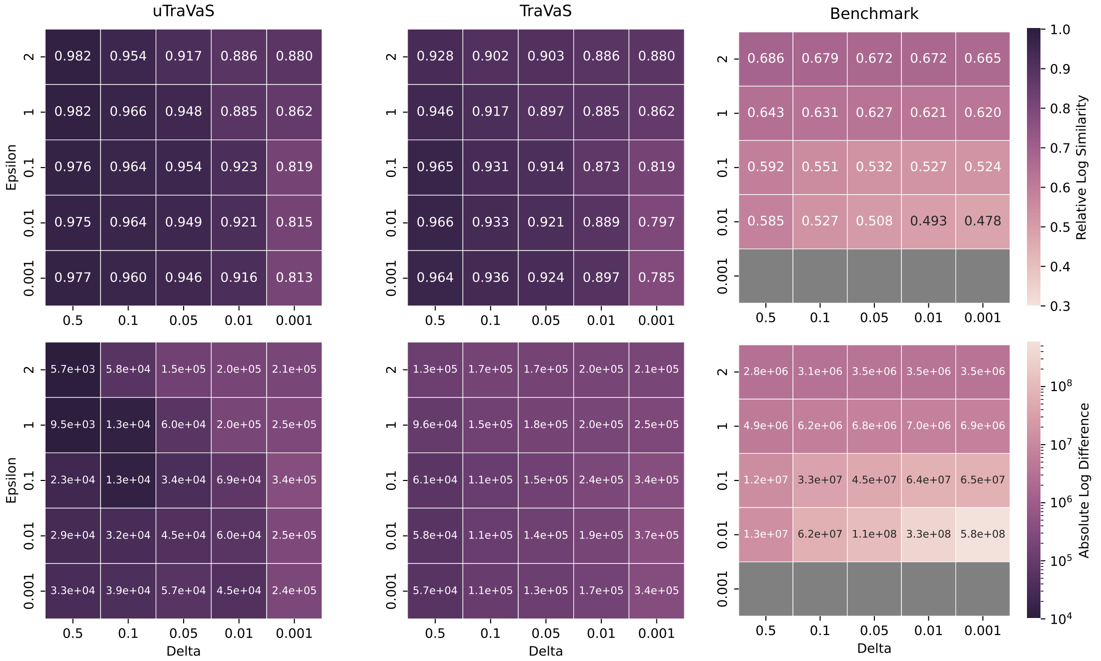

Figure 1.1: The *relative log similarity* and *absolute log difference* results of anonymized BPIC2013 event logs generated by *uTraVaS*, *TraVaS*, and our prefix-based benchmark method. Each value represents the mean of 10 algorithm runs.

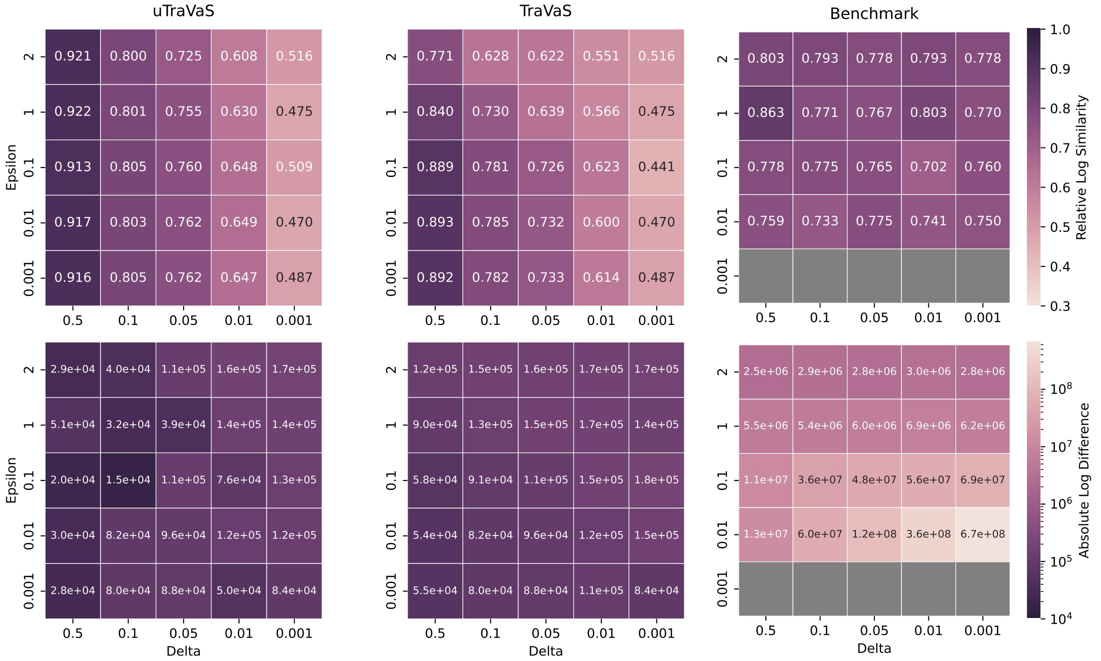

Figure 1.2: The *relative log similarity* and *absolute log difference* results of anonymized Sepsis event logs generated by *uTraVaS*, *TraVaS*, and our prefix-based benchmark method. Each value represents the mean of 10 algorithm runs.

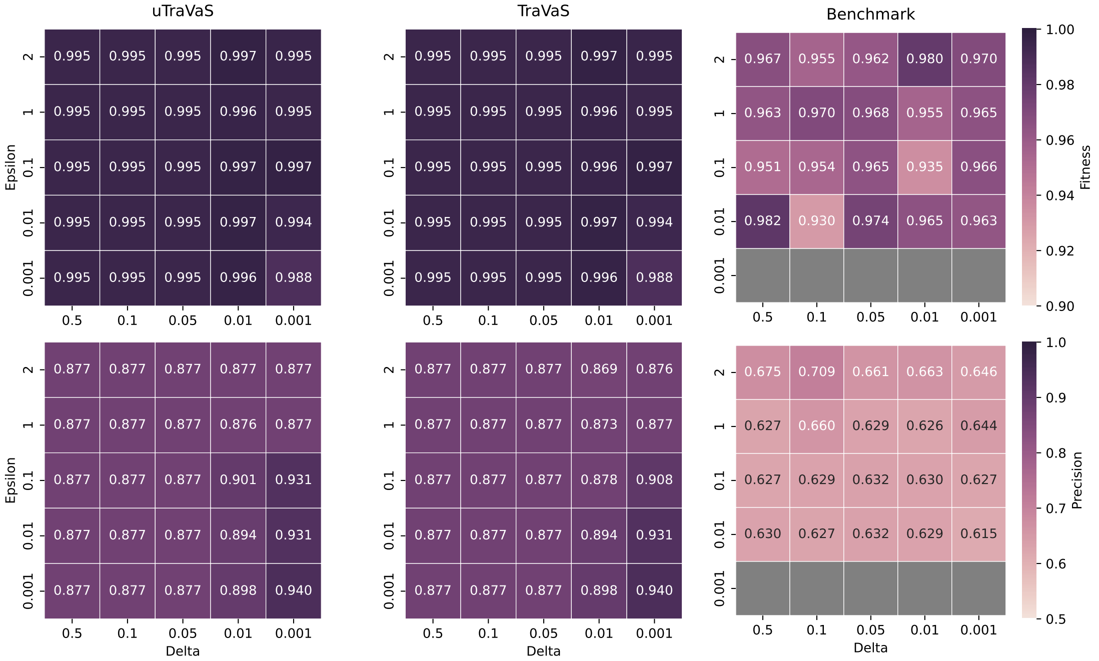

Figure 1.3: The *fitness* and *precision* results of anonymized BPIC2013 event logs generated by *uTraVaS*, *TraVaS*, and our prefix-based benchmark method. Each value represents the mean of 10 algorithm runs.

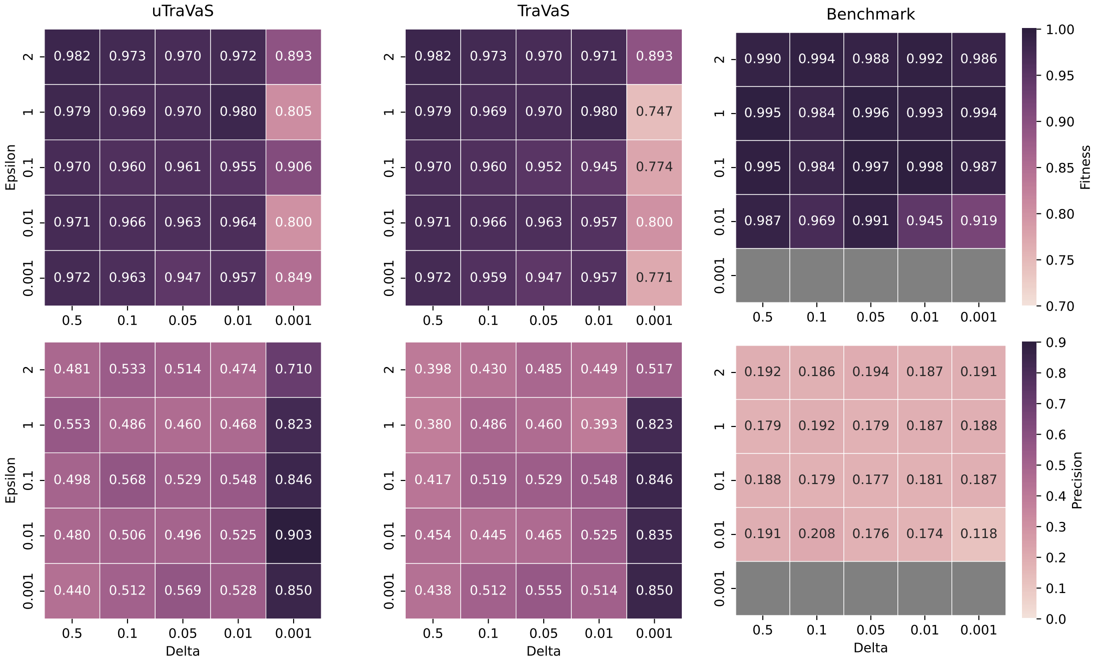

Figure 1.4: The *fitness* and *precision* results of anonymized Sepsis event logs generated by *uTraVaS*, *TraVaS*, and our prefix-based benchmark method. Each value represents the mean of 10 algorithm runs.

1.3 Note of Authorship

This document is part of the supplementary material created for the paper "TraVaS: Differentially Private Trace Variant Selection for Process Mining", written by Majid Rafiei, Frederik Wangelik and Wil M.P. Van der Aalst. Please contact *frederik.wangelik@rwth-aachen.de* for further information.

References

[1] C. Dwork, “Differential privacy: A survey of results,” in *Theory and Applications of Models of Computation, 5th International Conference*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Springer, 2008.

[2] M. Rafiei and W. M. P. van der Aalst, “Towards quantifying privacy in process mining,” in *Process Mining Workshops - ICPM 2020 International Workshops*, ser. Lecture Notes in Business Information Processing. Springer, 2020.