1 Computational Complexity

One essential difference of TraVaG and the diffusion-based generator (DDPM) compared to traditional differentially private selection-based anonymization techniques is their computational complexity structure. Whereas the training procedure complexity primarily scales with the input data, the network sizes, and the number of epochs, the actual repetitive task of privatized sample generation only represents efficient forward passes of multivariate random noise through the generator networks. The underlying matrix operations are computed in parallelized fashion on most modern hardware, leading to sufficiently fast down-stream applications. In the following, we present the most important network- and runtime characteristics of both approaches.

1.1 Network Characteristics

For our experiments, all CPU-based computations are performed using an Intel Core i7-8550U CPU (8 virtual cores at 1.8GHz) to represent classical end-user execution with limited computing power. Additionally, training runtime is compared to GPU-supported performance on a Nvidia Tesla P100 card (16Gb HBM2, 16GHz).

Table 1: TraVaG complexity characteristics

Attribute	BPIC12App	BPIC13	Sepsis
# networks	3	3	3
# layer	13	13	13
# parameters	10790	4767991	1637865
# epochs	10	10	10
GPU training time (sec)	50.1	100.4	38.7
CPU training time (sec)	88.2	3145.8	831.5
CPU sampling time (sec)	0.2	0.3	0.2

Table 2: DDPM complexity characteristics

Attribute	BPIC12App	BPIC13	Sepsis
# networks	1	1	1
# layer	5	5	5
# parameters	11781	9147599	8594514
# epochs	10	15	15
GPU training time (sec)	12.2	287.3	8.5
CPU training time (sec)	19.4	-	319.8
CPU sampling time (sec)	7.2	488.1	405.5

Tab. 1 and Tab. 2 show the main characteristics and runtime results of TraVaG and DDPM for all three event logs (BPIC12App, BPIC13, Sepsis). Due to its adversarial structure, TraVaG comprises a more complex architecture with 3 different networks and 13 layer in total. Nevertheless, the DDPM consists of larger networks and thus more parameters, particularly when the input dimension grows (note at BPIC13 and Sepsis data).[1] While *#epochs* denotes our optimal training epoch estimate, the CPU- and GPU training time refers to only 10 epochs to allow for comparison. At the training runtime, DDPM outperforms TraVaG on smaller datasets, but underperforms on larger datasets. This effect could be explained by the two-fold reverse- and forward diffusion process involved.[2] A clear disadvantage of DDPM is its sampling procedure. In contrast to the simple, parallelizable, one-shot forward generation of TraVaG, DDPM is limited to its sequential noise scheduler, leading to a clearly prolongated sampling time.[3]

1.2 Note of Authorship

This document is part of the supplementary material created for the paper *Releasing Differentially Private Event Logs Using Generative Models*, written by Frederik Wangelik, Majid Rafiei, Mahsa Pourbafrani and Wil M.P. Van der Aalst. Please contact *frederik.wangelik@rwth-aachen.de* for further information.

[1]The significant difference in parameter numbers for the different event logs mainly results from varying input dimensions (17 at BPIC12App, 846 at Sepsis, 1511 at BPIC13).

[2]With BPIC13, our DDPM model was unable to train on a CPU within reasonable time.

[3]The individual sequential compuations could also be computed by a GPU instead.