8. GAUDA - Supplementary

This chapter provides additional supplementary information for *GAUDA: Generative Adaptive Uncertainty-guided Diffusion-based Augmentation for Surgical Segmentation*.

8.1. GAUDA Pseudocode

A pseudocode formulation for training Bayesian downstream task models with GAUDA can be found in Figure 9. GAUDA serves as a training scheme with flexible options for the downstream task, downstream model and uncertainty estimation method.

```
**Data:** Training dataset $\mathcal{D}$, pre-trained class-conditional generative model $\mathcal{G}$, Bayesian down-stream model $\mathcal{F}_\theta$
**Input:** Sampling probability $p$, number of epochs $N$, batch size $B$, top-k uncertainty classes $k$, learning rate $\alpha$,
        validation frequency $V$
**Output:** Optimised model $\Phi^*_\theta$

Initialize $\mathcal{F}_\theta$ with pre-trained weights if available
for epoch $= 1$ to $N$ do
    for each batch $b$ in $\mathcal{D}$ do
        Sample mini-batch $\{x_i\}_{i=1}^B$ from $\mathcal{D}$
        for each $x_i$ in $\{x_i\}_{i=1}^B$ do
            if $\mathcal{X}_{syn}$ was synthesised before then
                With probability $p$, replace $x_i$ with synthetic data $x'_i \sim \mathcal{X}_{syn}$
            end
        end
        Compute loss $\mathcal{L}$ for $\mathcal{F}_\theta$ using batch $\{x_i\}_{i=1}^B$
        Update model weights $\theta$ using gradient descent with learning rate $\alpha$: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$
    end
    if epoch $\%V == 0$ then
        Perform validation on a held-out dataset
        Compute class-wise uncertainty estimates using Bayesian inference
        Identify top-k classes $\mathcal{C}_k$ with highest uncertainty
        Synthesise a new batch of data $\mathcal{X}_{syn}$ conditioned on classes $\mathcal{C}_k$ using $\mathcal{G}$
    end
end
Return optimised model $\mathcal{F}^*_\theta = \mathcal{F}_\theta$
```

Figure 9. **GAUDA Pseudocode.**

8.2. Uncertainty-based Sampling Versus Score-based Sampling

In this section, we analyse the effect of the predictive epistemic uncertainty as quantity for re-defining sampling weights in adaptive sampling. For that purpose, we define a simplified classification problem of two-dimensional points. As visualised in the top left plot of Figure 10, the data consists of two noisy classes depending on their centre distance (red and blue). The data shows a significant imbalance in the number of samples per class.

We deploy a simple neural network classifier with two fully connected layers, an intermediate feature size of 10 nodes and a ReLU activation + dropout with 50% chance. To obtain UE, we build an ensemble of 20 of such models.

We compare two training schemes to investigate the effect of different quantities for redefining sample weights. First, we use the validation accuracy analogously to the original AS formulation. Second, we use the epistemic uncertainty from the variance of the ensemble prediction.

As visualised in the top right and bottom plots of Figure 10, sampling based on UE yields faster convergence and improved generalisation capabilities, ultimately resulting in a testing accuracy improved by 6.1%.

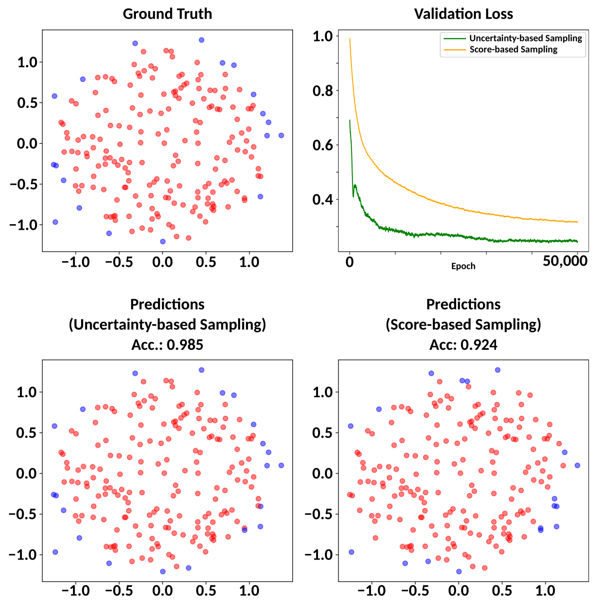

Figure 10. **Uncertainty- Versus Score-based Sampling.**

8.3. Pre-Train Augmentation Versus Online Augmentation

Adaptive augmentation of training data leads to a change in the data distribution, favouring underrepresented data points more and more during training. To demonstrate this, we first sample a fixed amount of additional samples of our simplified experimental data (doubling the number of examples). Second, we train the deep ensemble from Section 8.2 with the GAUDA scheme, adaptively augmenting the data based on the predictive epistemic uncertainty (again doubling the number of examples).

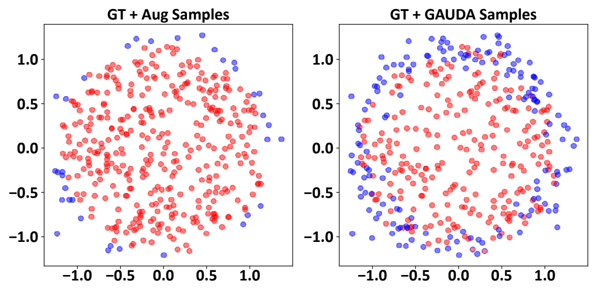

Figure 11. **Pre-train Versus Online Augmentation.**

Figure 11 shows adaptive online augmentation yields a significantly higher percentage of samples of the limited blue class compared to pre-train random augmentation.

8.4. Additional Segmentation Scores

The sample-wise and mean DICE and Average Precision (AP) scores for the surgical segmentation downstream task are visualised in Figure 12.

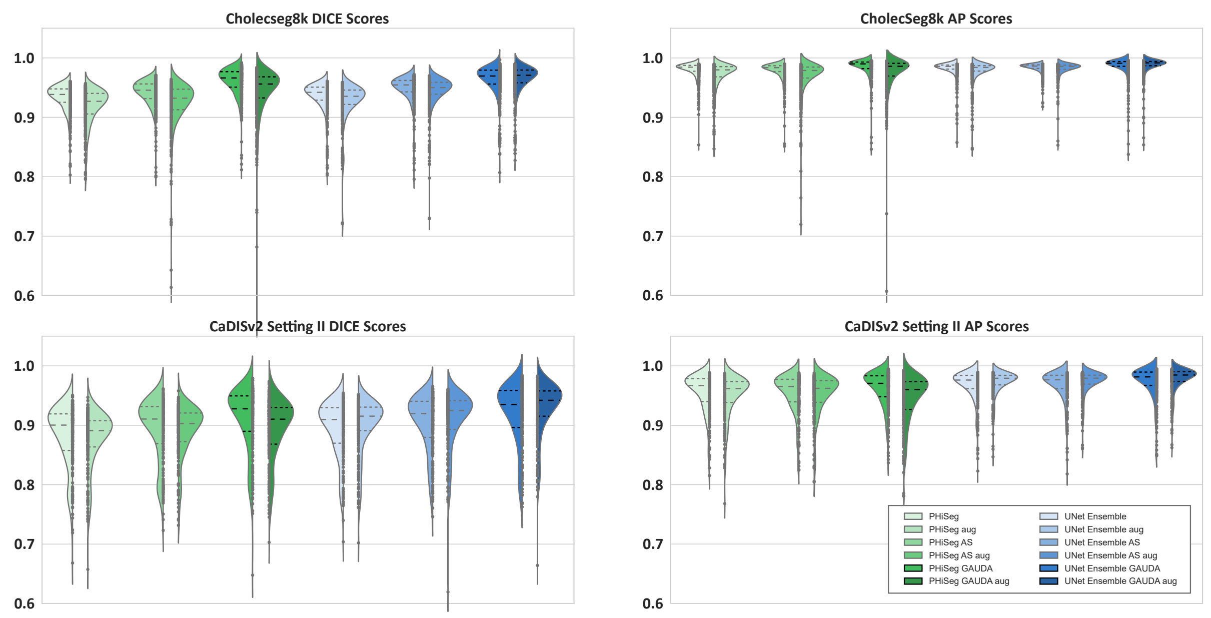

Figure 12. **Downstream DICE and AP Results.**

8.5. Failure Cases

Figure 13 displays examples of synthetic (image, mask) pairs with improvable quality. Notably, failures can occur in the form of noisy tool segmentation masks (left column), wrongly allocated or missing labels (middle column) and small inconsistent regions (right column). Yet, erroneous regions are small and can potentially be filtered out or improved with error propagation reduction mechanisms [43].

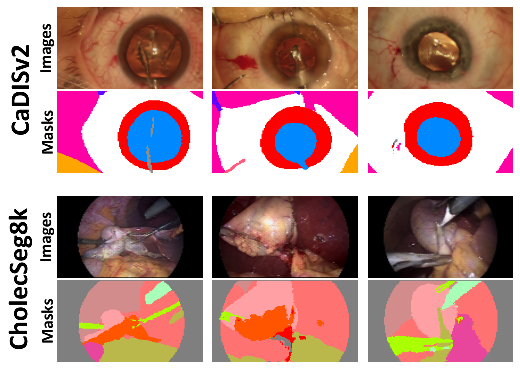

Figure 13. **Erroneous Synthetic Samples.**

8.6. Comparison to GAN-based Approaches

In Table 5, we compare the quantitative performance of our LDM ($\omega = 3.0$) to DatasetGAN [45]. The DatasetGAN implementation is based on an improved version from EditGAN [25]. It uses StyleGAN2 [18] with 'config-f', trained on $2e6$ random examples of both datasets, which was sufficient for convergence. For training the interpreter model, we used 50 annotated (image, mask) pairs and early stopping based on the loss progress.

Method	Dataset	FID (↓)	KID (↓)	RO IoU (↑)	SO IoU (↑)
LDM (ours)	CaDISv2	39.44	0.033 ± 0.004	0.755	0.635
DatasetGAN [45]	CaDISv2	66.99	0.057 ± 0.009	0.281	0.213
LDM (ours)	CholecSeg8k	56.80	0.041 ± 0.005	0.731	0.742
DatasetGAN [45]	CholecSeg8k	65.40	0.042 ± 0.007	0.114	0.145

Table 5. **Quantiative Comparison against DatasetGAN.**

Notably, our generative model surpasses DatasetGAN in terms of fidelity, but especially in RO and SO scores, indicating a superior semantic alignment between images and masks of generated pairs.

8.7. Computational and Resource Efficiency

Table 6 lists the number of training examples, the total training time and inference speed of each component of our proposed method, each component of DatasetGAN [45], as well as each training scheme. The reported numbers are averaged over datasets and downstream task models. The inference speed is reported for a single sample.

Method / Component	Num. Examples	Training Time	Inference Speed
Image VQ-GAN	$1.2e6$	11.4h	4ms
Mask VQ-GAN	$1.2e6$	20.8h	5ms
LDM	$2.0e7$	125.1h	36,891ms
Full Model (ours)	-	-	36,895ms
StyleGAN2	$2.0e6$	36.9h	290ms
StyleGAN Encoder	$6.0e5$	7.1h	9ms
DatasetGAN	$\leq 4.0e6$	0.25h	380ms
Downstream Default	$2.5e6$	15.3h	8ms
Adaptive Sampling	$2.5e6$	18.3h	8ms
GAUDA	$2.5e6$	22.6h	8ms

Table 6. **Training Times and Inference Speed.**