Tyche: Stochastic In-Context Learning for Medical Image Segmentation

Supplementary Material

A. Overview

We first present a brief overview of the analysis and information in this Supplemental Material.

Tyche Model Choices. We present additional data details, including for MegaMedical, multi-annotator data, and simulated data. We give additional details on the *Tyche-TS* training strategy, as well as the augmentations used at inference for *Tyche-IS*. We also detail how we trained each benchmark, and their respective limitations.

We provide an intuition behind the best candidate Dice loss, and show that with *Tyche*, it is possible to optimize for objectives that apply to the candidate predictions as a group, and show results when optimizing GED.

Tyche Analysis. For *Tyche-TS*, we investigate four aspects: noise, context set, number of predictions, and *Set-Block*. We show how different noise levels impact the predictions. Moreover, we give examples of how prediction changes when the context or the noise changes. We also show that the number of predictions at inference impacts the diversity of the segmentation candidates predicted.

For *Tyche-IS*, we investigate how different augmentations affect performance.

We also analyze a scenario where only few of annotated examples are available, using the LIDC-IDRI dataset. We compare *Tyche* with these samples in the context to PhiSeg, trained on these few annotated samples.

Further Evaluation. Given the nature of stochastic segmentation tasks, no single metric fits all purposes, and the optimal metric depends on the downstream goals. We provide *sample diversity* and *Hungarian Matching*, showing that *Tyche* performs well on these as well.

We additionally provide performances on 3 unseen datasets: CHASE [34], COBRE [2] and TotalSegmentator [132]. We also present performance per dataset and show that the relative performance of each model stays generally unchanged, even though the difficulty of each dataset is widely different.

We provide additional visualizations on how *Tyche* and the baselines perform for each datasets, both on single and multi-annotator data.

Data split. All analysis results in this supplemental material use the validation set of the out-of-distribution datasets, to avoid making modeling decisions on the test sets.

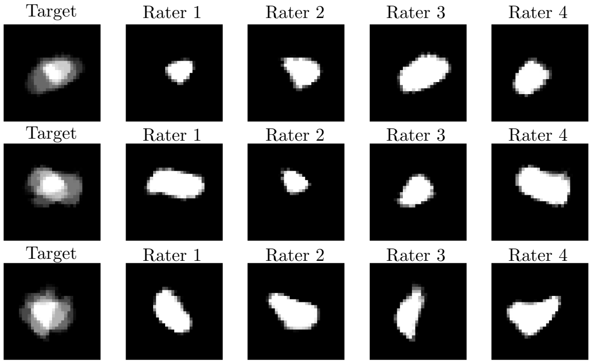

Figure 8. **Synthetic Multi-Annotated Examples.** Each blob is generated by deforming a white disk using a random smooth deformation field, each representing a different rater. The blobs are then averaged to form the target image to segment.

B. Tyche Model Choices

We present the implementation details for the data, our *Tyche* models, and the benchmarks.

B.1. Medical Image Data

Megamedical. We build on the dataset collection proposed in [20, 134], using the similar preprocessing methods. The complete list of datasets is presented Table 10.

Processing. Each image is normalized between 0 and 1 and is resized to 128 × 128. When the full 3D volume is available, we take two slices for each task: the slice in the middle of the volume and the slice with the largest count of pixels labelled for that task.

Task Definition. We consider a task as labelling a certain structure from a certain modality for a certain dataset. If a given medical image has different structures labeled, we consider each structure a different task. For 3D volumes, we consider each axis as a different task.

Synthetic Data. We use a set of synthetic tasks to enhance the performance of our networks, similar to [20]. Some examples are shown in Figure 9.

Synthetic Multi-Annotator Data. We use synthetic data to encourage diversity in our predictions. Figure 8 shows examples of blob targets with the corresponding simulated annotations.

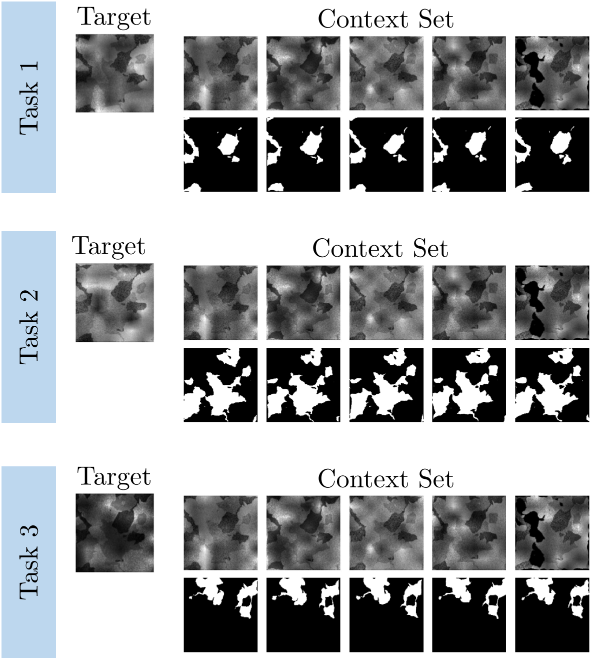

Figure 9. **Synthetic Data Examples.** Example of training images on the single annotator synthetic data. Each row is an example of a target-context pair corresponding to a different task.

Augmentations	p	Parameters
Random Affine	0.25	degrees $\in [-25, 25]$ translate$\in [0, 0.1]$ scale$\in [0.9, 1.1]$
Brightness Contrast	0.5	brightness$\in [-0.1, 0.1]$, contrast$\in [0.5, 1.5]$
Elastic Transform	0.8	$\alpha \in [1, 2.5]$ $\sigma \in [7, 9]$
Sharpness	0.25	sharpness$= 5$
Flip Intensities	0.5	None
Gaussian Blur	0.25	$\sigma \in [0.1, 1.0]$ k=5
Gaussian Noise	0.25	$\mu \in [0, 0.05]$ $\sigma \in [0, 0.05]$

(a) In-Task Augmentation

Augmentations	p	Parameters
Random Affine	0.5	degrees $\in [0, 360]$ translate$\in [0, 0.2]$ scale$\in [0.8, 1.1]$
Brightness Contrast	0.5	brightness$\in [-0.1, 0.1]$, contrast$\in [0.8, 1.2]$
Gaussian Blur	0.5	$\sigma \in [0.1, 1.1]$ $k = 5$
Gaussian Noise	0.5	$\mu \in [0, 0.05]$ $\sigma \in [0, 0.05]$
Elastic Transform	0.5	$\alpha \in [1, 2]$ $\sigma \in [6, 8]$
Sharpness	0.5	sharpness$= 5$
Horizontal Flip	0.5	None
Vertical Flip	0.5	None
Sobel Edges Label	0.5	None

(b) Task Augmentation

Table 5. **Set of augmentations used to train Tyche-TS.** We distinguish between augmentations aimed at increasing the diversity inside a task (Top) and the augmentations aimed at increasing the diversity of tasks (Bottom). An augmentation is applied with probability p (second column).

B.2. Training of Tyche-TS

We train *Tyche-TS* with a context size of 16 and a batch size of 4. We use a learning rate of 0.0001, Adam as an optimizer, and a kernel size of 3. We apply the augmentations shown Table 5. To obtain K segmentation candidates, we generate K noise samples $z_k \sim \mathcal{N}(0, \mathbb{I})$. We duplicate the input K times and concatenate each noise sample with a duplicated target to form K stochastic inputs to the network.

B.3. Network for Tyche-IS

For the *Tyche-IS* network, we use the baseline UniverSeg [20], trained with the same data as *Tyche-TS*, same batch size, and same context size. At test time, we use the augmentations in Table 6. Figure 10 shows example augmentation on different samples.

B.4. Benchmarks

We distinguish three types of benchmarks: in-context methods, that can take as input a context set, the interactive frameworks, and the task-trained specialized upper bounds.

In-context baselines. Because they were trained on medical data, we use SegGPT and SAM-Med2D as provided in their official release. We train from scratch UniverSeg and SENet. We use a batch size of 4 and the same set of data augmentation transforms as the ones used for *Tyche-TS* to encourage within task and across task diversity. Training UniverSeg on the same datasets as *Tyche* improves performances compared to the official UniverSeg release, as shown in Figure 11.

Specialized benchmarks. For the specialized models (CIDM, PhiSeg, and Probabilistic UNet), we train a model for each task, for a total of 20 tasks. For each task, we train three model variants, one for each different data augmentation scheme, shown Table 7. We select the model with the data augmentation strategy that does best on the *out-of-distribution* validation set. We use a batch size of 4. For CIDM and Probabilistic UNet, we use the official PyTorch release. We found Probabilistic UNet particularly unstable to train on some datasets when applying augmentations. To

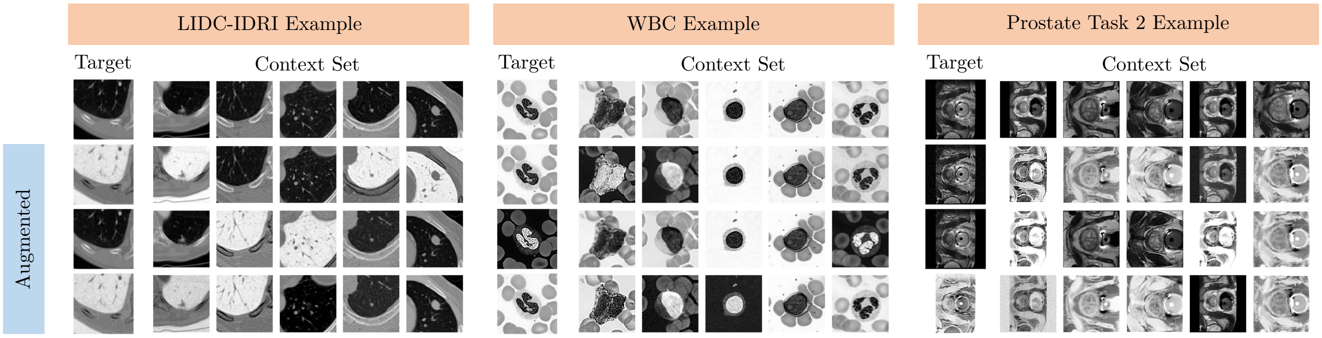

Figure 10. **Example Augmentations for Tyche-IS.** We show for samples from different datasets how three new segmentation candidates are generated. Starting from the top row, augmentations are applied to both the target and context set to obtain new prediction. Each row represents a new target-context pair. We do not show the corresponding labels as for *Tyche-IS*, we only augment with intensity-based transforms.

Tyche-IS **Augment.**	p	Parameters
Gaussian Blur	0.25	$\sigma \in [0.1, 1.0]$ $k = 5$
Gaussian Noise	0.25	$\mu \in [0, 0.05]$ $\sigma \in [0, 0.05]$
Flip Intensities	0.5	None
Sharpness	0.25	sharpness=5
Brightness Contrast	0.25	brightness$\in [-0.1, 0.1]$, contrast$\in [0.5, 1.5]$

Table 6. **Augmentations used for *Tyche-IS*.** We focus on intensity transforms, to avoid inverting the prediction. For each image, an augmentation is sampled with probability p.

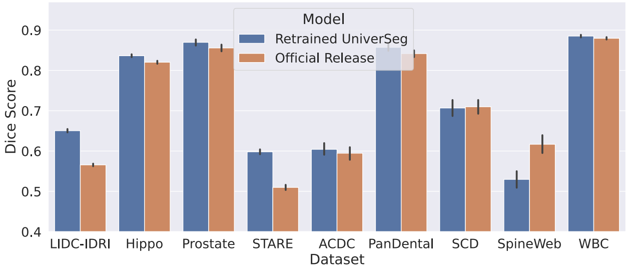

Figure 11. **Improvement of UniverSeg by training with more data.** By training UniverSeg on more data, we obtain a better average Dice score than the official UniverSeg release.

avoid diverging loss values, we had to initialize the Probabilistic UNet models from versions trained on datasets without augmentation. For PhiSeg, the official TensorFlow release was unstable, and we worked with the authors to run a more recent PyTorch version of their code. Per the authors' request, we smooth the predicted segmentations of CIDM to remove irregularities in the final segmentations.

Interactive Methods. We use the SAM-Med2D and SAM as interactive segmentation baselines. In our setting, user interaction is not available. We simulate it by averaging the ground truth labels from the context set and sample clicks and bounding boxes from averages.

We use SAM-Med2D's official release as it is an adaptation of SAM exclusively meant for 2D medical image segmentation tasks. We found empirically that SAM-Med2D performs best when 3 positive and 2 negative clicks are sampled. For the negative clicks, we sample areas inside the bounding box that are not covered by the context set.

We also fine-tune SAM on our data. We sample 5 positive clicks and 5 negative clicks uniformly. We also provide a bounding box and the average of the context label maps as input.

C. Advantages of candidates loss

C.1. Intuition behind best candidate loss

The best candidate loss (Eq. 8) only evaluates the candidate that yields the lowest loss. We provide intuition for this loss function (Figure 12).

Given as input a target x^t and a context $\mathcal{S}^t$, the model outputs K segmentation candidates $\hat{y}_j$. Despite potentially high target ambiguity, for each training iteration we use only one randomly sampled annotation y^t_r.

The loss element $\mathcal{L}_{Dice}(\hat{y}_k, y^t_r)$ captures volume overlap between a candidate prediction $\hat{y}_k$ and the rater annotation y^t_r. If a regular loss is used across all predictions, then *each* prediction tends towards the lowest expect cost over the space of possible segmentations for that image, leading to a mean segmentation among raters. This is particularly harmful when target ambiguity is high (high rater disagreement). Then, the mean segmentation may be very different than any one rater and may not be representative of the ambiguity. For the best candidate loss, the model is encouraged to make very different guesses for different segmentation candidates. This increases the chance that one prediction

Augmentation Set 1	p	Parameters
None	1	None

(a) No Augmentation. The images are given to the specialized model as is.

Light Augmentation	p	Parameters
Gaussian Blur	0.5	$\sigma \in [0.1, 1.5]$ $k = 7$
Gaussian Noise	0.5	$\mu \in [0, 0.1]$ $\sigma \in [0, 0.1]$
Variable Elastic Transform	0.25	$\alpha \in [1, 2]$ $\sigma \in [6, 8]$

(b) Light Augmentation. Gaussian bur, Gaussian noise, elastic transforms are each applied to the target with probability p.

Heavy Augmentation	p	Parameters
Gaussian Blur	0.5	$\sigma \in [0.1, 1.5]$ $k = 7$
Gaussian Noise	0.25	$\mu \in [0, 0.1]$ $\sigma \in [0, 0.1]$
Elastic Transform	0.25	$\alpha \in [1, 2]$ $\sigma \in [6, 8]$
Random Affine	0.5	degrees $\in [0, 360]$ translate$\in [0, 0.2]$ scale$\in [0.8, 1.1]$
Brightness Contrast	0.5	brightness$\in [-0.1, 0.1]$, contrast$\in [0.5, 1.5]$
Horizontal Flip	0.5	None
Vertical Flip	0.5	None
Sharpness	0.5	sharpness$= 5$

(c) Full Set of Augmentations. To form the last augmentation set, we add the following transforms: Affine transform, Sharpness, Brightness and Flips.

Table 7. **Set of Augmentations used to train the specialized benchmarks.** For evaluation, we select the model with the training augmentations that does best on the *out-of-distribution* validation set.

matches the rater being used as ground truth at that iteration.

C.2. Distribution loss function

Since *Tyche-TS* enables multiple candidates that can coordinate with one another, we can employ loss functions that apply to a collection of raters and predictions. We demonstrate this concept by fine-tuning *Tyche-TS*, using a GED^2 loss function.

Figure 13 shows that the corresponding model produces samples with lower GED^2, as expected.

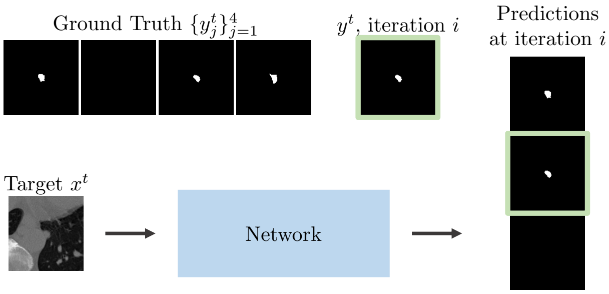

Figure 12. **Best candidate Loss Explanation.** Because of ambiguity in the target, there are multiple plausible ground truths. With a standard Dice loss, all the candidates segmentation converge to the mean segmentation. The best candidate loss enables the model to explore different plausible segmentation, since even one proposal matching the rater segmentation used at that iteration leads to a good loss value.

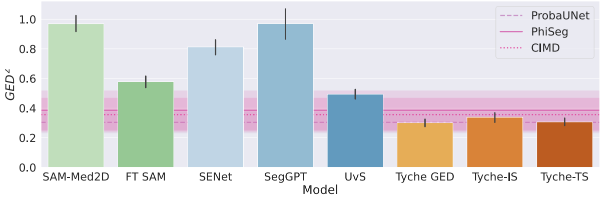

Figure 13. ***Tyche-TS* with GED loss** (yellow) improved test-time GED performance (lower is better).

D. Tyche Analysis

We first analyse the different components that lead to variability in *Tyche-TS*: the noise, the context set, the number of prediction in on forward pass, and the *SetBlock* mechanism. We then investigate how the size of the context set and the number of predictions impact performances. For *Tyche-IS*, we investigate different plausible sets of augmentations to apply. Finally, we show that *Tyche* can produce good results even in data scarcity settings when a stochastic methods trained solely on a few samples would have limited performance.

D.1. Diversity from noise

Figure 14 shows how three predictions vary depending on three noise variants: zero-noise, constant noise across candidates, and variable noise. Both zero noise and constant noise lead systematically to identical segmentation candidates. For each example of segmentation candidates, the context set is fixed.

Figure 16 shows that setting the noise to 0 for *Tyche-TS* produces similar performances to UniverSeg, both for GED and best candidate Dice score.

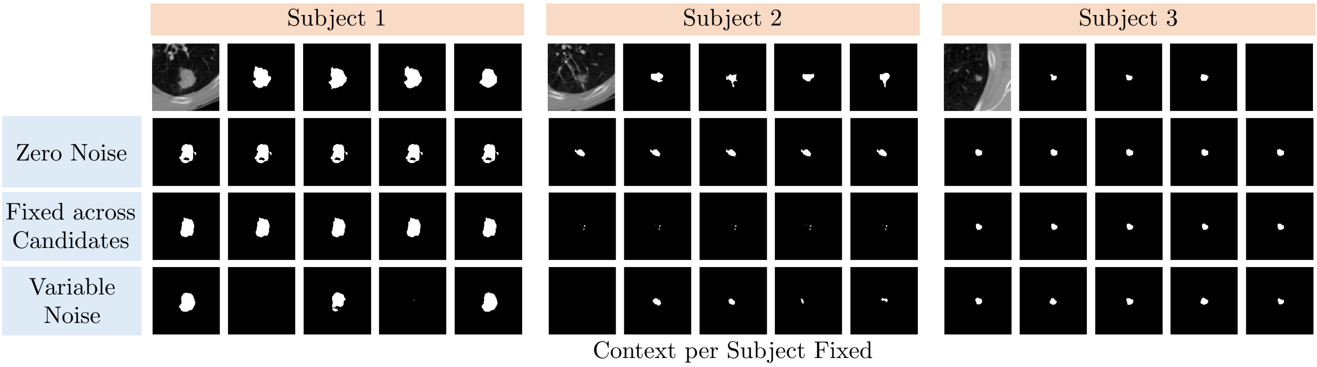

Figure 14. **Prediction variability as a function of injected noise.** Examples of predictions for three subjects with three different types of input noise. Top to Bottom: zero noise, Gaussian noise constant across segmentation candidates, and randomly sampled Gaussian noise. The context set is fixed. With random noise as an input *Tyche* can output diverse segmentation candidates.

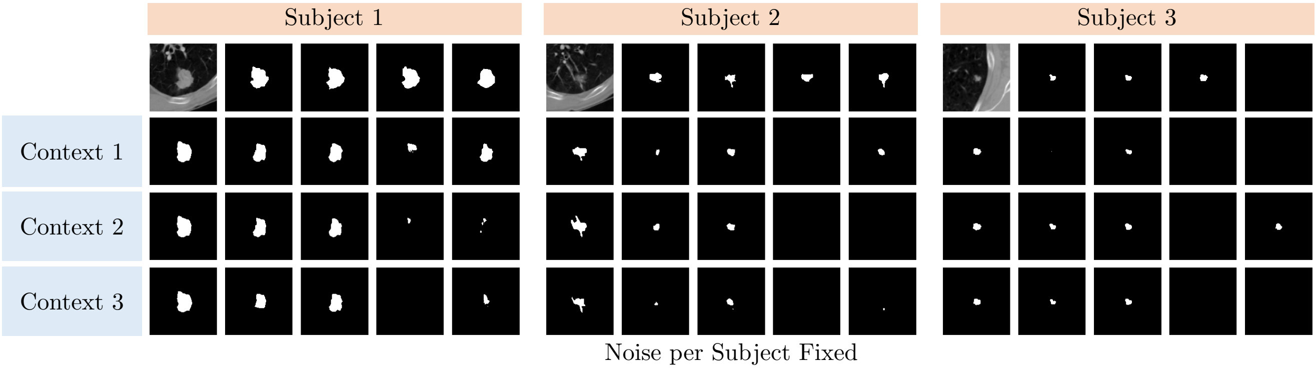

Figure 15. **Prediction variability as a function of the context set.** The context set significantly contributes to the variability of the output. For three different subjects, we show how the corresponding segmentation candidates are affected by different context sets. The random Gaussian noise given as input is fixed for each set of candidates.

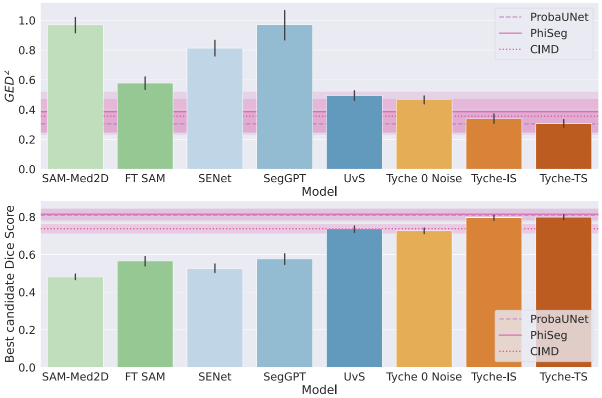

Figure 16. ***Tyche-TS* with no noise gives similar results to UniverSeg.** Setting the noise to 0 for *Tyche-TS* gives similar best candidate Dice score and similar GED^2 to UniverSeg. Top: Generalized Energy Distance. Bottom: Best candidate Dice score.

D.2. Diversity from context

Figure 15 shows example predictions for three subjects and three context sets. Different contexts lead to different predictions, even if the noise is the same, sometimes drastically. For instance, the predictions for subject 1 and context 3 contain a candidate with no annotations.

D.3. Diversity in the number of predictions

While *Tyche-IS* predicts each segmentation sample sequentially, *Tyche-TS* has a one-shot mechanism that predicts all the candidates at once. One may wonder how the number of predictions requested at inference impacts the output of *Tyche-TS*. Figure 17 shows that with the number of prediction set to 1, *Tyche-TS* loses a lot of its diversity compared to 5 predictions.

This is also shown quantitatively in Figures 19 and 20, where both best candidate Dice and Generalized Energy Distance improve as the number of predictions increases.

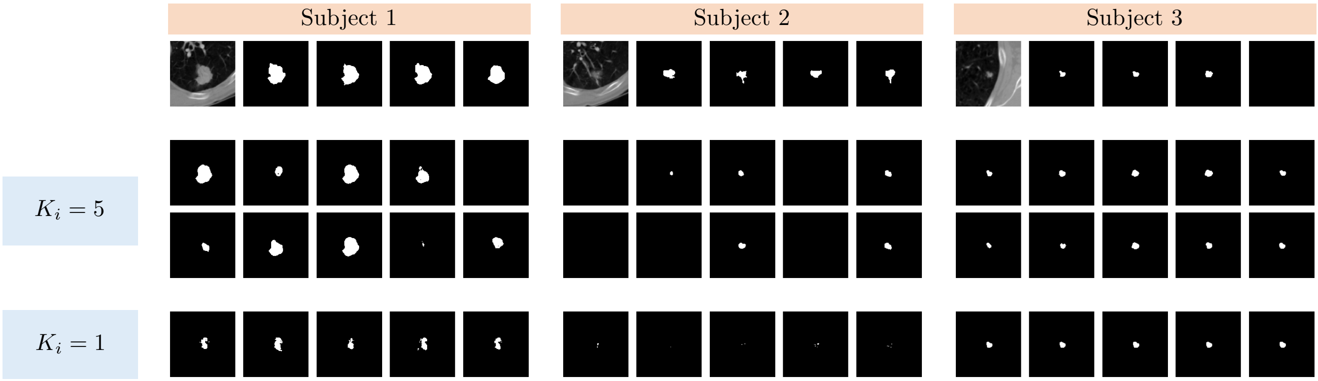

Figure 17. **Setting a number of predictions larger than 1 allows for more diversity.** The outputs of *Tyche-TS* are a lot more uniform when the number of prediction is set to 1 than when it is set to 5.

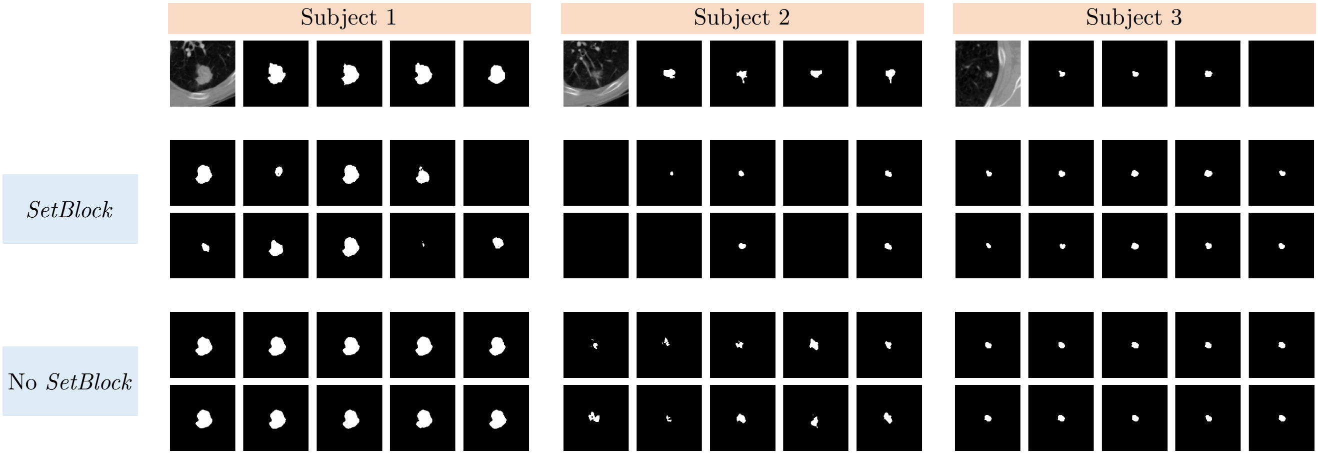

Figure 18. **Removing the SetBlock leads to less diversity in the output.** The segmentation candidates output by *Tyche-TS* are more diverse than the candidates of a *Tyche-TS* model trained without the *SetBlock* interaction.

D.4. Diversity from the SetBlock

Tyche-TS has an intrinsic mechanism to encourage the segmentation candidates to be diverse: *SetBlock*. Figure 18 visually compares the segmentations predicted with the mechanisms and the segmentations predicted by a *Tyche* model trained without the *SetBlock*. Without the *SetBlock*, the predictions output by *Tyche* are a lot less diverse.

D.5. Size of the Context Set

Generally, a larger context set improves performances. Figures 21 and 22 show that both best candidate Dice and Generalized Energy Distance improve as the size of the context set increases. However, the improvement decreases beyond 16 samples.

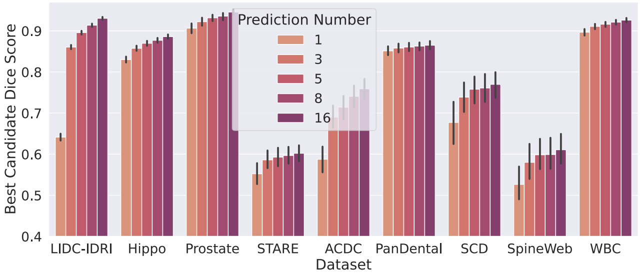

Figure 19. **Best candidate Dice score as the number of candidate prediction increases.** The largest improvements are usually obtained for a small number of predictions. The error bars represent the 95% confidence interval.

D.6. Augmentations

We study different augmentation for *Tyche-IS* and how the influence the quality of predictions. We consider four settings: no augmentation, only light augmentation such as Gaussian Noise and Gaussian Blur, described Table 6, the *Tyche-IS* augmentations shown Table 6 and the *Tyche-IS* with slightly stronger parameters. shown Table 9. The re-

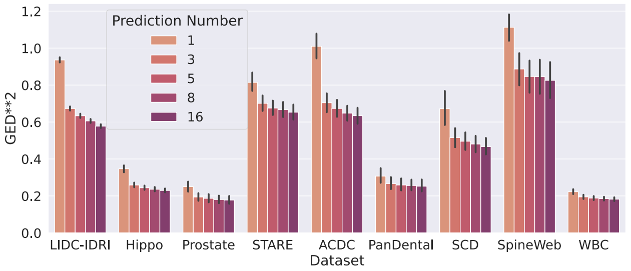

Figure 20. GED^2 **for Tyche as a function of the number of predictions,** K_i**.** Performances improve with the number of prediction but with diminishing returns.

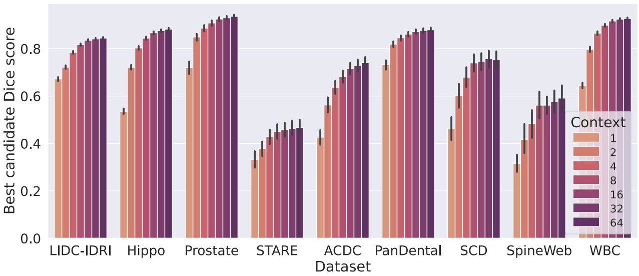

Figure 21. **Best candidate Dice score per dataset as context size increases.** A context size of 16 is already large enough to obtain a reasonable best candidate Dice. The error bars represent the 95% confidence interval.

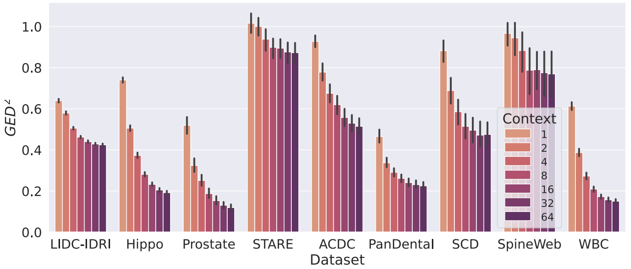

Figure 22. GED^2 **for Tyche as a function of the inference context size.** Performances improve as the context size increases but with diminishing returns.

sults are shown in Figure 23 both aggregated across datasets and for each dataset individually. Overall, adding augmentations improves the best candidate Dice score. However, it can degrades the quality of the predictions, for instance for STARE. The augmentation we selected for *Tyche-IS* is the most promising so far, without requiring inversion of the transform applied.

D.7. Few Shot Regime

We train PhiSeg on a subset of LIDC-IDRI, $\tilde{S}$, and examine how this network generalizes compared to Tyche, where the context set is $\tilde{S}$. We investigate four few-shot settings: 3, 5, 8 and 16. For each, we train 30 PhiSegs: 10 seeds to account for variability in our samples and three data augmentation regimes (none, light, normal). For each seed and

Blob	SetBlock	Std	Max. DSC(↑)	GED^2(↓)
✗			0.810 ± 0.01	0.349 ± 0.04
	✗		0.771 ± 0.02	0.425 ± 0.05
		✓	0.802 ± 0.01	0.425 ± 0.05
✓	✓		$\mathbf{0.811 \pm 0.01}$	$\mathbf{0.298 \pm 0.03}$

Target	CS	CS+	Max. DSC(↑)	GED^2 (↓)
✓			0.776 ± 0.02	0.477 ± 0.04
	✓		0.700 ± 0.02	0.410 ± 0.05
		✓	0.561 ± 0.02	0.867 ± 0.05
✓	✓		$\mathbf{0.813 \pm 0.01}$	$\mathbf{0.333 \pm 0.04}$
✓	✓	✓	0.808 ± 0.01	0.358 ± 0.04

Table 8. **Ablation Study for Tyche variants.** Top: *Tyche-TS*, without simulated multi-annotator data, with *SetBlock*, with Standard Deviation in *SetBlock*. Bottom: *Tyche TeS*, with Target, Context and Large Context augmentations.

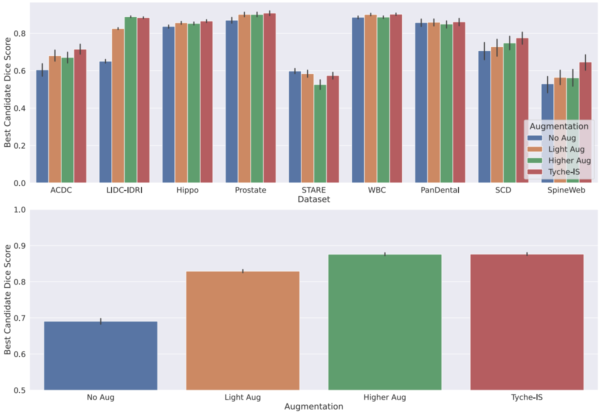

Figure 23. **Performance of different augmentation schemes for *Tyche-IS*.** We consider four augmentations: none, light, the one used in *Tyche-IS* and a stronger version of the later. Top: Per Dataset. Bottom: Overall. The one that we selected for *Tyche-IS* is the most promising.

***Tyche-IS* High Aug.**	p	Parameters
Gaussian Blur	0.25	$\sigma \in [0.5, 1.0]$ $k = 5$
Gaussian Noise	0.5	$\mu \in [0.4, 0.5]$ $\sigma \in [0.1, 0.2]$
Flip Intensities	0.5	None
Sharpness	0.25	sharpness=5
Brightness Contrast	0.25	brightness$\in [-0.1, 0.1]$, contrast$\in [0.5, 1.5]$

Table 9. **High augmentations used to validate *Tyche-IS*.** We focus on intensity transforms, to avoid inverting the prediction. For each image, an augmentation is sampled with probability p.

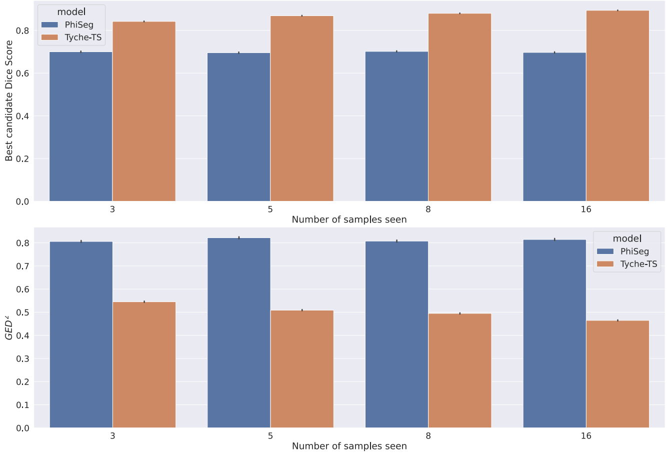

Figure 24. **Few-Shot Regime.** Comparison between *Tyche-TS* and PhiSeg trained on the few-shot examples. PhiSeg fails to learn from very few samples both on max. Dice score and GED, compared to *Tyche*.

each few-shot setting, we select the PhiSeg that does best on the validation set. Figure 24 shows that *Tyche* can leverage the data available much more effectively than PhiSeg, which fails to learn with so little samples.

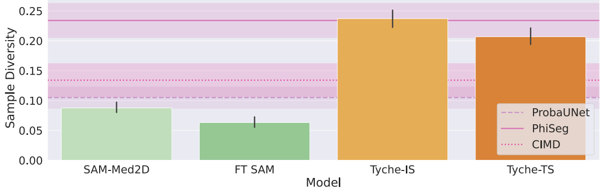

Figure 25. **Sample Diversity on Multi-Annotator Data.** Since the sample diversity of the deterministic methods is 0, we do not show them here. *Tyche-IS* produces the most diverse samples, while Fine-Tuned SAM has very low diversity, despite varying clicks and bounding box locations. Higher is better.

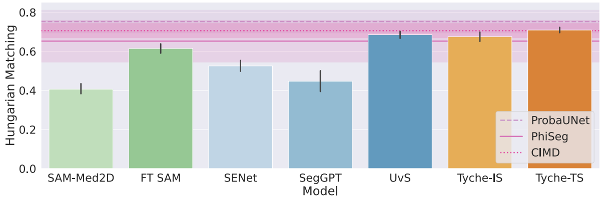

Figure 26. **Hungarian Matching on Multi-Annotator Data.** Both *Tyche-IS* and UniverSeg perform well. *Tyche-TS* performs best. Higher is better.

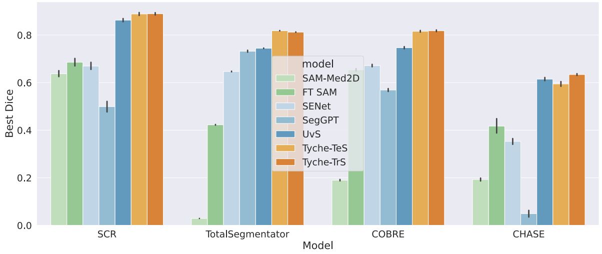

Figure 27. **Best Candidate Dice score on additional datasets.** Tyche variants outperform the interactive baselines and the in-context methods except for CHASE.

E. Further Evaluation

E.1. Performance on other datasets

We tested performance on three additional held-out datasets: SCR, TotalSegmentator and COBRE. We found similar performance trends. Because of the large number of structures in TotalSegmentator and in COBRE, we omitted the upper baselines. Figure 27 shows best candidate Dice for each method on the three datasets aggregated per task. We find that *Tyche* variants outperform the interactive and in-context baselinens. Tyche-TS and Tyche-IS are comparable except for CHASE where Tyche-TS performs better.

E.2. Performances on other Metrics

Evaluating the quality of different predictions can be challenging especially when different annotators are available. We proposed best candidate Dice score and Generalized Energy Distance. Some also analysed sample diversity [97], and Hungarian Matching [70, 136]. Sample diversity consists in measuring the agreement between candidate predictions $\hat{\mathcal{Y}}$, rewarding most diverse sets of candidates:

$$D_{SD}(\hat{\mathcal{Y}}) = \mathbb{E}\left[d(\hat{p}, \hat{p}')\right], \tag{17}$$

where $\hat{p}, \hat{p}' \sim \hat{\mathcal{Y}}$ and $d(\cdot,\cdot)$ is Dice score. One limitation of this metric is that it blindly rewards diversity without taking into account the natural ambiguity in the target. Ideally, when there is high ambiguity in the target, the samples are very diverse, and inversely, when there is low ambiguity, the segmentation candidates are not diverse.

In the context of stochastic predictions, Hungarian Matching [72] consists in matching the set of predictions with the set of annotations, so that an overall metric is minimized. We use negative Dice score. One limitation of this method is that it has to be adapted when the number of annotators does not match the number of prediction. The most widely used fix is to artificially inflate the number of predictions and the number of annotations to reach the least common multiple [70]. We use this strategy here as well.

Figures 25 and 26 show the performances for sample diversity and Hungarian Matching respectively.

E.3. Per Dataset Results

We show for each dataset and each method the best candidate Dice score. We also show for *Tyche* and the different benchmarks Generalized Energy Distance, Hungarian Matching and Sample Diversity for the datasets with multiple annotations.

Best candidate Dice score. Figure 28 shows the best candidate Dice score for the single-annotator datasets. *Tyche* performs well across datasets. Both versions of *Tyche* seem to dominate both types of benchmarks and be comparable to the upper bounds, except for one dataset: SpineWeb. We hypothesize that part of the performance drop is due to the nature of the structure to segment in SpineWeb: individual vertebra. Most of our training data contains single structures to segment.

Figure 29 shows the best candidate Dice score for the multi-annotator datasets. Similar conclusions can be drawn as for the single-annotator data. Some benchmarks are particularly sensitive to the data they are evaluated on, for instance SegGPT. This methods performs really well on the Prostate data but quite poorly on the STARE and the Hippocampus data. We assume that because SegGPT was designed for images of 448x448, our images of dimension 128x128 might affect performances.

Generalized Energy Distance. Figure 30 shows Generalized Energy distance for the multi-rater datasets.

Hungarian Matching. Figure 31 shows the Hungarian Matching metric for the multi-rater datasets. We find that UniverSeg performs particularly well. We suspect that because we have to artificially duplicate our samples to compute this metric, the resulting scenario favors methods that are closer to the mean.

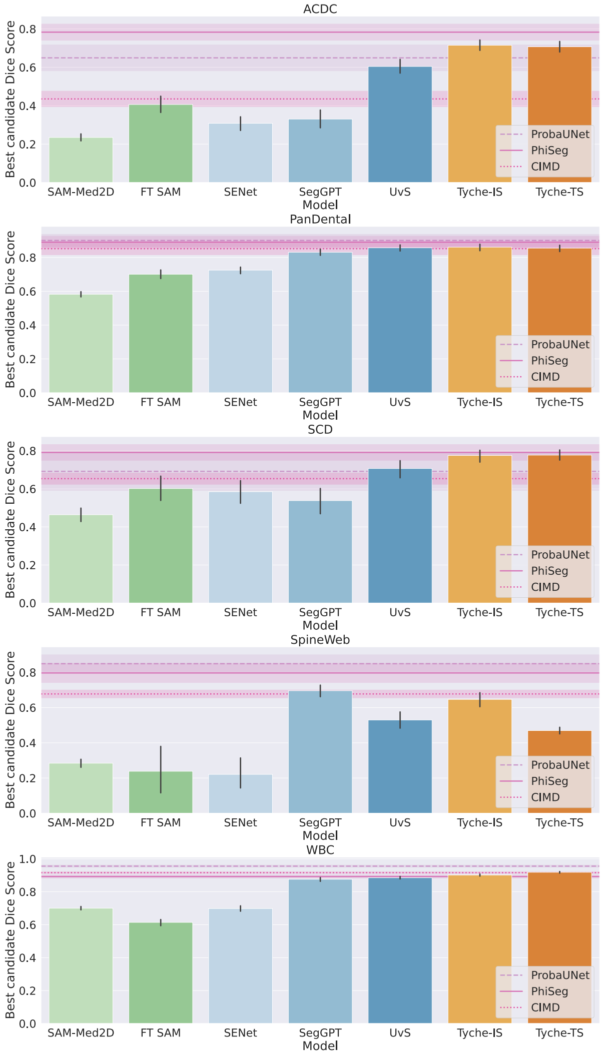

Figure 28. **Best candidate Dice score for the Single-Annotator Datasets.** Top to Bottom: ACDC, PanDental, SCD, SpineWeb and WBC. *Tyche* performs well in general except for SpineWeb.

Sample Diversity. Figure 32 shows sample diversity for the multi-annotator datasets. We only show the sample diversity for the methods outputing more than one segmentation candidate. For UniverSeg, SegGPT and SENet, the sample diversity is trivially 0.

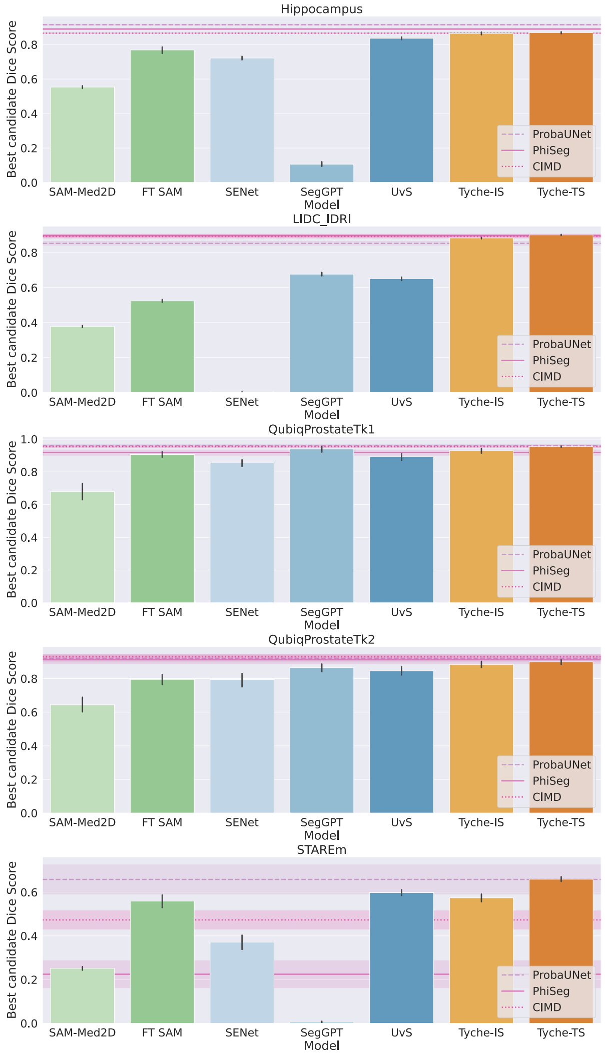

Figure 29. **Best candidate Dice score for Multi-Annotator Datasets** Top to bottom: Hippocampus, LIDC-IDRI, Prostate Task 1, Prostate Task 2 and STARE. *Tyche* performs well across datasets. (Higher is better.)

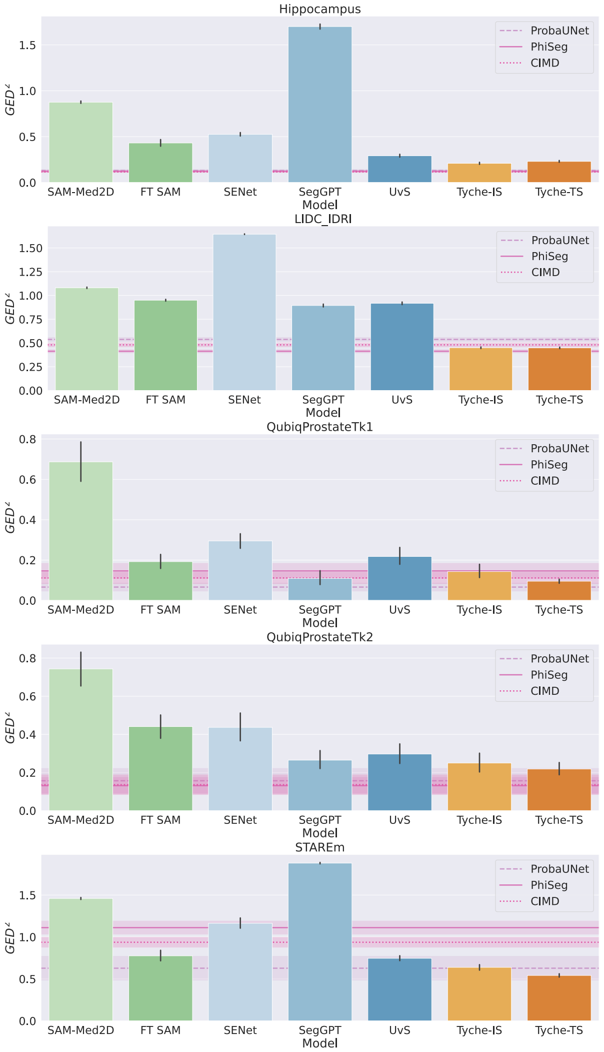

Figure 30. **Generalized Energy Distance for Multi-Annotator Datasets.** Top to bottom: Hippocampus, LIDC-IDRI, Prostate Task 1, Prostate Task 2 and STARE. *Tyche* performs well across datasets. (Lower is better.)

E.4. Additional Visualizations

We show additional visualization for *Tyche* frameworks as well as all the benchmarks.

Single-Annotator Data Visualizations for ACDC are shown Figure 33. We show two PanDental examples Figure 35 and Figure 36, one for each task. We show an example for SpineWeb Figure 34 and one for WBC Figure 38.

Multi-Annotator Data We show an example prediction for the Hippocampus data Figure 39 and one example prediction for STARE in Figure 40. We provide visualizations for each Prostate task Figures 41 and 42 respectively. Finally, we give two example visualizations for LIDC-IDRI Figures 43 and 44.

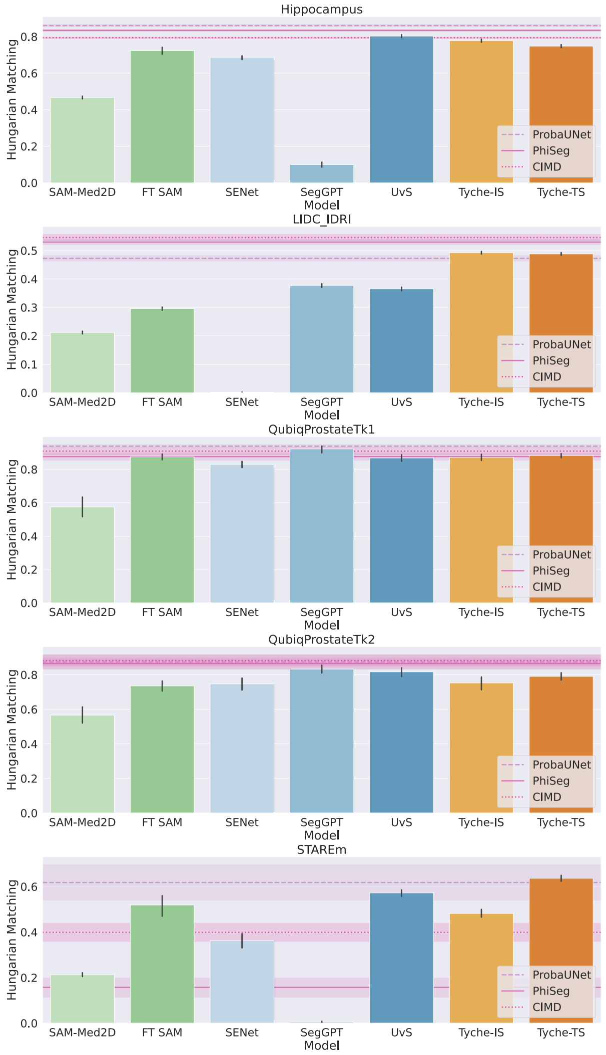

Figure 31. **Hungarian Matching Dice for Multi-Annotator Datasets.** Top to bottom: Hippocampus, LIDC-IDRI, Prostate Task 1, Prostate Task 2 and STARE. *Tyche* performs well across datasets. (Higher is better.)

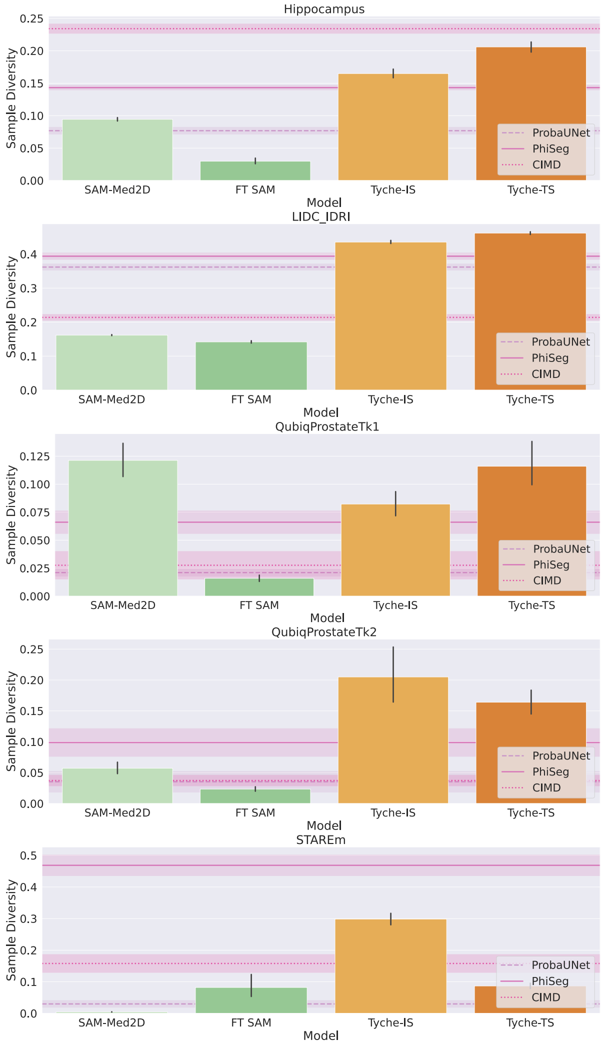

Figure 32. **Sample Diversity for Multi-Annotator Datasets.** Top to bottom: Hippocampus, LIDC-IDRI, Prostate Task 1, Prostate Task 2 and STARE. *Tyche* performs well across datasets. We only show methods with diversity greater than 0. (Higher is better.)

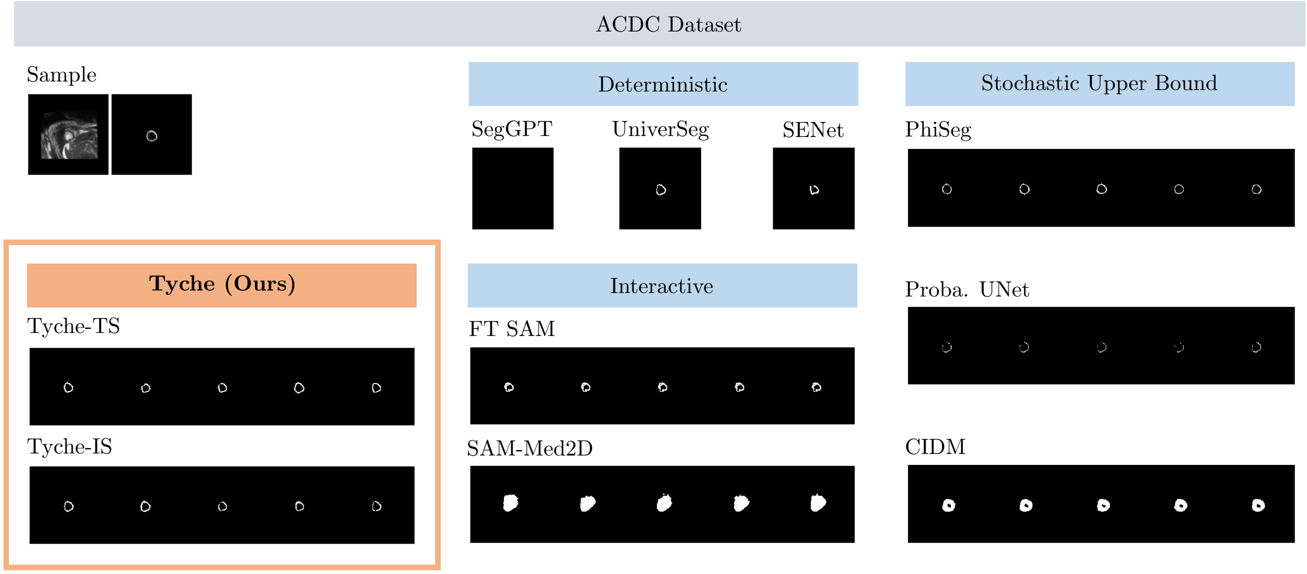

Figure 33. **Example Prediction for ACDC.**

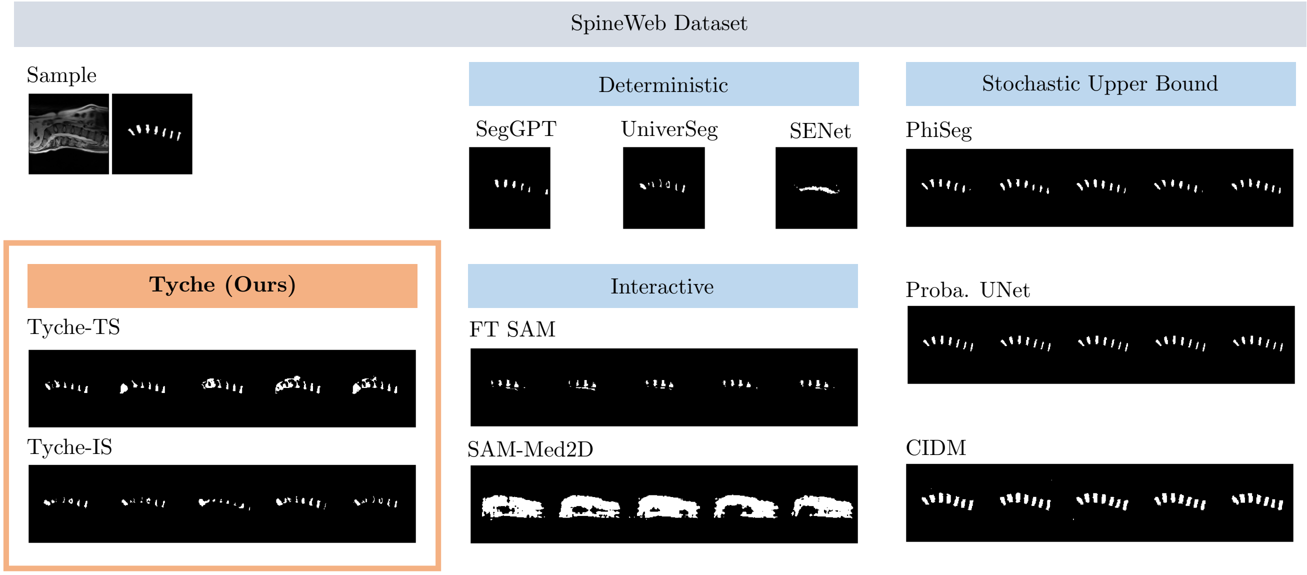

Figure 34. **Example Prediction for SpineWeb.**

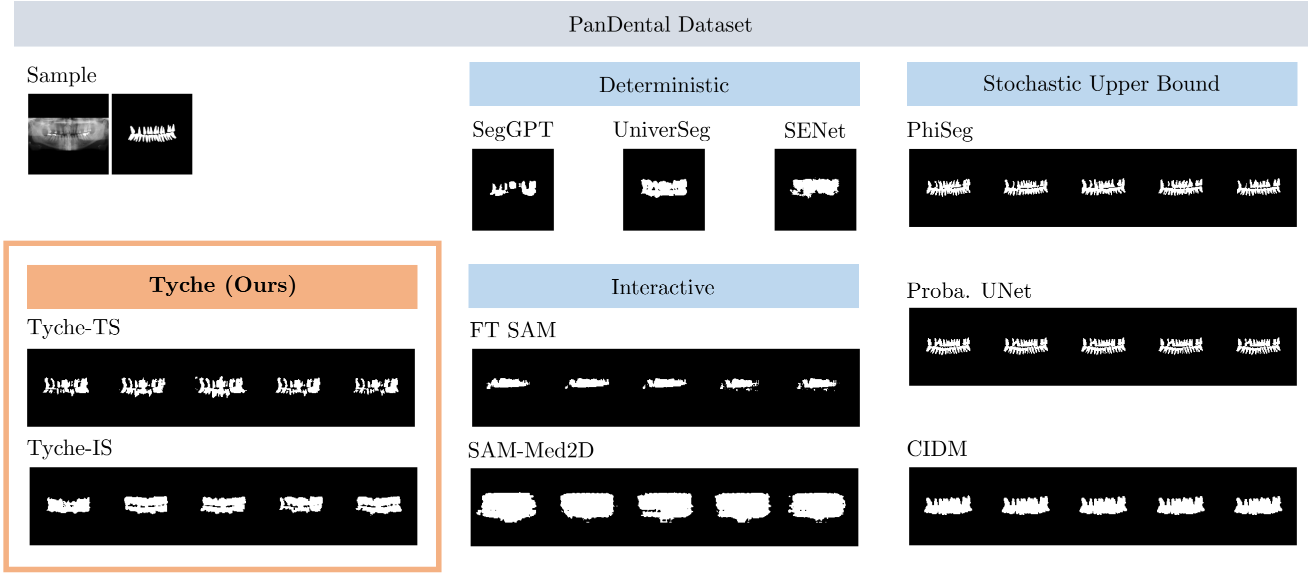

Figure 35. **Example Prediction for PanDental, task 1.**

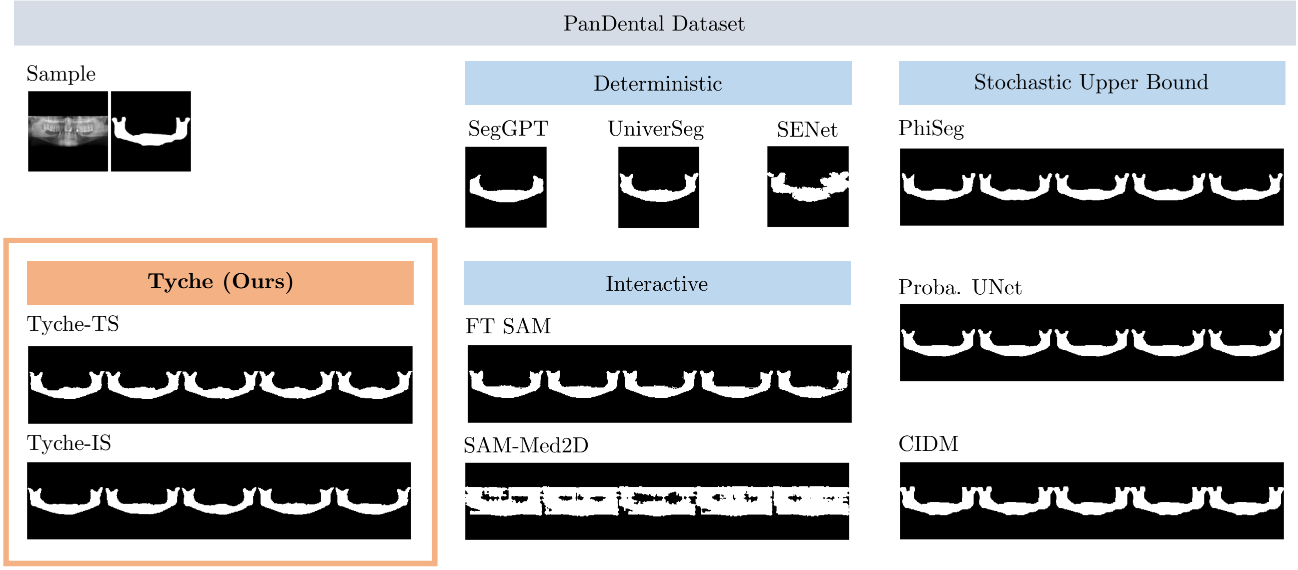

Figure 36. **Example Prediction for PanDental, task 2.**

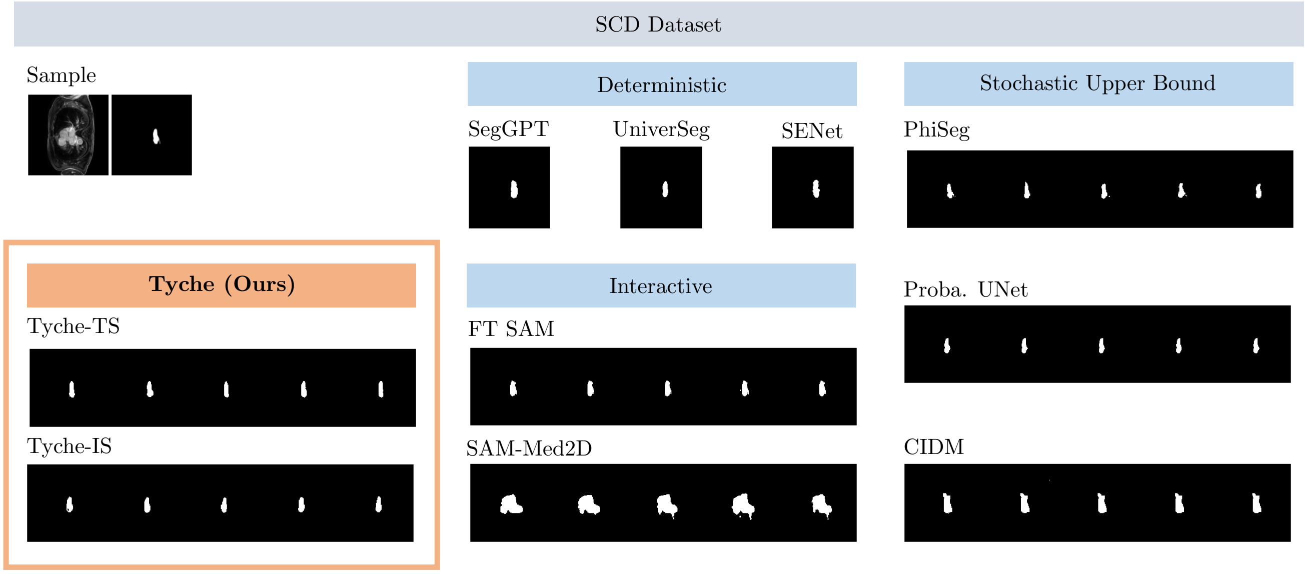

Figure 37. **Example Prediction for SCD.**

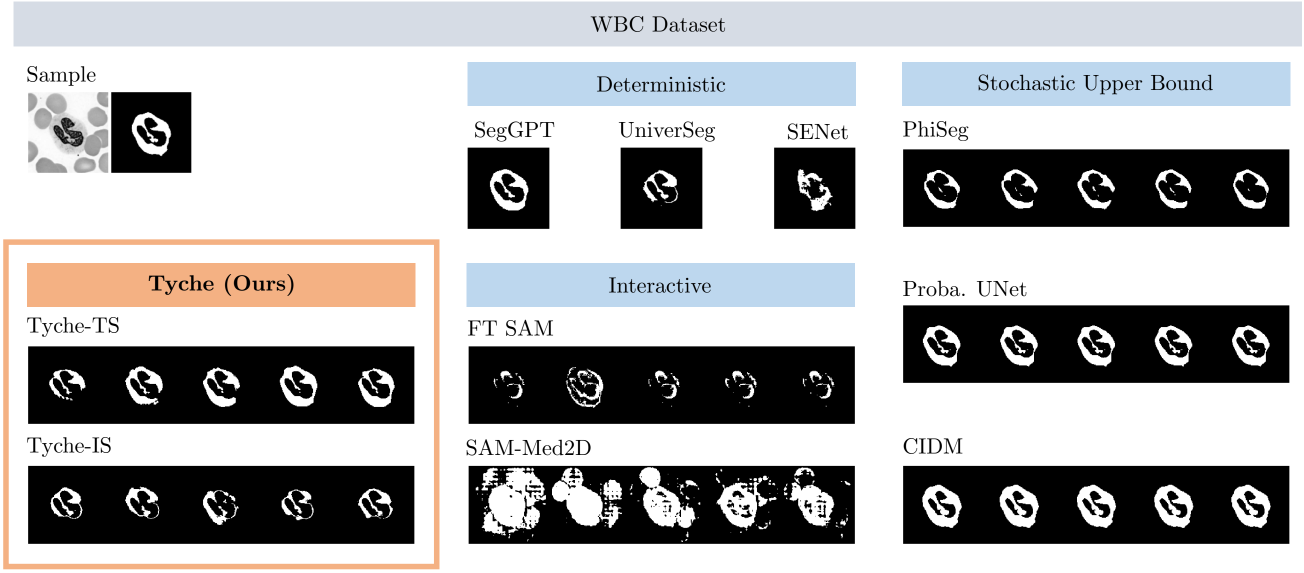

Figure 38. **Example Prediction for WBC.**

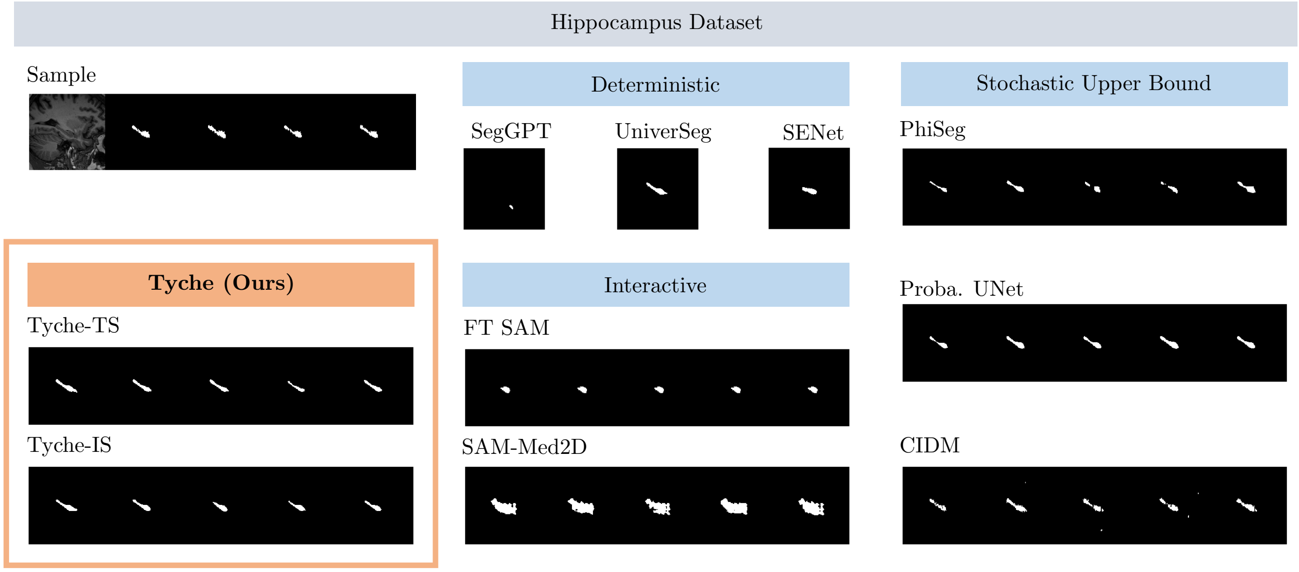

Figure 39. **Example Prediction for Hippocampus.**

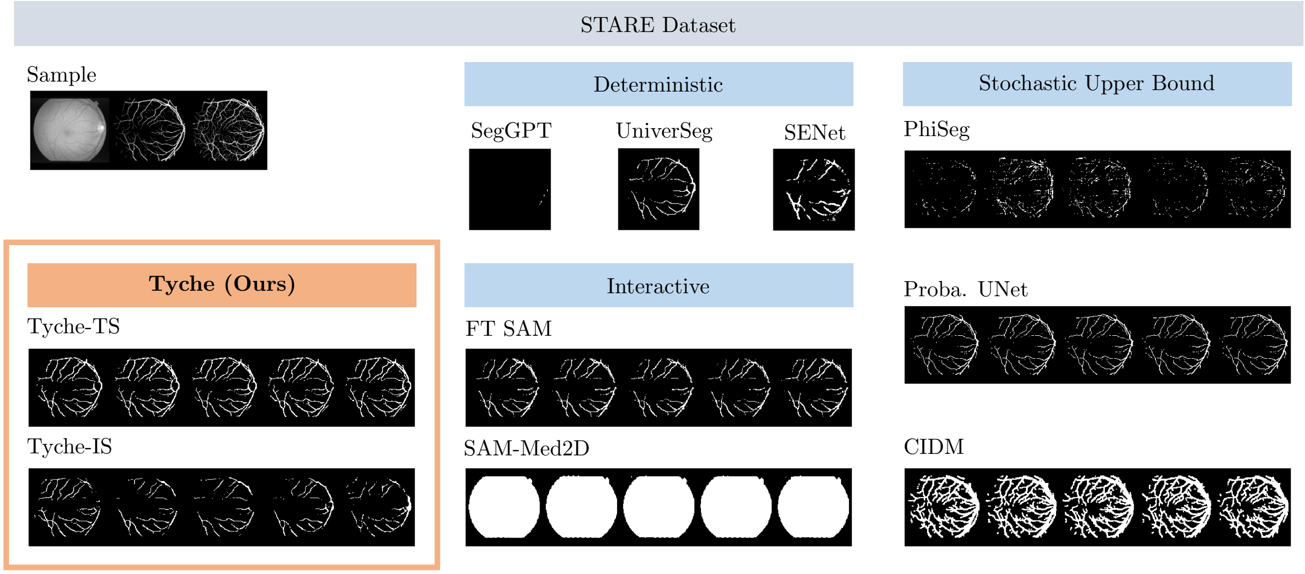

Figure 40. **Example Prediction for STARE.**

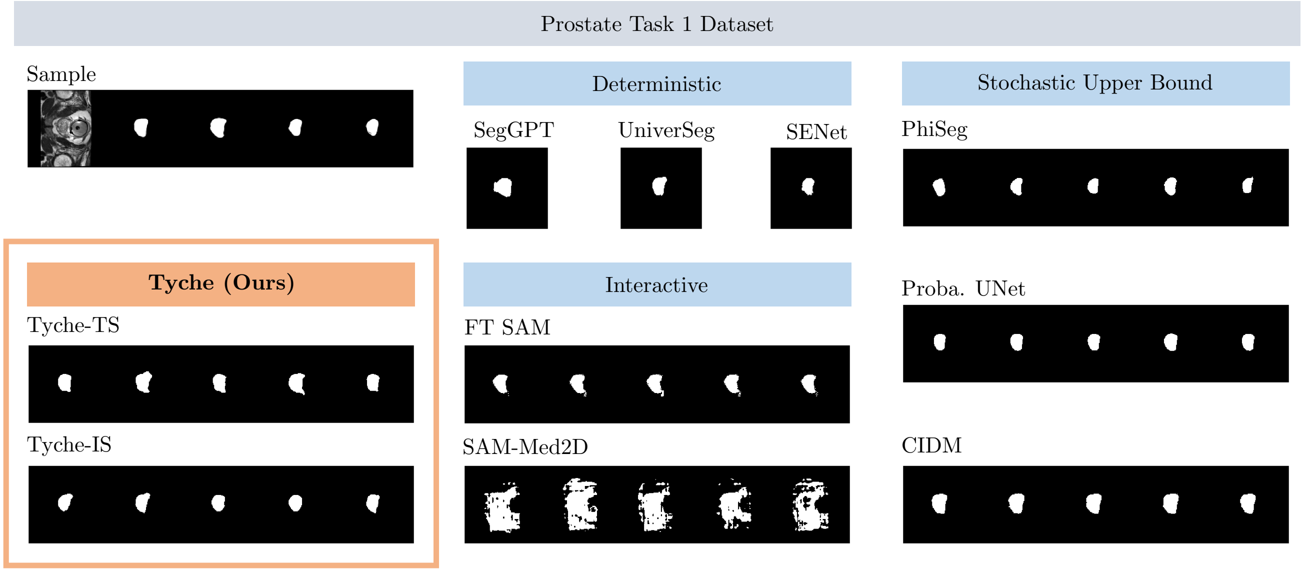

Figure 41. **Example Prediction for Prostate Task 1.**

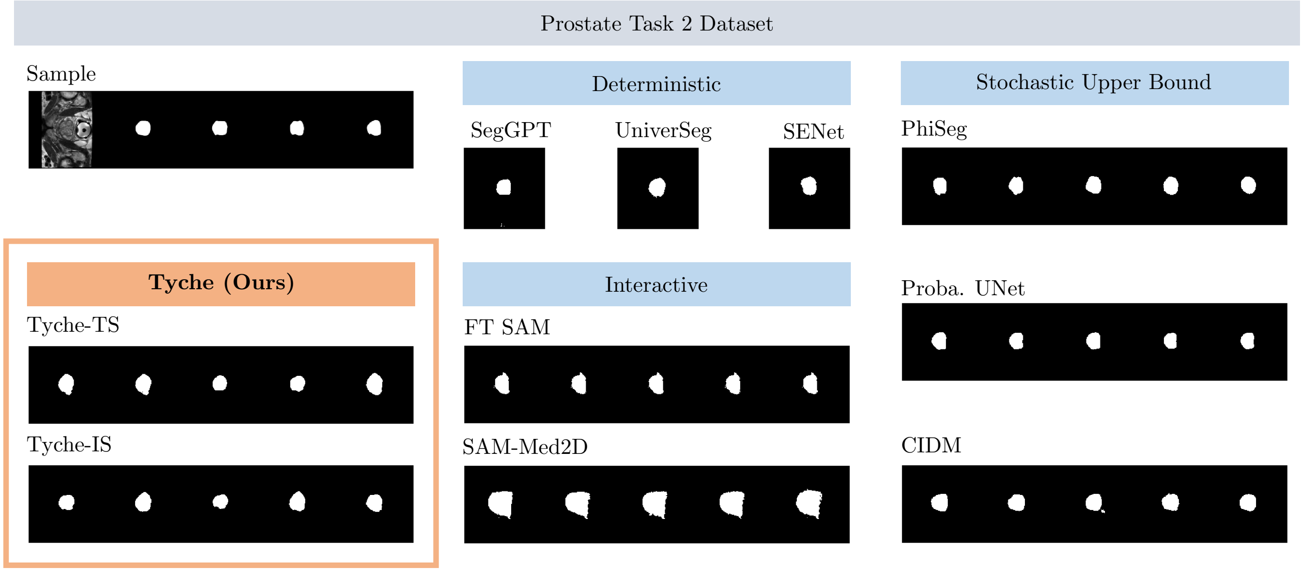

Figure 42. **Example Prediction for Prostate Task 2.**

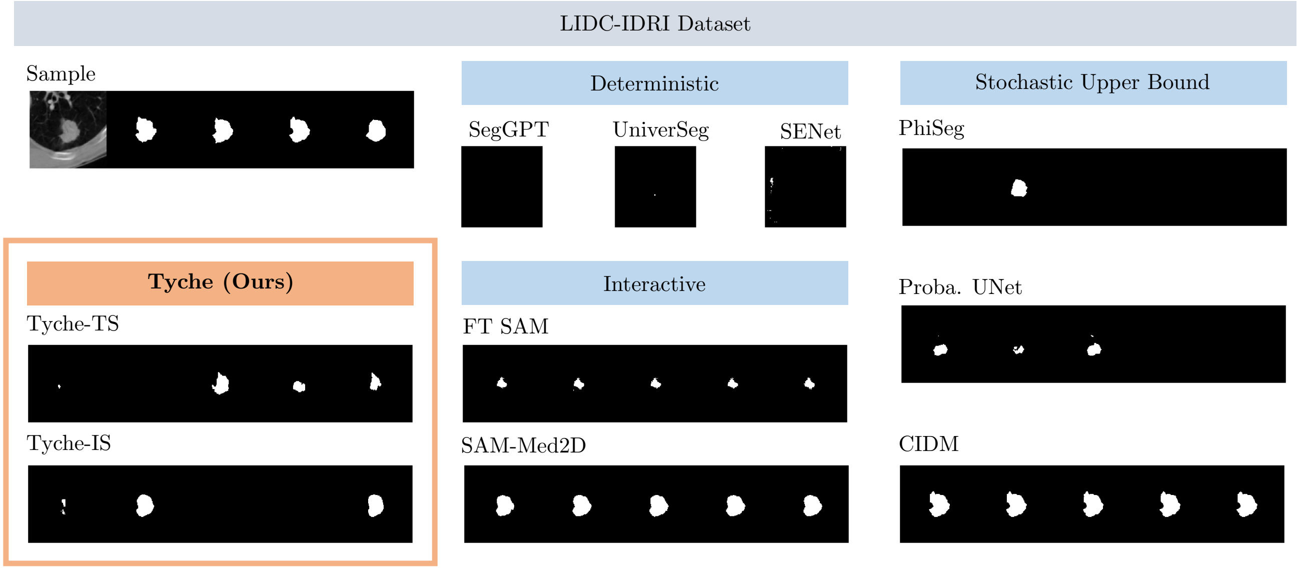

Figure 43. **Example Prediction for LIDC-IDRI.**

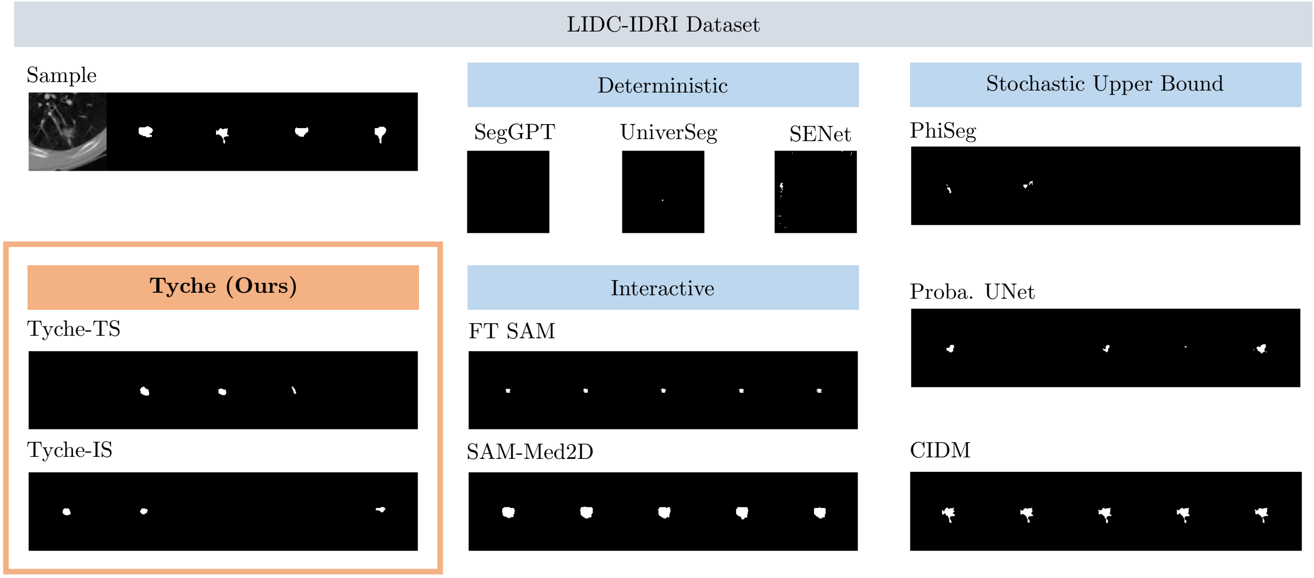

Figure 44. **Example Prediction for LIDC-IDRI.**

Dataset Name	Description	# of Scans	Image Modalities
ACDC [15]	Left and right ventricular endocardium	99	cine-MRI
AMOS [56]	Abdominal organ segmentation	240	CT, MRI
BBBC003 [85]	Mouse embryos	15	Microscopy
BBBC038 [21]	Nuclei images	670	Microscopy
BUID [3]	Breast tumors	647	Ultrasound
BrainDev. [39, 40, 74, 116]	Adult and Neonatal Brain Atlases	53	multi-modal MRI
BRATS [8, 9, 95]	Brain tumors	6,096	multi-modal MRI
BTCV [77]	Abdominal Organs	30	CT
BUS [139]	Breast tumor	163	Ultrasound
CAMUS [79]	Four-chamber and Apical two-chamber heart	500	Ultrasound
CDemris [58]	Human Left Atrial Wall	60	CMR
CHAOS [60, 62]	Abdominal organs (liver, kidneys, spleen)	40	CT, T2-weighted MRI
CheXplanation [112]	Chest X-Ray observations	170	X-Ray
CT-ORG[108]	Abdominal organ segmentation (overlap with LiTS)	140	CT
DRIVE [124]	Blood vessels in retinal images	20	Optical camera
EOphtha [28]	Eye Microaneurysms and Diabetic Retinopathy	102	Optical camera
FeTA [104]	Fetal brain structures	80	Fetal MRI
FetoPlac [11]	Placenta vessel	6	Fetoscopic optical camera
HMC-QU [29, 67]	4-chamber (A4C) and apical 2-chamber (A2C) left wall	292	Ultrasound
HipXRay [42]	Ilium and femur	140	X-Ray
I2CVB [80]	Prostate (peripheral zone, central gland)	19	T2-weighted MRI
IDRID [105]	Diabetic Retinopathy	54	Optical camera
ISLES [45]	Ischemic stroke lesion	180	multi-modal MRI
KiTS [44]	Kidney and kidney tumor	210	CT
LGGFlair [19, 93]	TCIA lower-grade glioma brain tumor	110	MRI
LiTS [17]	Liver Tumor	131	CT
LUNA [117]	Lungs	888	CT
MCIC [38]	Multi-site Brain regions of Schizophrenic patients	390	T1-weighted MRI
MSD [120]	Collection of 10 Medical Segmentation Datasets	3,225	CT, multi-modal MRI
NCI-ISBI [18]	Prostate	30	T2-weighted MRI
OASIS [48, 90]	Brain anatomy	414	T1-weighted MRI
OCTA500 [81]	Retinal vascular	500	OCT/OCTA
PanDental [1]	Mandible and Teeth	215	X-Ray
PAXRay [114]	Thoracic organs	880	X-Ray
PROMISE12 [84]	Prostate	37	T2-weighted MRI
PPMI [91]	Brain regions of Parkinson patients	1,130	T1-weighted MRI
ROSE [88]	Retinal vessel	117	OCT/OCTA
SCD [106]	Sunnybrook Cardiac Multi-Dataset Collection	100	cine-MRI
SegTHOR [76]	Thoracic organs (heart, trachea, esophagus)	40	CT
SpineWeb [141]	Vertebrae	15	T2-weighted MRI
ToothSeg [52]	Individual teeth	598	X-Ray
WBC [142]	White blood cell and nucleus	400	Microscopy
WMH [73]	White matter hyper-intensities	60	multi-modal MRI
WORD [87]	Organ segmentation	120	CT

Table 10. **Collection of datasets in MegaMedical 2.0**. The entry number of scans is the number of unique (subject, modality) pairs for each dataset.

Dataset Name	**Description**	**# of Scans**	**Image Modalities**
LIDC-IDRI [5]	Lung Nodules	1,018	CT
QUBIQ [94]	Brain, kidney, pancreas and prostate	209	MRI T1, Multimodal MRI, CT
STARE [49]	Blood vessels in retinal images	20	Optical camera

Table 11. **Multi-Annotator Data**. The entry number of scans is the number of unique (subject, modality) pairs for each dataset.