A Appendix

A.1 Tissue Modeling & Rendering

The Arctique dataset has been engineered to replicate the complexity of Histopathological Hematoxylin & Eosin (H&E) stained colon tissue images. This section describes the key steps involved in generating the synthetic dataset, encompassing macro-structure modeling, cell placement, sectioning, staining, and rendering.

Macro-structure Modeling We first generate a surface mesh object in Blender representing a large section of colonic tissue with intricate epithelial crypts to serve as the foundation for our dataset. Specifically, we mimic the rod-shaped crypts. These are arranged on a hexagonal grid, see Figure 6a. To add more detail the resulting mesh is then perturbed along the xy-plane using Blenders noise implementation. The crypts surface mesh serves as basis to build the volumes which we will use later to place cell objects, namely the stroma (the space in-between the crypts) and ring-shaped volumes within the crypts for placing epithelial nuclei and goblets.

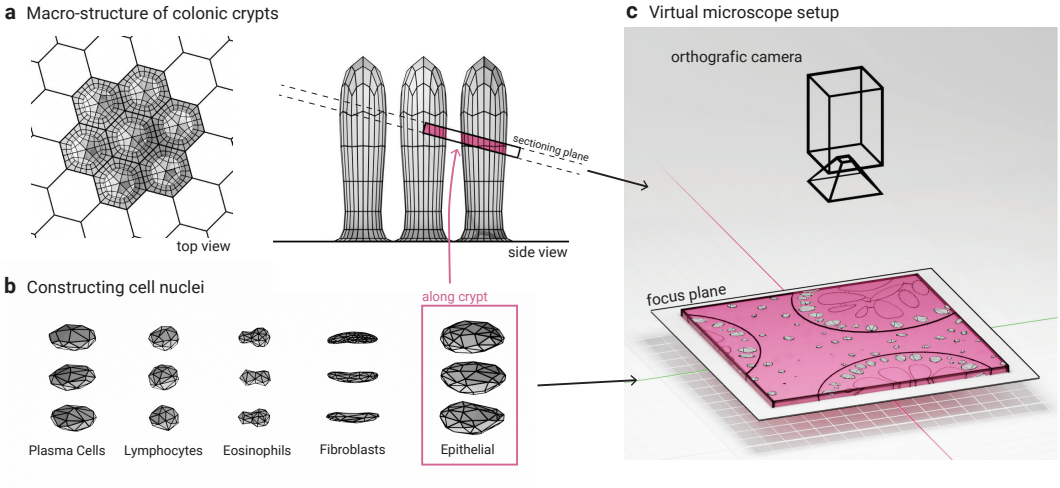

Figure 6: **Dataset Modeling**: (a) The base surface of colonic tissue is modeled as multiple rod-shaped crypts organized in a hexagonal pattern. This model is then sectioned along a plane to produce a synthetic tissue sample. (b) Five relevant cell types and their nuclei are modeled to create the final segmentation masks. (c) After populating the sectioned tissue sample with cells and applying appropriate staining, a digital image is generated using a virtual microscope setup.

Sectioning To simulate the 2D nature of histopathological images, we generate smaller digital tissue sections akin to real-world tissue slices. We intersect the macro-structure volumes with a thin rectangular 3d slice representing the real-world tissue dimensions (see Figure 6a) resulting in cut cells and nuclei, see Figure 6c. The location and orientation of the sectioning plane is chosen randomly for each image resulting in a procedurally generated dataset with diverse appearances, see Figure 6a.

Cell Placement From the sectioned macrostructure model we generate separate 3D volumes for the *stroma* and *epithelium*, which are then populated with procedurally generated cells, see Figure 6b.

The distribution of natural cells in the stroma typically exhibits an almost random structure. We model four common cell types, namely plasma cells, lymphocytes, eosinophils and fibroblasts. Each cell consists of its nucleus and the surrounding cytoplasm. Each cell type is distinguished by controllable parameters such as its typical diameter, elongation, and nucleus shape. We procedurally model each cell as an ellipsoid and apply slight deformations using a random noise parameter resulting in realistic cell shapes. We then distribute these cells uniformly throughout the stroma tissue.

The epithelium consists of an outer surface populated with epithelial cells and an inner surface populated with goblet cells which exhibit a characteristic ring- or flower-like structure when the tissue is sectioned. For this, we first sample points on the surface of the epithelium volume in order to achieve a slightly irregular hexagonal lattice structure. A cell type specific radius parameter hereby

controls the minimal lattice distance of these points. The points are then used as seeds to compute the vertices, egdes and faces of the resulting 3D Voronoi regions using the `scipy.spatial.Voronoi` library ([27], [17]). From these vertices, egdes and faces we generate a mesh for each polyhedral Voronoi region. The epithelial and goblet cells are then created by placing a deformed best-fitting ellipsoid into each Voronoi region.

Staining We replicate the characteristic staining colors of H&E stained images based on Blenders volumetric shaders. These mimic absorption and scattering in volumes with a set density. We first set the following staining parameters for individual objects such as cell nuclei, cell cytoplasm and the surrounding tissue:

- **Staining Hue**: Specifies the color hue for the staining of the objects. For example, each cell type has a unique hue value to differentiate it visually in the images.
- **Staining Intensity**: Determines the intensity of the staining for each object. Higher values result in more prominently stained cells, facilitating their identification and segmentation. Conversely, lighter staining can be used to introduce uncertainty at the image level, as it makes cell nuclei and cytoplasm less distinguishable from the surrounding tissue. In 7a, we exemplary display varying nuclei staining intensities.
- **Staining Noise**: Introduces variability in the staining intensity to mimic real-world staining irregularities. This parameter helps in creating more realistic synthetic images by adding slight variations to the staining across different cells.

Rendering We simulate the lighting conditions of a light microscope by using an area light in Blender. This light source accurately mimics the illumination provided by a microscope's light. The final scene is then rendered by performing raytracing from a virtual camera placed above the light object and tissue slice, see Figure 6c. By adjusting the camera's focal plane, we achieve a depth-blurring effect typical of histopathological light microscopy images. This workflow allows us to generate both 2D images and high-resolution 3D stacks. Additionally, the synthetic image generation provides precise pixel-wise semantic and instance masks, serving as exact ground truth.

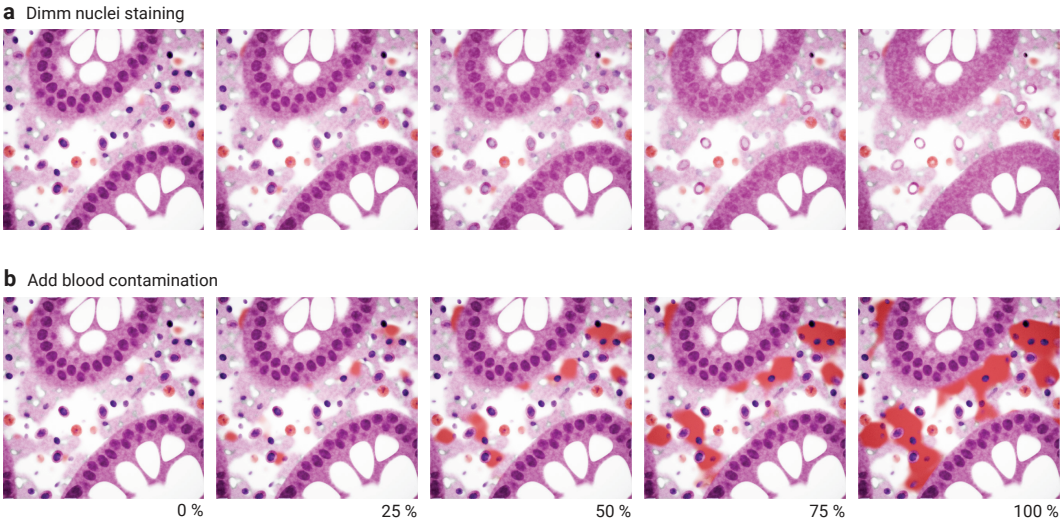

Figure 7: **Parameter-Sliders** allow the create different versions of the same scene.

Controllable Parameters The dataset generation process incorporates the following adjustable parameters to facilitate control over rendering outcomes. Many of these parameters allow to generate different versions of the same scene (consistent placement of cells and nuclei), as depicted in 7.

- **tissue thickness**: Controls the thickness of the digital tissue section. A thicker tissue allows for overlapping nuclei in the resulting image due to the higher depth.
- **tissue size**: Determines the overall size of the tissue sample. This parameter sets the dimensions of the tissue area to be rendered.

- **tissue color**: Specifies the color of the tissue in RGBA format. This parameter defines the appearance of the tissue under simulated staining conditions.
- **tissue intensity**: Adjusts the intensity the color intensity of the tissue. Higher values result in more prominently stained tissue which makes nulcei less indentifiable.
- **tissue ripping**: Simulates the ripping artifacts common to real-world scenarios. This parameter controls the occurence and strength of tissue ripping. Higher values introduce more irregularities in the tissue structure, mimicking real histological variations.
- **nuclei intensity**: Determines the staining intensity of cell nuclei. Higher values result in darker, more easily identifiable nuclei.
- **cell density**: Controls the density of cells within the stroma tissue.
- **cell type ratios**: Specifies the ratios of different cell types within the tissue. This parameter determines the relative abundance of lymphocites, plasma cells, eosinophils and fibroblasts.
- **cell shape mixing factor**: This parameter influences the blending of cell shapes across different cell types. Specifically, it enables linear interpolation of cell shapes between plasma cells and lymphocytes, allowing for controlled variation and uncertainty evaluation. This facilitates the creation of intermediate cell shapes, which can be useful for testing segmentation models under varying conditions.

A.2 The Zero-Shot Segmentation

To quantify and demonstrate Arctique's ability to replicate the complexity of an H&E dataset, we train HoVer-NeXt (HN) [3, 23] as the state-of-the-art reference model on four datasets, three of which are variants of Arctique, to perform zero-shot learning on a panoptic segmentation dataset.

Starting from the HoVer-Net model setup in [11], [23] simplifies the pipeline by substituting the binary nuclei segmentation map by the 3-class-nuclei center-background prediction map (BCB-map) and combining two instance segmentation decoders into a single decoder. Maps are created on-the-fly from instance segmentation masks. The architecture backbone is a U-Net architecture [22] with an EfficientNet-V2 encoder [25].

The strengths of the workflow primarily lie in its pre-processing and post-processing steps. To address the class imbalance in Lizard, [23] employ class-based importance sampling using per-pixel statistics, along with a focal loss regularized by a sample-based class prior [2]. Additionally, an extensive hyperparameter search for the watershed algorithm is conducted during inference to accommodate the varying shapes and sizes of the objects.

HoVer-NeXt (HN) updates the model with a ConvNeXt-v2 encoder, which demonstrates competitive results across various benchmarks [30]. In our experiments for Section 2, we utilize the ConvNeXt-v2 Tiny. We replace the focal loss regularized by a sample-based class prior with a standard focal loss function [18], and the Whole Slide Images (WSI) routine is not required.

Training For the experiments in Section 2, HN is trained for 200000 steps with a batch size of 12 using the AdamW optimizer with a weight decay of 0.0001. We employ a cosine-annealing learning rate schedule that ranges from $1\mathrm{e}{-4}$ to $1\mathrm{e}{-8}$. For the encoder, we utilize ImageNet pre-trained ConvNeXt-v2 encoders from the `pytorch-segmentation-models (timm)` [29]. All encoders are trained with 50% dropout, while the decoder does not use dropout. The two arms of instance and semantic segmentation have distinct losses:

- for the instance arm, the center point vector predictions are trained with MSELoss and the BCB-map with cross entropy loss;
- for the class prediction arm, the Focal Loss (with parameter $\gamma = 2.0$) is employed.

The Focal Loss and the MSE-Loss are summed and weighted with a pre-set weighting parameters ($\lambda = 0.02$). Model selection is done via best validations metrics specific to the dataset instead of lowest validation loss. Given the challenges of zero-shot learning, we opt to use the F1 score per class instead of Panoptic Quality (PQ) or multiclass Panoptic Quality (mPQ). A discussion on metrics will follow in subsequent subsections.

Based on [3] and [26] we apply HED color augmentations, hue saturation and brightness variation, random noise and Gaussian blurring. We also include random rotation, flipping, mirroring , zoom,

scale, shear, translate and elastic transform. Due to the specific sizes of the nuclei, masks and images of Arctique, in pre-processing phase the images were resized to 256×256 using nearest neighbor interpolation. This approach ensures that the features remain unchanged during training in relation to the size of the objects in the images. Additionally, the α parameter of the elastic transformation is magnified to enhance the realism of the cell borders.

Inference Following the adaptation of the model to the Arctique dataset, there is no longer a need to utilize parts of the Whole Slide Imaging (WSI) framework for inference. To reduce overfitting risk and enhance generalization, we apply Test-Time Augmentation (TTA) during both training and inference phases. For further details on TTA, please refer to Section 3. Briefly, our TTA procedure includes HED color augmentation, mirroring, and 90° rotation to mitigate potential negative effects, such as those caused by excessive Gaussian blur.

HN enables tiles to be center-cropped and stitched together to create larger regions, facilitating parallel processing and metric computation. Based on individual class thresholds, we generate foreground areas and seed points in the BVB map, which are then processed using a watershed algorithm to extract nuclei instances. Small holes in instances are removed, and any false merges are addressed. The parameters previously validated for the Lizard dataset were reused to configure Arctique. Classes are assigned based on a majority vote, and instances are filtered according to class-specific size thresholds determined through hyperparameter search on the validation set.

Lizard and Arctique Datasets Description

1. **Lizard ($\mathcal{L}$)** The Lizard dataset is an H&E based nuclei segmentation and classification dataset for large-scale colorectal cancer (CRC) and normal colon tissues with six classes respectively corresponding to index masks from 1 to 6: neutrophils, epithelial cells, lymphocytes, plasma cells, eosinophils, and connective tissue cells [10]. Raw 4981 H&E images are available as pre-cropped (with overlap) 256×256 tiles. It combines multiple datasets form several institutes and has 459179 total annotated nuclei, but it is highly imbalanced, particularly with neutrophils and eosinophils. Additionally, 84% of the dataset is background. Using a train-validation-test split of $80 - 10 - 10$, we create the benchmark datasets on which we train and evaluate the baseline model.
2. **Arctique ($\mathcal{A}$):** The Arctique dataset is designed to closely replicate the complexity and features of H&E-stained colon tissue images. In its latest version, it provides pre-rendered 512×512 images optimized for segmentation task training and evaluation, complete with detailed masks (2D and 3D), metadata detailing cellular object characteristics, and rendering parameters for scene regeneration. For this experiment, a subset of 1450 items was curated, split into training and validation sets with a $90 - 10$ ratio, with an additional test set of 50 samples, all equipped with instance and semantic masks.
3. **Arctique with additional noise ($\mathcal{A}_n$):** This dataset is a more complex subset of Arctique, incorporating numerous elements that define its complexity. Multiple variations are uniformly applied using the out-of-distribution noise "slider" inherent to the dataset creation process. Potential variations include adjusting the quantity of blood cells in the scene, modifying the intensity of the nuclei, changing the size of the epithelial cells, applying an overall hue shift, and altering the strength of the red tones in the eosinophils. The number of samples, image size, and train-validation-test split remain consistent with those of $\mathcal{A}$. The hypothesis behind using this more comprehensive dataset is to showcase its ability to enable generalized learning, thereby reducing overfitting while maintaining similar performance levels.
4. **Arctique with less complexity ($\mathcal{A}_{dm}$):** This dataset is a simplified subset of Arctique, with many complexity-defining elements removed. The images are generated by mapping pixel values along the depth axis (from which the 3D image is sliced) to a dark color resembling a stained cell nucleus, while the background is assigned a color closely matching that of the surrounding tissue. The number of samples, image size and the train-val-test split are the same as for $\mathcal{A}$. The hypothesis behind using this simplified dataset is to demonstrate that the inherent complexity of Arctique allows the model to effectively learn and segment the intermediate features of cellular objects.

Evaluation Metrics [6] argue that PQ should not be used for evaluating nuclei segmentation and classification because the small size of nuclei renders Intersection over Union (IoU) overly sensitive to coarse annotations. This sensitivity can lead to misleading evaluations, especially when the

annotations lack precision. Given the intricate structures of nuclei, even slight misalignments in annotations can significantly impact IoU calculations, skewing the results and not accurately reflecting model performance. As a result, we do not report PQ for comparison, despite it being monitored during training and evaluation.

For binary detection we use F1 score and Matthews Correlation Coefficient (MCC). The detection method is based on the distance-based matching approach [24], an alternative to the widely used Intersection-over-Union (IoU). Then, we evaluate the detections using balanced accuracy and F1 Score. Detection metrics for Lizard and Arctique are evaluated on 248×248 center crops for consistency and to avoid having to detect nuclei with their center outside of the tile.

In the first experiment, we compare the baseline model $\widehat{f}_{\mathcal{L}}$, trained and evaluated on the Lizard dataset, with a model $\widehat{f}_{\mathcal{A}}$ pre-trained and inferred on the Arctique dataset $\mathcal{A}$. Compared to models trained on more complex data, $\widehat{f}_{\mathcal{A}_n}$, and less complex data, $\widehat{f}_{\mathcal{A}_{dm}}$, the baseline model $\widehat{f}_{\mathcal{L}}$ achieves a higher F1 score overall. However, when examining specific classes, the model with higher complexity, $\widehat{f}_{\mathcal{A}_c}$ shows a higher F1 for lymphocytes and a lower overall Hausdorff distance, confirming the expectation that training with noise is equivalent to applying a form of regularization [4]. A real correlation between predictor and ground truth is recorded for $\widehat{f}_{\mathcal{A}}$ and $\widehat{f}_{\mathcal{A}_n}$ respectively for epithelial cells and lymphocytes. In line with our assumptions, however, the worst metrics are reported by $\widehat{f}_{\mathcal{A}_{dm}}$. The above-mentioned results are available in in Figure 2 and are based on inference over 5 rounds, each with 16 TTA.

A.3 Additional Segmentation Tasks

The two segmentation tasks we address to analyze their predictive uncertainty primarily differ in the number of output channels, while all perform pixel-wise predictions:

FG-BG-Seg: Here the goal is to distinguish between foreground and background. The model outputs two channels: one for the background (BG $= 0$) and one for the foreground (FG $= 1$).

Sem-Seg: Here the goal is to distinguish between five different types of cells and background [19]. The model outputs five channels: one for the background (BG $= 0$) and five for plasma cells, lymphocytes, eosinophils, fibroblasts and epithelial cells.

For the label-noise and image-noise manipulations outlined in Section 2, full-data inference is conducted on *FG-BG-Seg* and *Sem-Seg*.

The Segmentation Backbone In this paragraph, we provide a detailed explanation of the model backbones, which serve as the fundamental building blocks for uncertainty estimation. It is important to note that we did not conduct exhaustive hyperparameter search. Rather, we opted for default values as long as they produced reasonable results. Further exploration of more optimal training settings remains an open direction for future works.

For all the evaluated segmentation tasks, we use a standard U-Net [22] architecture with five convolutional blocks and ReLU activations. We add optional Dropout-layers after each convolutional block.

All segmentation models are trained on Arctique, using the cross-entropy loss and the Adam optimizer, featuring a learning rate of $5\mathrm{e}{-4}$ and weight decay of $5\mathrm{e}{-4}$. The batch size is set to 16 and the number of epochs to 200. To ensure consistency in comparing UQ methods and performing zero-shot learning, each model incorporates dropout layers with a rate $p^{dropout} = 0.5$, and only flips and 90 degree rotations are used as augmentation during training.

A.4 The Prediction Models

We test for different model predictions to derive corresponding measures of uncertainty, guided by the extensive UQ evaluation work outlined in [14]. Specifically, we compare four methods: *Monte-Carlo Dropout* [7], *Deep Ensembles* [16], *Test Time Augmentation* [28], and *Maximum Softmax Response* (MSR).

Following the notation in Section 3, we generate M probability maps $\mathrm{P}(\widehat{y}^m_{kj} = c)$, for each pixel coordinate (k, j), by sampling M realizations $\widehat{y}^m$ from the predictive distribution $p(\widehat{y}|x, \mathcal{D})$, which is

inferred using the estimated set of weights $\widehat{\omega}$, and applying the Softmax function to the corresponding logits. This process is used for all methods except MSR. In the following, we provide a more detailed outline of the sampling procedure for each of the UQ models.

Monte-Carlo Dropout (MCD) For the U-Net, dropout layers are activated during test time, and the model generates $M = 10$ predictions for each test image by passing it through the network 10 times. In each forward pass, the weight matrix is randomly masked, resulting in a distinct function being drawn for each prediction. This justifies why MCD is theoretically linked to the quantification of epistemic uncertainty.

Deep Ensembles (DE) For the ensemble models, we train $M = 5$ U-Net backbones, each instantiated and trained according to the scheme described in A.3, but initialized with different random seeds. During test time, the image is passed through each of these models and their outputs are combined. We select $M = 5$ as it offers a balanced trade-off between training cost and performance gain.

Test Time Augmentation (TTA) For the test time augmentations, we generate predictions by applying various combinations of vertical and horizontal flips, along with Gaussian blur, to each test input. Each augmented version, as well as the original unmodified input, is passed through the model to produce predictions. Prior to applying the Softmax function, inverse transformations are needed to align the predictions with the original object orientation [20]. Thereby, the process results in $M = 16$ forward passes (3 flipping possibilities, each with the same probability of occurrence and a small Gaussian noise).

Maximum Softmax Response (MSR) For the U-Net, a single set of scores is generated for each input image, as the stochastic inference paradigm is not applicable in this context.

A.5 The Uncertainty Derivation

In this subsection, we derive the uncertainty estimates for the MCD, DE and MSR models for the tasks *FG-BG-Seg* and *Sem-Seg*.

Probabilistic approach After generating the probability maps, we use the mean over the M realizations as the prediction, $\bar{p}_{\widehat{y}_{kj}}(c) = M^{-1} \sum_m p^m_{\widehat{y}_{kj}}(c)$[1], and the (Shannon) entropy as the uncertainty measure, assuming the test-time inputs are independent. The pixel-wise predictive uncertainty is thus calculated as,

$$pu(\widehat{y}_{kj}) = \mathbb{H}[\widehat{y}_{kj}|x, \mathcal{D}] = -\sum_{c=1}^{C} \bar{p}_{\widehat{y}_{kj}}(c) \log \bar{p}_{\widehat{y}_{kj}}(c) \tag{2}$$

for the class labels $c = 1, \ldots, C$. Following [15], we compute the aleatoric component of the uncertainty based on the decomposition Eq.(1). This is achieved by calculating the entropy of each m-th pixel-wise prediction and then averaging across samples,

$$au(\widehat{y}_{kj}) = \mathrm{E}_{\widehat{\omega}\sim p(\omega|\mathcal{D})}[\mathbb{H}[\widehat{y}_{kj}|x, \widehat{\omega}]] = -M^{-1} \sum_{m=1}^{M} \sum_{c=1}^{C} p^m_{\widehat{y}_{kj}}(c) \log p^m_{\widehat{y}_{kj}}(c). \tag{3}$$

Finally, the epistemic component (or mutual information in Eq.(1)) is quantified through simple subtraction,

$$eu(\widehat{y}_{kj}) = pu(\widehat{y}_{kj}) - au(\widehat{y}_{kj}). \tag{4}$$

Eq. (4) shows how the predictive entropy upper-bounds the epistemic uncertainty.

Deterministic approach As detailed above, the MSR is an example of a deterministic model and therefore it is not possible to generate multiple samples from it. In light of Eq.s (2) and (3), observe that $\mathbb{H}[\widehat{y}_{kj}|x, \mathcal{D}] = \mathrm{E}_{\widehat{\omega}\sim p(\omega|\mathcal{D})}[\mathbb{H}[\widehat{y}_{kj}|x, \widehat{\omega}]]$, thus indicating that no meaningful decomposition into aleatory and epistemic components can be obtained. In this case, we define a computationally cheaper alternative to the predictive entropy [12] via Maximum Softmax Response,

$$pu^{msr}(\widehat{y}_{kj}) = 1 - msr(\widehat{y}_{kj}|x, \widehat{\omega}) = 1 - \max_c \ \mathrm{P}(\widehat{y}_{kj} = c|x, \widehat{\omega}) \tag{5}$$

where $\widehat{\omega}$ is a point estimate and not a random variable. A promising direction for future research is to integrate the separation of aleatoric and epistemic components even in the deterministic approach. This separation can be achieved by inferring the epistemic uncertainty from the penultimate layer, as proposed by [21].

[1] From this point forward, a more streamlined notation is used.

A.6 A Note on Test Time Augmentation

The TTA model uniquely combines characteristics of both the deterministic and the probabilistic approaches. While it maintains the deterministic nature by not treating the parameter set $\widehat{\omega}$ as a random variable, it also embraces probabilistic elements by introducing diversity through stochastic transformations applied to the test inputs.

Under the same hypotheses as in [14], we consider a model that employs label-preserving transformations T, whose support is defined on the input space $\mathcal{T}$. This leads us to define the predictive distribution as $p(Y = y|x, \mathcal{D}) = \mathrm{E}_{t(x)\sim p(\mathcal{T})}[p(y|x, t(x), \mathcal{D})]$. In this context, the sampling procedure described in A.4 involves taking the expected value of the empirical predictive distribution $p(\widehat{y}|x, \mathcal{D})$ with respect to the distribution of transformations $p(t)$, rather than with respect to the distribution of model parameters $p(\widehat{\omega}|\mathcal{D})$.

Eq. (1) can thus be rewritten as,

$$\underbrace{\mathbb{H}[Y|x, \mathcal{D}]}_{\text{Predictive Unc. (PU)}} = \underbrace{\mathbb{I}[Y; t(x)|x, \mathcal{D}]}_{\text{Epistemic Unc. (EU)}} + \underbrace{\mathrm{E}_{p(\mathcal{T})}[\mathbb{H}[Y|x, t(x)]]}_{\text{Aleatoric Unc. (AU)}}. \tag{6}$$

Therefore, the expected pixel-wise entropy over the augmentations is supposed to give information about the amount of AU in the prediction for a new x,

$$au(\widehat{y}_{kj}) = \mathrm{E}_{t(x)\sim p(\mathcal{T})}[\mathbb{H}[\widehat{y}_{kj}|x, t(x)]] = -M^{-1} \sum_{m=1}^{M} \sum_{c=1}^{C} p^m_{\widehat{y}_{kj}}(c) \log p^m_{\widehat{y}_{kj}}(c), \tag{7}$$

and the mutual information between the augmentation variable $t(x)$ and the predicted label per pixel $\widehat{y}_{kj}$ can again be obtained using the formulation from Eq. (4). Given that our model remains invariant to transformations encountered during training [1], the mutual information between prediction and augmentation for a new test input would not be zero. However, if we were to augment the test image and include it in the training set for retraining, the mutual information would become zero. Hence, the term in Eq. (7) is reducible by adding new training points, thereby affirming the initial hypothesis that it represents epistemic uncertainty. This probabilistic approach tailored to TTA also facilitates the applications of the same inference functions delineated in A.5.

A.7 The Aggregation Strategies

The aggregation strategy is a unique component for UQ in semantic and instance segmentation tasks, unlike classification problems where it is not required. Motivated by the analysis in [14], we investigate three distinct strategies for aggregating pixel-wise uncertainty maps into image-wise uncertainty scores. All results presented in Section 3 are based on aggregating pixel-level uncertainty into a single scalar value using,

Threshold level aggregation For threshold aggregation, we must find a level above which pixels are deemed "uncertain". Intuitively, uncertainty is typically highest at object borders, correlating with object size.

First, the mean foreground ratio across all predicted segmentations in the validation set is so determined,

$$\alpha = \frac{\#\text{foreground prediction pixels}}{\#\text{pixels}} \tag{8}$$

The probability quantile is then calculated as $p = 1 - \alpha$, which is then used to compute the empirical quantile $\widehat{Q}_{u(\widehat{y}_{kj})}(p)$ for new predictions. Here u can represent $pu(\widehat{y}_{kj})$, $au(\widehat{y}_{kj})$ or $eu(\widehat{y}_{kj})$. To approximate the "true" quantile $Q_{u(\widehat{y}_{kj})}(p)$, albeit at a slightly higher computational cost, we numerically solve for $u(\widehat{y}_{kj})$ such that its cumulative distribution function matches the specified probability p. These thresholds tend to be slightly larger than empirical estimates, but for the analysis in Section 3, the differences in final results are minimal. Hence, empirical quantiles suffice for practical purposes. Lastly, we only consider uncertainty scores that exceed $\widehat{Q}_{u(\widehat{y}_{kj})}(p)$, and then compute the average of these scores.

For further comparison, we also implement the following two aggregation strategies:

Image level aggregation Uncertainty scores for all pixels are summed per image. This simple approach is widely used in the literature, e.g [5, 9, 13]. [14] shows how in cases of segmentation with a single object the size of the object correlates positively with the final uncertainty score. To obtain a measurement independent of the size of the nuclei present in our images we divide the scores by the predicted image size.

Patch level aggregation employs a sliding window of size 10^d (where d represents the image dimensionality) to aggregate uncertainties within the window. The image-level uncertainty score is determined by selecting the patch with the highest uncertainty. Based on the statistics of our dataset, we select a sliding window of 200 pixels (roughly 14^2).

For generating Figure 3, 4 in Section 3, and Figure 8, 9, 10 in A.8, we compute as last step the average of the image-wise scores across the test dataset, assuming that the test images are independently and identically distributed (i.i.d.).

A.8 Comparison of Alternative Aggregation Strategies

In this subsection, we present variations of Figure 3 and 4 from Section 3, employing different aggregation strategies, namely *patch level* and *image level aggregation*.

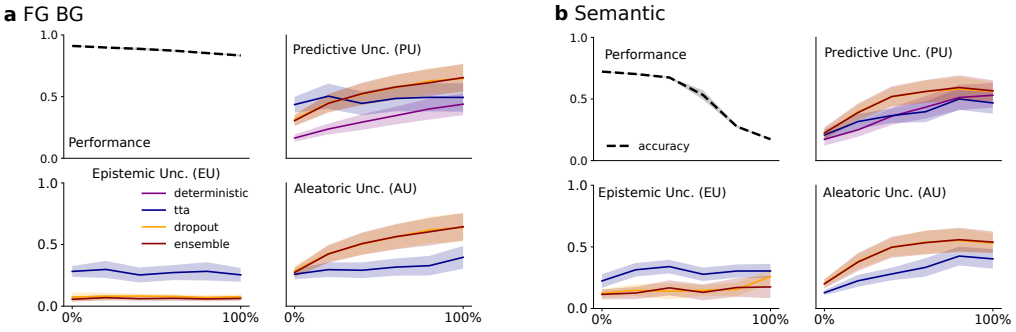

Figure 8: **Variant of Figure 3 using patch-level aggregation** (a) Effect of noisy label shapes on FG-BG-Seg: The four panels correspond to model performance across noise levels (x-axis) as measured by accuracy and dice score, predictive uncertainty for all four UQ methods, and aleatoric and epistemic uncertainty for DE, TTA and MCD. (b) Effect of noisy class labels on Sem-Seg: sub-panels analogous to (a).

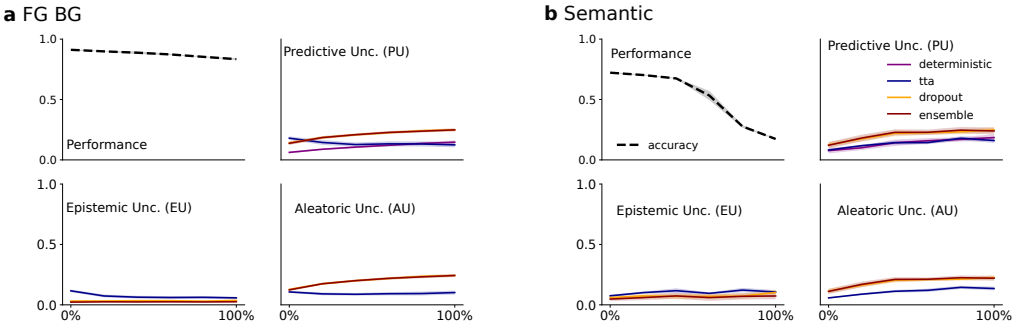

Figure 9: **Variant of Figure 3 using image-level aggregation** Structure of sub-panels is identical to Figure 9

Overall, we observe that the *patch level aggregation* results in significantly higher variance and generally larger values compared to *image level aggregation*. Despite these differences in magnitude, the two approaches yield qualitatively similar results. Specifically, the ranking of Uncertainty

Quantification (UQ) methods remains consistent across both strategies, and all methods show non-decreasing behavior as noise levels increase. When compared to the *threshold-level aggregation* strategy, the ranking of UQ methods remains the same. However, the differences between the methods become more pronounced, indicating that threshold-level aggregation is better suited to accentuate performance disparities among different UQ methods.

As highlighted by [14], the choice of aggregation strategy is often influenced by the nature of the task and the characteristics of the dataset. The primary motivation for selecting the *threshold level aggregation* for Figure 3, 4 is its lower standard deviation in the estimates, which suggests more stable and reliable results. Moreover, the *image level aggregation* method has shown, in various instances, to be somewhat heuristic. It is often unclear whether certain conclusions drawn from this method are due to the quality of the uncertainty estimates or simply the size of the foreground objects [9, 5]. This ambiguity further supports the preference for using alternative strategies in our analysis. Developing methods to more comprehensively compare the three aggregation strategies presents an intriguing direction for future research.

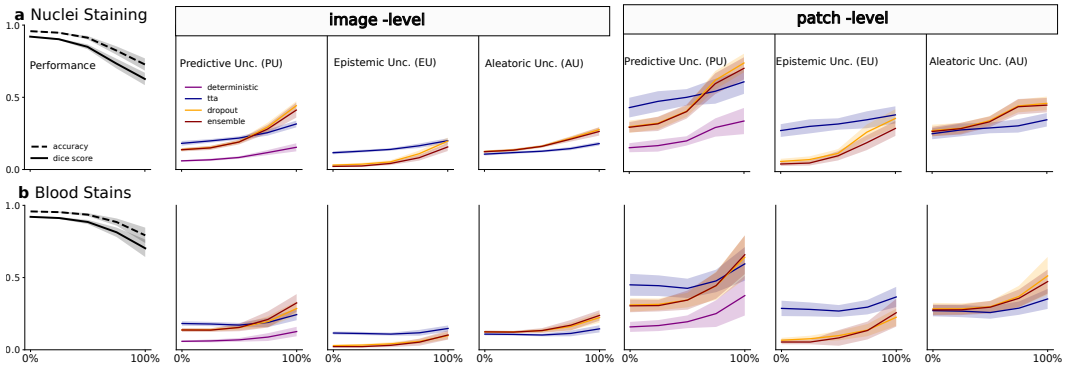

Figure 10: **Variant of Figures 4(b) and 4(c) using image- and patch-level aggregation**. As in figure 4 we distinguish between decreasing intensity of nuclei staining (a) and increasing prevalence of red-spots (b). Sub-panels show the effect of the image-level manipulations on for a FG-BG-Seg. model trained on exact labels. From left to right are shown: prediction performance of the segmentation model, predictive, epistemic and aleatoric uncertainty aggregated on the image level and predictive, epistemic and aleatoric uncertainty aggregated on the patch level.

A.9 Applicability To Classification Tasks

While our primary experiments emphasize UQ for image segmentation, this section highlights the broader potential of our dataset. We focus on segmentation due to the significant challenges associated with obtaining accurate pixel-wise annotations. Nonetheless, the dense segmentation labels and procedural metadata in Arctique can also be leveraged to derive image-level labels, extending its applicability to classification tasks.

As a proof-of-concept, we generated binary labels from the segmentation masks to indicate the presence or absence of specific cell types. Utilizing Arctique's variational framework, we modified cell appearances and analyzed how these alterations influenced classification performance and uncertainty scores, see Figure 11.

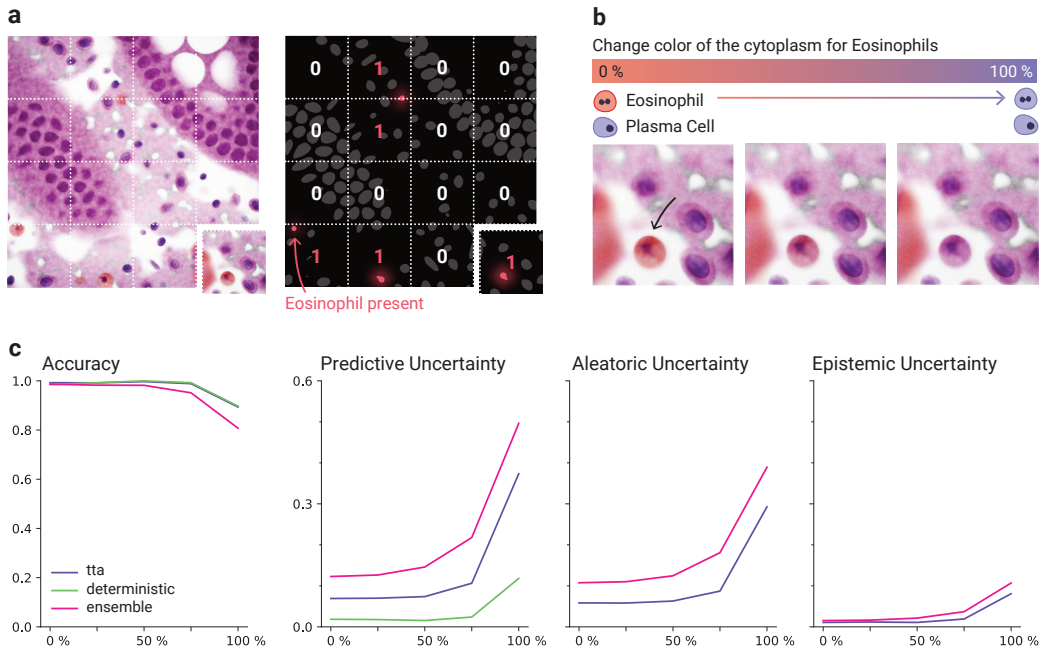

Figure 11: Arctique for Classification: From the existing segmentation masks we devise an exemplary classification task, namely detecting the presence/absence of eosinophils. a) Constructing image-level labels: We divide the image into smaller patches ensuring the desired cell type is not present on all patches. Then we use the mask to obtain binary class labels. b) We use the Arctique interface to vary the staining color of the eosinophilic cytoplasm from red to purple at test-time. c) As noise levels increase in this variation, accuracy declines, while both epistemic and aleatoric uncertainty increase.

References

[1] Sur les fonctions convexes et les inégalités entre les valeurs moyennes. Acta Mathematica, 30(1):175–193, 1906.

[2] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 15379–15389, 2021.

[3] Elias Baumann, Bastian Dislich, Josef Lorenz Rumberger, Iris D Nagtegaal, Maria Rodriguez Martinez, and Inti Zlobec. Hover-next: A fast nuclei segmentation and classification pipeline for next generation histopathology. In Medical Imaging with Deep Learning, 2024.

[4] Chris M. Bishop. Training with Noise is Equivalent to Tikhonov Regularization. Neural Computation, 7(1):108–116, 01 1995.

[5] Steffen Czolbe, Kasra Arnavaz, Oswin Krause, and Aasa Feragen. Is segmentation uncertainty useful? In Information Processing in Medical Imaging: 27th International Conference, IPMI 2021, Virtual Event, June 28–June 30, 2021, Proceedings 27, pages 715–726. Springer, 2021.

[6] Adrien Foucart, Olivier Debeir, and Christine Decaestecker. Panoptic quality should be avoided as a metric for assessing cell nuclei segmentation and classification in digital pathology. Scientific Reports, 13(1):8614, 2023.

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, pages 1050–1059. PMLR, 2016.

[8] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III au2, and Kate Crawford. Datasheets for datasets, 2021.

[9] Camila Gonzalez, Karol Gotkowski, Andreas Bucher, Ricarda Fischbach, Isabel Kaltenborn, and Anirban Mukhopadhyay. Detecting when pre-trained nnu-net models fail silently for covid-19 lung lesion segmentation. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert, editors, Medical Image Computing and Computer Assisted Intervention – MICCAI 2021, pages 304–314, Cham, 2021. Springer International Publishing.

[10] Simon Graham, Mostafa Jahanifar, Ayesha Azam, Mohammed Nimir, Yee-Wah Tsang, Katherine Dodd, Emily Hero, Harvir Sahota, Atisha Tank, Ksenija Benes, et al. Lizard: a large-scale dataset for colonic nuclear instance segmentation and classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 684–693, 2021.

[11] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. Medical Image Analysis, 58:101563, 2019.

[12] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. CoRR, abs/1610.02136, 2016.

[13] Alain Jungo, Fabian Balsiger, and Mauricio Reyes. Analyzing the quality and challenges of uncertainty estimations for brain tumor segmentation. Frontiers in Neuroscience, 14, 2020.

[14] Kim-Celine Kahl, Carsten T Lüth, Maximilian Zenk, Klaus Maier-Hein, and Paul F Jaeger. Values: A framework for systematic validation of uncertainty estimation in semantic segmentation. arXiv preprint arXiv:2401.08501, 2024.

[15] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? Advances in neural information processing systems, 30, 2017.

[16] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems, 30, 2017.

[17] Hugo Ledoux. Computing the 3d voronoi diagram robustly: An easy explanation. In 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007), pages 117–129, 2007.

[18] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 2999–3007. IEEE Computer Society, 2017.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[20] Nikita Moshkov, Botond Mathe, Attila Kertesz-Farkas, Reka Hollandi, and Peter Horvath. Test-time augmentation for deep learning-based cell segmentation on microscopy images. Scientific Reports, 10(1):5068, 2020.

[21] Janis Postels, Hermann Blum, César Cadena, Roland Y. Siegwart, Luc Van Gool, and Federico Tombari. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. ArXiv, abs/2012.03082, 2020.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, pages 234–241. Springer, 2015.

[23] Josef Lorenz Rumberger, Elias Baumann, Peter Hirsch, Andrew Janowczyk, Inti Zlobec, and Dagmar Kainmueller. Panoptic segmentation with highly imbalanced semantic labels. In 2022 IEEE International Symposium on Biomedical Imaging Challenges (ISBIC), pages 1–4, 2022.

[24] Korsuk Sirinukunwattana, Shan E. Ahmed Raza, Yee-Wah Tsang, David R. J. Snead, Ian A. Cree, and Nasir M. Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. IEEE Transactions on Medical Imaging, 35:1196–1206, 2016.

[25] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pages 10096–10106. PMLR, 2021.

[26] David Tellez, Geert J. S. Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. Medical image analysis, 58:101544, 2019.

[27] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, Ilhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy. Scipy 1.0-fundamental algorithms for scientific computing in python. CoRR, abs/1907.10121, 2019.

[28] Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. Neurocomputing, 338:34–45, 2019.

[29] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[30] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. pages 16133–16142, 06 2023.

B Dataset access

The current version of our dataset, as well as the complete version history, can be accessed via https://doi.org/10.5281/zenodo.11635056

In particular, the following link provides access to 50,000 training and 1,000 test images along with their corresponding instance and semantic masks, including 400 additional exemplary variations corresponding to 50 test images: https://zenodo.org/records/12704955

The dataset used for the results presented in this paper, including noisy variations, is provided here: https://zenodo.org/records/14016860

The complete code for reproducing the dataset, creating variations of the same dataset using parameter sliders and evaluating uncertainty quantification methods can be found here: https://github.com/Kainmueller-Lab/arctique

C Metadata access

Metadata for the dataset is provided by Zenodo, which offers export options in a variety of standard formats to facilitate easy integration and citation.

D Author Statement and Confirmation of Data License

The authors of this work bear all responsibility in case of any violation of rights, misuse of data, or other related issues.

Our final contribution encompasses three key components: the complete synthetic dataset, the codebase responsible for generating this dataset, and the codebase utilized for evaluation purposes. In adherence to Blender's GNU General Public License (GPL), we specify the following licensing arrangements:

Dataset & Evaluation Codebase: MIT License
The dataset itself and the codebase utilized for evaluation purposes will be licensed under the MIT License. This license allows for maximum flexibility and ease of use, enabling others to freely utilize, modify, and distribute the dataset and associated evaluation codebase without significant restrictions.

Codebase for Image Generation: GNU General Public License v3.0 (GPLv3)
The codebase responsible for generating the synthetic images interacts directly with Blender's API, thus forming a derivative work of Blender. Therefore, in accordance with Blender's GPL license, this codebase must also be distributed under the terms of the GNU General Public License v3.0 (GPLv3). This ensures that modifications and distributions of the image generation codebase are also subject to the principles of copyleft, ensuring the continued openness and availability of the codebase.

E Dataset Documentation

In this section we answer the Datasheet for Datasets questionnaire [8] to document the Arctique dataset. It contains information about motivation, composition, collection, preprocessing, usage, licensing as well as hosting and maintenance plan.

E.1 Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.
The dataset was created to simulate histopathological images for the purpose of developing and evaluating segmentation models. The specific task in mind was to create a dataset that reflects the complexity of Hematoxylin & Eosin (H&E) stained colon tissue in light microscopy, addressing the need for a reliable and reproducible dataset that can be used for training and testing machine learning models in this domain.

Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?
This dataset was created in a collaboration of the Kainmueller Lab from the Max-Delbrueck-Center for Molecular Medicine in the Helmholtz Association (MDC) and Helmholtz Imaging in Berlin, as well as the Institute of Pathology from the Charité, Universitätsmedizin Berlin, Corporate Member of Freie Universität Berlin and Humboldt-Universität zu Berlin.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.
Funding: German Research Foundation (DFG) Research Training Group CompCancer (RTG2424), DFG Research Unit DeSBi (KI-FOR 5363, project no. 459422098), DFG Collaborative Research Center FONDA (SFB 1404, project no. 414984028), DFG Individual Research Grant UMDISTO (project no. 498181230), Synergy Unit of the Helmholtz Foundation Model Initiative, Helmholtz Einstein International Berlin Research School In Data Science (HEIBRiDS).

E.2 Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.
The instances in the dataset are digitally generated images akin to histopathological images that represent various cell types within colonic tissue. There are multiple types of instances, including images of tissue sections and corresponding pixel-wise semantic and instance masks.

How many instances are there in total (of each type, if appropriate)?
The full dataset includes 50,000 training images and 1,000 test images, each with corresponding instance and semantic masks, along with 400 additional variations based on 50 test images. Additionally, we provide the specific dataset used in our experiments, comprising 1,500 synthetic images without noise, 1,500 with added noise, and 1,500 depth mask images, along with a range of noisy variations.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).
The dataset is a synthetic representation and not a sample from a larger set. It is designed to simulate the diversity and complexity of real-world histopathological images.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.
Each instance consists of raw image data of tissue sections, along with ground-truth pixel-wise semantic and instance masks.

Is there a label or target associated with each instance? If so, please provide a description.
Yes, each instance has corresponding ground-truth pixel-wise semantic and instance masks that serve as labels.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.
No information is missing from the individual instances.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.
Relationships between individual instances are not applicable in this dataset.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.
The dataset can be split into training, validation, and testing sets as needed. The specific splits are left to the user's discretion based on their experimental setup.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.
The dataset is synthetically generated to minimize errors and noise. However, simulated noise and artifacts are intentionally included to reflect real-world conditions.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a dataset consumer? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.
The dataset is self-contained and does not rely on external resources.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description.
No, the dataset does not contain confidential data.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.
No, the dataset does not contain such data.

Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.
No, the dataset does not identify subpopulations.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.
No, it is not possible to identify individuals from the dataset.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals race or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.
No, the dataset does not contain sensitive data.

E.3 Collection Process

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If the data was reported by subjects or indirectly inferred/derived from other data,

was the data validated/verified? If so, please describe how.
The data for each instance was directly observable, consisting of synthetically generated images of digitally modeled colonic tissue using the rendering software Blender. Validation was performed by comparing generated images with real histopathological images to ensure similarity.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)? How were these mechanisms or procedures validated?
The image generation was carried out using the 3D rendering software Blender and Python scripts. The procedure was validated by comparing the generated images with real histopathological images to ensure visual and structural similarity.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?
The dataset is not a sample from a larger set; it consists entirely of synthetically generated images. Moreover, additional images can be produced using scripts with adjustable parameters.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?
The data collection process was carried out by a team of researchers. Compensation details are not applicable as this was part of their research and development activities.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.
The timeframe for generating the dataset depends on the available computing resources. Specifically, the image generation process alone spanned about one week, while the entire script generation process extended over approximately six months.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.
No ethical review processes were conducted as the dataset does not involve human subjects or personal data.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?
The data was not collected from individuals; it was synthetically generated.

Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.
Not applicable, as the dataset does not involve human subjects.

Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.
Not applicable, as the dataset does not involve human subjects.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).
Not applicable, as the dataset does not involve human subjects.

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.
Not applicable, as the dataset does not involve human subjects.

E.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing

of missing values)? If so, please provide a description. If not, you may skip the remaining questions in this section.
Yes, preprocessing and labeling were done. After generating the synthetic images, exact pixel-wise semantic and instance masks were created to label each cell type accurately. Additionally, the images were reviewed and minor cleaning steps were performed to remove any artifacts from the rendering process.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.
There is no "raw" data as the images and labels were generated algorithmically.

Is the software that was used to preprocess/clean/label the data available? If so, please provide a link or other access point. No such software was employed.

E.5 Uses

Has the dataset been used for any tasks already? If so, please provide a description.
Yes, the dataset has been used for training and evaluating segmentation models for histopathological images. The generated images and masks help in benchmarking and improving the performance of these models.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.
No, there is no such repository.

What (other) tasks could the dataset be used for?
The dataset could be used for a variety of tasks including but not limited to:
- Training and evaluating segmentation and classification models (both 2D and 3D)
- Evaluating uncertainty quantification methods
- Evaluating Explainable AI (XAI) methods
- Evaluating sampling strategies for Active Learning (AL)
- Research on domain adaptation in histopathological imaging

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a dataset consumer might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other risks or harms (e.g., legal risks, financial harms)? If so, please provide a description. Is there anything a dataset consumer could do to mitigate these risks or harms?
The dataset is synthetically generated, ensuring no real patient data is involved, thus avoiding privacy concerns. However, users should be aware that while the dataset is designed to closely mimic real histopathological images, it may not capture all the nuances of actual tissue samples. Ensuring models trained on this data are validated on real-world data is crucial to mitigate potential biases.

Are there tasks for which the dataset should not be used? If so, please provide a description.
The dataset should not be used for any applications requiring actual patient data or clinical decision-making without further validation. It is also not suitable for tasks that require genetic or molecular level analysis as the dataset does not contain such information.

Any other comments?
The synthetic nature of the dataset allows for extensive control and reproducibility, making it a valuable resource for research and development in medical imaging. However, users should complement their studies with real-world data to ensure the applicability and robustness of their models.

E.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?
Yes, the dataset will be distributed to third parties outside of the entity to encourage further research and development.

How will the dataset be distributed (e.g., tarball on website, API, GitHub)? Does the dataset have a digital object identifier (DOI)?
The dataset including version history is publicly available via Zenodo. The complete code for reproducing the dataset, creating variations of the same dataset using parameter sliders and evaluating uncertainty quantification methods is available on GitHub. See Section B for URLs.

When will the dataset be distributed?
The dataset will be published and distributed upon acceptance of the paper.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?
Yes, the dataset will be distributed under the MIT license upon publication. There are no fees associated with this license.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances?
No third parties have imposed IP-based or other restrictions on the data associated with the instances.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?
No export controls or other regulatory restrictions apply to the dataset or to individual instances.

E.7 Maintenance

Who will be supporting/hosting/maintaining the dataset?
The dataset will be supported, hosted, and maintained by the research team that created it.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?
Please reach out to the corresponding authors.

Is there an erratum? If so, please provide a link or other access point.
There is no erratum currently available. Any future errata will be posted on the dataset's GitHub repository which will be published upon acceptance.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to dataset consumers (e.g., mailing list, GitHub)?
The dataset will be maintained and kept up-to-date using version control in Zenodo. All updates and new versions will be communicated through Zenodo and the GitHub repository. See Section B for URLs.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.
This dataset does not relate to people, so there are no applicable limits on the retention of the data.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to dataset consumers.
Older versions of the dataset will not be actively maintained, but they will remain accessible through Zenodo. Obsolescence of older versions will be communicated to dataset consumers via Zenodo.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to dataset consumers? If so, please provide a description.
We encourage others to reach out to our research group to contribute to the dataset. All contributions will be validated and verified by our team before integration. Updates and contributions will be communicated to dataset users through the repository.

F Dataset Format and Reading Explanation

F.1 Dataset Structure

The Arctique dataset includes 50,000 training images and 1,000 test images, each with corresponding instance and semantic masks, along with 400 additional variations based on 50 test images. It is split into training and test sets, each containing directories for images and masks. There is an additional directory.

```
arctique
├── test
│   ├── images
│   ├── masks
│   │   ├── cytoplasm_indexing
│   │   ├── instance_3d_indexing
│   │   │   └── <ID>
│   │   ├── instance_indexing
│   │   ├── semantic_indexing
│   │   └── <task>_noise_<label_noise>
│   ├── metadata
│   └── parameters
├── train
│   ├── images
│   ├── masks
│   │   ├── cytoplasm_indexing
│   │   ├── instance_3d_indexing
│   │   │   └── <ID>
│   │   ├── instance_indexing
│   │   └── semantic_indexing
│   ├── metadata
│   └── parameters
└── variations
    └── <variation_type>
        └── <variation_type>_<img_noise>
            ├── images
            ├── masks
            │   ├── cytoplasm_indexing
            │   ├── instance_3d_indexing
            │   │   └── <ID>
            │   ├── instance_indexing
            │   ├── semantic_indexing
            │   └── <task>_noise_0
            ├── metadata
            └── parameters
```

F.2 Dataset description

Images

The `images` directory contains all synthetically generated images stored as PNG files. Each image has a resolution of 512x512 pixels with RGB channels and is named "img_<ID>", where <ID> is a unique integer identifier for each image.

Masks

The `masks` directory includes subdirectories containing various masks related to the images.

- `cytoplasm`: Contains 2D semantic masks for the cell cytoplasm. Each mask corresponds to an image named "<ID>.tif", where "<ID>" is the identifier for that image. The mask file is named using the same identifier.

- `instance_3d`: Contains a directory for each image, named "<ID>". Inside each directory, there is a 3D stack numpy file representing the instance IDs in a 3D volumetric array. Additionally, it includes a sequence of 2D instance segmentation masks, named "slice_<ID>_<slice_count>.png", each representing equidistant slices through the 3D volume along the depth axis.
- `instance`: Contains 2D instance masks for the cell nuclei. Each mask corresponds to an image named "<ID>.tif", and the mask file is named with the same identifier.
- `semantic`: Contains 2D semantic masks for the cell nuclei. Similar to the instance masks, each mask corresponds to an image named "<ID>.tif", with the mask file named using the same identifier.
- `<task>_noise_<label_noise>`: Contains 2D masks for the cell nuclei, where individual masks have been deformed and/or removed according to a specified probability, indicated by "<label_noise>". These modifications align with the specific requirements of the task labeled as "<task>". The `<task>_noise_0` directory consistently contains the original, unmodified masks. Each mask corresponds to an image named "<ID>.png", with the mask file named using the same identifier.

Note that all semantic masks appear as black images when viewed with a standard image viewer. This is because the cell type IDs, ranging from 1 to 6, are used as greyscale values, which appear dark in the images.

Metadata

The `metadata` directory contains JSON metadata files named "metadata_<ID>" for each image. Each JSON file includes a list of Python dictionaries, one for each cell object visible in the image. Each dictionary contain the following keys:

- **ID**: Unique identifier for the object
- **ID_Type**: ID of cell type. Possible values: 1 for epithelial cells, 2 for goblet cells, 3 for plasma cells, 4 for lymphocytes, 5 for eosinophils, and 6 for fibroblasts.
- **Type**: Short name of cell type. Possible values: "EPI" for epithelial cells, "LYM" for lymphocytes, "PLA" for plasma cells, "FIB" for fibroblasts, "EOS" for eosinophils, and "GOB" for goblet cells.
- **Cellpart**: Specific part of the cell, either "Nucleus", "Cytoplasm" or "Goblet"
- **Cellname**: Unique name of that object
- **Staining_color**: RGBA color used for staining
- **Staining_intensity**: Intensity of the staining, between 0 and 100
- **Location**: 3D location of the object
- **Location_pixel**: Pixel coordinates of the object in the image

Parameters

The `parameters` directory contains JSON files named "parameters_<ID>", which detail the parameters used to generate each image. Each JSON file is a Python dictionary with all the parameter values necessary to reproduce the scene. The detailed parameters are:

- **seed**: The random integer seed used for reproducibility.
- **gpu_device**: The GPU device ID to be used for rendering.
- **gpu**: Boolean indicating whether to use GPU acceleration.
- **output_dir**: The directory where rendered images will be saved.
- **start_idx**: The starting index for naming rendered images.
- **n_samples**: The number of tissue samples to generate.
- **tissue_thickness**: The thickness of the tissue section.
- **tissue_size**: The size of the tissue section.

- **tissue_color**: The color of the tissue staining in RGBA format.
- **tissue_location**: The location of the tissue sample in 3D space.
- **tissue_padding**: The padding around the tissue sample. Is used to reduce the complexity of the whole macrostructure model to a smaller section.
- **tissue_rips**: The number of occuring tissue ripping instances.
- **tissue_rips_std**: The standard deviation of occuring tissue ripping instances.
- **stroma_intensity**: The intensity of stroma tissue staining.
- **noise_seed_shift**: The shift in noise seed.
- **stroma_density**: The density of stroma cells.
- **ratios**: The ratios of different cell types.
- **nuclei_intensity**: The intensity of cell nuclei staining.
- **mix_factor**: The shape mixing factor for plasma cell nuclei. 0 for pure plasma cell type shape, 1 for pure lymphocyte shape
- **epi_rescaling**: The rescaling factor for epithelial cells.
- **mix_cyto**: The shape mixing factor for plasma cytoplasm. 0 for pure plasma cell type shape, 1 for pure lymphocyte shape

Variations

The directory `variations` contains subdirectories named `<variation_type>`, each corresponding to specific aspects of the images which have been intentionally manipulated, while leaving other elements unchanged. Within each `variations_type` subdirectory, there are additional `<variation_type>_<img_noise>` subdirectories.

These subdirectories contain a subset of previously described `image`, `mask`, `parameters`, and `metadata` subdirectories, where the targeted aspect's parameter has been adjusted to the value `<img_noise>`. For instance, if the manipulation pertains to staining the cell nuclei, the parameter that is modified is **nuclei_intensity**. The detailed explanation of the parameters available for manipulation is provided in the Subsection F.2.

It is important to note that the structure of the `mask` directory presented here exclusively contains the `<task>_noise_0` directory. This directory specifically holds the original, unmodified masks related to the task influenced by the chosen `<variation_type>`.