A Hierarchical Probabilistic U-Net
for Modeling Multi-Scale Ambiguities

Simon A. A. Kohl[1,2], **Bernardino Romera-Paredes**[1], **Klaus H. Maier-Hein**[3],
Danilo Jimenez Rezende[1], **S. M. Ali Eslami**[1], **Pushmeet Kohli**[1], **Andrew Zisserman**[1],
and **Olaf Ronneberger**[1]

[1] DeepMind, London, UK
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
[3] Division of Medical Image Computing, German Cancer Research Center, Heidelberg, Germany
`{simonkohl,brp,danilor,aeslami,pushmeet,zisserman,olafr}@google.com`
`k.maier-hein@dkfz.de`

Abstract

Medical imaging only indirectly measures the molecular identity of the tissue within each voxel, which often produces only ambiguous image evidence for target measures of interest, like semantic segmentation. This diversity and the variations of plausible interpretations are often specific to given image regions and may thus manifest on various scales, spanning all the way from the pixel to the image level. In order to learn a flexible distribution that can account for multiple scales of variations, we propose the Hierarchical Probabilistic U-Net, a segmentation network with a conditional variational auto-encoder (cVAE) that uses a hierarchical latent space decomposition. We show that this model formulation enables sampling and reconstruction of segmenations with high fidelity, i.e. with finely resolved detail, while providing the flexibility to learn complex structured distributions across scales. We demonstrate these abilities on the task of segmenting ambiguous medical scans as well as on instance segmentation of neurobiological and natural images. Our model automatically separates independent factors across scales, an inductive bias that we deem beneficial in structured output prediction tasks beyond segmentation.

1   Introduction

In real world applications the recorded measurements are often not sufficient to predict a single outcome. Instead, we often have a large manifold of potential interpretations, and we can only use the measurements to narrow down this manifold to a smaller manifold. The modelling of the remaining ambiguities and uncertainties is often challenging, especially in high-dimensional outputs, like segmentation maps. This particularly arises in the fine-grained distinctions that need to be made in medical images. For example a lesion in a CT scan can have identical shape and gray values independently of whether it consists of cancer or healthy cells in the real world (see Fig. 1a). In the same way the true shape of the structure might not be resolvable due to fuzzy borders, occlusions and noise. Similar ambiguities also appear in natural images, e.g. think of cats and dogs lying under a sofa with only parts of their fur being visible. An uncertainty-aware segmentation algorithm can assign a 50% cancer / 50% non-cancer probability (or 50% cat / 50% dog probability) to each pixel [1, 2], but depending on the downstream task this pixel-wise probability is not sufficient, because it only provides the marginals of the high-dimensional probability distribution. For example, in a potential clinical application a clinician with access to additional non-imaging data can select the correct segmentation, or a number of segmentation hypotheses can be presented to a subsequent

classification network to assign a diagnosis to each possible interpretation of the medical scan [3]. Several algorithms have been proposed that provide samples from the output distribution (here: consistent segmentation maps instead of pixel-wise samples). They are based on ensembles [4], networks with multiple heads [5–8], or image-conditional generative models [9–13] such as cVAEs [14–18], as in the Probabilistic U-Net [19]. These type of approaches work well for a single object in the image or for other global variations (like different segmentation styles, e.g. more narrow or more inclusive outlining), but do not scale to images containing multiple objects with uncorrelated variations. There exist several approaches which use hierarchical latents to produce rich probability distributions [20–26], but this concept has not yet been used in the context of segmentation or image-to-image translation.

Here we propose a 'Hierarchical Probabilistic U-Net' (the HPU-Net) that overcomes these issues. Similar to the existing Probabilistic U-Net [19] it combines a segmentation U-Net [27] with a cVAE. Instead of global latent variables we use a coarse-to-fine hierarchy of latent variable maps (see Fig. 1) that are injected into the synthesis path of the U-Net at the corresponding resolutions.

Our main contributions are: (1) A generative model for semantic segmentation able to learn complex-structured conditional distributions equipped with a latent space that scales with image size. This results in (2) Compared to prior art, strongly improved fidelity to fine structure in the models' samples and reconstructions. (3) Improved modelling of distributions over segmentations including independently varying scales and locations, as demonstrated in its ability to generate instance segmentations. (4) Automatic learning of factors of variations across space and scale.

We demonstrate the improved quality of the segmentations on a lung lesion segmentation task. Furthermore we show the ability of the model to learn highly complex probability distributions, by presenting an instance segmentation task, where we ask the model to label ('colorize') each instance consistently with a random instance id. We test this ability on neuronal structures in EM images as well as on car instances in natural images. Finally we show that the model is also capable of predicting consistent segmentations with corresponding uncertainties in a blacked out region of the image. In a medical application this could be used to predict disease progression by applying a 4D version of the proposed network to time series (3 spatial axes and 1 time axis), where the blacked out part corresponds to the unknown future development of the disease. The model will be open-sourced at `https://github.com/project_repo`[1].
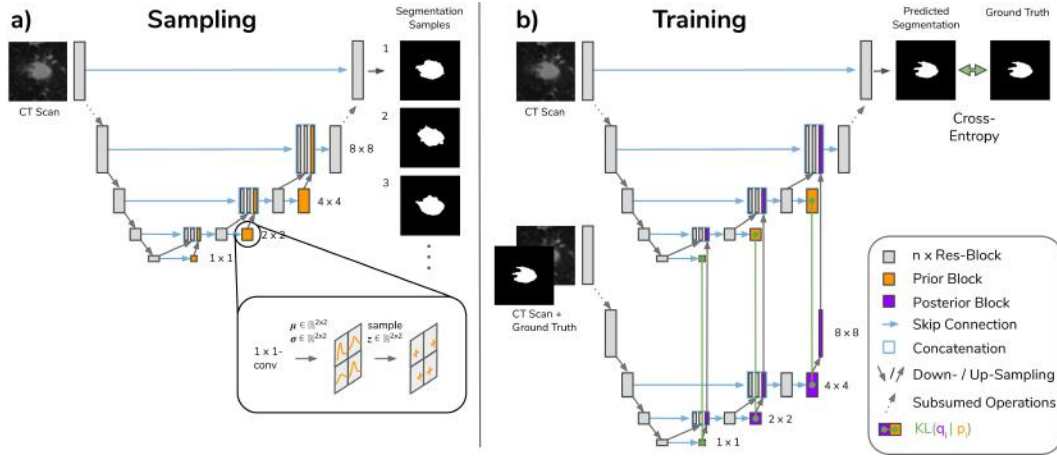

Figure 1: The Hierarchical Probabilistic U-Net. The model is based on a U-Net and adds a hierarchy of spatially arranged Gaussian distributions that is interleaved with the U-Net's decoder. (**a**) Sampling process: For each iteration of the network latents $\mathbf{z}_i$ at scale i (slim orange blocks) are successively sampled from the prior when going up the hierarchy towards increasing resolutions. (**b**) Training process illustrated for one training example: During training samples $\mathbf{z}_i$ from the posterior (slim purple blocks) are injected into the U-Net's decoder and used to reconstruct a given segmentation. Green connections: loss functions. For more details see Sec. 2 and Appx. D.

---

[1]The url will be updated once available.

2   Network Architecture and Learning Objective

The standard Probabilistic U-Net (abbreviated as 'sPU-Net') models segmentation ambiguities using a low-dimensional, image global latent vector, that is sampled from a separate 'prior net' and is combined with U-Net features by means of a shallow network of 1×1-convolutions [19]. As we show below, this image-global latent space heavily constrains the granularity at which the output space can be modelled. While our proposed architecture also combines a U-Net with a cVAE, it instead employs a hierarchical latent space that resides in the U-Net's decoder. A hierarchical decomposition yields a much more flexible generative model that can further easily model top-down dependencies. E.g. the global part can model the patient's genetic predisposition for a certain disease, while the local parts can model indiscernible tissue types, or fuzzy borders at different scales. The spatial arrangement of the latent variables further enables the network to easily model local independent variations (like multiple lesions). Due to the fully-convolutional architecture, it can also generalize from few to many lesions at arbitrary locations. Beside these fundamental extensions, we additionally removed the separate prior net and instead use U-Net internal features to predict the parameters of the prior distributions (as in [18]), which results in parameter and run-time savings. For the network to employ the full hierarchy, we further found it crucial to minimize obstructions between latent scales by introducing (pre-activated) res-blocks [28] (as discussed in Appx. D and in line with [22, 25]).

Sampling The architecture's main feature is its highly flexible parameterization of the conditional prior that it employs. This prior is composed of a) a deterministic feature extractor that computes features at spatial resolutions up to scale L (counted with ascending resolution) for the given input image X and b) a cascade of distributions interleaved with the U-Net's decoder, that allows to hierarchically sample latents. In a conventional U-Net, the U-Net decoder's features of every resolution are up-sampled and then concatenated with the features of the U-Net's encoder from the respective resolution above [27]. In our proposed architecture there is one additional step at each scale of the latent hierarchy: Conditioned on the decoder features of each scale $i \leq L$, we sample a spatial grid of latents $\mathbf{z}_i$ and concatenate it with the input decoder features, before the usual up-sampling and concatenation with encoder features from above takes place, see Fig. 1a. The latents of each scale i thus depend on the input image X and on all latents of scales $i' < i$ that have already been sampled lower in the hierarchy, which we collectively denote as $\mathbf{z}_{<i} := (\mathbf{z}_{i-1}, ..., \mathbf{z}_0)$. At each scale with spatial dimensions $H_i \times W_i$ the model uses conditional Gaussian distributions with mean $\boldsymbol{\mu}_i^{\text{prior}} \in \mathbb{R}^{H_i \times W_i}$ and variance $\boldsymbol{\sigma}_i^{\text{prior}} \in \mathbb{R}^{H_i \times W_i}$. The means and variances are predicted by 1×1-convolutions for each spatial position of that scale. Sampling from the corresponding Gaussian distribution results in the spatial latents $\mathbf{z}_i \in \mathbb{R}^{H_i \times W_i}$:

$$\mathbf{z}_i \sim \mathcal{N}\big(\boldsymbol{\mu}_i^{\text{prior}}(\mathbf{z}_{<i}, X), \boldsymbol{\sigma}_i^{\text{prior}}(\mathbf{z}_{<i}, X)\big) =: p(\mathbf{z}_i | \mathbf{z}_{<i}, X). \tag{1}$$

Our experiments did not benefit from going beyond scalar latents at each spatial location, which however is a choice that one might want to make depending on the application. The hierarchical (ancestral) sampling results in a joint distribution for the prior that decomposes as follows:

$$P(\mathbf{z}_0, ..., \mathbf{z}_L | X) = p(\mathbf{z}_L | \mathbf{z}_{<L}, X) \cdot ... \cdot p(\mathbf{z}_0 | X). \tag{2}$$

Every run of the network yields a segmentation hypothesis $Y' = S(X, \mathbf{z})$ for the given image (where $\mathbf{z} = (\mathbf{z}_L, ..., \mathbf{z}_0)$ and S stands for the segmentation network), which is illustrated in Fig. 1a. Note that only the U-Net's decoder (including the hierarchical sampling) needs to be rerun to produce the next segmentation samples for the same image. The number of latent scales L is a hyper-parameter and typically chosen smaller than the full number of scales of the U-Net; our models use $L = 3$ (4 scales).

Training As is standard practice for VAEs, the training procedure aims at maximizing the so-called evidence lower bound (ELBO) on the likelihood $p(Y|X)$, where in our case Y is a segmentation and X is an image. This requires to model a variational posterior $Q(.|X, Y)$ that depends on both X and Y. The structure matches with that of the prior:

$$\mathbf{z}_i \sim \mathcal{N}\big(\boldsymbol{\mu}_i^{\text{post}}(\mathbf{z}_{<i}, X, Y), \boldsymbol{\sigma}_i^{\text{post}}(\mathbf{z}_{<i}, X, Y)\big) =: q(\mathbf{z}_i | \mathbf{z}_{<i}, X, Y), \tag{3}$$

$$Q(\mathbf{z}_0, ..., \mathbf{z}_L | X, Y) = q(\mathbf{z}_L | \mathbf{z}_{<L}, X, Y) \cdot ... \cdot q(\mathbf{z}_0 | X, Y). \tag{4}$$

The posterior Q is modeled in form of a separate network with the same hierarchical topology in which for each scale $i \leq L$, we compute conditional Gaussian distributions with mean $\boldsymbol{\mu}_i^{\text{post}} \in \mathbb{R}^{H_i \times W_i}$ and variance $\boldsymbol{\sigma}_i^{\text{post}} \in \mathbb{R}^{H_i \times W_i}$. During training, samples $\mathbf{z} \sim Q$ are fed into the U-Net's decoder

(as illustrated in the bottom half of Fig. 1b) with the aim of learning to reconstruct the given input segmentation Y. The reconstruction objective ($\mathcal{L}_{\text{rec}}$) is formulated as a cross-entropy loss between the prediction Y' and the target Y (below formulated as a pixel-wise categorical distribution P_c). Additionally there is a Kullback-Leibler divergence $D_{\text{KL}}(Q||P) = \mathbb{E}_{\mathbf{z} \sim Q}\left[\log Q - \log P\right]$, that assimilates P and Q (more details in Appx. A). Our ELBO objective with a relative weighting factor β thus amounts to

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{\mathbf{z} \sim Q}\big[- \log P_c(Y|S(X, \mathbf{z}))\big] + \beta \cdot \sum_{i=0}^{L} \mathbb{E}_{\mathbf{z}_{<i} \sim Q} D_{\text{KL}}\big(q_i(\mathbf{z}_i|\mathbf{z}_{<i}, X, Y)||p_i(\mathbf{z}_i|\mathbf{z}_{<i}, X)\big).$$
(5)

Minimizing $\mathcal{L}_{\text{ELBO}}$ leads to sub-optimally converged priors in our experiments. For this reason we make use of the recently proposed $GECO$-objective [29] that adds in a constraint on the reconstruction term and thus dynamically balances it with the KL terms from above:

$$\mathcal{L}_{\text{GECO}} = \lambda \cdot \Big(\mathbb{E}_{\mathbf{z} \sim Q}\big[-\log P_c(Y|S(X, \mathbf{z}))\big] - \kappa\Big) + \sum_{i=0}^{L} \mathbb{E}_{\mathbf{z}_{<i} \sim Q} D_{\text{KL}}\big(q_i(\mathbf{z}_i|\mathbf{z}_{<i}, X, Y)||p_i(\mathbf{z}_i|\mathbf{z}_{<i}, X)\big),$$
(6)

where κ is chosen as the desired reconstruction loss value and the Lagrange multiplier λ is updated as a function of the exponential moving average of the reconstruction constraint. This formulation initially puts high pressure on the reconstruction and once the desired κ is reached it increasingly moves the pressure over on the KL-term. For more details we refer to Appx. D and the literature [29].

We additionally perform an online hard-negative mining, specifically, we only back-propagate the gradient of the kth percentile of the worst pixels of the batch [30], $\mathcal{L}_{\text{rec}} \rightarrow \texttt{top\_k\_mask}\,(\mathcal{L}_{\text{rec}})$. We chose $k = 0.02$ (the worst 2% pixels) in all experiments of the HPU-Net and stochastically pick the kth percentile [31] (we sample from a Gumbel-Softmax distribution [32] over $\mathcal{L}_{\text{rec}}$ per pixel).

3 Results

The sPU-Net has established significant performance advantages over other approaches in segmenting ambiguous images [19]. With this work we aim at improving on the flexibility of the sPU-Net to model complex output interdependencies as well as segmentation fidelity. To this end we report results for a segmentation task of CT scans showing potential lung abnormalities annotated by four expert graders (examples are shown in Fig. 2). We further consider the task of segmenting individual instances, i.e. inferring a latent id for each object in an image, and use it to assess the models' ability to capture correlated pixel-uncertainty. We use the EM dataset of the SNEMI3D challenge (published in [33]), which contains instance segmentations of neuronal cells (examples are shown in Fig. 4) and further probe our model's performance on the segmentation of car instances on natural street scenes from Cityscapes (see Fig. 6). For training details in the respective tasks we refer to Appx. D.

Table 1: Test set results. Mean and standard deviation are calculated from results of 10 random model initializations and 1000 bootstraps with replacement. Data splits are defined in Appx. C.

Dataset	Metric	Probabilistic U-Net (re-implementation)	Hierarchical Probabilistic U-Net
a) LIDC	IoU_{rec}	0.75 ± 0.04	$\mathbf{0.97 \pm 0.00}$
	Hungarian-matched IoU	0.50 ± 0.03	$\mathbf{0.53 \pm 0.01}$
	Hungarian-matched IoU (subset B)	0.37 ± 0.07	$\mathbf{0.47 \pm 0.01}$
b) SNEMI3D	IoU_{rec}	0.13 ± 0.03	$\mathbf{0.60 \pm 0.00}$
	Rand Error	0.52 ± 0.10	$\mathbf{0.06 \pm 0.00}$
c) Cityscapes Car Instances	IoU_{rec}	-	0.62
	Rand Error	-	0.13

3.1 Performance Measures

We are interested in assessing how well the conditional distribution produced by the respective generative model and the given ground-truth distribution agree, which we measure in terms of the intersection-over-union (IoU) based *Generalized Energy Distance* (GED^2) and the *Hungarian-matched IoU*. Furthermore, we would like to measure an upper-bound on the fidelity of the models'
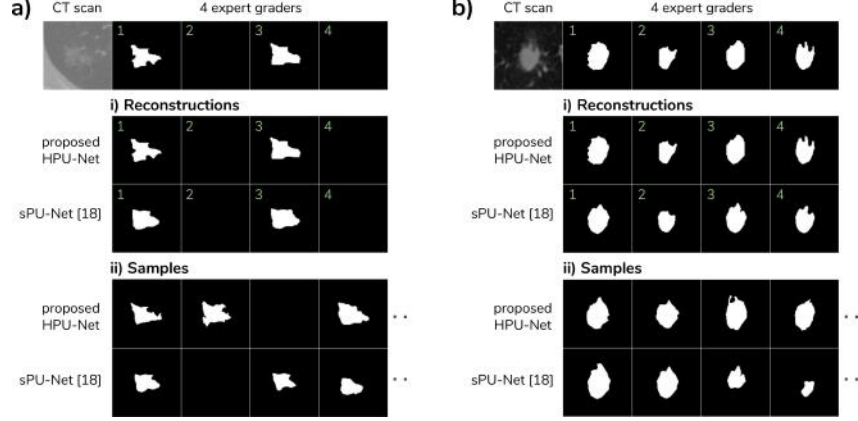
Figure 2: Two example CT scans with the 4 available expert gradings from LIDC-IDRI. (**i**) Reconstructions of the 4 graders and (**ii**) Sampled segmentations. Note that the gradings can be empty, as foreground annotations correspond to supposed abnormal cases only. More cases in Fig. 9 and 10.

samples, i.e. how accurately the models are able to produce fine segmentation structure and detail, for which we employ the *reconstruction IoU*, $\text{IoU}_{\text{rec}}(Y, Y')$ where $Y' = S(X, \boldsymbol{\mu}^{\text{post}}(X, Y))$. To assess the instance segmentation performance we employ the metric used in the SNEMI3D challenge, the *Rand Error*. For more details and definitions of all metrics, we refer to Appx. B.

3.2 LIDC: Segmentation of Ambiguous Lung Scans

The LIDC-IDRI dataset [34–36] contains 1018 lung CT scans from 1010 lung patients with manual lesion segmentations from four experts. We process and use the data in the exact same way as [19], see Appx. C, i.e. the models are fed lesion-centered 2D crops of size 128×128 for which at least one grader segmented a lesion, resulting in 8882 images in the training set, 1996 images in the validation set and 1992 images in the test set.

The LIDC results are reported in Table 1a. The HPU-Net performs better in terms of the Hungarian-matched IoU (and in terms of $\text{GED}^2 = 0.27 \pm 0.01$), while showing a largely improved reconstruction fidelity, that amounts to a near perfect posterior reconstruction of $\text{IoU}_{\text{rec}} = 0.97$. Retraining the sPU-Net with an identical training set-up as in [19], we obtain an unsatisfactorily low value of 0.75 for the foreground-restricted reconstruction IoU (IoU_{rec}) and recapture [19]'s GED^2 of 0.29 (re-implementation: $\text{GED}^2 = 0.32 \pm 0.03$). We additionally evaluate the models on the test subset of samples for which all 4 graders agree on the presence of an abnormality ('subset B', see Appx. C), exposing the HPU-Net's significantly improved ability to capture shape variations (see also Appx. E).

The HPU-Net's capacity to faithfully learn segmentation distributions with high reconstruction and sample fidelity is also qualitatively evident. Fig. 2 compares samples from both models given a pair of CT scans of prospective lung abnormalities. The hierarchical model exhibits enhanced local segmentation structure. Its samples reflect the difficulty to pin-down the boundary of normal vs. abnormal tissue from the image alone (Fig. 2a) and also whether or not the salient structure is abnormal. The sPU-Net's samples on the other hand appear much coarser and 'blobby' (Fig. 2b). In order to explore how the model leverages the hierarchical latent space decomposition, we can use the predicted means $\boldsymbol{\mu}_i^{\text{prior}}$ for some scales instead of sampling. Fig. 3a shows samples for the given CT scans resulting from the process of sampling from the full hierarchy, i.e. from 4 scales in this case. Fig. 3b,c show the resulting samples when sampling from the most global or most local scale only. The hierarchical latent space appears to induce the anticipated bias: the global scales determine the coarse structure, which in this case includes the decision on whether or not the structure at hand is abnormal, while the more local scales fill in appropriate local annotations.
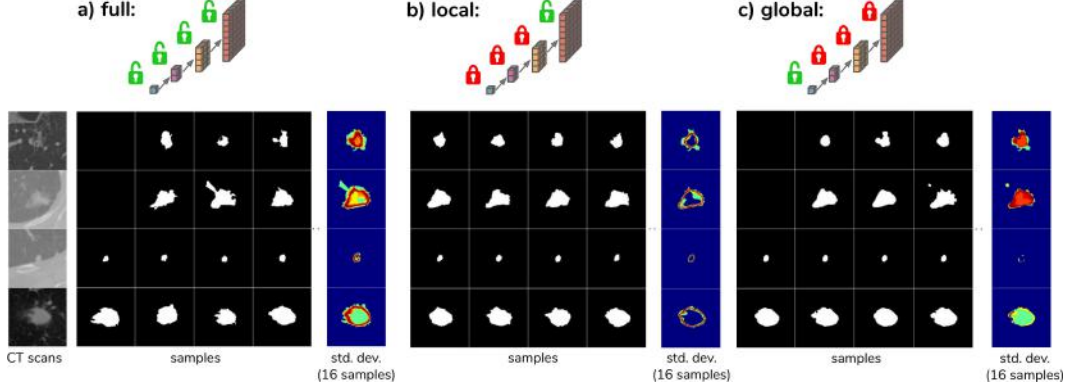
5

Figure 3: HPU-Net samples and standard deviations across 16 samples given the CT scans on the left. Sampling from (**a**) the full hierarchy, (**b**) from only the most local latent scale and (**c**) from only the most global scale while fixing the respectively remaining scales to their predicted means $\boldsymbol{\mu}_i^{\text{prior}}$. Observe in the standard deviations how the local latents alter fine details, mostly at the boundaries, while the global latents can flick the presence of coarser abnormality segmentations on and off.
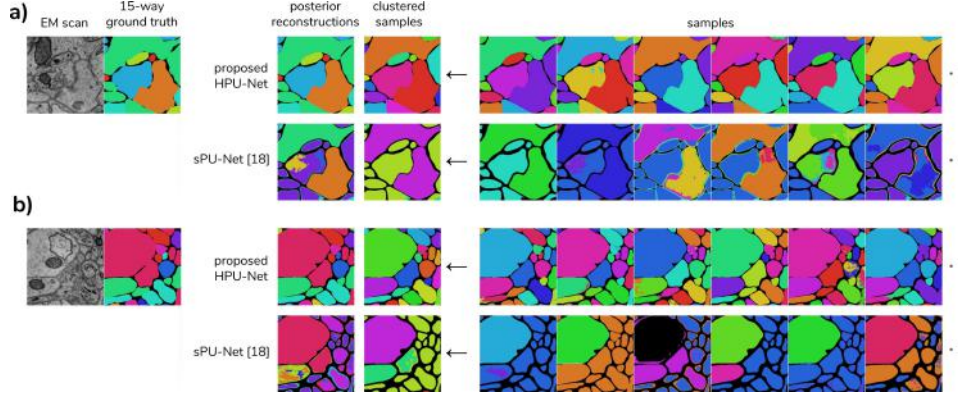

Figure 4: Instance segmentation of neurons. From left to right: EM images from SNEMI3D, the ground-truth mapped to 15 random instance ids, the corresponding posterior reconstructions, predicted instance segmentation after clustering as well as 6 samples. Color denotes instance id (one of 15) and background is shown in black. For more examples see Fig. 11 and 12 in the appendix.

3.3 SNEMI3D: Generative Instance Segmentation of Neurites

As a second dataset we use the SNEMI3D challenge dataset that is comprised of a fully annotated 3D block of a sub-volume of mouse neocortex, imaged slice by slice with an electron microscope [33]. We crop 2D patches of size 256×256 resulting in 1280 images for training, 160 for validation and 160 for testing (for more details see Appx. C). During training we randomly map the instance ids of the cells to one of 15 labels. Because the number of individual cells per image can surmount this number, the training task does not necessitate a unique predicted instance id for every cell, which is why we aggregate a number of samples for a given image to obtain a final instance segmentation. For this purpose we employ a greedy Hamming distance based clustering across 16 samples followed by a light-weight post-processing, detailed in Algorithm 1 and Appx. H.

The HPU-Net, in this case using four latent scales, displays both a strong reconstruction fidelity, $\text{IoU}_{\text{rec}} = 0.60$, as well as a very low Rand error $= 0.06$. Although we want to caution against a direct comparison between results obtained on our smaller test set (in 2D) against those from the official test set (in 3D), it is interesting to put an eye on the official leader-board, where the best dedicated algorithms reach a Rand Error of ~ 0.025 (e.g. [37]) and the human baseline achieved a value of 0.059[2]. For the sPU-Net the parameter settings used in [19] (on other datasets) did not

---

[2]http://brainiac2.mit.edu/SNEMI3D/leaders-board

6

produce satisfactory results. Even when matching the number of global latents of a 4-scale HPU-Net ($\sum_{i=0}^{3} 2^{2i} = 85$), the sPU-Net struggles with reconstructing instance segmentations of neurites and likewise scores badly in terms of the Rand Error, see Table 1c).

From Fig. 4 it is evident that the HPU-Net is able to sample coherent instance segmentations of these amorphous structures with largely varying size and shape, resulting in faithful instance segmentations when clustered across samples. In contrast, the sPU-Net has a hard time accommodating for the independently varying instances and also fails to coherently segment individual instances which is apparent in its samples, the clustering thereof and its reconstructions.

Extrapolation Task In order to further explore the expressiveness of the proposed generative model, we train it to generate extrapolated segmentations given masked images. The masked parts are maximally ambiguous and sensible ways of extrapolating need to be inferred from the unmasked regions. Samples and reconstructions are shown in Fig. 5. To be able to visualize the extrapolations across samples we feed in both the image and the ground-truth segmentation of the unmasked region to the prior, so that it can fix the found instance ids (which is not required for this to work). We observe that the model's generative structure can produce convincing extrapolations, note how the model preserves scale and appearance of unmasked instances, e.g. large cells are more likely to cover larger areas in the masked region and slim cells remain slim and elongated, see third row of samples.
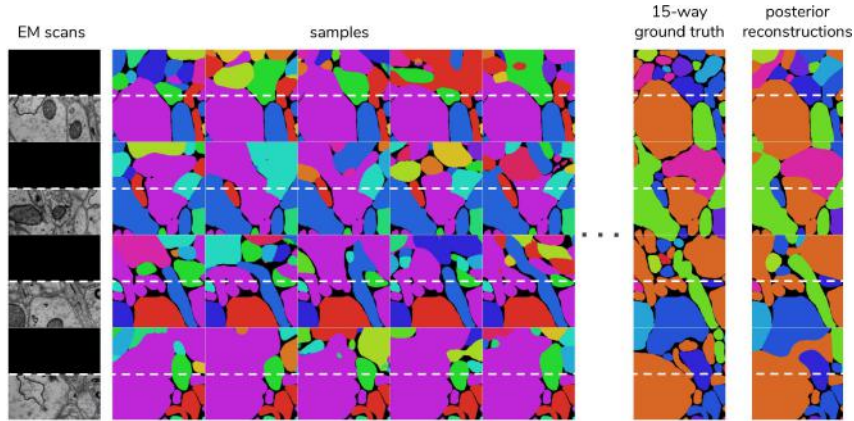

Figure 5: Generative extrapolation on masked EM images with the HPU-Net. Areas above the dashed line in each row correspond to the masked part. Colors denote instance ids (one of 15) with black for background segmentation.

3.4 Cityscapes Cars: Generative Instance Segmentation of Cars

In order to test our model's ability to coherently flip independent regions on natural images, we evaluate it on the task of segmenting car instances on Cityscapes. We train our model to segment all 19 Cityscapes classes while introducing additional alternative car classes that are randomly flipped during training. We run on half-resolution, i.e. 512×1024 (more details in Appx. C and D). At test time we cluster 32 samples per image (see Appx. G). Results on the official validation set (our held-out test set) are reported in Table 1d). Employing 4 latent scales (with the highest latent resolution at 16×32), we reach an $\text{IoU}_{\text{rec}} = 0.62$, a Rand error $= 0.12$ and a $\text{AP}_{50} = 46.8$, without ensembling or any other test-time augmentations. While these results are not competitive with top-performing bounding box regressors such as Mask R-CNN [38] ($\text{AP}_{50} = 68.3$ on the Cityscapes test set), which are tailored towards instance segmentation of boxy objects, we observe an arguably solid out-of-the-box performance of the HPU-Net. Direct comparison may further suffer from the post-hoc computation of object-level confidence scores which we are required to carry out for the AP-metric and which bounding-box regressors on the other hand can optimize for during training.

Fig. 6 shows predicted and ground truth instance segmentations for five scenes. This task is difficult as aside from varying factors such as appearance and illumination, cars nestled along the road can be heavily occluded and individual cars can cover anything between tiny to large regions of the image. Nonetheless our proposed model can sample individual instance segmentations with good coherence,
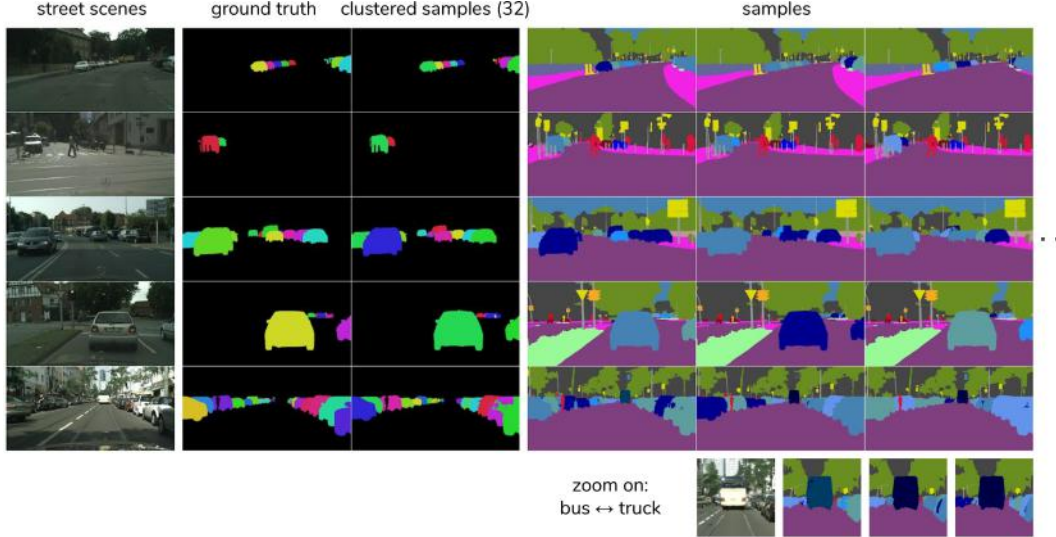
Figure 6: Generative instance segmentation of cars on our Cityscapes test set on 512×1024 resolution with the HPU-Net. Last row that shows crops zoomed in on samples from above. More examples can be found in Fig. 13 and 14.

resulting in strong instance segmentations. Interestingly the model also picks up on ambiguity that is naturally present in the data, e.g. the samples in the first row show coherent flips between parts of the *road* and *sidewalk* and the last row shows coherent flips between *bus* and *truck* annotation for the bus at the end of the road (for which we provide zoomed crops of size 200×200 in Fig. 6). Fig. 15 shows samples and the standard deviations when sampling from only the most local versus only the most global latent scales. It is apparent that the local latents affect small and distant cars while the global latents control more global factors such as cars close to the observer. This shows that also on this large scale natural image data, the model has learned to separate scales.

4 Discussion

Targeted at the segmentation of ambiguous medical scans, prior work (the sPU-Net [19]) learns an image-global distribution that allows to sample consistent segmentation hypotheses. As we show here, this model however suffers from poor sample and reconstruction fidelity and breaks down altogether in more complex scenarios such as instance segmentation. This work proposes a much improved model, the HPU-Net, which shows clear quantitative and qualitative evidence for its advantages over the prior art. Our proposed model uses a much more flexible generative model and further profits from advances such as improved training procedures for VAEs and efficient hard-negative mining (which we ablate in Appx. F). The hierarchical latent space formulation enables to model ambiguities at all scales and affords the learning of complex output interdependencies such as e.g. coherent regions of pixels as found in the task of instance segmentation.

In addition to presenting high-quality results on the segmentation of ambiguous lung CT scans, we achieve strong out of the box performance in instance segmentation of both neurobiological images as well as natural images of street scenes, showing the flexibility and amenability of the proposed model to such tasks. While state-of-the-art deterministic bounding-box regressors [38, 39] still perform significantly better on car instance segmentation, they are predominantly based on a pixel-wise refinement of bounding-boxes and are not designed for overlapping or intertwined instances as found in neurobiological instances. Our generative approach could be a way to directly perform dense object-level segmentation, which has recently attracted attention [40–44].

The HPU-Net's samples are indicative of model uncertainty for ambiguous cases that it has seen during training, which is expected to benefit prospective down-stream tasks. As such the expressed model uncertainty is valid within the data distribution only and, like many others, the model is not aware if and when it fails out-of-distribution [45]. Aside from allowing to capture multiple scales

of variations simultaneously, the latent hierarchy further imposes an inductive bias that mirrors the structure of many medical imaging problems, in which global information can affect top-down decision making, i.e. local annotations in our case. We show this trait in our lung CT scan experiments, where the model learns to separate variations at different scales. Here our model automatically opts to take the decision as to whether the given structure may be abnormal at its most global scale, while reserving more local decisions for local latents, see Fig. 3. A similar decomposition is apparent on natural images (Fig. 15). In terms of KL cost, it is more expensive to model global aspects locally, which in combination with the hierarchical model formulation itself, is the mechanism that puts into effect the separation of scales. Disentangled representations are regarded highly desirable across the board and the proposed model may thus also be interesting for other down-stream applications or image-to-image translation tasks.

In the medical domain the HPU-Net could be applied in interactive clinical scenarios where a clinician could either pick from a set of likely segmentation hypotheses or may interact with its flexible latent space to quickly obtain the desired results. The model's ability to faithfully extrapolate conditioned on prior observations could further be employed in spatio-temporal predictions, such as e.g. predicting tumor therapy response.

Acknowledgments

References

[1] Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv:1511.02680 (2015)

[2] Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems. (2017) 5580–5590

[3] De Fauw, J., Ledsam, J.R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., Askham, H., Glorot, X., O'Donoghue, B., Visentin, D., et al.: Clinically applicable deep learning for diagnosis and referral in retinal disease. Nature medicine **24**(9) (2018) 1342

[4] Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems. (2017) 6405–6416

[5] Batra, D., Yadollahpour, P., Guzman-Rivera, A., Shakhnarovich, G.: Diverse m-best solutions in markov random fields. In: European Conference on Computer Vision, Springer (2012) 1–16

[6] Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., Batra, D.: Why m heads are better than one: Training a diverse ensemble of deep networks. arXiv preprint arXiv:1511.06314 (2015)

[7] Lee, S., Prakash, S.P.S., Cogswell, M., Ranjan, V., Crandall, D., Batra, D.: Stochastic multiple choice learning for training diverse deep ensembles. In: Advances in Neural Information Processing Systems. (2016) 2119–2127

[8] Rupprecht, C., Laina, I., DiPietro, R., Baust, M., Tombari, F., Navab, N., Hager, G.D.: Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In: International Conference on Computer Vision (ICCV). (2017)

[9] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint (2017)

[10] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. (2017) 2223–2232

[11] Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems. (2017) 465–476

[12] Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems. (2017) 700–708

[13] Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. arXiv preprint arXiv:1903.07291 (2019)

[14] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the 2nd international conference on Learning Representations (ICLR). (2013)

[15] Jimenez Rezende, D., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on Machine Learning (ICML). (2014)

[16] Kingma, D.P., Jimenez Rezende, D., Mohamed, S., Welling, M.: Semi-supervised learning with deep generative models. In: Neural Information Processing Systems (NIPS). (2014)

[17] Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Advances in neural information processing systems. (2015) 3483–3491

[18] Esser, P., Sutter, E., Ommer, B.: A variational u-net for conditional appearance and shape generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 8857–8866

[19] Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J.R., Maier-Hein, K., Eslami, S.A., Rezende, D.J., Ronneberger, O.: A probabilistic u-net for segmentation of ambiguous images. In: Advances in Neural Information Processing Systems. (2018) 6965–6975

[20] Gregor, K., Besse, F., Rezende, D.J., Danihelka, I., Wierstra, D.: Towards conceptual compression. In: Advances In Neural Information Processing Systems. (2016) 3549–3557

[21] Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. In: Advances in neural information processing systems. (2016) 3738–3746

[22] Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M.: Improving variational inference with inverse autoregressive flow.(nips), 2016. URL http://arxiv. org/abs/1606.04934

[23] Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. arXiv preprint arXiv:1701.05517 (2017)

[24] Menick, J., Kalchbrenner, N.: Generating high fidelity images with subscale pixel networks and multidimensional upscaling. arXiv preprint arXiv:1812.01608 (2018)

[25] Maaløe, L., Fraccaro, M., Liévin, V., Winther, O.: Biva: A very deep hierarchy of latent variables for generative modeling. arXiv preprint arXiv:1902.02102 (2019)

[26] De Fauw, J., Dieleman, S., Simonyan, K.: Hierarchical autoregressive image models with auxiliary decoders. arXiv preprint arXiv:1903.04933 (2019)

[27] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2015. Volume 9351 of LNCS., Springer (2015) 234–241

[28] He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision, Springer (2016) 630–645

[29] Rezende, D.J., Viola, F.: Taming vaes. arXiv preprint arXiv:1810.00597 (2018)

[30] Wu, Z., Shen, C., Hengel, A.v.d.: Bridging category-level and instance-level semantic image segmentation. arXiv preprint arXiv:1605.06885 (2016)

[31] Nikolov, S., Blackwell, S., Mendes, R., De Fauw, J., Meyer, C., Hughes, C., Askham, H., Romera-Paredes, B., Karthikesalingam, A., Chu, C., et al.: Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy. arXiv preprint arXiv:1809.04430 (2018)

[32] Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)

[33] Kasthuri, N., Hayworth, K.J., Berger, D.R., Schalek, R.L., Conchello, J.A., Knowles-Barley, S., Lee, D., Vázquez-Reina, A., Kaynig, V., Jones, T.R., et al.: Saturated reconstruction of a volume of neocortex. Cell **162**(3) (2015) 648–661

[34] Armato, I., Samuel, G., McLennan, G., Bidaut, L., McNitt-Gray, M.F., Meyer, C.R., Reeves, A.P., Clarke, L.P.: Data from lidc-idri. the cancer imaging archive. http://doi.org/10.7937/K9/TCIA.2015.LO9QL9SX (2015)

[35] Armato, S.G., McLennan, G., Bidaut, L., McNitt-Gray, M.F., Meyer, C.R., Reeves, A.P., Zhao, B., Aberle, D.R., Henschke, C.I., Hoffman, E.A., et al.: The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. Medical physics **38**(2) (2011) 915–931

[36] Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al.: The cancer imaging archive (tcia): maintaining and operating a public information repository. Journal of digital imaging **26**(6) (2013) 1045–1057

[37] Lee, K., Zung, J., Li, P., Jain, V., Seung, H.S.: Superhuman accuracy on the snemi3d connectomics challenge. arXiv preprint arXiv:1706.00120 (2017)

[38] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. (2017) 2961–2969

[39] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. (2017) 2980–2988

[40] Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. arXiv preprint arXiv:1801.00868 (2018)

[41] Kulikov, V., Yurchenko, V., Lempitsky, V.: Instance segmentation by deep coloring. arXiv preprint arXiv:1807.10007 (2018)

[42] Kulikov, V., Lempitsky, V.: Instance segmentation of biological images using harmonic embeddings. arXiv preprint arXiv:1904.05257 (2019)

[43] Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet: A unified panoptic segmentation network. arXiv preprint arXiv:1901.03784 (2019)

[44] Chen, X., Girshick, R., He, K., Dollár, P.: Tensormask: A foundation for dense object segmentation. arXiv preprint arXiv:1903.12174 (2019)

[45] Nalisnick, E., Matsukawa, A., Teh, Y.W., Gorur, D., Lakshminarayanan, B.: Do deep generative models know what they don't know? arXiv preprint arXiv:1810.09136 (2018)

[46] Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics (NRL) **2**(1-2) (1955) 83–97

[47] Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the society for industrial and applied mathematics **5**(1) (1957) 32–38

[48] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 3213–3223

[49] Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association **66**(336) (1971) 846–850

[50] Arganda-Carreras, I., Turaga, S.C., Berger, D.R., Cireşan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J.M., et al.: Crowdsourcing the creation of image segmentation algorithms for connectomics. Frontiers in neuroanatomy **9** (2015) 142

[51] Nunez-Iglesias, J., Kennedy, R., Plaza, S.M., Chakraborty, A., Katz, W.T.: Graph-based active learning of agglomeration (gala): a python library to segment 2d and 3d neuroimages. Frontiers in neuroinformatics **8** (2014) 34

Appendices

A KL-Divergence between Posterior and Prior

The Kullback-Leibler terms in Eq. 5 and 6 come about as follows:

$$D_{\mathrm{KL}}(Q||P) = \mathbb{E}_{\mathbf{z} \sim Q}\left[\log Q - \log P\right] \tag{7}$$

$$= \int_{\mathbf{z}_0,...,\mathbf{z}_L} \prod_{j=0}^{L} q(\mathbf{z}_j|\mathbf{z}_{<j}) \sum_{i=0}^{L} \left[\log q(\mathbf{z}_i|\mathbf{z}_{<i}) - \log p(\mathbf{z}_i|\mathbf{z}_{<i})\right] d\mathbf{z}_0...d\mathbf{z}_L, \tag{8}$$

$$\text{using} \int \phi(\mathbf{z}_i) \prod_{j=0}^{L} q(\mathbf{z}_j|\mathbf{z}_{<j}) d\mathbf{z}_0...d\mathbf{z}_L = \int \phi(\mathbf{z}_i) \prod_{j=0}^{i} q(\mathbf{z}_j|\mathbf{z}_{<j}) d\mathbf{z}_0...d\mathbf{z}_i : \tag{9}$$

$$= \sum_{i=0}^{L} \int_{\mathbf{z}_0,...,\mathbf{z}_i} \prod_{j=0}^{i} q(\mathbf{z}_j|\mathbf{z}_{<j}) \left[\log q(\mathbf{z}_i|\mathbf{z}_{<i}) - \log p(\mathbf{z}_i|\mathbf{z}_{<i})\right] d\mathbf{z}_0...d\mathbf{z}_i \tag{10}$$

$$= \sum_{i=0}^{L} \int_{\mathbf{z}_0,...,\mathbf{z}_i} \prod_{j=0}^{i-1} q(\mathbf{z}_j|\mathbf{z}_{<j}) \, q(\mathbf{z}_i|\mathbf{z}_{<i}) \left[\log q(\mathbf{z}_i|\mathbf{z}_{<i}) - \log p(\mathbf{z}_i|\mathbf{z}_{<i})\right] d\mathbf{z}_0...d\mathbf{z}_i \tag{11}$$

$$= \sum_{i=0}^{L} \mathbb{E}_{\mathbf{z}_{<i} \sim Q} D_{\mathrm{KL}}(q(\mathbf{z}_i|\mathbf{z}_{<i})||p(\mathbf{z}_i|\mathbf{z}_{<i})), \tag{12}$$

where for improved clarity we omit X and X, Y as conditional arguments to p and q respectively. For brevity our notation additionally subsumes $q(\mathbf{z}_0) := q(\mathbf{z}_0|\mathbf{z}_{-1})$ and similar for $p(\mathbf{z}_0)$. For our choice of posterior and prior distribution (see Eq. 1 and 3) the KL-terms above can be evaluated analytically. The expectations in Eq. 5 and 6 using samples $\mathbf{z} \sim Q$ are performed with a single sampling pass.

B Performance Measures

B.1 Distribution Agreement

We report how well the distribution produced by the respective generative model and the given ground-truth distribution agree on the LIDC dataset. In real world scenarios such as the LIDC dataset, the ground-truth distribution is only known in terms of a set of samples. One way to measure the agreement between two distributions that only requires samples as opposed to knowledge of the full distributions is by means of the **Generalized Energy Distance** (GED^2, also referred to as Maximum Mean Discrepency). This kernel-based metric was employed in [19] using $1 - \mathrm{IoU}(Y, Y')$ as a distance kernel, where IoU is the intersection over union metric between two segmentations. We found this measure inadequate in such cases where the models' samples only poorly match the ground truth samples, since the metric then unduly rewards sample diversity, regardless of the samples' adequacy. As an alternative that appears less vulnerable to such pathological cases, we propose to use the Hungarian algorithm [46, 47] to match samples of the model and the ground-truth. The Hungarian algorithm finds the optimal 1:1-matching between the objects of two sets, for which we use $\mathrm{IoU}(Y, Y')$ to determine the similarity of two samples. We report the match as the **Hungarian-matched IoU**, i.e. the average IoU of all matched pairs and duplicate both sets so that their number of elements matches their least common multiple. As empty segmentations can be valid gradings in the LIDC dataset we need to define how the IoU enters the distribution metrics for the case of correctly predicted absences, which is detailed below.

B.2 Reconstruction Fidelity

The reconstruction fidelity is an upper bound to the fidelity of the conditional samples. In order to asses this upper bound on the fidelity of the produced segmentations we measure how well the model's posteriors are able to reconstruct a given segmentation in terms of the IoU metric, i.e. we report the **reconstruction IoU**, $\mathrm{IoU}_{\mathrm{rec}}(Y, Y')$ where $Y' = S(X, \boldsymbol{\mu}^{\mathrm{post}}(X, Y))$. Whenever we employ the IoU-metric, i.e. also when it enters the measures for distribution agreement, we calculate it with respect to the stochastic foreground classes only. We further do not calculate it globally across all the test set pixels (as is regularly done in semantic segmentation challenges, e.g. in Cityscapes [48]), but calculate it across the pixels of each image and then average across all test set images. For these reasons the question arises how to deal with a correctly predicted absence of a class in an image, a case for which the IoU metric is undefined (the denominator would be 0). For the LIDC dataset, empty ground-truth segmentations can be a valid grading which is why we follow [19] and define a correctly predicted absence as IoU = 1. In the SNEMI3D and Cityscapes instance segmentation tasks we do not want to evaluate whether a model correctly predicts a class' absence, which is why we correct class absences do not enter the mean IoU of an image, while wrongly predicted absences are penalized (in practice we perform

a 'NAN-mean' over the classes of interest). In the Cityscapes case we additionally make use of the provided ignore-masks, keeping unlabeled pixels out of the evaluations.

B.3 Instance Segmentation

In order to score how well the predicted instance segmentations (the instance clusters) agree with the ground truth, we calculate the **Rand Error**. This measure is defined as $1 - F$-score, where the precision and recall values that enter the F-score are determined from whether pixel pairs between the ground truth clustering and a predicted clustering belong to the same segment (positive class) or different segments (negative class) [49, 50]. We use the foreground-restricted version as employed in the SNEMI3D challenge[3].

On Cityscapes instance segmentation we additionally report the **Average Precision** (AP). It is based on object level scoring and defined as the area under the precision recall curve for all predicted object detections. To span the precison recall curve, an object level score that quantifies a model's confidence in the 'objectness' of its prediction is required. For our car instance segmentation experiments we employ the Cityscapes evaluation scheme[4], reporting AP_{50} and AP, the average precision when requiring predictions to match above a thresholded $IoU_{thres} > 0.50$ and when averaging across multiple such thresholds (10 different overlaps ranging from 0.5 to 0.95 in steps of 0.05), respectively. To artificially obtain object-level scores we average the softmax scores of all stochastic classes across samples and pixels of a predicted instance mask [41].

C Dataset Details

LIDC-IDRI The LIDC-IDRI dataset [34–36] contains 1018 lung CT scans from 1010 lung patients with manual lesion segmentations from four experts. We use the same setup as in [19], i.e. we employ the annotations from a second reading and employ the same data splits (722 patients for training, 144 patients for validation and 144 patients for the test set). The data is then cropped to 2D images of shape 180×180 pixels, resulting in 8882 images in the training set, 1996 images in the validation set and 1992 images in the test set. Crops are centered at positions for which at least one grader indicates a lesion. In the LIDC dataset, empty foreground segmentations can be viable expert gradings, which is why we employ $IoU = 1$ when a model sample agrees in such cases. IoU_{rec} is an average of the reconstructions of all four gradings. As in [19] we only use those lesions that were specified as a polygon (outline) in the XML files of the LIDC dataset, disregarding the ones that only have center of shape. That is, according to the LIDC paper only such lesions that are larger than 3mm are used, with smaller ones filtered out as they are regarded clinically less relevant [35]. We filter out each Dicom file whose absolute value of SliceLocation differs from the absolute value of ImagePositionPatient[-1]. Finally we assume that two masks from different graders correspond to the same lesion if their tightest bounding boxes overlap.

LIDC-IDRI subset B For 'Subset B' we consider only those test set cases, which have annotations by all 4 graders, i.e. all graders agree on the presence of an abnormality. This results in 638 images, so close to a third of the full test set.

SNEMI3D As a second dataset we use the SNEMI3D challenge[5] dataset that is comprised of a fully annotated 3D block of a sub-volume of mouse neocortex, imaged slice by slice with an electron microscope [33]. This stack is $1024 \times 1024 \times 100$ voxels large, comes at a voxel size of $6 \times 6 \times 29\,\mathrm{nm}^3$ and contains a total of 400 fully annotated neurite instance annotations. We use the first 80 z-slices as our training dataset, the adjacent 10 slices as a validation set and the remaining 10 slices as a test set to report results on. We crop non-overlapping patches of size $256 \times 256 \times 1$ resulting in 1280 images for training, 160 for validation and 160 for testing. During training we randomly map the instance identifiers (ids) of the cells to one of 15 labels, thereby treating the instance id of the cells as latent information that the networks need to model. Because the number of individual cells per image can surmount this number, the training task does not necessitate a unique predicted instance id for every cell. This means that in order to obtain a predicted instance segmentation at test time, we need to aggregate a number of samples for a given image. For this purpose we employ a greedy Hamming distance based clustering across n samples followed by a light-weight post-processing, detailed in Algorithm 1 of Appx. G and Appx. H (we chose $n = 16$ and Hamming distance threshold $\alpha = 16$).

Cityscapes As a third dataset we use the Cityscapes street scene dataset that comes with both dense category segmentations, as well as with instance segmentations for a number of categories. The official Cityscapes training dataset (with fine annotations) comprises 2975 images. We employ the official validation set of 500 images as a test set to report results on, and split off 274 images (corresponding to the 3 cities of Darmstadt, Mönchengladbach and Ulm) from the official training set as an internal validation set. At test time, we cluster 32 samples per image (see Appx. G), using a threshold of $\alpha = 32$.

---

[3]Available as `adapted_rand_error` in the python package gala [51].

[4]Official evaluation code can be found here.

[5]http://brainiac2.mit.edu/SNEMI3D/home

D   Architecture and Training Details

Architecture The Hierarchical Probabilistic U-Net implementations employed for the different tasks differ in their number of processing scales, the depth of each such scale and the number of latent scales employed. The architecture as employed during the sampling process is depicted in Fig. 1a. The setup for the training process is depicted in Fig. 1b) and includes the separate posterior net that is required during training.

At each processing scale of the HPU-Net we employ a stack of n pre-activated residual blocks [28] (grey blocks in Fig. 1), where n determines the depth of that scale. For both the LIDC and SNEMI3D experiments we use $n = 3$ residual blocks and for the Cityscapes experiment we use $n = 2$ residual blocks at each processing scale of the U-Net's encoder and decoder respectively. Similar to [22, 25], we find the use of unobstructed connections (in our case res-blocks) between latent scales of the hierarchy to be crucial for the lower scales to be employed by the generative model. Without the use of res-blocks the KL-terms between distributions (indicated by green connecting lines in Fig. 1) at the beginning of the hierarchy often become ~ 0 early on in the training, essentially resulting in uninformative and thus unused latents.

In each res-block the residual feature map is calculated by means of a series of three 3×3-convolutions, the first of which always halves the number of the feature maps employed at the present scale, such that the residual representations live on a lower dimensional manifold. At the end of the residual branch a single (un-activated) 1×1-convolution projects the features back to the number of features of the given scale. The resulting residual is then added to the skipped feature map, which is skipped forward (i.e. left untouched) unless the number of feature maps is set to change, in which case it is projected by a 1×1-convolution. This happens only at transitions that change the feature map resolutions. For down-sampling of feature maps we use average pooling and upsample by using nearest neighbour interpolation. As described in Sec. 2, the spatial grid of latent variables is sampled at the end of each U-Net decoder scale that is part of the hierarchy and concatenated to the final feature map produced at this scale, before both are up-sampled.

The number of latent scales is chosen empirically such as to allow for a sufficiently granular effect of the latent hierarchy. For the tasks and image resolutions considered here, we found 3 - 5 latent scales to work well. The number of processing scales is chosen such that a smallest possible spatial resolution is achieved in the bottom of the U-Net. For the square images in LIDC and SNEMI3D this means a resolution of 1×1 and for the Cityscapes task the minimum resolution is 1×2 (in this case we however employ 2×4, which is detailed below). The employed separate posterior mirrors the number of scales and the number of feature maps of the corresponding components in the U-Net, see the bottom part of Fig. 1b. Its only architectural difference is its first convolutional layer, which processes the input image concatenated with the corresponding one-hot segmentation along the channel axis. All weights of all models are initialized with orthogonal initialization having the gain (multiplicative factor) set to 1, and the bias terms are initialized by sampling from a truncated normal with $\sigma = 0.001$.

Training The HPU-Net is trained using the GECO-objective (Eq. 6) and a stochastic top-k reconstruction loss. As described in Sec. 2, the kth percentile employed for the top-k objective is fixed across tasks to 2% of each batch's pixels. The GECO-objective aims at matching a reconstruction target value κ. For each experiment we chose κ sufficiently low so as to correspond to a strong reconstruction performance while resulting in a training schedule that is not dominated by the reconstruction term over the entire course of the training (e.g. if κ is chosen too high, the Lagrange multiplier λ, and thus the learning pressure it exerts, mounts and remains on the reconstruction term rather then moving over on the KL terms). The desired behavior of the reconstruction objective $\mathcal{L}_{\text{rec}}$ and the Lagrange multiplier λ can be observed in Fig. 7 and Fig. 8, where λ rises until $\mathcal{L}_{\text{rec}}$ matches κ, after which λ drops and the pressure on the KL-terms increases.

In contrast to the regular cross-entropy employed in semantic segmentation, the reconstruction error here is not averaged but summed over individual pixels (before being averaged across batch instances). This is because the likelihood is assumed to factorize over the pixels of an image and so their log-likelihood is summed over. For comparability we however report $\mathcal{L}_{\text{rec}}$ and κ per pixel (e.g. in Fig. 7, Fig. 8 and in Table 2).

The precise training setups for each of the tasks and models are reported below. Note that the training objectives for all models encompass an additional weight-decay term that is weighted by a factor of $1e^{-5}$.

D.1   LIDC-IDRI Lung CT scans

During training on LIDC, image-grader pairs are drawn randomly. Similar to what was done in [19], we apply random augmentations[6] to the image and label tiles (180×180 pixels size) including random elastic deformation, rotation, mirroring, shearing, scaling and a randomly translated crop that results in a tile size of 128×128 pixels. Any padding added to the images and labels during the augmentation process is masked from the loss during training.

---

[6]We use the code available at https://github.com/deepmind/multidim-image-augmentation/.

In order to evaluate the Probabilistic U-Net on additional metrics than those employed in [19], we retrain a re-implementation of the model with the exact same hyperparameters and setup as in [19], i.e we employ a 5-scale model, with three 3×3-convolutions per encoder and decoder-scale, a separate prior and posterior net that mirror the used U-Net's encoder as well as 6 global latents and three final 1×1 convolutions. Moreover we employ an identical ELBO-formulation ($\beta = 1$), train with identical batch-size of 32, number of iterations ($240k$) and learning rate schedule $0.5e^{-5} \rightarrow 1e^{-6}$.

On LIDC, the HPU-Net uses 8 latent scales resulting in a global 1×1-'U-Net bottom' and 3 res-blocks per encoder and decoder scale. The base number of channels is 24 and until the fourth down-sampling the number of channels is doubled after each down-sampling operation, resulting in a maximum width of 192 channels. The U-Net's decoder mirrors this setup. We train the HPU-Net with an initial learning rate of $1e^{-4}$ that is lowered to $0.5e^{-5}$ in 4 steps over the course of $240k$ iterations. The employed batch-size is 32. The HPU-Net is trained with the GECO-objective using $\kappa = 0.05$.
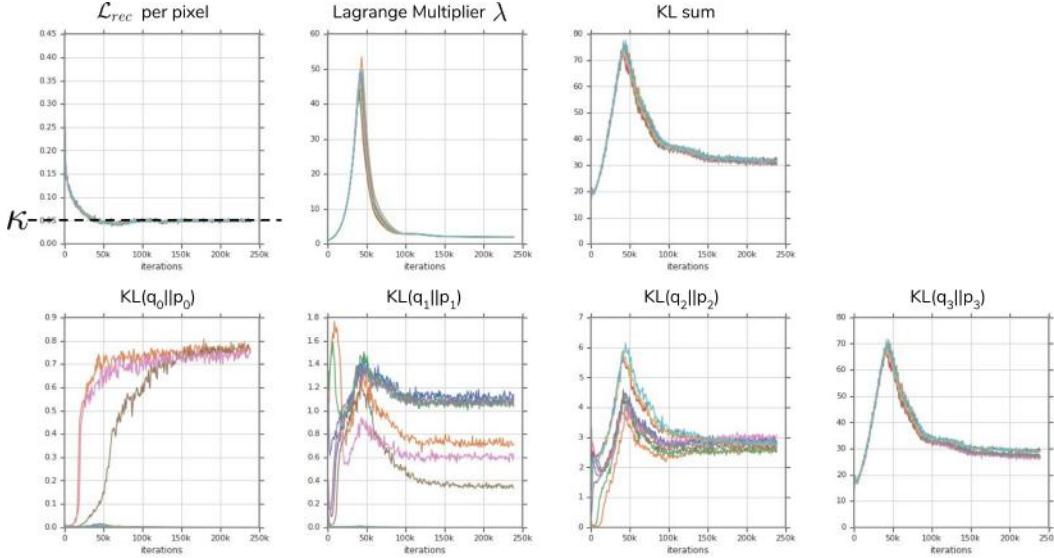

Figure 7: Components of the learning objective in the course of the LIDC training for 10 random initializations.

Fig. 7 shows how the top-k reconstruction term $\mathcal{L}_{\text{rec}}$, the Lagrange multiplier λ, as well as the individual KL-terms (and their sum) progress in the course of training for the 10 random model initializations reported in Table 1. As mentioned above, GECO structures the dynamics such that λ puts pressure on $\mathcal{L}_{\text{rec}}$ until it reaches its target value κ. After that the training objective holds the reconstruction term at κ while optimizing for lower overall Kullback-Leibler divergence ('KL'). The KL is a measure for how much more information the posterior distribution carries compared to the prior, a quantity that we aim to minimize. Note that the KL-sum is very similar for all models, but the way the KL splits across the hierarchy can differ. The models that end up using the global latents profit from a slightly lower overall KL indicating that this decomposition is more efficient, e.g. it is more efficient not to repeat global information in the local latents when it is already provided by global latents etc.

D.2 SNEMI3D neocortex EM slices

During training on SNEMI3D we randomly sample a latent (class) id for each cell in each image. We limit the number of instance ids to 15 and just like on LIDC we apply random augmentations including random elastic deformation, rotation, mirroring, shearing, scaling and a randomly translated crop. Any padding added to the images and labels during the augmentation process is masked from the loss during training.

For the standard Probabilistic U-Net we employ a 9 scale architecture and a base number of 24 channels, that until the 4th down-sampling, is doubled after each down-sampling operation, resulting in a maximum width of 192 channels. The sPU-Net again uses three 3×3-convolutions per encoder and decoder scale, while the HPU-Net employs three res-blocks. The HPU-Net also employs 32 base channels, a total of 9 scales interleaved with four (scalar) latent scales, resulting in a total of 85 latents. This is also the number of global latents that we used for the sPU-Net, since employing low numbers of latents, such as ~ 10 as proposed in [19] never converged (even working with 85 global latents does not make for a very stable training). Both models are trained for $450k$

iterations with a batch-size of 24, and an initial learning rate of $1e^{-4}$ that is lowered to $1e^{-7}$ in 5 steps. The HPU-Net is trained with the GECO-objective using $\kappa = 1.20$.
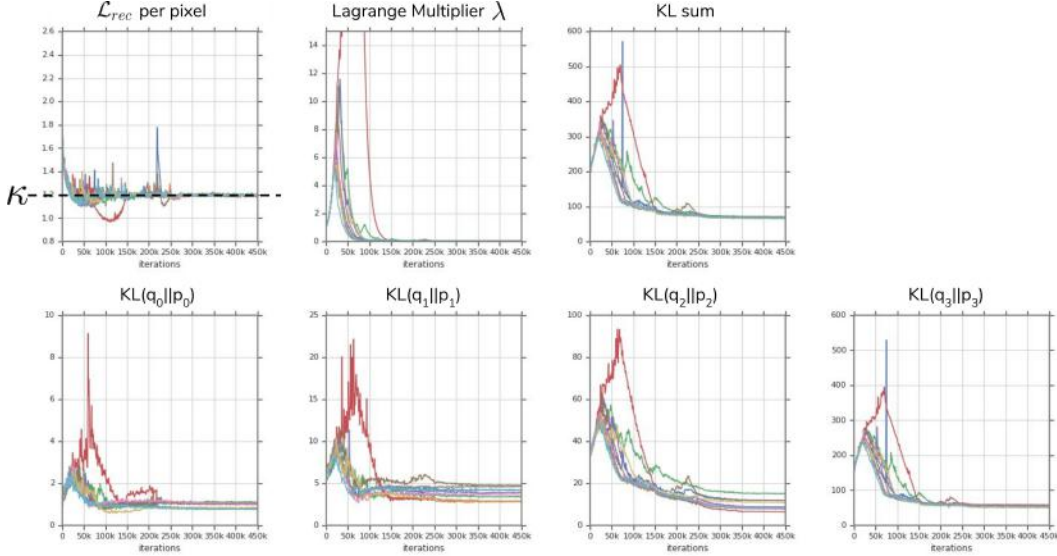

Figure 8: Components of the learning objective in the course of the SNEMI3D training for 10 random initializations.

Fig. 8 again shows how the top-k reconstruction term $\mathcal{L}_{rec}$, the Lagrange multiplier λ, as well as the individual KL-terms (and their sum) progress in the course of training for the 10 random model initializations reported in Table 1. Again the KL sums to a similarly low value across models with different decompositions across the four scales.

D.3 Cityscapes Car Instances

We resample the Cityscapes images and labels to half-resolution, i.e. 512×1024. During training we randomly sample a (latent) instance id for each car in the image, where we limit the total number of car ids to 5. We apply random deformations including random color augmentations, elastic deformation, rotation, mirroring, shearing, scaling and a randomly translated crop. Any padding added to the images and labels during the augmentation process is masked from the loss during training alongside any such pixels that are marked as part of the 'ignore'-class in the dataset (pixels that can't be attributed to one of the provided 19 classes).

We train a HPU-Net with 9 scales, resulting in a 2×4-'U-Net bottom' and 4 latent scales. Using another scale (so 5 latent scales and a number of 10 overall scales) did not significantly change the results and due to the image aspect ratio of 1:2, does not result in a fully global latent scale either. The employed model uses two res-blocks for each encoder and decoder scale and we train the model with a batch-size of 128 for $100k$ iterations using TPU accelerators and spatial batch partitioning. We use an initial learning rate of $2e^{-4}$ that is halved after $70k$ iterations. The base number of channels is 32 and until the fourth down-sampling the number of channels are doubled after each down-sampling operation, resulting in a maximum width of 256 channels. The HPU-Net is trained with the GECO-objective using $\kappa = 0.77$.

E GED^2 on LIDC subset B

On 'Subset B' the sPU-Net gets a GED$^2 = 0.52 \pm 0.09$ while the HPU-Net achieves as GED$^2 = 0.38 \pm 0.02$. Both values result from the set of 10 models used for the LIDC results in Table 1 (again using 1000 bootstraps with replacement).

F Ablation Study

In order to show the effect of some of the main choices we made for the model and the loss formulation, we perform an ablation study on the LIDC lung abnormalities segmentation task. All models are trained with the same training setup and hyper parameters as used in the LIDC experiments (described in Appx. D), if not stated differently in the following. First we evaluate the importance of the latent hierarchy. We train 10 random

initializations for a model with a global latent scale in the 'U-Net's bottom' that otherwise employs the same model topology as the HPU-Net that we employ on LIDC. For this model we use 85 global latents, i.e. the same number of total latents that the 4-scale hierarchical model employs. In order to arrive at a comparable reconstruction IoU, we found it necessary to raise the reconstruction target κ above the value of 0.05 (employed for the other models) to a value of $\kappa = 0.15$. As reported in Table 2, this model performs significantly worse than the HPU-Net in terms of both GED^2 and the Hungarian-matched IoU, while also suffering from a loss in reconstruction fidelity. As a second model configuration we consider a model with the same topology as the employed HPU-Net, however employing only its most local scale of latents (a spatial grid of size 8×8). The idea is to assess to what degree the latents lower in the hierarchy help coordinate the sampling from the last, most finely resolved grid of latents. The results in Table 2 show another significant decrease in the model's ability to match the ground truth distribution, suggesting that the hierarchy indeed is an important model choice enabling the strong performance in terms of GED^2 and the Hungarian-matched IoU. Lastly we quantify the effect of employing a top-k loss for the hierarchical model. The last row in Table 2 shows the positive effect that the top-k loss formulation has on the reconstruction IoU (IoU_{rec}), while allowing to keep the same level of distribution match (there is a slight increase in Hungarian-matched IoU when ablating the top-k loss, it is however insignificant across 10 random initializations).

Table 2: Ablation study on LIDC-IDRI. All results are reported on our test set and the given means and standard deviations are taken across 10 random initializations of the same respective model setup and 1000 bootstraps with replacement each. The values reported for κ are normalized per pixel and for comparison the LIDC results reported in Table 1 are shown in the first row of this table.

model + loss formulation	IoU_{rec}	GED^2	Hungarian-matched IoU
4-scale hierarchy + GECO ($\kappa = 0.05$) + top-k (k=0.02)	0.97 ± 0.00	0.27 ± 0.01	0.53 ± 0.01
local latents + GECO ($\kappa = 0.05$) + top-k (k=0.02)	0.97 ± 0.00	0.34 ± 0.01	0.45 ± 0.01
global latents + GECO ($\kappa = 0.15$) + top-k (k=0.02)	0.94 ± 0.02	0.40 ± 0.02	0.37 ± 0.02
4-scale hierarchy + GECO ($\kappa = 0.05$)	0.94 ± 0.00	0.27 ± 0.01	0.54 ± 0.01

G   Hamming Distance based Greedy Clustering

Result: Instance Segmentation $\mathbf{I} \in \mathbb{Z}^{H \times W}$.
Parameters: n: number of samples, α: threshold.

1 Retrieve n sample segmentations $\mathbf{Y}_i^{\mathrm{prob}} \in \mathbb{R}^{H \times W \times C}$; $\mathbf{Y}_i^{\mathrm{prob}} \leftarrow S(X, \mathbf{z}_i)$, $\mathbf{z}_i \sim P(.|X)$;
2 Transform samples to one-hot $\mathbf{Y}_i \in \mathbb{Z}^{H \times W \times C}$, $\mathbf{Y}_i \leftarrow \mathrm{one\_hot}(\mathrm{argmax}(\mathbf{Y_i}^{\mathrm{prob}}))$;
3 Concatenate samples over channels $\mathbf{Y} \in \mathbb{Z}^{H \times W \times nC}$; $\mathbf{Y} \leftarrow \mathrm{concat}([\mathbf{Y}_0, ..., \mathbf{Y}_n])$;
4 Initialize Instance Segmentation $\mathbf{I} \in \mathbb{Z}^{H \times W}$; $\mathbf{I} \leftarrow [[-1, \dots], \dots]$;
5 Inititalize set of unassigned pixels $\mathcal{U} = \mathrm{where}(\mathbf{I} == -1)$;
6 Initialize background one-hot vector $\mathbf{b} \in \mathbb{Z}^{C \times 1}$; $\mathbf{b} \leftarrow \mathrm{one\_hot}(\text{background label})$;
7 Initialize protoype $\mathbf{p} \in \mathbb{Z}^{nC \times 1}$ with the prototype of the background class $\mathbf{p} \leftarrow \mathrm{concat}([\mathbf{b}, \mathbf{b}, \dots])$;
8 Inititialize cluster id $c = 0$;
9 **while** $|\mathcal{U}| > 0$ **do**
10   **if** *c==0* **then**
11   Do nothing, as $\mathbf{p}$ is initially assigned to background class prototype;
12   **else**
13   Draw a random pixel from the set of unassigned pixels $i \sim \mathcal{U}$;
14   Use the one-hot sample vector of this pixel as the cth cluster's prototype $\mathbf{p} \leftarrow \mathbf{Y}[i]$;
15        $\mathbf{I}[i] \leftarrow c$;
16   Drop i from set of unassigned pixels $\mathcal{U} \leftarrow \mathcal{U} \setminus \{i\}$;
17   **end**
18   **foreach** $j \in \mathcal{U}$ **do**
19   Retrieve one-hot sample vector of the pixel j as $\boldsymbol{\nu} \leftarrow \mathbf{Y}[j]$;
20   Calculate Hamming distance $d = \mathrm{hamming\_distance}(\boldsymbol{\nu}, \mathbf{p})$;
21   **if** $d \leq \alpha$ **then**
22          $\mathbf{I}[j] \leftarrow c$;
23          $\mathcal{U} \leftarrow \mathcal{U} \setminus \{i\}$;
24   **end**
25   **end**
26     $c \leftarrow c + 1$;
27 **end**

Algorithm 1: Hamming distance based greedy clustering, which makes use of the assumption that pixels of the same object vary together across samples. Pseudo-code used to get instance segmentations from segmentation samples $\mathbf{Y}_i$ for a given image X of size $H \times W$. The employed algorithm assigns pixels to clusters based on the Hamming distance between a cluster's prototype and the pixel's vector representation. The Hamming distance is a simple count of element-wise mismatches. Both vectors consist of the respective pixels' sampled class labels in one-hot form, i.e. for n samples and C classes, they have length nC. The algorithm proceeds in a greedy manner, i.e. once no more matches satisfying an upper bound on the distance to the current prototype are found, a new prototype is randomly picked from the remaining unassigned pixels. Sampling at random rather than picking the next available pixel minimizes the clustering run-time (which is $\mathcal{O} \leq (HW)^2$) and the likelihood of picking cluster prototypes from object boundaries. The algorithm starts with assigning pixels to a provided background class label. This assures that cluster $c = 0$ always corresponds to the background class, but is not strictly necessary, alternatively the algorithm can omit the case distinction in line 10ff.

H   Instance Segmentation Post-Processing

For the instance segmentation experiments we post-process the clustered samples to remove tiny regions that sometimes appear at segmentation borders. For each cluster (instance) found via Algorithm 1, we check whether it survives an erosion operation with an $n \times n$-structure element. If the given erosion eliminates the cluster, we replace each pixel within the cluster in question by its majority neighbour label. The majority neighbour label is determined from a $m \times m$-box centered at the given pixel. The cluster label that is to be replaced as well as background labels are ignored while finding the majority label. If this results in 0 valid neighbour labels, we keep the current pixel's label in SNEMI3D and use the background label in Cityscapes. In both SNEMI3D and Cityscapes, we chose $n = 5$ and $m = 11$. Painting in the majority label is carried out on the fly.

I   Model Samples, Reconstructions and (on SNEMI3D and Cityscapes) Instance Clusterings
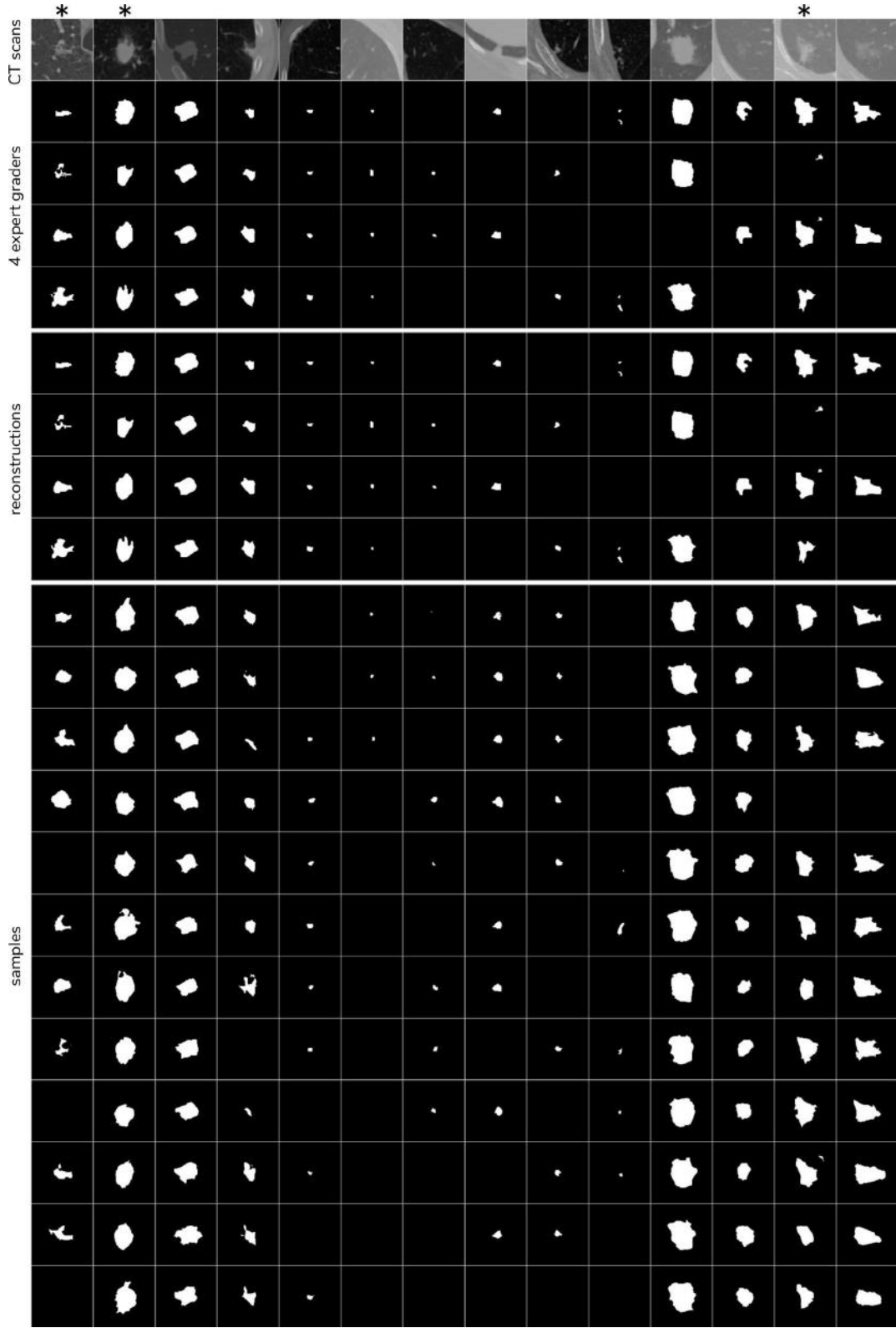
Figure 9: Qualitative results of the Hierarchical Probabilistic U-Net on our LIDC-IDRI test set. An *
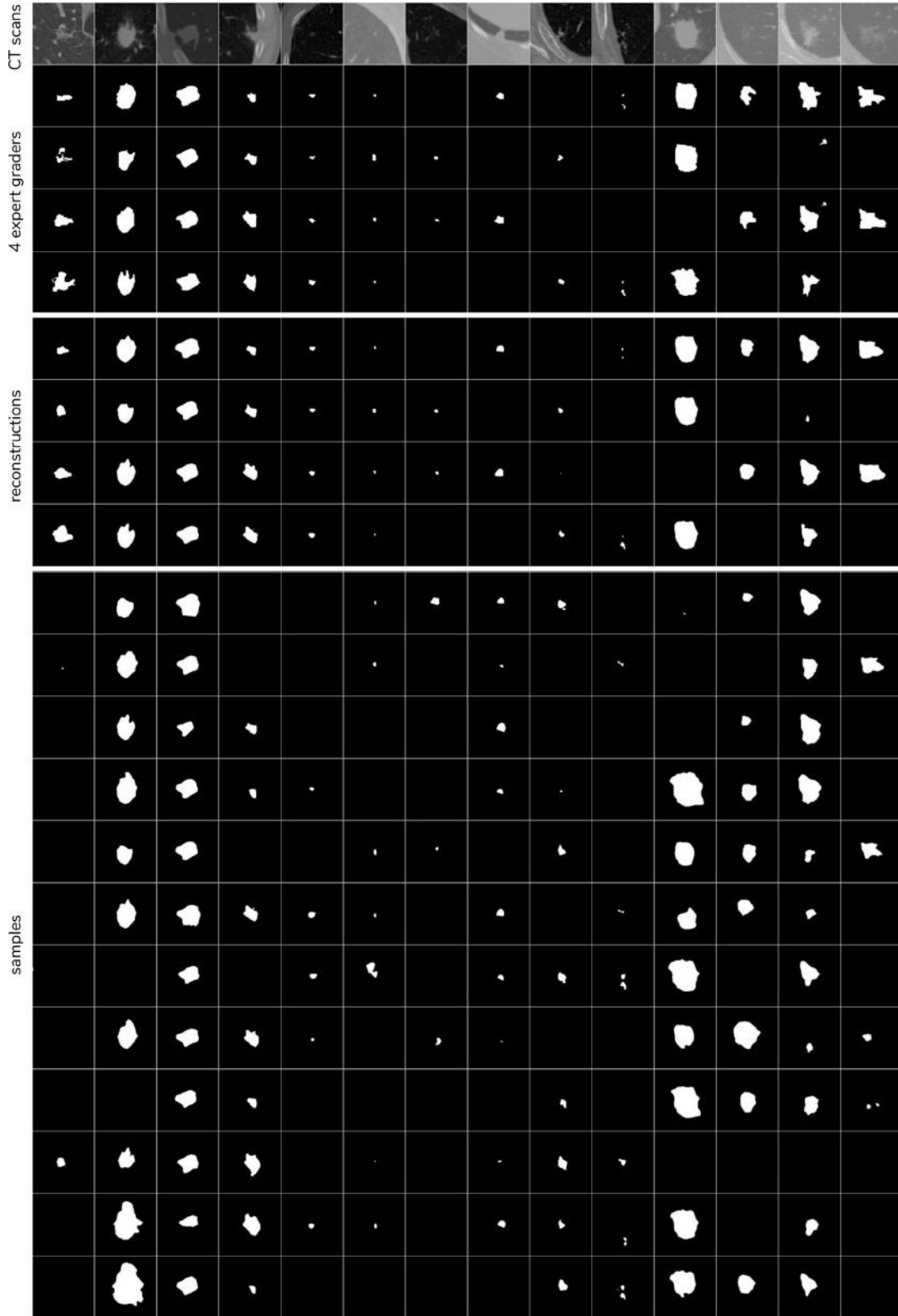denotes cases that we also use in Fig. 3.

Figure 10: Qualitative results of the Standard Probabilistic U-Net on our LIDC-IDRI test set.

EM scans

15-way
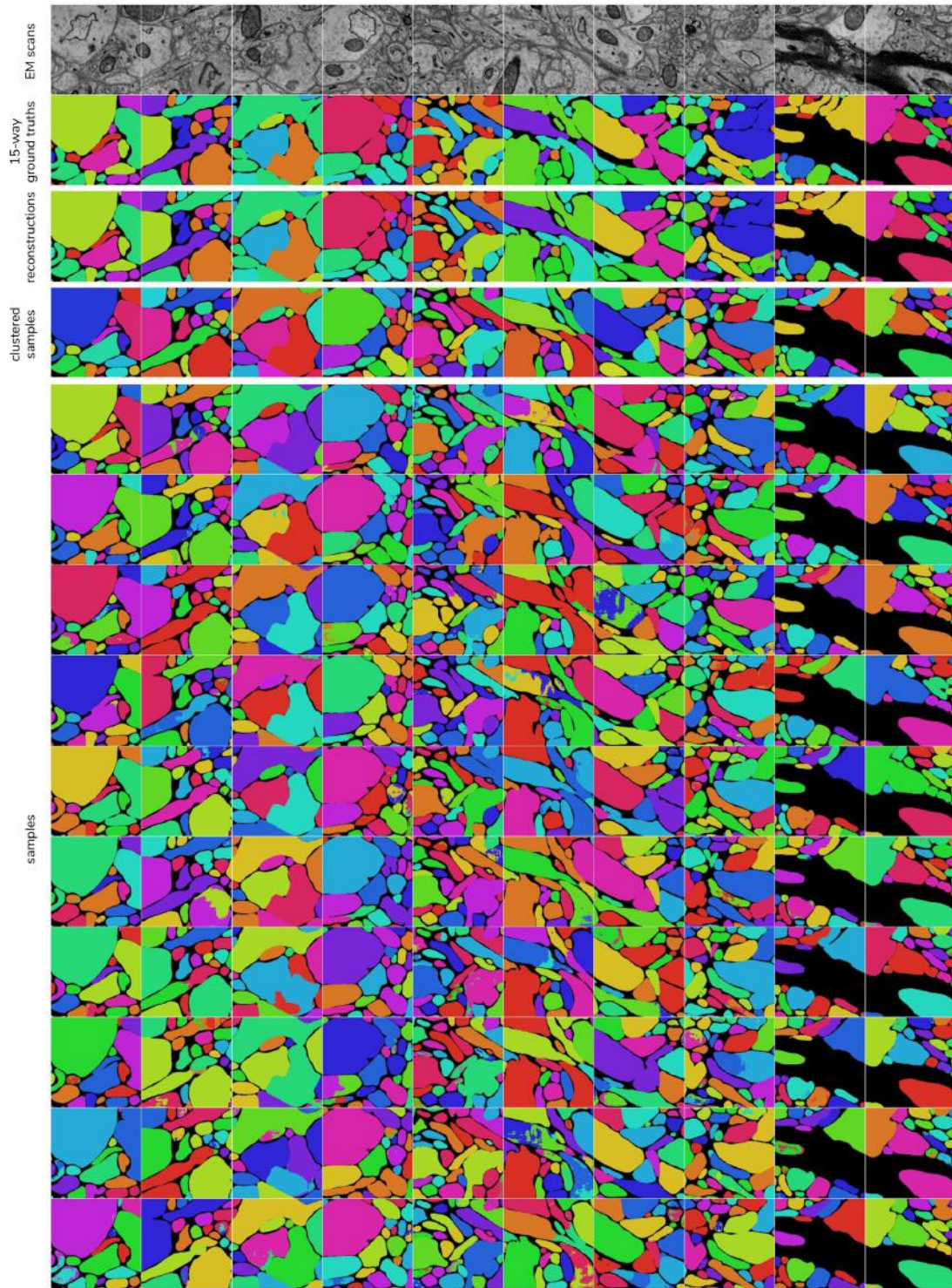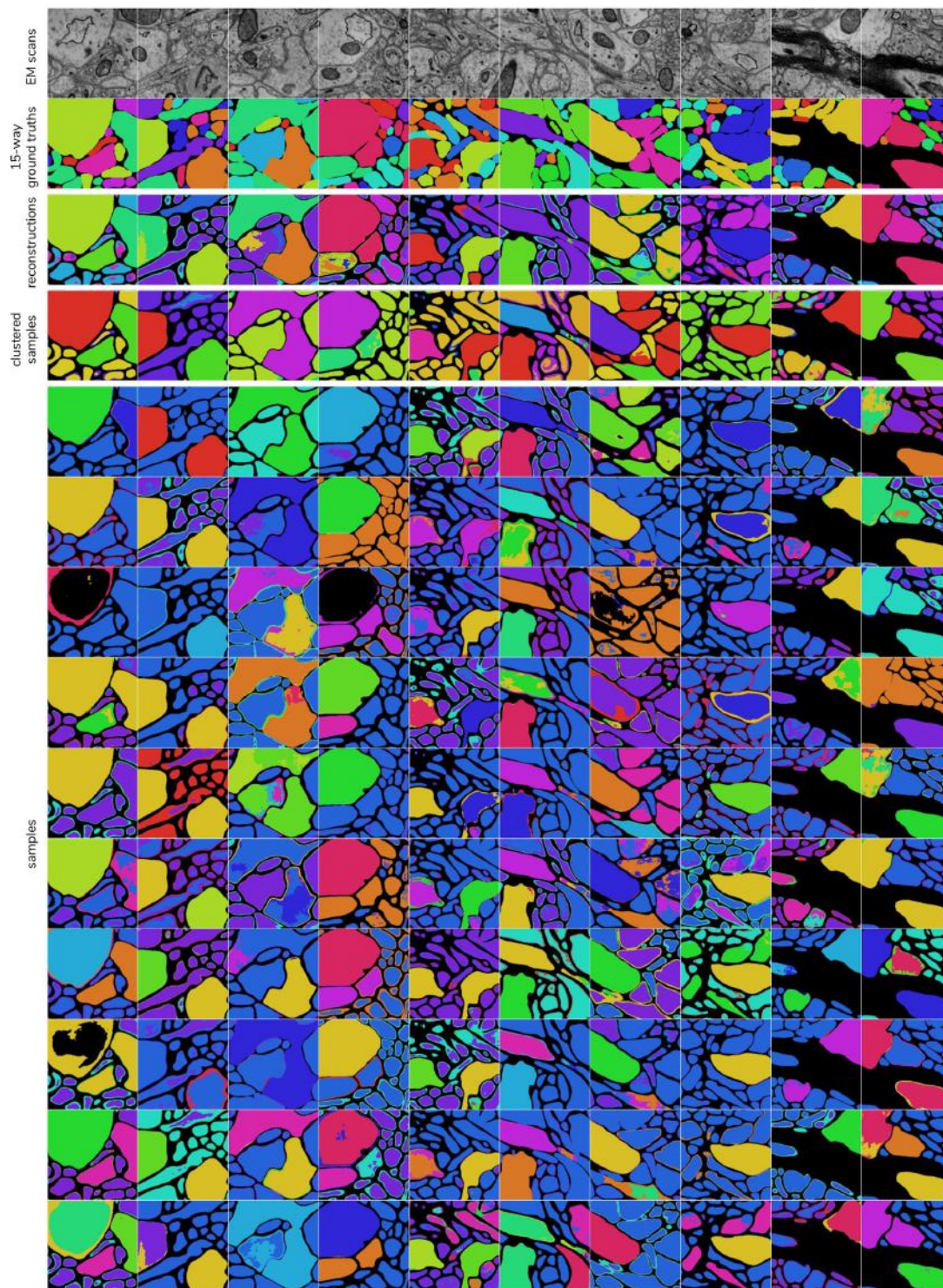ground truths

reconstructions

clustered
samples

samples

Figure 11: Qualitative results of the Hierarchical Probabilistic U-Net on our SNEMI3D test set.

Figure 12: Qualitative results of the Standard Probabilistic U-Net on our SNEMI3D test set.
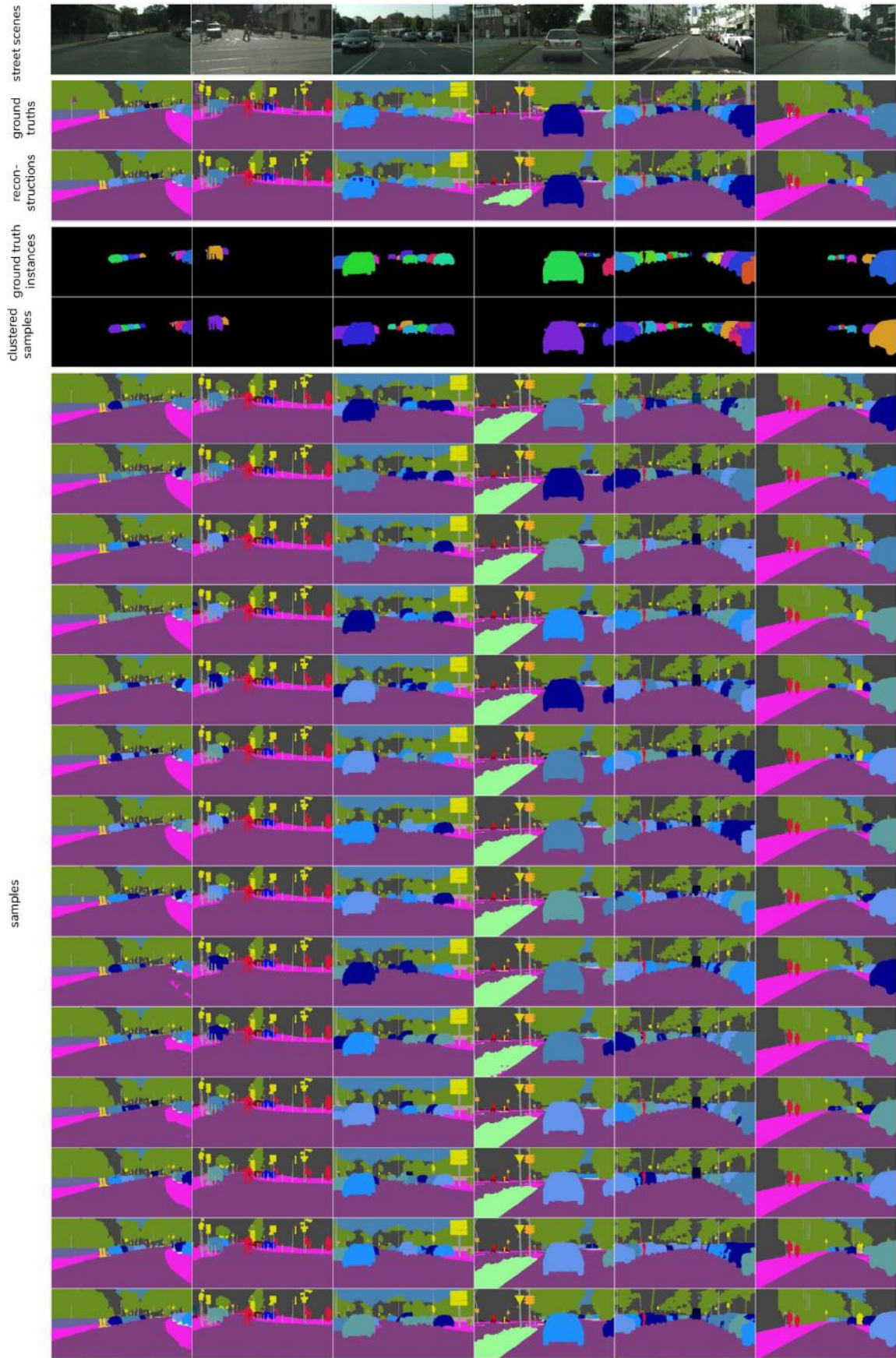
Figure 13: Qualitative results of the Hierarchical Probabilistic U-Net on our test set for the Cityscapes car instance task trained with 5 distinct latent car ids on resolution 512×1024. The 5 car ids take on different shades of blue. The samples show good consistency across individual car instances resulting in high-quality instance segmentations, see the $4th$ row from the top. Note how the model flips other natural ambiguous regions aside from cars e.g. *street $\leftrightarrow$ sidewalk* in the first scene and *truck $\leftrightarrow$ bus* in the second last.
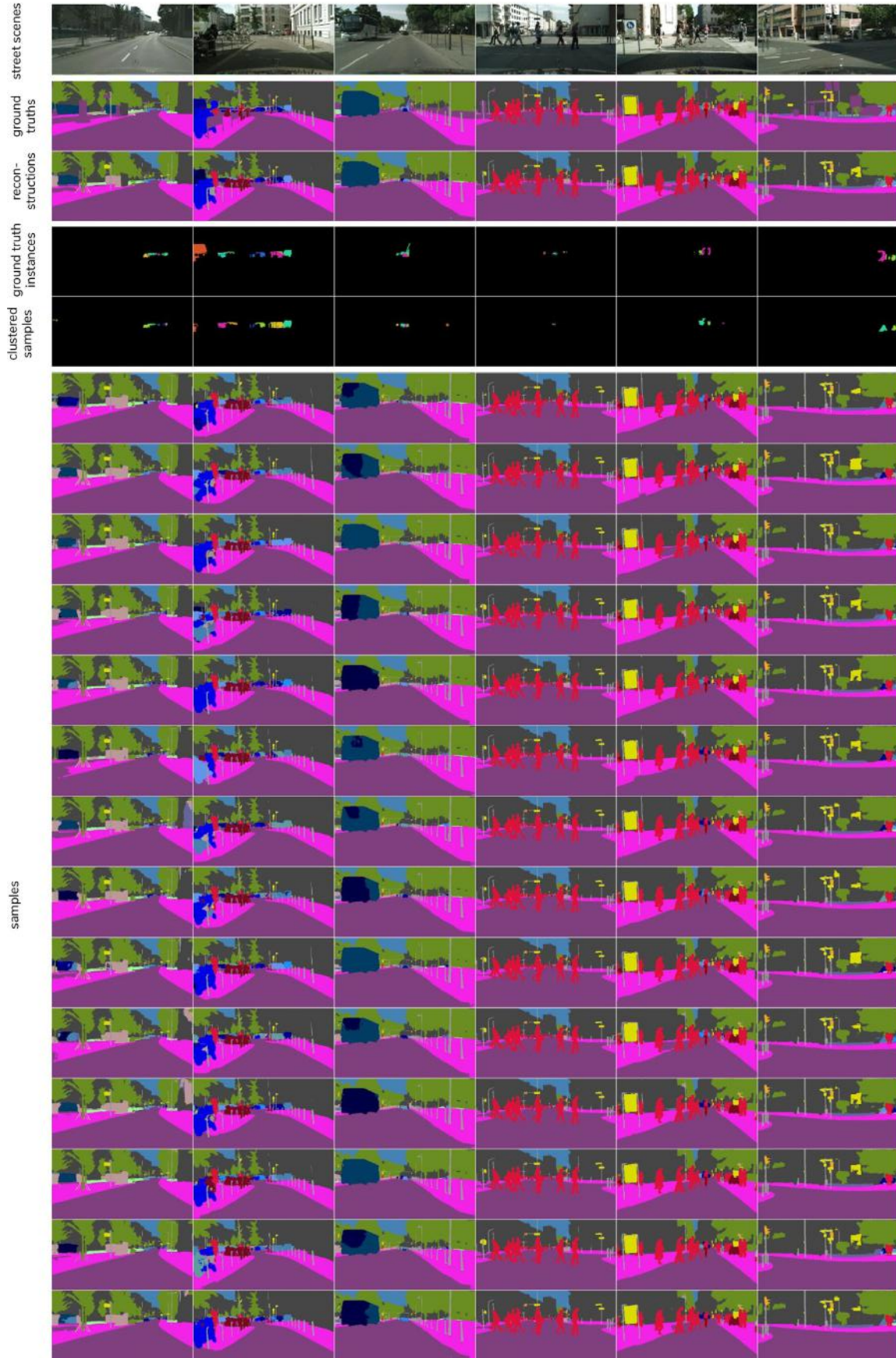
23

Figure 14: Qualitative results of the Hierarchical Probabilistic U-Net on our test set for the Cityscapes car instance task trained with 5 distinct latent car ids on resolution 512×1024. The 5 car ids take on different shades of blue. Here we show the top difficult cases in the test set in terms of the Rand error, which shows the difficulty of segmenting individual cars when they are very distant in the scene or heavily occluded.
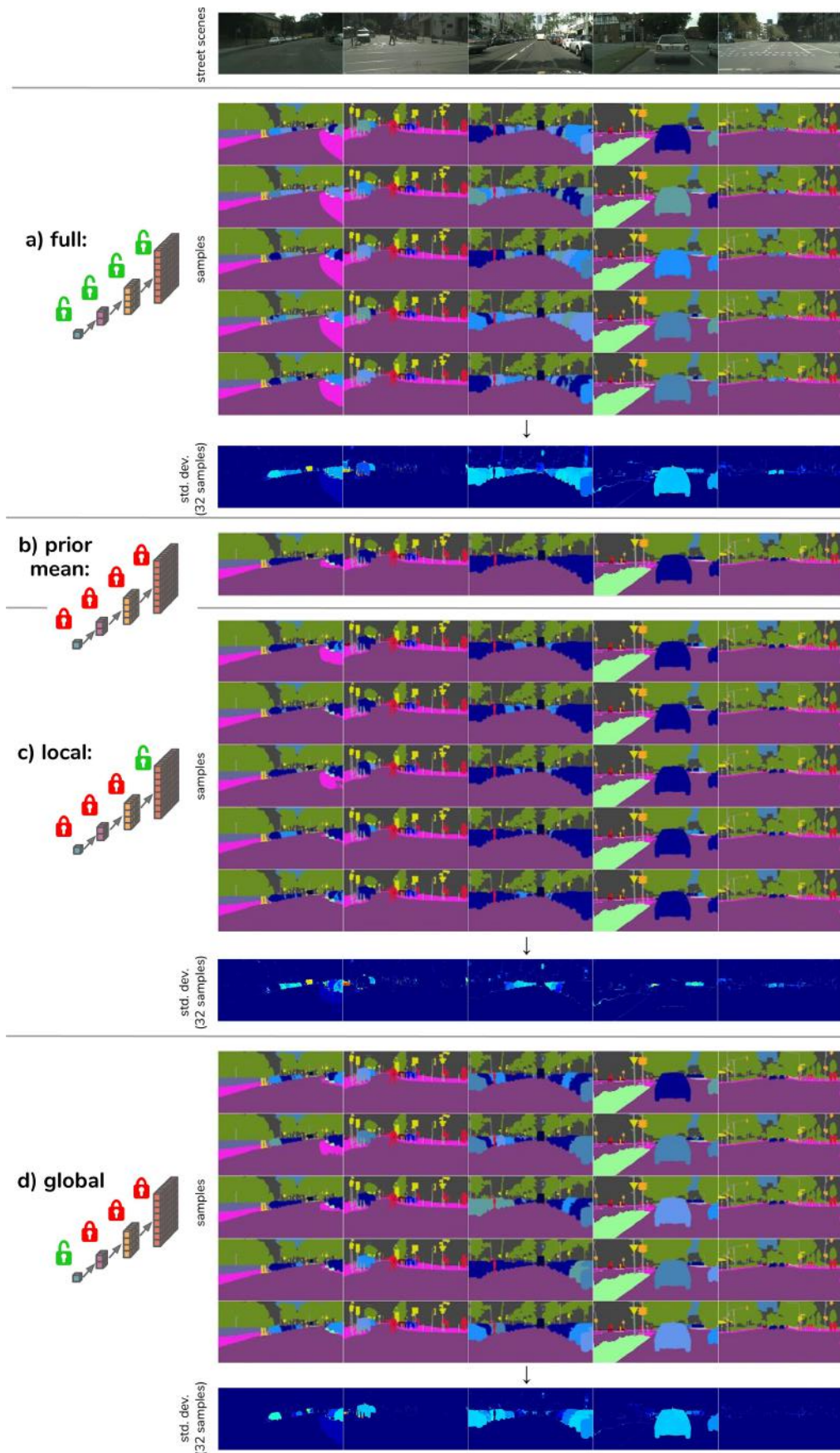
Figure 15: Qualitative results of the Hierarchical Probabilistic U-Net on our test set for the Cityscapes car instance task trained with 5 distinct latent car ids on resolution 512×1024. (**a**) Samples and standard deviation (std. dev.) across 32 samples when sampling from the full hierarchy. (**b**) Predictions from the prior mean. (**c**) Samples and std. dev. when sampling only from the most local scale. (**d**) Samples and std. dev. when sampling only from the most global scale. Note how the global and local scales affect the instance mask generation almost complementarily.