LEARNING TO DOWNSAMPLE FOR SEGMENTATION OF ULTRA-HIGH RESOLUTION IMAGES

Anonymous authors
Paper under double-blind review

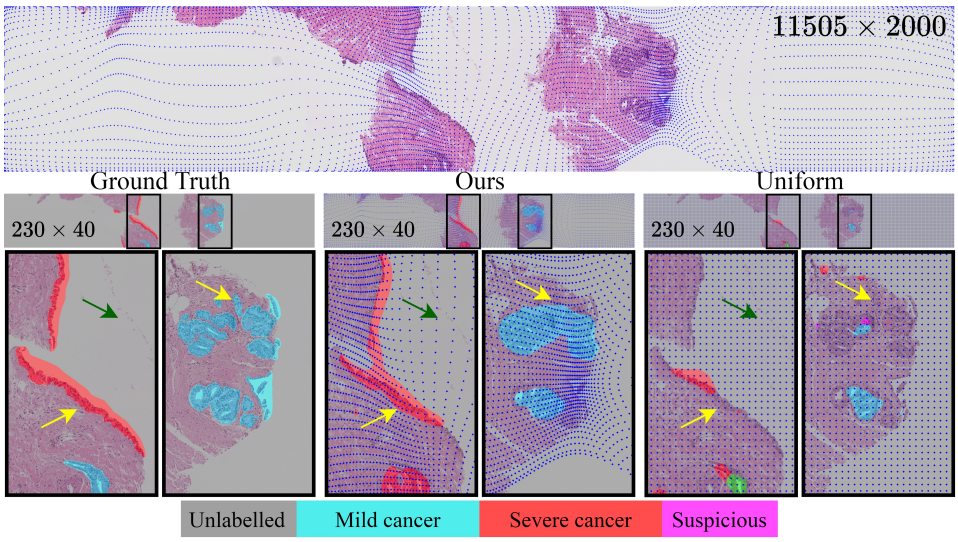


Figure 1: **Semantic segmentation with learnt deformed downsampling on cancerous histology image.** We propose a method for learning to downsample large images for better low-cost segmentation. We adapt sampling to the memory budget according to the difficulty of pixel-level segmentation by *deformation module*, a lightweight end-to-end learnable "downsampler" that can be flexibly integrated into existing semantic segmentation architectures. Top: Our learnt downsampling (blue dots on the images) adapts sampling density according to the semantic importance at each location. Middle: Segmentation performed on low-resolution downsampled images. Bottom: Compare to uniform downsampling (referred to as "Uniform"), our deformed downsampling samples more densely at difficult regions (yellow arrows) and ignore image content that does not contribute (green arrows) thereby leading to more accurate segmentation.

ABSTRACT

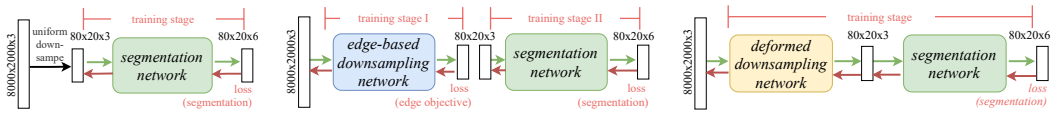
Many computer vision systems require low-cost segmentation with deep learning either because of the enormous size of images or limited computational budget. Common solutions uniformly downsample the input image to meet memory constraints, *assuming all pixels are equally important*. However this assumption does not hold when segmentation difficulty varies spatially, and hence compromises the performance (see the arrow pointed regions in Figure 1 where uniform downsample fail). We propose learning the spatially varying downsampling strategy jointly with segmentation offers advantages in segmenting large images with a limited computational budget (see Figure 1). We formulate the problem as learning the sampling density distribution depends on the local context. To avoid over-sampling at trivial regions like the background, we propose an edge-based sampling target to regularise training. Our experiments show that this method consistently learns sampling locations preserving more information. On benchmarks, we demonstrate superior segmentation accuracy and cost-performance trade-off compared to both uniform downsampling and two recent downsampling methods. Code will be available here.

1 INTRODUCTION

Many computer vision applications such as auto-piloting, geospatial analysis and medical image processing rely on good semantic segmentation over ultra-high resolution images. Exemplary applications include urban scene analysis with camera array images ($>25000\times14000$ pixels)(Wang et al., 2020), geospatial analysis with satellite images ($>5000\times5000$ pixels) (Maggiori et al., 2017) and histopathological analysis with whole slide images ($>10{,}000\times10{,}000$ pixels) (Srinidhi et al., 2019). Challenges arise when applying deep learning (DL) methods on those ultra-high resolution images, like those in Srinidhi et al. (2019); Chen et al. (2019); Marin et al. (2019), due to GPU memory constraints.

To speed up performance and adapt to memory requirements or data transmission latency, standard pipeline (Figure 2 (a)) uniformly downsample both input images and labels, then train the segmentation network at lower resolution, whose predictions are upsampled to the original resolution at testing. However the uniform downsampling wrongly assumes all pixels are equally important thus compromises the performance of applying DL approaches that have proved powerful on standard-sized images without downsampling preprocessing, like those in Chen et al. (2016); Sun et al. (2019). To tackle such problem, Marin et al. (2019) recently postulated that for better segmentation quality more pixels should be picked near semantic boundaries and formulated it as objective to independently train a content-adaptive downsampling network prior to the training of segmentation network, hence we referred to as "edge-based" and summarise in Figure 2 (b), which achieved state-of-the-art in low-cost segmentation accuracy. However the "edge-based" downsampling is not end-to-end optimised whose sampling density distribution is fixed to empirical single “best fit” hyper-parameter.

In this work, we start with a motivational experiment to investigate the barriers of current SOTA. We demonstrate that the segmentation performance of the "edge-based" method (Marin et al., 2019) is sub-optimal, as the optimal sampling density distribution varies spatially according to different local patterns. Motivated by this finding, we then introduce *deformation module*, a lightweight end-to-end trainable "downsampler" with *segmentation module* to adapt sampling density according to the local segmentation difficulty (Figure 2 (c)). The downsampled image guided by the *deformation module* results in a deformed low-resolution image and hence the name. The sampling operation is not directly differentiable, hence we formulate the *deformation module* to predict sampling density at each location to shift each sampling coordinate in a convolutional form. This idea is inspired by Recasens et al. (2018) who proposed a jointly learnable downsampler for image classification. However, without further assumptions, jointly learn downsampling to optimise pixel-level segmentation loss is ill-posed because the model can learn to densely sample both input and label at trivial locations like background. We propose to simulate a label-edge based sampling target to regularise the training, inspired by Marin et al. (2019). We demonstrate the general utility of our approach using two public and one local datasets from different domains. DeepGlobe (Demir et al., 2018) and a local Prostate Cancer Histology dataset ("PCa-Histo") are datasets consisting of ultra high-resolution aerial and whole slide pathological images. We also use the Cityscapes dataset (Cordts et al., 2016) since it provides a familiar platform for demonstration and evaluation. In all datasets, we show that a lightweight implementation of the *deformation module* can more flexibly adjust the sampling to content hence consistently boost the segmentation performance with little extra computational cost.



(a) uniforml (b) edge-based (c) ours

Figure 2: Comparing our jointly trained downsampling pipeline against independently trained "edge-based" (Marin et al., 2019) and determinstic "uniform" downsamplings. Low resolution prediction are uniform/non-uniform upsampled to original resolution at inference.

2 RELATED WORKS

Fix-sized sampling window restricted information "visible" to the network. Various multi-scale architectures have shown the importance of sampling multiple-scale information, either at input patches (He et al., 2017; Jin et al., 2020) or feature level (Chen et al., 2014; 2016; Hariharan et al., 2015), but less efficient because multiple re-scaling or inference steps. Dilated convolution (Yu

& Koltun, 2015) shift sampling locations of convolutional kernel at multiple distances to collect multiscale information therefore avoided inefficient re-scaling. Such shifting is however limited to fixed geometric structures of the kernel. Deformable convolutional networks (DCNs) (Dai et al., 2017) learn sampling offsets to augment each sampling location at feature level. Although DCNs share similarities with our approach, they are complementary to ours because we focus on learning optimal image downsampling as pre-processing so that keeps computation to a minimum at segmentation and a flexibly approach can be plug-in to existing segmentation networks.

Learnt sampling methods have been developed for image classification, arguing better image-level prediction can be achieved by an improved sampling while keeps computation cost low. Spatial Transformer Networks (STN) (Jaderberg et al., 2015) introduce a layer that estimates a parametrized affine, projective and splines transformation from an input image to recover data distortions and thereby improve image classification accuracy. Recasens et al. (2018) proposed to jointly learn a saliency-based network and "zoom-in" to salient regions when downsampling an input image for classification. Talebi & Milanfar (2021) jointly optimise pixel value interpolated (i.e. super-resolve) at each fixed downsampling location for classification. However, an end-to-end trained downsampling network has not been proposed so far for per-pixel segmentation. Joint optimising the downsampling network for segmentation is more challenging than the per-image classification, as we experimentally verified, due to potential oversampling at trivial locations and hence we propose to simulate a sampling target to regularise the training.

On learning the sampling for image segmentation, post-processing approaches (Kirillov et al., 2020; Huynh et al., 2021) refining samplings at multi-stage segmentation outputs are complementary to ours, but we focusing on optimising sampling at inputs so that keeps computation to a minimum. Non-uniform grid representation modifying the entire input images to alleviate the memory cost, such as meshes (Gkioxari et al., 2019), signed distance functions (Mescheder et al., 2019), and octrees (Tatarchenko et al., 2017). However none of those approaches is optimised specifically for segmentation. Recently Marin et al. (2019) proposed to separately train an "edge-based" downsampling network, to encourage denser sampling around object edges, hence improving segmentation performance at low computational cost and achieving state-of-the-art. However the "edge-based" approach is sub-optimal with sampling density distribution fixed to manual designed objective, the distance to edge, rather than segmentation performance. We verify this point empirically in Section 3.2 and hence propose our jointly trained downsampling for segmentation.

3 METHODS

In this section, we first formulate the basic components associated with this study in Section 3.1. We then perform a motivation experiment in Section 3.2 to illustrate the impact of manual tuned objective, particularly the sampling density around edge, on segmentation performance and its spatial variation across the image. Motivated by the finding, we propose the *deformation module*, the jointly trained downsampler for segmentation, in Section 3.3. To ease the joint optimisation challenge of oversampling at the trivial locations, a regularisation term is proposed in Section 3.4.

3.1 PROBLEM FORMULATION

Sampling method. Let $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ be a high-resolution image of an arbitrary size H,W,C. A sampler G takes $\mathbf{I}$ as input and computes a downsampled image $\hat{\mathbf{I}} = G(\mathbf{I})$, where $\hat{\mathbf{I}} \in \mathbb{R}^{h \times w \times C}$. Consider a relative coordinate system[1] such that $\mathbf{I}[u, v]$ is the pixel value of $\mathbf{I}$ where $u, v \in [0,1]$. And an absolute coordinate system such that $\hat{\mathbf{I}}[i, j]$ is the pixel value of $\hat{\mathbf{I}}$ at coordinates (i, j) for $i \in \{1,2,...h\}$, $j \in \{1,2,...w\}$. Essentially, the sampler G computes a mapping between (i, j) and (u, v). Practically, sampler G contains two functions $\{g^0, g^1\}$ such that:

$$\hat{\mathbf{I}}[i, j] := \mathbf{I}[g^0(i, j), g^1(i, j)]^2 \tag{1}$$

Segmentation and non-uniform upsampling. In this work we discuss model agnostic downsampling and upsampling methods, therefore any existing segmentation model can be applied. Here we denote

[1] A relative instead of absolute coordinate system is selected for sampling be calculated in a continues space.

[2] The "uniform" approach will have sampler $G_u = \{g_u^0(i, j) = (i-1)/(h-1), g_u^1(i, j) = (j-1)/(w-1)\}$.

the segmentation network S_ϕ, parameterised by ϕ, that takes as an input a downsampled image $\hat{\mathbf{I}}$ and makes a prediction $S_\phi(\hat{\mathbf{I}})$ in the low-resolution space. At testing, the upsampling process consists of a reverse sampler $G^{-1}()$ that reverse mapping each pixel at coordinates (*i, j*) from the sampled image $\hat{\mathbf{I}}$ back to coordinates (*u, v*) in the high-resolution domain. Then an interpolation function $\Pi()$ is applied to calculate the missing pixels to get the final prediction $\mathbf{P} = \Pi(G^{-1}(S_\phi(\hat{\mathbf{I}})))$. The nearest neighbour interpolation is used as $\Pi()$ in this work.

Disentangle the intrinsic interpolation error. Typical evaluation of downsampled segmentation with *Intersection over Union* (IoU) is performed after non-uniform upsampling, between the final prediction $\mathbf{P}$ and the label $\mathbf{Y}$. Hence $IoU(\mathbf{P}, \mathbf{Y})$ incorporated both segmentation error and interpolation error. To improve the interpretability of the joint trained system, we propose to disentangle the intrinsic interpolation error from the segmentation error by assuming a perfect segmentation at the downsampled space and calculates the error introduced after interpolation. In specific we calculate $IoU(\mathbf{Y}^{'}, \mathbf{Y})$, where $\mathbf{Y}^{'} = \Pi(G^{-1}(G(\mathbf{Y})))$, indicting the same non-uniform downsampling and upsampling processes applied to the label $\mathbf{Y}$.

3.2 Motivational study: investigating the barrier of the SOTA

The "edge-based" approach (Marin et al., 2019), separately train a non-uniform down-sampler G_e minimizing two competing energy terms: "sampling distance to semantic boundaries" and "sampling distance to uniform sampling locations", as the first and second term in equation 2, where $\mathbf{b}[i, j]$ is the spatial coordinates of the closest pixel on the semantic boundary. A temperature term λ is used to balance the two energies, whose value is empirically recommend to 1 by Marin et al. (2019) and decided the sampling density around edge.

$$E(G_e(i,j)) = \sum_{i,j} \|G_e(i,j) - \mathbf{b}[i,j]\|^2 + \lambda \sum_{i,j} \|G_e(i,j) - G_e(i',j')\|^2 \tag{2}$$

The first part of our work performs an empirical analysis to investigate a key question: "How does the empirically tuned temperature term λ affect the segmentation performance?". We hence perform binary segmentation on the Cityscapes dataset (Cordts et al., 2016). We generate a set of simulated "edge-based" samplers each at different sampling density around edge (i.e. λ) [3] and either directly calculate $IoU(\mathbf{Y}^{'}, \mathbf{Y})$ (Figure 3 (a)) or train segmentation with each fixed sampler to calculate IoU (Figure 3 (b)). It is clear that there is no "one-size-fits-all" sampling density configuration that leads to the best performance for all individual classes, neither for intrinsic interpolation error ($IoU(\mathbf{Y}^{'}, \mathbf{Y})$) nor segmentation error (IoU), which is also verified visually in Figure 3 (c). These observations altogether imply that the SOTA "edge-based" sampling scheme with a pre-set sampling density around the edge is sub-optimal, highlighting the potential benefits of a more intelligent strategy that can adapting sampling strategy to informative regions according to local patterns.

3.3 Deformation module: learn non-uniform downsampling for segmentation

Motivated by the above finding, we introduce the *deformation module*, a data-driven sampling method that adapts sampling density at each location according to its importance to the downstream segmentation task. Figure 4 provides a schematic of the proposed method.
In specific, for each high-resolution image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$, we compute its lower resolution version $\mathbf{I}_{\text{lr}} \in \mathbb{R}^{h_d \times w_d \times C}$. The *deformation module*, D_θ, parametrised by θ, takes the low-resolution image $\mathbf{I}_{\text{lr}}$ as input and generates a deformation map $\mathbf{d} = D_\theta(\mathbf{I}_{\text{lr}})$, where $\mathbf{d} \in \mathbb{R}^{h_d \times w_d \times 1}$, that predicts the sampling density at each pixel location. Next, the deformed sampler G_d is constructed by taking both $\mathbf{I}$ and $\mathbf{d}$ as input and computes the downsampled image $\hat{\mathbf{I}} = G_d(\mathbf{I}, \mathbf{d})$, where $\hat{\mathbf{I}} \in \mathbb{R}^{h \times w \times C}$. The downsampled image $\hat{\mathbf{I}}$ is then fed into the segmentation network to estimate the corresponding segmentation probabilities $\hat{\mathbf{P}} = S_\phi(\hat{\mathbf{I}})$. During training we calculate the downsampled label $\hat{\mathbf{Y}} = G_d(\mathbf{Y}, \mathbf{d})$ with the same deformed sampler and jointly optimise the parameters $\{\theta, \phi\}$ of both the *deformation module* and the *segmentation module* by minimise the segmentation specific loss function $\mathcal{L}_s(\mathbf{I}, \mathbf{Y}; \theta, \phi) = \mathcal{L}_s(\hat{\mathbf{P}}, \hat{\mathbf{Y}})$.

[3] We simulate the impact of λ to sampler as shown by Marin et al. (2019), with more details in Appendix A.6. The reason is our experiments used original non-square inputs instead of the square input reported by Marin et al. (2019).

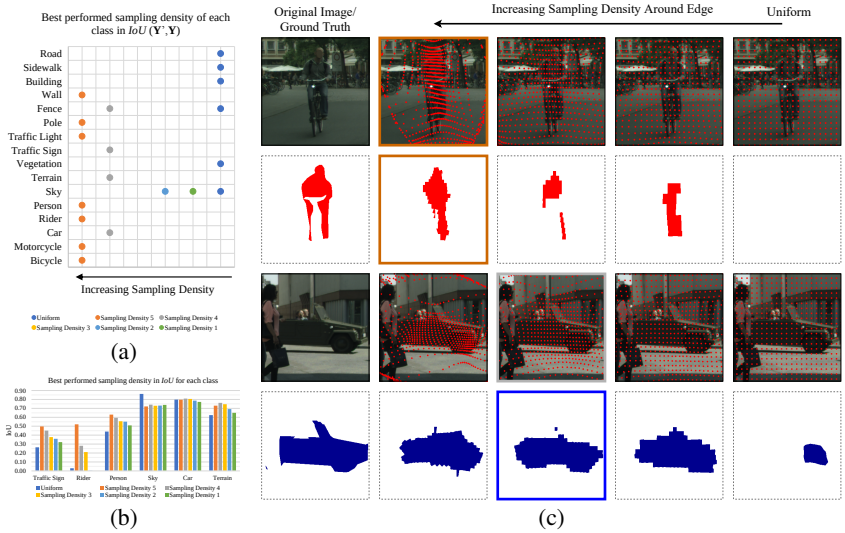


Figure 3: **Optimal sampling density varies over locations.** We demonstrate this by simulating a set "edge-based" samplers each at different sampling density around edge, and evaluate class-wise performance in (a) $IoU(\mathbf{Y}',\mathbf{Y})$ and (b) IoU. Best performed sampler for each class is shown as each dot in (a). Such variation also been observed visually in (c), where ground truth and predictions with various sampling density for two example classes are illustrated. Sampling locations are masked on original image in red dots. Motivational experiments are performed on binary segmentation on $1/10^{th}$ subset of Cityscapes (Cordts et al., 2016) with downsampled size of 64×128 given input of 1024×2048. Three under-represented classes are not presented.

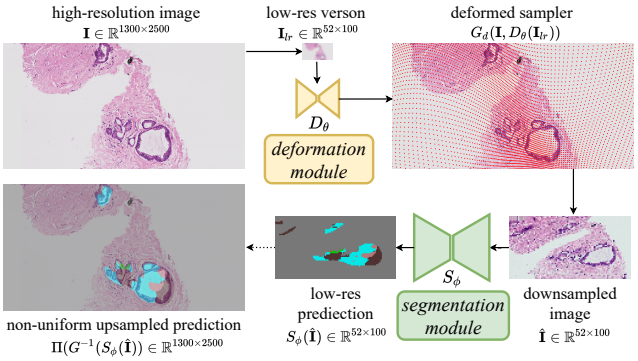


Figure 4: **Architecture schematic.** The *deformation module* takes a uniformly downsampled low-resolution version $\mathbf{I}_{\text{lr}}$ of the input image $\mathbf{I}$ and produces the deformation map $\mathbf{d} = D_\theta(\mathbf{I}_{\text{lr}})$ that predicts sampling importance weights at different locations. The deformed sampler G_d then samples with different density at each location of the input image $\mathbf{I}$ according to the deformation map $\mathbf{d}$, where sampling locations are shown as red dots masked on the image. The low-resolution downsampled image $\hat{\mathbf{I}} = G_d(\mathbf{I}, \mathbf{d})$ is then fed into the segmentation module which estimates the corresponding segmentation probabilities $S_\phi(\hat{\mathbf{I}})$ in the low-resolution space. At inference time, the prediction is non-uniformly upsampled to the original space in a deterministic manner.

Following earlier definitions in equation 1, the key is to construct and learn the deformed sampler G_d such that $\boldsymbol{u}[i,j] = g^0(i,j)$, $\boldsymbol{v}[i,j] = g^1(i,j)$. The idea is to construct the deformed sampler G_d who samples $\mathbf{I}$ denser at high-importance regions based on the sampling density predicted in the deformation map $\mathbf{d}$. Inspired by Recasens et al. (2018) we consider the sampling location of each pixel (i, j) is pulled by each surrounding pixel (i', j') by an attractive force. The attractive force is defined to be: 1) proportional to the sampling density $\mathbf{d}(i', j')$; 2) degrade away from the center pixel (i, j) by a factor defined by a distance kernel $k((i,j),(i', j'))$ and 3) applies within a certain distance.

Practically the distance kernel is a fixed Gaussian, with a given standard deviation σ and square shape of size $2\sigma+1$. The composition of forces applied to a central pixel (i,j) can then be calculated in the convolutional form hence the two deformed sampler functions $\{g_d^0,g_d^1\}$ are defined as equation 3. The standard deviation σ decided the distance the attraction force can act on, and the degree of force is learnt though the deformation map. We perform ablation study investigate the impact of σ in Section A.1.

$$g_d^0(i,j)=\frac{\sum_{i',j'}\mathbf{d}(i',j')k_\sigma((i,j),(i',j'))i'}{\sum_{i',j'}\mathbf{d}(i',j')k_\sigma((i,j),(i',j'))},\qquad g_d^1(i,j)=\frac{\sum_{i',j'}\mathbf{d}(i',j')k_\sigma((i,j),(i',j'))j'}{\sum_{i',j'}\mathbf{d}(i',j')k_\sigma((i,j),(i',j'))} \tag{3}$$

This formulation holds certain desirable properties that fit our goal: 1) the learnable *deformation module* produces a deformation map with varying sampling density at different regions, the need for this was illustrated in the motivational studies in Section 3.2; 2) the *deformation module* can naturally fit CNN architecture and preserve differentiability for it to be jointly trained with downstream segmentation loss. This way the sampling density at each location is optimised by segmentation performance rather than separately trained based on manual designed objective as the "edge-based" approach (Marin et al., 2019)).

3.4 Regularisation of the joint-trained sampling

Our method is similar to Recasens et al. (2018), which however is only explored for image classification tasks. Transferring the method proposed in Recasens et al. (2018) to segmentation not only needs reformulating the problem as described in Section 3.1, optimising the joint system at the pixel level is not trivial due to potential oversampling at easy to segment regions like background. This unwanted behaviour is caused by jointly optimise the downsampling parameters θ with both image $\mathbf{I}$ and label $\mathbf{Y}$ as input so that the segmentation loss $\mathcal{L}_s(\mathbf{I},\mathbf{Y};\theta,\phi)=\mathcal{L}_s(\hat{\mathbf{P}},\hat{\mathbf{Y}})$ can be calculated in low-resolution so that keeps computation to a minimum. Such setting would encourage oversampling at easy rather than difficult locations to reduce loss, which contradict our goal. We also experimentally verified the naive adaptation of the method from Recasens et al. (2018) to segmentation (red bars in Figure 5) not performing satisfactionally. To discourage the network learning trivial solutions, we simulate a target sampling density map to regularise training. Inspired by Marin et al. (2019) that object edge may be informative for segmentation, we use the simulated target to encourage denser sampling around the edge. Different to the approach of Marin et al. (2019) 1) we directly simulate a target deformation map without separate training to carry its main message that edge information is useful and 2) fit into the joint training system excludes the heavy dependency to a specific parameter λ.

In specific, let $\mathbf{Y}_{lr}\in\mathbb{R}^{h_d\times w_d\times 1}$ be the uniformly downsampled segmentation label at same size with the deformation map $\mathbf{d}$, the edge loss $\mathcal{L}_e(\mathbf{I}_{lr},\mathbf{Y}_{lr};\theta)=MSE(\mathbf{d},\mathbf{d}_t)$, where $\mathbf{d}$ is the predicted deformation map $d=D_\theta(I_{lr})$, and $\mathbf{d}_t$ is the target deformation map $\mathbf{d}_t=Edge(Gaus(Y_{lr},\delta))$[4]. To this end, we jointly optimise the parameters $\{\theta,\phi\}$ corresponds to the *deformation module* and the *segmentation module* respectively by the single learning objective as equation 4, where the first term is the segmentation specific loss (e.g. cross entropy + $L2$ weight-decay) and the second term is the edge loss. We add a weight term γ for both losses have comparable magnitudes. We note different to λ in equation 2, γ can better adapt sampling location to difficult regions because it is is balancing the edge-loss and segmentation loss (end task loss) in an end-to-end setting, while λ is balancing the two manual designed sampling targets in a separately trained downsampling network. We also evaluate impact of γ in Section A.1.

$$\mathcal{E}(\mathbf{I},\mathbf{Y};\theta,\phi)=\mathcal{L}_s(\mathbf{I},\mathbf{Y};\theta,\phi)+\gamma\mathcal{L}_e(I_{lr},\mathbf{Y}_{lr};\theta) \tag{4}$$

4 Experiments and Results

In this section, we evaluate the performance of our deformed downsampling approach on three datasets from different domains as summarised in Table 1, against three baselines. We evaluate the segmentation performance with IoU and verify the quality of the sampler by looking at intrinsic

[4] $Edge()$ is an edge detection filter by a convolution of a specific 3×3 kernel [[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]] and $Gaus()$ is Gaussian blur with radius (Standard deviation of the Gaussian kernel) $\delta=1$ to encourage sampling close to the edge. To avoid the edge filter pickup the image boarder, we padding the border values prior to applying the edge filter.

interpolation error with $IoU(\mathbf{Y}^{'},\mathbf{Y})$. We show quantitatively and qualitatively in Section 4.1. For a fair comparison, we additionally perform experiments at the same experimental conditions with square input and compare reported results from Marin et al. (2019) in Section 4.2.

Model and training. *deformation module* is defined as a small CNN architecture comprised of 3 convolution layers. The segmentation network was defined as a deep CNN architecture, with HRNetV2-W48 (Sun et al., 2019) used in all datasets, and PSP-net (Zhao et al., 2017) as a sanity check in the Cityscapes dataset. We employ random initialisation like as in He et al. (2015), the same training scheme (Adam (Kingma & Ba, 2014), the focal loss as segmentation loss and MSE loss for the edge-loss with full details in Appendix A.8) unless otherwise stated.

Baselines. We compare our method either with single segmentation loss ("Ours-Single loss") or adding edge-loss ("Ours-Joint loss") against three baselines: 1) "uniform" downsampling; 2) the "edge-based" [5] and 3) "interpolation", the jointly learnt method for pixel value interpolation at fixed uniform downsampling locations (Talebi & Milanfar, 2021), although interpolation is not our original goal we add this joint learnt method for reference in Section 4.2, whose implementation details given in Section A.2.

Table 1: Dataset summary. More details in Appendix A.7

Dataset	Content	Resolution (pixels)	Number of Classes
Cityscape (Cordts et al., 2016)	Urban scenes	2048×1024	19
DeepGlobe (Demir et al., 2018)	Aerial scenes	2448×2448	6
PCa-Histo (local)	Histopathological	$1968\pm216\times9392\pm4794$	6

4.1 Quantitative and qualitative performance analysis

We plot $mIoU$ and $mIoU(\mathbf{Y}^{'},\mathbf{Y})$ in Figure 5 comparing our method against "uniform" and "edge-based" baselines on all three datasets. Figure 6 further investigate IoU at per-class level, aiming to understand the impact of different object sizes reflected by different class frequency. Figure 7 examine the impact of downsampled size and ask could our method learn a downsampling strategy enable better trade-off between performance and computational cost.

In Figure 5, "Ours-Single loss" represents the independent contribution of the proposed joint training system with single segmentation loss which performs better than "uniform" baseline on 2/3 tested datasets. Adding the proposed *edge loss*, "Ours-Joint loss" consistently performs best with 3% to 10% higher absolute $mIoU$ over the "uniform" baseline. The performance improvement from "Ours-Single loss" to "Ours-Joint loss", represents the contribution of *edge loss* within proposed joint training framework, which vary over datasets suggests that the informativeness of edge is data dependent, hence an end-to-end system is needed to better adapt sampling to content. Besides, $mIoU(\mathbf{Y}^{'},\mathbf{Y})$ does not always agree with $mIoU$, indicting segmentation error and interpolation error are not tightly bounded therefore separately trained sampler may fail.

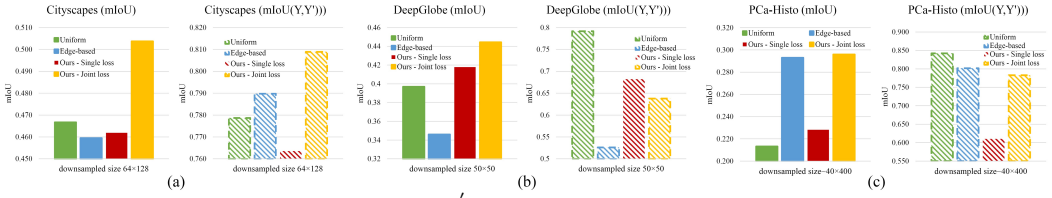


Figure 5: Comparing $mIoU$ and $mIoU(\mathbf{Y}^{'},\mathbf{Y})$ of our joint trained downsampler, either with single segmentation loss ("Ours-Single loss") or additional edge loss ("Ours-Joint loss"), versus two baseline downsampling methods on three datasets.

At class-wise, as in Figure 6, "Ours-Joint loss" performs up to 25% better in absolute IoU over baselines. Despite improving all low-frequency classes representing small object over both baselines,

[5] Note this is an approximation of method from Marin et al. (2019), follow same reason and scheme as declared in motivational Section 3.2, which is the best-performed result from a set segmentation networks each trained with a simulated "edge-based" samplers represents with different λ

"Ours-Joint loss" also improves "edge-based" baseline in high-frequency classes representing large objects and leads to either better or close to the "uniform" baseline. Results suggests the joint trained method generalise "edge-based" sampling better to not only small but large objects.

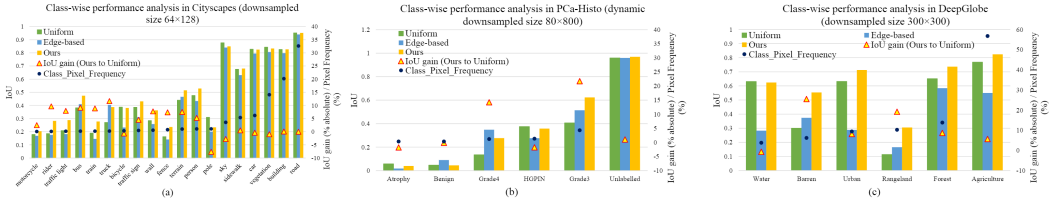


Figure 6: Class-wise IoU on three datasets. IoU gain indicates improvements from our method over "uniform" baseline. Classes are ordered with increasing pixel frequency which is also indicated in each plot.

One key motivation of this work is enable a better trade-off between performance and computational cost, and this is verified in Figure 7. Our method shows consistently saving up to 90% computational cost than the "uniform" baseline, across a set of different downsampling sizes (down to 0.0004 pixel sampled from original) over all three datasets. We notice $mIoU$ does not increase monotonically on DeepGlobe dataset (Demir et al., 2018) with increased cost in Figure 7 (b), this indicates high resolution is not always preferable but a data-dependent optimal tradeoff exists and an end-to-end adjustable downsampler is needed. The $mIoU(\mathbf{Y}^{'},\mathbf{Y})$ results show our method can adjust sampling priority depends on available computational resource, by compromise the interpolation error but focus on improving segmentation at the low-cost end in Figure 7 (b) and (c). Quantitative evidences all together show our method can efficiently learn where to "invest" the limited budget of pixels at downsampling to achieve the highest overall return in segmentation accuracy.

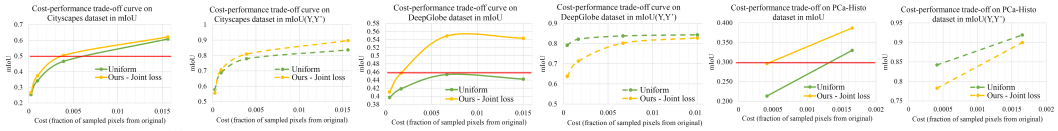


Figure 7: Cost-performance trade-offs in $mIoU$ and $mIoU(\mathbf{Y}^{'},\mathbf{Y})$ on three datasets. Cost determines the amount of computation required at segmentation. The red lines indicates at same performance our method can save 33%, 90% and 70% from the "uniform" baseline on the three datasets respectively.

Visual results from Figure 8 and Figure 9 show our method picked the advantage of the "edge-based" approach by sampling denser at small objects while further generalise with better sparse sampling on larger objects (see caption for details). Visual evidence also suggests the object edges are not always informative (see window edges in the wall class of Figure 8 misguided the edge-based sampler under-sampling of surrounding textures and lead to miss-prediction, and similarly as bushes in forest region as Figure 9). We therefore further discuss how boundary accuracy affects segmentation accuracy in the next section.

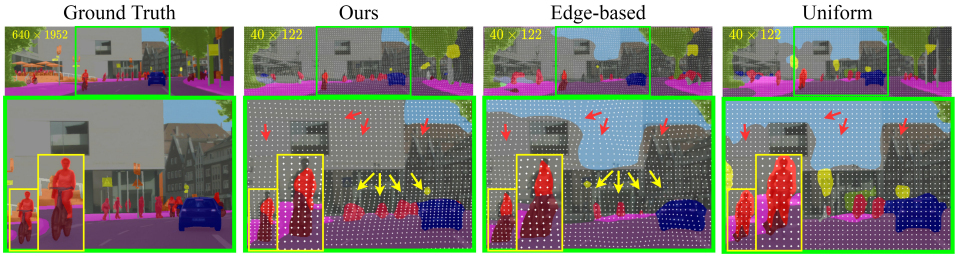


Figure 8: Examples on Cityscapes (Cordts et al., 2016) comparing our method against both baselines, where segmentation is performed on 16 times downsampled images (at each dimension). Predictions are masked over and sampling locations are shown in white dots. Yellow/ red arrows indicated regions denser/ sparser sampling helped to segment rider (red)/ sky (blue) classes, respectively.

4.2 A FAIR COMPARISON TO THE SOTA AND BOUNDARY ACCURACY ANALYSIS

In this section we perform experiments at same experimental condition on Cityscapes dataset and compare to reported results from Marin et al. (2019). The "interpolation" (Talebi & Milanfar, 2021)

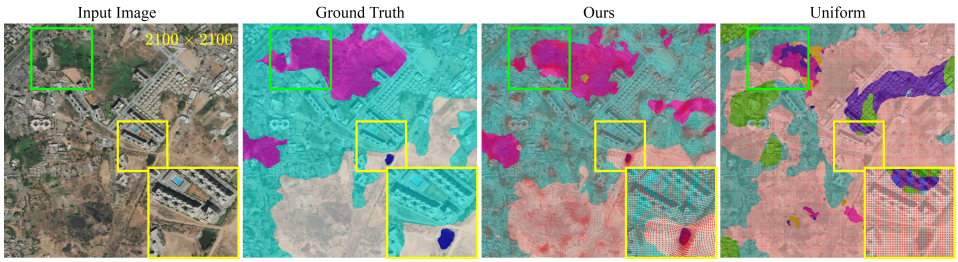
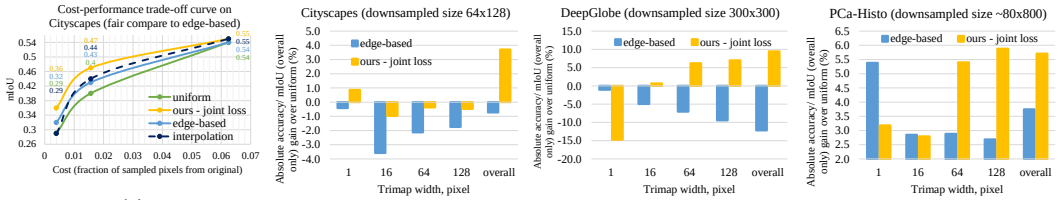


Figure 9: Qualitative example on DeepGlobe dataset (Demir et al., 2018) where segmentation performed on 8 times downsampled image (at each dimension). Predictions are masked over and sampling locations are shown in red dots. Yellow/ green boxed region indicated regions denser/ sparser sampling helped segmenting water (blue)/ forest (purple) classes, respectively.

baseline is also compared. Figure 10 (a) shows our method outperforms all three baselines at all trade-off downsample sizes by upto 4% in absolute $mIoU$ over the second best result and achieves better cost-performance trade-off saving upto 47% calculation. Figure 10 (a) also suggests "interpolation" approach is less effective at small downsampling while its benefits increase as downsampling size increases. Visual comparison to "interpolation" baseline are provided in Appendix Section A.2.

We introduced the edge-loss to regularize the training of the joint system, inspired by Marin et al. (2019). Here we measure boundary precision and ask 1) how does it contribute to overall segmentation accuracy? and 2) how differently do the joint system balance boundary accuracy and overall segmentation accuracy than the separate system (Marin et al., 2019)? We adopt trimap following Kohli et al. (2009); Marin et al. (2019) computing the accuracy within a band of varying width around boundaries in addition to $mIoU$, for all three datasets in Figure 10 (b). We found: 1) the "edge-based" baseline is optimal close to the boundary, but its performance does not consistently transfer to overall $mIoU$; 2) Our joint learnt downsampling shows can identify the most beneficial sampling distance to invest sampling budget and lead to the best overall $mIoU$; 3) The most beneficial sampling distances learnt by our approach are data-dependent that can close to the boundary (i.e. Cityscapes) or away (i.e. DeepGlobe and PCa-Histo); 4) it also suggests our method is particularly useful when the labelling is based on higher-level knowledge, i.e. Deepglobe and PCa-Histo dataset.



(a) (b)

Figure 10: (a) Experiments performed at comparable condition to reported "edge-based" results by Marin et al. (2019) on Cityscapes dataset (Cordts et al., 2016), i.e. applying central 1024×1024 crop pre-processing and train with PSP-net (Zhao et al., 2017); (b) Absolute accuracy difference between our method and "edge-based" baseline near semantic boundaries and overall $mIoU$ differences (shown as overall) on the three datasets.

5 Conclusion

We introduce an approach for learning to downsample ultra high-resolution images for segmentation tasks. The main motivation is to adapt the sampling budget to the difficulty of segmented pixels/regions. We empirically illustrate SOTA method (Marin et al., 2019) been limited by sampling location fixed constrained to manual designed edge objective, and hence motivate our end-to-end trained method. We illustrate simple extend the method of (Recasens et al., 2018) to segmentation does not work, and propose to avoid trivial solutions by incorporating an edge-loss to regularize the training. Although our edge-loss, despite a simpler approximation, share the same spirit with Marin et al. (2019) we demonstrate our jointly trained method generalises sampling more robustly especially when label edges are less informative and consistently leads to a better cost-performance trade-off. Our method is light weighted and can be flexibly combined with the existing segmentation method without modifying architecture.

REFERENCES

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3640–3649, 2016.

Wuyang Chen, Ziyu Jiang, Zhangyang Wang, Kexin Cui, and Xiaoning Qian. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8924–8933, 2019.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9785–9795, 2019.

Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 447–456, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Chuong Huynh, Anh Tuan Tran, Khoa Luu, and Minh Hoai. Progressive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16755–16764, 2021.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

Chen Jin, Ryutaro Tanno, Moucheng Xu, Thomy Mertzanidou, and Daniel C Alexander. Foveation for segmentation of mega-pixel histology images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 561–571. Springer, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9799–9808, 2020.

Pushmeet Kohli, Philip HS Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 3226–3229. IEEE, 2017.

Dmitrii Marin, Zijian He, Peter Vajda, Priyam Chatterjee, Sam Tsai, Fei Yang, and Yuri Boykov. Efficient segmentation: Learning downsampling near semantic boundaries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2131–2141, 2019.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 51–66, 2018.

Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. Deep neural network models for computational histopathology: A survey. *arXiv preprint arXiv:1912.12378*, 2019.

Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

Hossein Talebi and Peyman Milanfar. Learning to resize images for computer vision tasks. *arXiv preprint arXiv:2103.09950*, 2021.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2088–2096, 2017.

Xueyang Wang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, Liuyu Xiang, Zerun Wang, Guiguang Ding, David Brady, Qionghai Dai, et al. Panda: A gigapixel-level human-centric video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3268–3278, 2020.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

A APPENDIX

Here we provide additional results and various ablation studies and implementation details that have not been presented in the main paper. We submit our code in an anonymous repository at the discussion.

CONTENTS

A.1 SENSITIVITY ANALYSIS ON HYPERPARAMETERS

We perform sensitivity analysis on four hyperparameters in our system, on Cityscapes Cordts et al. (2016) and DeepGlobe Demir et al. (2018) datasets in Table 2 to Table 5. In specific, Table 2 investigates the standard deviation σ of the kernel k in equation 3, which impact the distance the attraction force can be act on. Table 3 and Table 4 investigates the the kernel size and input size of proposed *deformation module*, which impact model capability and amount of context to the *deformation module*. Table 5 investigates edge-loss weight γ in equation 4 which impact the weight of edge regularisation. In general our method been stable and robust across all tested hyparameters (with absolute $mIoU$ variation within 3%).

Table 2: The sensitivity analysis of the hyperparameter of the kernel k in equation 3: the standard deviation σ.

Cityscapes (1024^2 to 96^2 pixels)			
σ (pixels)	15	26	32
$mIoU$	0.35	0.34	0.32
DeepGlobe (2448^2 to 100^2 pixels)			
σ (pixels)	20	25	33
$mIoU$	0.43	0.45	0.43

Table 3: The sensitivity analysis of the hyperparameter of the kernel size in proposed *deformation module*.

Cityscapes (1024^2 to 96^2 pixels)			
kernel size	3×3	5×5	7×7
$mIoU$	0.36	0.35	0.35
DeepGlobe (2448^2 to 100^2 pixels)			
kernel size	3×3	5×5	7×7
$mIoU$	0.45	0.45	0.46

Table 4: The sensitivity analysis of different low-res input sizes to the proposed *deformation module*.

Cityscapes (1024^2 to 64^2 pixels)					
low-res input size	32^2	48^2	64^2	80^2	96^2
$mIoU$	0.35	0.35	0.34	0.33	0.36
DeepGlobe (2448^2 to 50^2 pixels)					
low-res input size	50^2	75^2	100^2	200^2	300^2
$mIoU$	0.44	0.42	0.45	0.42	0.43

Table 5: The sensitivity analysis of the edge-loss weight γ in equation 4

Cityscapes (1024^2 to 64^2)			
γ	50	100	200
$mIoU$	0.33	0.35	0.35
DeepGlobe (2448^2 to 100^2)			
γ	50	100	200
$mIoU$	0.45	0.45	0.43

A.2 IMPLEMENTATION AND VISUAL ILLUSTRATION OF THE "INTERPOLATION" BASELINE

The "interpolation" baseline is implemented by replacing our *deformation module* with the *resizing network* proposed by Talebi & Milanfar (2021). The number of residual blocks (r) and the number of convolutional filter (n) are two key hyperparameters of *resizing network*, hence we perform each of our experiment with suggested hyparameter combinations by Talebi & Milanfar (2021), and select the best performed results to represent "interpolation" baseline, with all results provided in Table 6.

Table 6: The "interpolation" baseline performance with different combinations of number of residual blocks (r) and the number of convolutional filter (n) for the *resizing network*, at each of the three downsampling sizes on cityscapes dataset (1024×1024). Results measured in $mIoU$.

Downsampling size	Blocks / Filters	$r=1$	$r=2$	$r=3$	$r=4$
64×64	$n=16$	0.29	0.27	0.27	0.27
	$n=32$	0.29	0.29	0.29	0.29
128×128	$n=16$	0.42	n/a	n/a	0.43
	$n=32$	0.43	n/a	n/a	0.44
256×256	$n=16$	0.54	n/a	n/a	0.54
	$n=32$	0.55	n/a	n/a	0.55

To better understand the why our method works better than the "interpolation" baseline at small downsampling size, we plots visual examples in Figure 11. The results when the sampling budget is limited, jointly learning where to "invest" the limited sampling locations is a more effective strategy, while with more sampling budgets available the learning to "interpolation" approach would also work, as previous trade-off experiments (Figure 10 (a)) have confirmed.

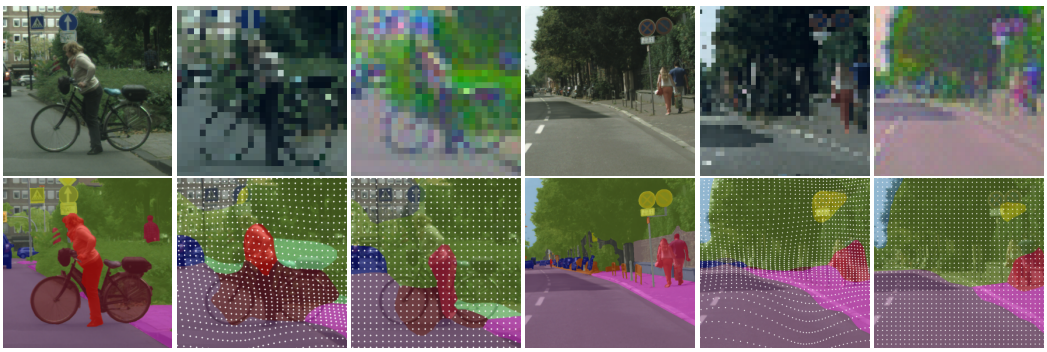


Figure 11: Visual comparison of our jointly trained downsampling against "interpolation" baseline (Talebi & Milanfar, 2021) on Cityscapes dataset where 1024 inputs are downsampled to 64. Sampling locations are masked over prediction as white dots.

A.3 MORE VISUAL EXAMPLES AT EXTREME DOWNSAMPLING RATES

A visual overview of applying our method on Cityscapes are given in Figure 12. To further verify how our method performs at extreme downsampling on histology images, in Figure 13, we show visual examples when a downsampling rate of 50 times at each dimension is applied to the PCa-Histo dataset. Consistent with quantitative results, our method can still perform well comparing to ground truth and significantly better than "uniform" baseline at such extreme downsampling rate, which leads to better cost-performance trade-off. Our method performs especially well for the most clinically important classes, Gleason Grade 3 and Gleason Grade 4, the separation of these two classes is often referred to as the threshold from healthy to cancer. The sampling locations shown as blue dots in Figure 13 also indicated our method learnt to sample densely at clinical important and challenge regions (e.g. cancerous

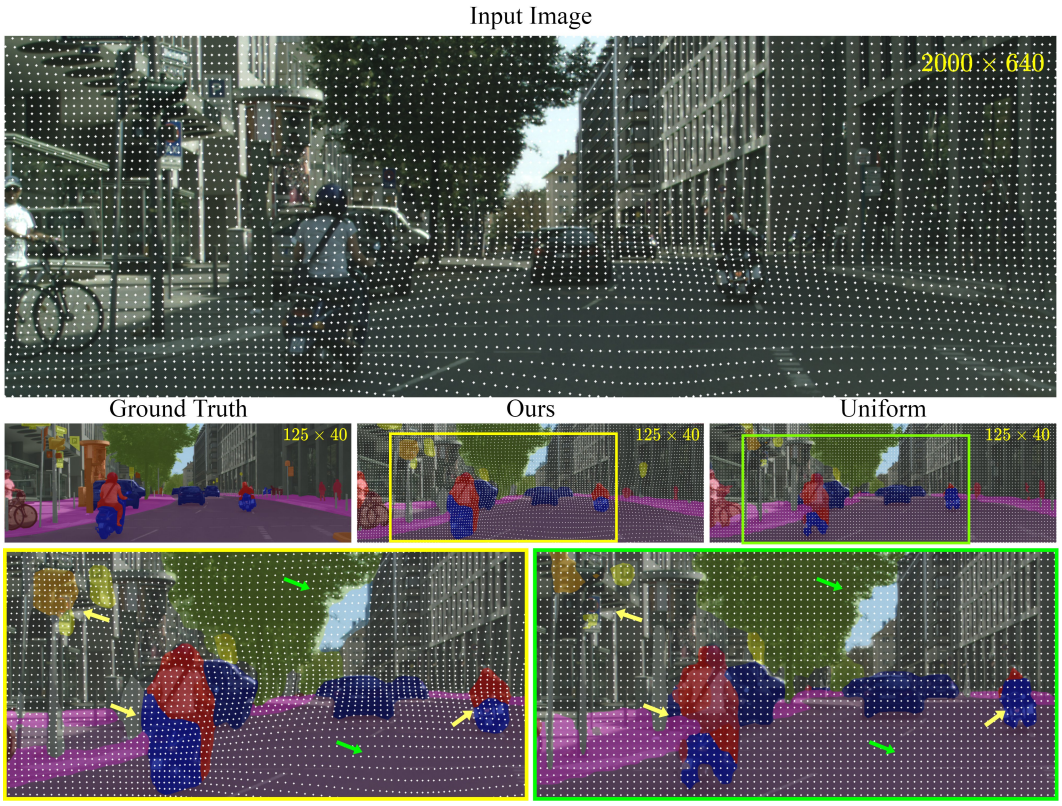


Figure 12: **Semantic segmentation with learnt deformed downsampling on Cityscapes.** Our method learns non-uniform downsampling (white dots on the images) a high-resolution image (top-row) to preserve information that guides segmentation and ignore image content that does not contribute (green arrows) thereby enabling low-cost semantic segmentation on low-resolution downsampled images. This strategy helps identify small regions such as the people and traffic signs in the figure; see masked ground truth and predictions in middle-row. Compared to "Uniform" baseline "Ours" samples are more densely at important semantic regions (yellow arrows) and leads to more accurate segmentation (see bottom-row).

nuclei and tissues) while sampling sparsely at less informative regions (e.g. unlabelled/background), which lead to consistent improvement over uniform sampling on segmentation accuracy.

A.4 Investigate the learnt class distribution

To verify our *deformation module* has learnt to downsample denser at important regions/classes effectively, we monitor class frequency change after downsampling, as shown in Figure 14. To have a direct evaluation of whether a class has been sampled denser or sparser comparing to its original frequency, we calculate the class frequency change as the ratio between sampled and original image.

The results indicated the *deformation module* learnt to adjust sampling strategy so that more pixels are "invested" to important regions/classes and less from less informative region/classes, and lead to optimal overall segmentation accuracy. For example, the *deformation module* learnt to sample less (shown as negative sampling frequency ratio) in the "Background" class in the PCa-Histo dataset (Figure 14 (a)) and "road" and "sky" classes in the cityscapes (Cordts et al., 2016) dataset (Figure 14 (b)), in both cases those classes can be considered as potential "background" with common knowledge. However, our method also shows ability to adjust sampling strategy on more complex dataset where "background class" is less obvious. For example, in DeepGlobe (Demir et al., 2018) dataset there is no obvious "background class", but the "Urban" class has been learnt to sample much denser than other classes (Figure 14 (c)), and leads to boosted segmentation accuracy (see Figure 5 and Figure 7 in the main paper). The learnt distribution on the DeepGlobe dataset also verified our argument that the manual designed *sample denser around object edge strategy* as the "edge-based" (Marin et al., 2019) method may fail when object boundary is less informative, which is justified in the main results.

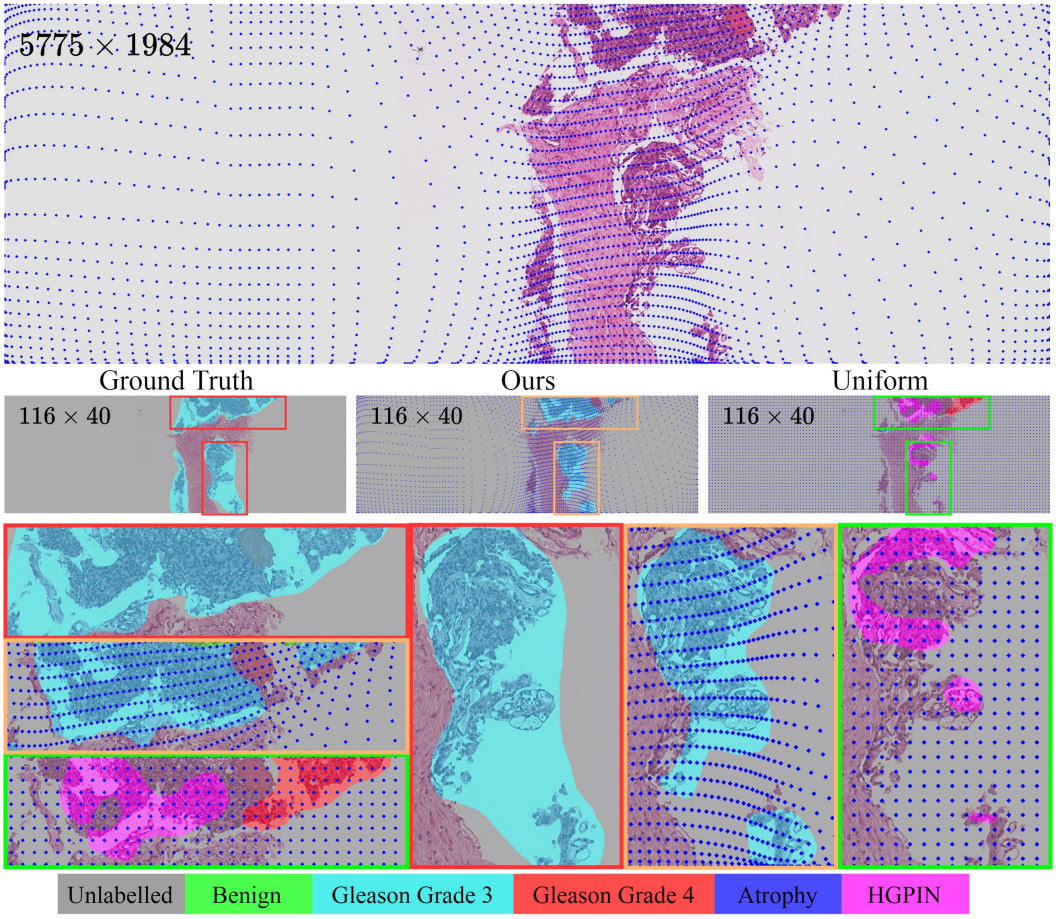


Figure 13: Qualitative example on PCa-Histo dataset (1) where segmentation performed on 50 times (at each dimension) downsampled image. Predictions are masked over and sampling locations are shown in blue dots.

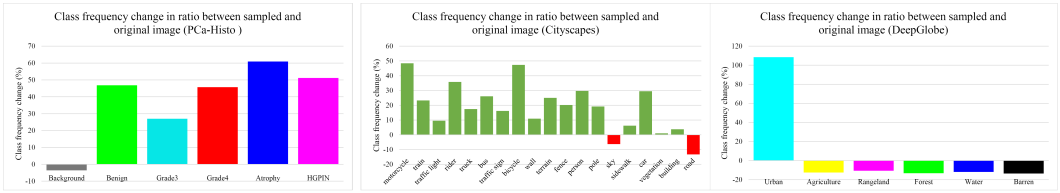


Figure 14: Class frequency change in the ratio between sampled and original image. The higher the value the denser the learnt sampling. Our method shows able to learn to adjust sampling either when foreground and background classes are clear (PCa-Histo and Cityscapes (Cordts et al., 2016)) and when less clear (DeepGlobe (Demir et al., 2018)). Results are from three datasets: (a) a local prostate cancer histology dataset (PCa-Histo) at dynamic downsampled size of 80×800 pixels (b) Cityscapes (Cordts et al., 2016) at downsampled size of 64×128 pixels and (c) DeepGlobe (Demir et al., 2018) at downsampled size of 300×300 pixels.

A.5 Stage training: how much contribution from the deformation module and the edge loss

To better understand the compounded contribution from the *segmentation loss* and the *edge loss*, we break the training into two stages, and at each stage we optionally switch on/off either the loss terms to separately investigate their contribution, on the Cityscapes (Cordts et al., 2016) dataset. First, we separately train each module with a single loss, referred to as "stage - single loss" in Figure 15. Specifically, we singularly train the downsampler, the *deformation module*, while freezing the

segmentation module in the first stage with single *edge loss*, and in the second stage, we freeze the *deformation module* and only train the *segmentation module* with single *segmentation loss*. The "stage - single loss" strategy is a two-stage training strategy that is most similar to the original "edge-based" (Marin et al., 2019) and its results represents the contribution of the "edge loss" term in separate training setting. On top of "stage - single loss", we modify the second stage by jointly train both module with both the *segmentation loss* and the *edge loss*, referred to as "stage - joint loss". The "stage - joint loss" result represents the contribution from the joint system. The results in Figure 15 shows that contribution from "edge loss" in a separately trained setting is limited, where "stage - single loss" under-performed than "uniform" by 6% in $mIoU$. While contribution from our joint trained system is significant, where "stage - joint loss" significantly improved "stage - single loss", and performs close to "uniform" and slightly better than "ours - single loss" and "edge-based".

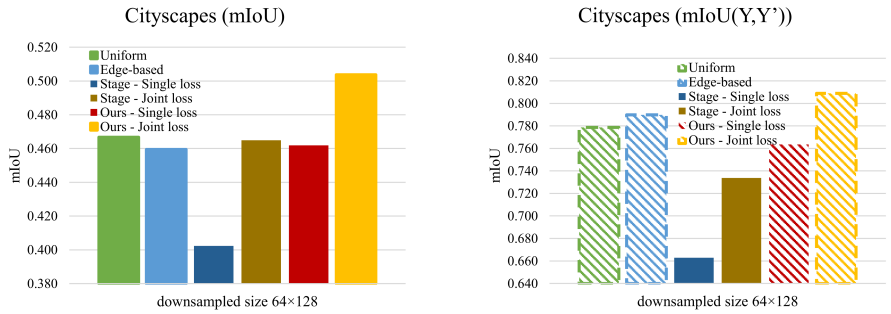


Figure 15: Comparison between two-stage training versus joint training with our approach. A key insight is the "edge loss" in a separate trained setting ("stage - single loss") does not work, while our joint trained system ("stage - joint loss") can significantly improve performance. Results for "uniform" and "edge-based" baselines are included as references. Results are from Cityscapes datasets at the downsampled size of 64×128.

A.6 SIMULATED "EDGE-BASED" RESULTS

We direct simulate different sampling density around edges by expanding sampling locations away from label edge at different distances in a Gaussian manner to mimic the effect of λ on sampling as shown by Marin et al. (2019). In specific, we calculate edge from label then apply gaussian blur to generate a simulated deformation map to guide sampling in our framework. As declared in the main text, our "edge-based" baseline is the best-performed segmentation network trained from a set of simulated "edge-based" downsamplers, each with a fixed sampling density around the edges. Here we present full segmentation accuracy with the set directly simulated "edge-based" downsamplers, for each of the three datasets in Figure 16.

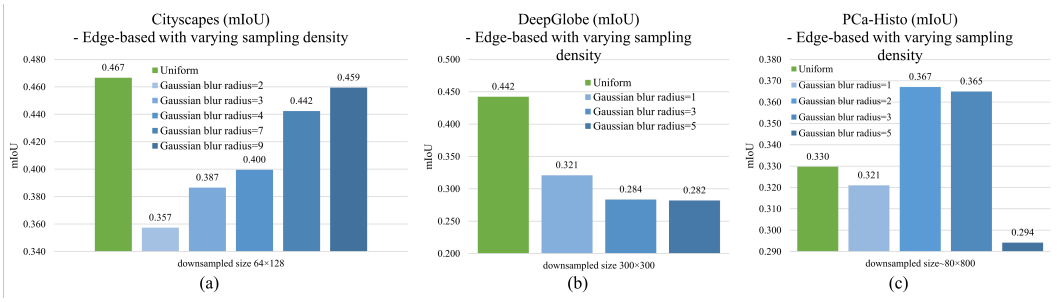


Figure 16: Simulated "edge-based" results, with a set of directly simulated "edge-based" downsamplers, each with a different fixed sampling density determined by Gaussian blur radius. The best-performed result is selected and referred to as the "edge-based" result presented in the main paper. Results are from three datasets: (a) Cityscapes (Cordts et al., 2016) (b) DeepGlobe (Demir et al., 2018) and (c) a local prostate cancer histology dataset (PCa-Histo).

A.7 DATASETS

In this work, we verified our method on three segmentation datasets: DeepGlobe aerial scenes segmentation dataset (Demir et al., 2018), Cityscapes urban scenes segmentation dataset (Cordts et al., 2016) and PCa-Histo medical histopathological segmentation dataset.

The **DeepGlobe (Demir et al., 2018)** dataset has 803 high-resolution (2448×2448 pixels) images of aerial scenes. There are 7 classes of dense annotations, 6 classes among them are used for training and evaluation according to Demir et al. (2018). We randomly split the dataset into train, validate and test with 455, 207, and 142 images respectively.

The **Cityscapes (Cordts et al., 2016)** dataset contains 5000 high-resolution (2048×1024 pixels) urban scenes images collected across 27 European Cities. The finely-annotated images contain 30 classes, and 19 classes among them are used for training and evaluation according to Cordts et al. (2016). The 5000 images from the Cityscapes are divided into 2975/500/1525 images for training, validation and testing.

The **PCa-Histo:** dataset contains 266 ultra-high resolution whole slide images, from 33 distinct biopsies. The size of the images ranged from 250800 pixels to 40880000 pixels. Each pixel is annotated into one out of six classes: Unlabelled, Benign, Gleason Grade 3, Gleason Grade 4, Atrophy, HGPIN (see Table 7 for explanations of each class). Among the six classes, informative classes are underrepresented comparing to Unlabelled class (see Table 7). We random split the dataset into 200 training, 27 validation and 39 test images. This dataset has a varying size, with height (the short axis) range between 229 to 2000 pixels (1967.5 ± 215.5 pixels), width (the long axis) range between 3386 to 20440 pixels (9391.9 ± 4793.8 pixels) and Aspect Ratio (AP) range between 1.69 to 33.9 (5.04 ± 3.54). The downsampling size of this dataset is calculated as follows: with a dynamic downsample size of for example 80×800, for each image, we calculate two candidates downsample sizes by either downsample the short axis to 80 or the long axis to 800 while in either case we keep AP unchanged. Then we select the one to downsample size that has fewer total pixels. For example, for an image with a size 368×7986 pixels, we have candidate downsample sizes of 200×4340 pixels or 36×800 pixels, and we will use downsample size of 36×800 in our experiment. Also we apply batch size per GPU node of 1 because of the varying downsampled size of each image.

Table 7: Pixel frequency distribution of PCa-Histo dataset and explanation of each class

Label	Background	Benign	Gleason Grade 3	Gleason Grade 4	Atrophy	HGPIN
% pixels in dataset	93.99	0.97	1.23	3.35	0.26	0.20
Explanation	all unlabelled pixels	healthy	mild cancer	aggressive cancer	healthy noise	precursor legion

The prostate biopsies of the PCa-Histo dataset are collected of 122 patients. The identity of the patients is completely removed, and the dataset is used for research. The biopsy cores were sliced, mounted on glass slides, and stained with H&E. From these, one slice of each of 220 cores was digitised using a Leica SCN400 slide scanner. The biopsies were digitised at 5, 10 and 20 magnification. At 20 magnification (which is the magnification used by histopathologists to perform Gleason grading), the pixel size is 0.55 μm^2.

A.8 NETWORK ARCHITECTURES AND IMPLEMENTATION DETAILS

Architectures: The *deformation module* is defined as a small CNN architecture comprised 3 layers each with 3×3 kernels follower by BatchNorm and Relu. The number of kernels in each respective layer is $\{24,24,3,1\}$. A final 1×1 convolutional layer to reduce the dimensionality and a softmax layer for normalisation are added at the end. All convolution layers are initialised following He initialization (He et al., 2015). The *segmentation module* was defined as a deep CNN architecture, with HRNetV2-W48 (Sun et al., 2019) applied in all datasets, and PSP-net (Zhao et al., 2017) as a sanity check in the Cityscapes dataset. The segmentation network HRNetV2-W48 is pre-trained on the Imagenet dataset as provided by Sun et al. (2019). In all settings, the size of the deformation map $\mathbf{d}$ is either consistent or twice the size of the downsampled image.

Training: For all experiments, we employ the same training scheme unless otherwise stated. We optimize parameters using Adam (Kingma & Ba, 2014) with an initial learning rate of 1×10^{-3} and $\beta = 0.9$, and train for 200 epochs on DeepGlobe, 250 epoches on PCa-Histo dataset, and 125 epochs on Cityscapes dataset. We use a batch size of 4 for the Cityscapes dataset (Cordts et al., 2016) and the DeepGlobe dataset (Demir et al., 2018) and batch size of 2 for the PCa-Histo dataset. We employ a step decay learning rate policy. We scale the *edge loss* 100 times so it is in the same scale with the *segmentation loss*. During the training of the segmentation network we do not include upsampling stage but instead, downsample the label map. For all datasets, we take the whole image to downsample as input, without any crop. During training, we augment data by random left-right flipping. All networks are trained on 2 GPUs from an internal cluster (machine models: gtx1080ti, titanxp, titanx, rtx2080ti, p100, v100, rtx6000), with syncBN.