



Universiteit  
Leiden

The Netherlands

## Using deep learning models in image processing

Xiong, Z.

### Citation

Xiong, Z. (2026, May 19). *Using deep learning models in image processing*.  
Retrieved from <https://hdl.handle.net/1887/4303646>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4303646>

**Note:** To cite this publication please use the final published version (if applicable).

## Chapter 5

# Advances in Kidney Biopsy Lesion Assessment through Dense Instance Segmentation

**Based on the following publication:**

Zhan Xiong, Junling He, Pieter Valkema, Tri Q. Nguyen, Maarten Naesens, Jesper Kers, and Fons J. Verbeek. Advances in Kidney Biopsy Lesion Assessment Through Dense Instance Segmentation. *Artificial Intelligence in Medicine*, 164:103111, 2025



## Chapter Summary

Renal biopsies are the gold standard for diagnosing kidney disease. Lesion scores made by renal pathologists are semi-quantitative and exhibit high inter-observer variability. Automating lesion classification within segmented anatomical structures can provide decision support in quantification analysis, thereby reducing inter-observer variability. Nevertheless, classifying lesions in regions-of-interest (ROIs) is clinically challenging due to (a) a large amount of densely packed anatomical objects, (b) class imbalance across different compartments (at least 3), (c) significant variation in size and shape of anatomical objects, and (d) the presence of multi-label lesions per anatomical structure. Existing models cannot address these complexities efficiently in a generic manner. This chapter presents an analysis for a **generalized solution** to datasets from various sources (pathology departments) with different types of lesions. Our approach utilizes two sub-networks: dense instance segmentation and lesion classification. We introduce **DiffRegFormer**, an end-to-end dense instance segmentation sub-network designed for multi-class, multi-scale objects within ROIs. Combining diffusion models, transformers, and RCNNs, DiffRegFormer is a computationally friendly framework that can efficiently recognize over 500 objects across three anatomical classes, i.e., glomeruli, tubuli, and arteries, within ROIs. On a dataset of 303 ROIs from 148 Jones' silver-stained renal Whole Slide Images (WSIs), our approach outperforms previous methods, achieving Average Precisions of 52.1% (detection) and 46.8% (segmentation). Moreover, our lesion classification sub-network achieves 89.2% precision and 64.6% recall on 21889 object patches out of the 303 ROIs. Lastly, our model demonstrates direct domain transfer to PAS-stained renal WSIs without fine-tuning.

**Keyword:** Renal Pathology   Dense Instance Segmentation   Diffusion Model   Regional Transformers   Multi-label Lesion Classification

## 5.1 Introduction

The evaluation of expert pathologists of kidney biopsies remains the gold standard for diagnosing and staging renal diseases [43]. Although biopsies digitalized into Whole Slide Images (WSIs)<sup>1</sup> have facilitated obtaining a visual morphological assessment of different anatomical structures for disease categorization, high-quality diagnostic assessments heavily depend on the correct lesion quantification manually annotated by pathologists across structures within a biopsy. Figure 5.1 shows an example of an annotated region-of-interest (ROI)<sup>2</sup> within a biopsy that contains hundreds of densely packed tissue objects. The annotation would cost a skilled expert around 2-4 hours for a complete biopsy. Due to the complexity and time-consuming nature of this task, there is a strong need for automated structure annotation and tools for lesion classification to facilitate further quantification, offload annotation time, and reduce intra- / inter-observer variability [9, 591].

Deep learning based instance segmentation algorithms have demonstrated significant capabilities on biomedical datasets [738, 42, 308, 562, 139, 782, 407, 474, 179]. However, developing a generic framework for renal pathology remains a challenge. This challenge can be broken down into four technical gaps and two practical issues. The four gaps (see Figure 5.1) are: (1) densely packed structures, up to 1000, per ROI; (2) considerable variation in size and shape of objects (e.g., arteries can be up to 100 times larger than that of tubuli); (3) class imbalance (e.g., the tubulointerstitial area occupies more than 70% on average in healthy and diseased renal parenchyma [42]); (4) each anatomical structure may present multiple lesions. The two practical issues are: (a) how to fuse multiple datasets with variation in staining to exploit scarce annotations fully; (b) readiness for extensibility, i.e., cost-effective adaptation to new lesion types from expanding datasets in clinical scenarios. Simultaneously addressing these difficulties requires a universal framework that includes efficiency, staining style (domain) transfer [187, 641, 316], and flexibility for continuous learning [830, 140].

Prior studies have demonstrated strong capabilities for lesion classification via dense instance segmentation, but their paradigms lack scalability and adaptability to datasets with potential changes in lesion composition. Some approaches [253, 42, 562, 437, 139] adopted a two-step process: semantic segmentation followed by dataset-specific post-processing to obtain final instance masks, which cannot be scaled to large-scale datasets

<sup>1</sup><https://www.mbfbioscience.com/whole-slide-imaging-analysis/>

<sup>2</sup>To prevent any confusion with 'RoI' in RCNNs, we will hereafter use the term 'ROI' solely in the context of renal biopsies. To maintain clear distinctions in RCNN discussions, we will adopt the terms 'candidate regions' or 'regional proposals'.

## 5.1. Introduction

---

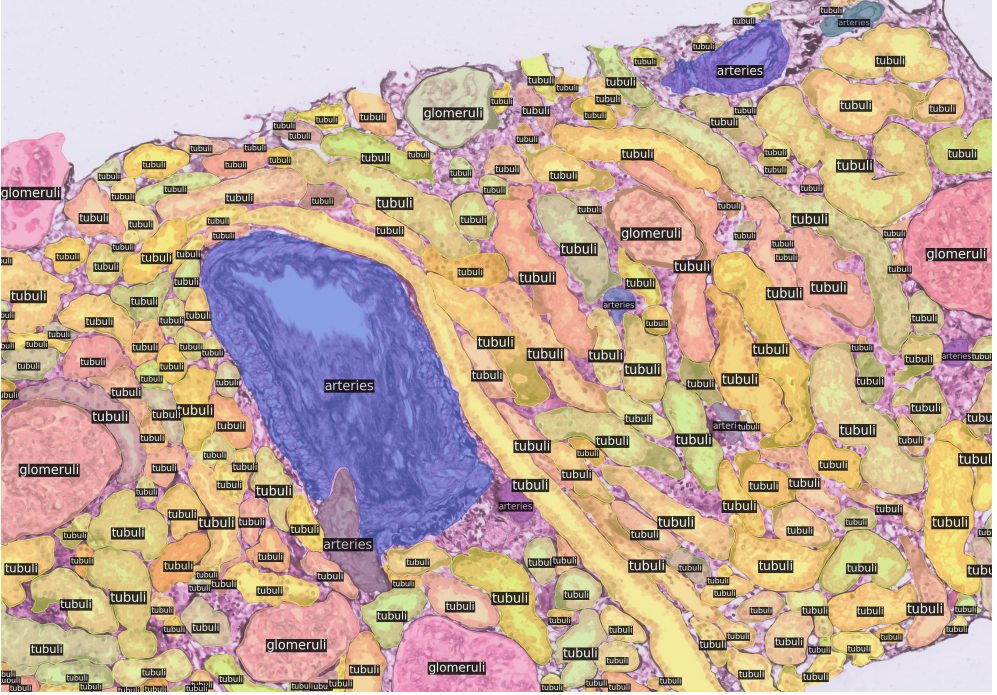
with dense objects of various shapes. Besides, each lesion is assigned to a single semantic class, leading to difficulty with multi-label lesion classification and potential changes in lesion combinations. Specifically, adding or removing lesions requires a complete redesign and retraining of the model. Lastly, increasing the number of lesion types requires adding more segmentation maps for prediction, which is significantly resource-intensive.

Detection-based models with regional convolution neural networks (RCNNs) [301, 308] can efficiently detect dense anatomical structures and separately design lesion classification heads for each class. They can therefore adopt a plug-and-play mechanism and adapt to lesion changes by replacing the corresponding sub-modules and maximally reusing the others. However, these variants are not scalable to multi-class objects with varying scales and shapes because they rely on predefined bounding box anchors, limiting their application to a single class with less variation in scale and shape, e.g., glomeruli.

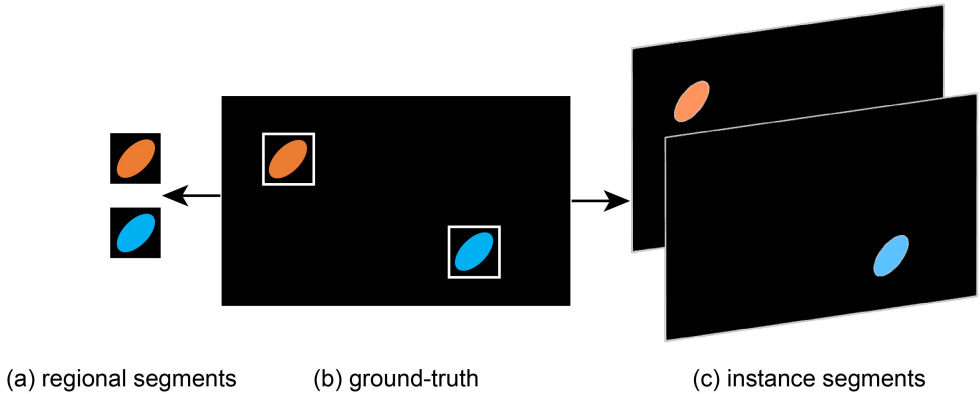
Recently, transformers have been the prevailing anchor-free approach for multi-class object detection across various scales and shapes. Transformer-based instance segmentation leverages attention mechanisms and learns latent representative embeddings (queries) from global contextual features to process highly varied objects. However, existing models [53, 105, 573] are resource-intensive for dense structures in large-scale datasets. That is due to two shortages: (1) a large number of static queries from one embedding per object; (2) low instance map occupancy from one object per instance map (depicted in Figure 5.2c). Hence, they are limited to processing classes with sparse objects, like glomeruli.

In response to these challenges, we propose a generic and extensible system for dense instance segmentation and lesion classification on large-scale datasets, robust to changes in lesion combinations. Our design has two key components: (a) A novel dense instance segmentation sub-network that recognizes basic anatomical structures, i.e., glomeruli, tubuli, and arteries; (b) a lesion classification sub-network with a set of independent heads that predict lesions for each class. It is crucial to separately modularize lesion classification and dense instance segmentation for the big picture. The segmentation of basic anatomical structures is a universal foundation for all visual assessment systems. However, lesion classification is a task-driven downstream application that may vary depending on the interests of users. Therefore, the robustness of dense instance segmentation to local changes in lesion classification heads is beneficial for the extensibility of our system and for expanding datasets in clinical scenarios.

More specifically, for dense instance segmentation, we propose a novel end-to-



**Figure 5.1:** An illustration depicts a manually annotated ROI of a kidney biopsy, focusing on **glomeruli**, **tubuli**, and **arteries**. There are three primary challenges encountered in clinical renal biopsies: (1) a large number of objects closely touching each other; (2) the significant variation in size and shape among different instances; (3) the distribution of classes is heavily biased.



**Figure 5.2:** Comparison between regional segments and instance segments. (a) cropped instance maps from bounding boxes tightly surrounding each object; (b) ground-truth bounding boxes and instance segments within one image; (c) separated entire instance maps per object.

## 5.1. Introduction

---

end approach, named **DiffRegFormer**, which effectively combines the advantages of diffusion models, RCNNs, and transformers to tackle dense objects across multiple classes and scales. Diffusion models [260] generate bounding box proposals from Gaussian noise, and therefore eliminating the need for pre-defined anchors; RCNNs efficiently crop dense instance maps into dense regions which have very high occupancy rate w.r.t. the bounding boxes (see Figure 5.2. (a)); Cross-attention mechanisms with dynamic queries [65, 367, 808, 109, 368] extract long-range contextual features and enables robust representation of objects across varying scales. DiffRegFormer is not just a simple assembly of the aforementioned techniques. Such assemblies may suffice for sparse instance segmentation on datasets like MSCOCO [415]. They, however, fail in the context of dense instance segmentation of kidney biopsy ROIs. The crux lies in the proposals, which are essentially Gaussian noise during the early training stage of diffusion models. Those noisy candidates, consequently, hinder training due to accumulating errors. Our ablation study analyzes these effects in detail (see section 5.4.4) and highlights the importance of our specific designs. Specifically, we introduce the following **key innovations** to address the challenges for dense instance segmentation:

- **Regional features:** Similar to RCNNs, feature maps are cropped using generated proposals and converted to dynamic queries for efficient long-range dependency modeling; this is robust to large-scale variations.
- **Feature disentanglement:** Instead of using shared feature maps, we redesign the model structure to generate separate feature maps for the bounding box decoder and mask decoder; this is crucial to stabilizing the training of the mask decoder.
- **Class-wise balanced sampling:** Unlike conventional sampling methods, we propose a novel sampling approach. The key difference is to select **class-wise balanced** positive samples among the **ground-truth** boxes instead of the proposal ones.

In conclusion, based on these key designs, the proposed model is a generic and extensible approach for large-scale datasets with minimal overhead. To the best of our knowledge, DiffRegFormer is the first end-to-end framework that combines a diffusion-based method with a transformer in an RCNN-style architecture to efficiently process dense objects within ROIs for multi-scale and multi-class tasks. Figure 5.3 depicts a flow chart of the dense instance segmentation sub-network.

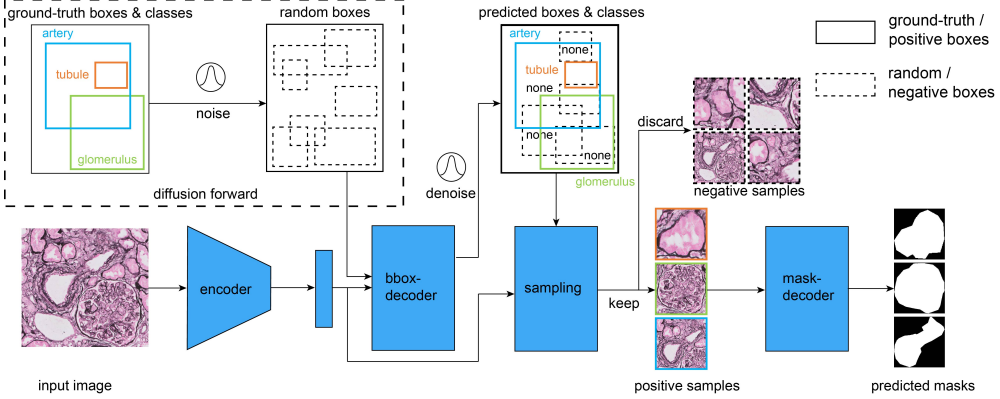
Each anatomical structure has a dedicated classification head for the lesion classifier, since lesion classes can vary across datasets. In Figure 5.6, we input the croppings of anatomical structures and resized them to the same size. Each head is trained only with patches of the specific structures and learns to predict multi-label lesions over each class. Our dataset currently only contains sclerotic glomeruli and atrophic tubuli. However, the lesion classifier is extensible to the expanded large-scale dataset at minimal cost, because adding, removing, or replacing heads does not disrupt the other modules.

Our work aims to propose a **generic technical solution** that has flexibility and scalability for automating lesion classification on dense anatomical structures in clinical scenarios. It works as the foundation for further quantification analyses. The main contributions of this chapter are the following:

- We propose the first end-to-end dense instance segmentation that effectively combines diffusion models, RCNNs, and transformers for multi-class, multi-scale objects. More importantly, our model can directly process ROIs of kidney biopsies, avoiding patch splitting.
- We propose novel designs to address the accumulative errors caused by diffusion models at an early stage and stabilize the training process.
- We propose a novel class-wise balance sampling method to improve detection and segmentation performance.
- Instead of the static queries used in transformers, we convert regional features into dynamic queries and model the long-range dependencies between regional features while avoiding performance deterioration on dense objects.
- We compare DiffRegFormer with previous models that can process multi-class and multi-scale objects within ROIs in an end-to-end manner. Our model outperforms the previously published models in evaluating Jones’ silver-stained images.
- We show that DiffRegFormer has the potential of stain (domain)-agnostic detection for PAS-stained images without stain-specific fine-tuning.
- Our lesion classifier can achieve multi-lesion classification. In addition, our plug-and-play strategy can flexibly adapt to large-scale datasets while accommodating potential changes in lesion combinations with minimal overhead.

The remainder of the chapter is organized as follows. In section 5.2, all relevant work is introduced, and we elaborate on all improvements w.r.t. previous research. In section

## 5.2. Related Work



**Figure 5.3:** Our **DiffRegFormer** is a one-stage anchor-free method. Instead of predefined anchors, we add Gaussian noise to ground-truth boxes and generate a fixed-sized set of random bounding boxes (bbox). With feature maps extracted from the encoder, the bbox-decoder iteratively learns to denoise and predicts class-wise candidate boxes. Due to the uneven distribution of candidates, we propose a sampling module that effectively discards negative samples while maintaining a balanced proportion of positive samples to facilitate fast convergence. Finally, we generate the output instance masks based on the selected positive samples in the mask-decoder. For simplicity of illustration, we choose only one anatomical object per class (artery, tubule, glomerulus).

5.3, we describe each component of our model in detail. In section 5.4, we demonstrate the extensive evaluation of our model, including comparison experiments and ablation studies. In section 5.5, we describe the advantages and limitations of our pipeline and show possible future follow-up research.

## 5.2 Related Work

**Semantic Segmentation with Post-Processing:** In the context of lesion classification of dense structures for renal biopsies, semantic segmentation with post-processing has drawn substantial attention. Previous studies [253, 42, 562, 139] treat anatomy-wise lesions (e.g., atrophic tubuli) as semantic classes alongside basic anatomical structures (e.g., arteries, glomeruli). An auxiliary **border** class is often used during training to facilitate splitting semantic masks into distinct instances. This approach has demonstrated notable performance. However, challenges are faced in scalability and adaptability to large-scale datasets with potential changes in lesion composition. These issues are at hand as (1) Reliance on ad-hoc procedures (e.g., thresholding and morphological operations [200]) for border generation reduces generalizability. (2) Due to the inflexibility of dataset change, any modification to lesion types necessitates

model redesign and retraining. (3) Computational cost scales with lesion classes in the dataset. (4) The difficulty in modeling long-range correlations between objects of varying size due to splitting ROIs into tiles. (5) It cannot tackle multi-label lesion classification. In contrast to these issues, our end-to-end framework addresses them by directly modeling dense objects within ROIs using attention mechanisms.

**Diffusion Model:** Diffusion models, a class of deep generative models [260, 492], learn to approximate complex distributions through iterative denoising. Despite success in data generation [278, 499, 374], natural language processing [380], audio processing [518], and self/weakly-supervised learning [170], their application to dense instance segmentation remains limited. Initial attempts have been to introduce diffusion models to segmentation tasks [85, 209]. The adaptation to dense instance segmentation, however, remains challenging. From our ablation studies, we formulate two primary challenges: (1) The potential dominance of the diffusion modules over feature learning in shared feature maps destabilizes the training of instance prediction modules. (2) Accumulative errors from the diffusion module’s early training stages impair the accuracy of dense object predictions. Our approach introduces innovative strategies to overcome these challenges, enabling the first successful implementation of a diffusion model for dense instance segmentation.

**Transformer:** Transformer-based approaches introduce an anchor-free paradigm in an end-to-end manner for instance segmentation, leveraging their capability to capture long-range dependencies and aggregate contextual information. In general, the dependency extraction process is called an attention mechanism, and each context feature is treated as a query for object representation. Early transformer-based methods [53, 105, 573] rely on static queries to model all objects within a dataset. These approaches are incompatible with dense object scenarios because the static query set inevitably expands with dataset size. Recent works [367, 808, 109, 368] advancement towards dynamic queries, allowing for temporary modeling of objects within a mini-batch. In this manner, the complexity of processing dense objects is reduced. However, their instance segment representation remains inefficient for dense instance segmentation<sup>3</sup> due to the expensive representation of instance segments (Figure 5.2.(c)) with low occupation rate. Our DiffRegFormer addresses this by integrating region proposals with transformers for improved representation efficiency.

---

<sup>3</sup>We conducted experiments on our dataset from the official code of both. Compared to our approach, they required more than twice the GPU memory at a batch size of 2 and 500 queries. That is not infeasible for applications with dense objects.



### 5.3. Methods

---

**RCNN:** The Regional Convolutional Neural Network (RCNN) framework [197, 545] effectively generates candidate regions (i.e., proposals) for dense objects. In instance segmentation, RCNN-based methods use mask modules to predict binary instance masks within proposals. Early variants [245, 284, 301, 308] relied on class-agnostic proposals followed by class-specific mask prediction, constrained by predefined anchors. More recent approaches [177] have explored the mapping from proposals to static queries, employing cross-attention mechanisms for instance mask prediction. However, this paradigm faces scalability issues in dense object segmentation due to the computational overhead associated with static queries (see QueryInst in Table 5.2). Our method adopts dynamic queries, balancing computational efficiency and scalability for dense instance segmentation.

In summary, our work combines diffusion modeling and attention mechanisms into an RCNN-style framework that addresses the scalability, adaptability, and efficiency challenges inherent to existing approaches for dense instance segmentation in renal biopsies.

## 5.3 Methods

In this section, we first introduce the basic conceptions for the diffusion models. Then, details of all core components are elaborated. Finally, implementation information is added for the training and inference stages.

### 5.3.1 Preliminaries

**Diffusion model.** Diffusion approaches [583, 548] are a family of deep generative models inspired by the principles of non-equilibrium thermodynamics [581, 278]. Their operation is conceptualized as a  $T$ -step sequential process that iteratively transitions from an initial state  $Z_0$  to a final state  $Z_T$ , which is pure noise. Diffusion forward involves progressively adding Gaussian noise at each transition. Due to the special properties of the Gaussian distribution [260], it is possible to sample any intermediate noisy state  $Z_t$  directly:

$$q(Z_t|Z_0) = \mathcal{N}(Z_t|\sqrt{\bar{\alpha}_t}Z_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (5.1)$$

where  $Z_0$  is the original data,  $Z_t$  ( $t \leq T$ ) is the latent state at step  $t$ , and  $\bar{\alpha}_t$  is the cumulative noise variance schedule [85]. Notably, the diffusion forward process does

not introduce trainable parameters. The objective of a diffusion model is to learn the reverse process. A neural network,  $f_\theta(Z_t, t)$ , approximates the recovery of the original distribution  $Z_0$  from a noisy state  $Z_t$ . Training optimizes this network by minimizing a loss function to denoise at an arbitrary intermediate state  $t$ :

$$\mathcal{L} = \frac{1}{2} \|f_\theta(Z_t, t) - Z_0\|^2 \quad (5.2)$$

where  $Z_0$  represents the ground-truth and  $Z_t$  is the noisy input. During inference, the model  $f_\theta$  reconstructs from pure noise  $Z_T$  to the predicted data  $Z_0$  through a series of reverse updates [583] with a step-size  $s$ :  $Z_T \rightarrow Z_{T-s} \rightarrow \dots \rightarrow Z_0$ . See [583] for a detailed derivation.

### 5.3.2 Dense Instance Segmentation Model

Iterative proposal generation in DiffRegFormer requires multiple model executions during inference. That poses a computational burden if the entire model runs on the raw image at each training step. To address this issue, we decouple the image encoder from the remaining modules. This strategy enables the encoder to extract multi-scale feature maps from the raw input image only once. Then, subsequent modules operate iteratively to refine box and mask predictions from an initial state of Gaussian noise, conditioned on these deep features. Moreover, to facilitate the mask-decoder training, we introduce a novel sampling module that randomly selects positive samples from each class, ensuring a balanced distribution of class-wise regional features. That facilitates the learning of object representations at lower frequencies in the dataset.

In summary, the DiffRegFormer comprises four components:

- **Encoder:** Processes the raw input image once to extract multi-scale feature maps for the bbox-decoder and mask-decoder separately.
- **Bbox-decoder:** Utilizes the extracted feature maps to iteratively refine bounding box predictions.
- **Sampling Module:** Stabilizes the training by randomly selecting class-wise balanced positive samples for the mask-decoder.
- **Mask-decoder:** Iteratively refines mask predictions based on the sampled regional features.

### 5.3. Methods

---

#### Encoder

The image encoder extracts high-level feature maps from the raw image for subsequent decoders. To generate multi-scale feature maps, it utilizes a ResNet [248] pre-trained on ImageNet [138] as its backbone, followed by a Feature Pyramid Network (FPN) [414]. After FPN, a stack of  $3 \times 3$  convolutional operations is applied to produce two separate feature maps, thereby decoupling the bbox-decoder and mask-decoder.

#### Bbox-decoder

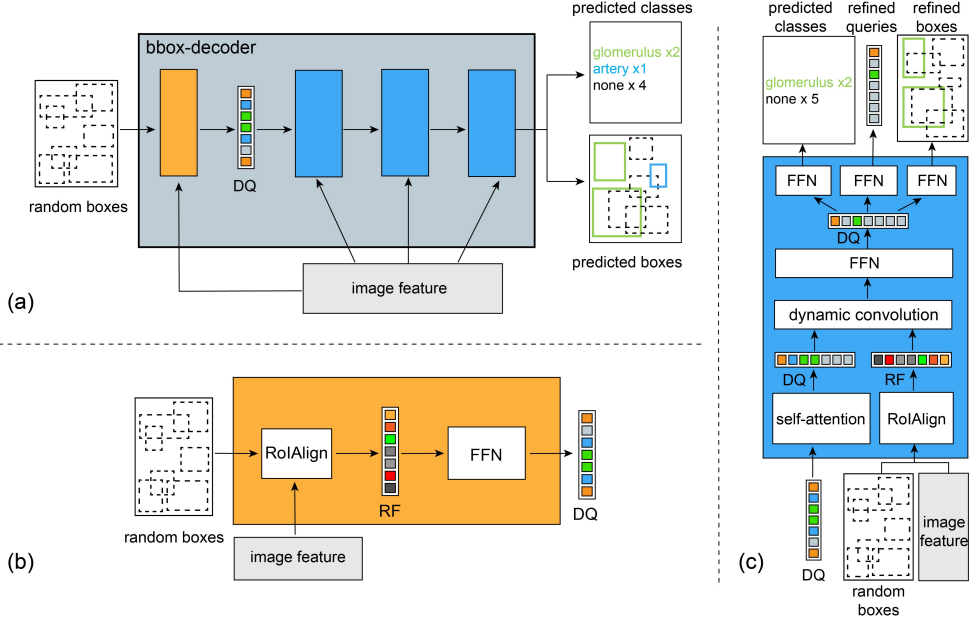
The bbox-decoder builds upon DiffusionDet [85]. As shown in Figure 5.4 (a), a set of random boxes and multi-scaled feature maps is first taken from the encoder as input. Then, dynamic queries (Figure 5.4b) are initialized from a regional pooling operator (RoIAlign [245]) and a feed-forward network (FFN) [642]. These queries undergo iterative refinement through multiple stages. Each stage takes proposal boxes and dynamic queries from the previous stage, generating refined dynamic queries, box predictions, and class predictions for the next stage. Figure 5.4c illustrates one detailed refinement stage. First, new regional features (cropped by RoIAlign) interact with dynamic queries (after a self-attention module) via dynamic convolution [97], emphasizing regions likely containing objects while suppressing others. Next, the enhanced regional features are processed by FFNs to produce refined box predictions, box-wise classifications, and queries, respectively.

#### Sampling

Conventional RCNN models use intersection over union (IoU) to classify a proposal box as positive if its IoU with a ground-truth box is  $\geq 0.5$  [245]. This approach, however, is ill-suited to diffusion methods, particularly in early training stages when they predominantly produce noise. That can complicate convergence or even cause failure due to error accumulation. Further, conventional IoU-based methods are susceptible to class imbalance, biasing the learning process towards frequently occurring objects and neglecting rare instances. We therefore propose a novel sampling method to address these issues. We first replace proposal boxes with ground-truth boxes to ensure a sufficient supply of positive samples. That is justified because the bbox-decoder aims to predict boxes that closely approximate the ground-truth. We then divide the ground-truth bounding boxes into groups by class. Within each group, we randomly select up to  $n$  boxes as positive samples without repetition. The core idea is that objects of each class that have similar shapes and appearances can be mapped to a cluster. Consequently, it allows us to learn instance feature representations with a

small sample size  $n$ . The rule especially applies to the tubular class, which accounts for most of our dataset. This sampling strategy offers three distinct advantages:

1. **Rare Instance Learning:** The mask-decoder can effectively learn representations of infrequently occurring instances whenever they appear in the input, mitigating the risk of being overwhelmed by more common classes.
2. **Randomize Frequent Instance:** Learning on a randomized subset of frequent instances prevents model overfitting.
3. **Training Stability:** Employing selected ground-truth boxes as positive samples promotes stability and accelerates mask training.

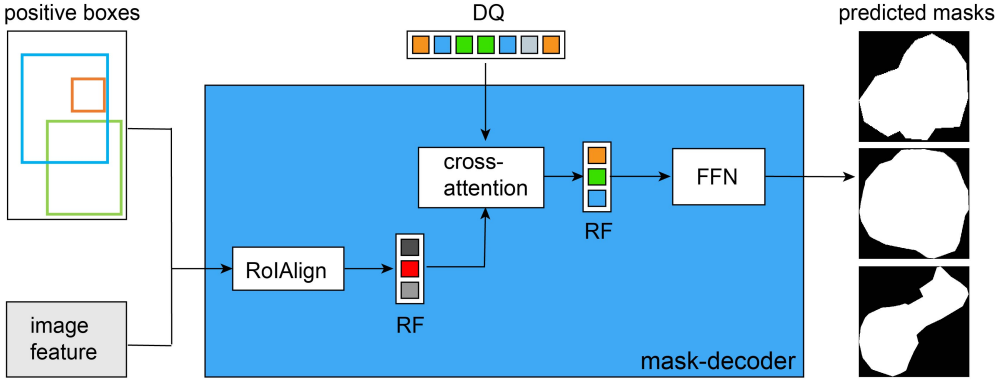


**Figure 5.4:** The bounding box (bbox)-decoder takes multi-scale feature maps and a set of random boxes as input. Then, the predicted classes and boxes will be output iteratively. (a) The module comprises an initialization head for dynamic queries (orange rectangle) and multiple box refinement heads (blue rectangles). (b) In the orange rectangle in (a), the initial dynamic queries are generated via a RoIAlign pooling operator and a feed-forward network (FFN). (c) Each box refinement module (one blue rectangle in (a)) takes the previous stage's dynamic queries and proposal boxes as input, generating predictions and refined dynamic queries for the next stage. Abbreviations: dynamic queries: DQ, feed-forward network: FFN, regional features: RF.

### 5.3. Methods

#### Mask decoder

The mask decoder iteratively refines mask predictions by using regional features cropped from multi-scale feature maps, based on boxes selected by the sampling module. Instead of using static queries [105, 106], we compute cross-attention between dynamic queries and positive regional features. That highlights effective queries and filters, enhancing regional features by focusing on pixels within positive regions, thereby facilitating accurate per-pixel instance score prediction (Figure 5.5).

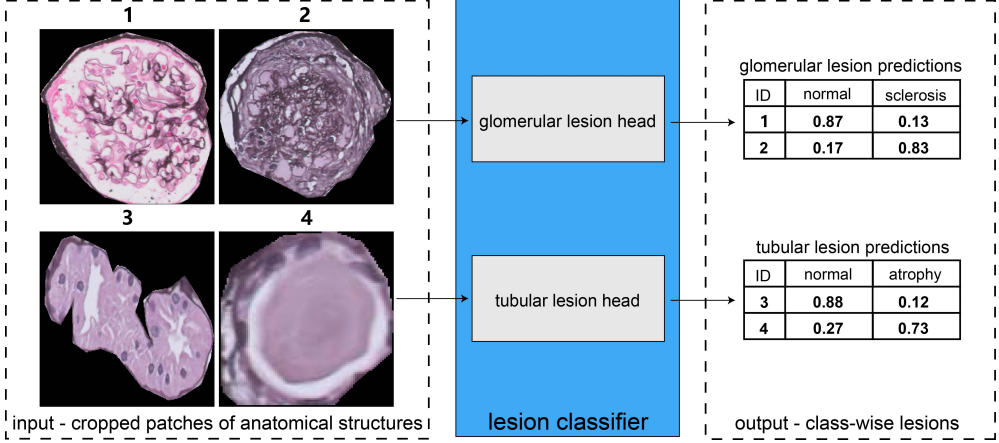


**Figure 5.5:** The mask-decoder takes multi-scale feature maps and a set of positive boxes as input and predicts instance masks. The dynamic queries interact with regional features via cross-attention and highlight only pixels within proposal boxes. Final instance masks are generated from the enhanced regional feature maps. Abbreviations include DQ: dynamic queries, FFN: feed-forward network, and RF: regional features.

#### 5.3.3 Lesion classifier

The lesion classifier comprises multiple independent lesion prediction heads. Each head specializes in a specific anatomical structure (e.g., the glomerulus) and processes cropped images of that structure to predict probabilities for multi-label lesions. Figure 5.6 illustrates this design. The architecture of a single head consists of three convolutional blocks, followed by a max pooling operator. Each block contains a  $3 \times 3$  convolution, batch normalization, and ReLU activation. We currently implement two prediction heads: one for glomeruli (predicting sclerosis) and one for tubuli (predicting atrophy).

Notably, the modular design of the instance segmentation component allows lesion prediction heads to operate as independent plug-ins. This modularity is crucial for seamlessly adapting the model to the expanding datasets, particularly in clinical scenarios. It allows for the straightforward integration or removal of prediction heads



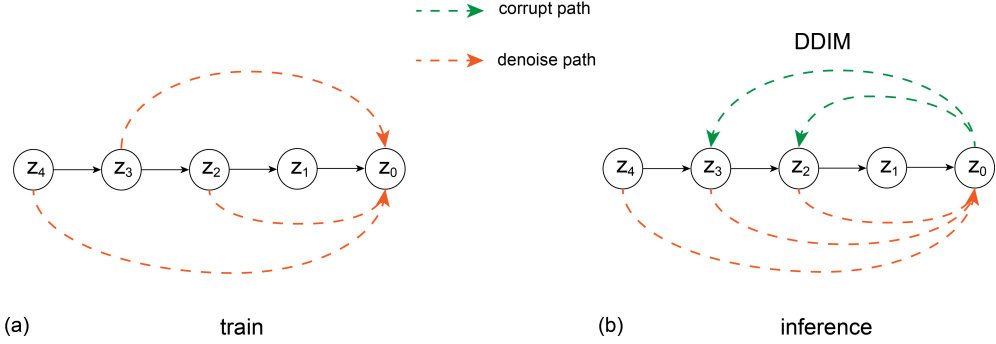
**Figure 5.6:** The lesion classifier consists of 2 prediction heads, one dedicated to glomeruli and the other to tubuli. It takes croppings of anatomical structures and outputs the probability of class-wise lesions.

with minimal impact on the overall model architecture. For example, if new annotations for arteries become available, a dedicated prediction head can be trained to classify arterial lesions and integrate them into the model without modifying existing components. Likewise, this design extends the model’s diagnostic scope to include additional lesions within a given structure (e.g., tubuli) by simply replacing an existing prediction head with a newly trained one on expanded data. This design promotes the model’s versatility and long-term adaptability within clinical scenarios where the scope of identifiable lesions may increase.

### 5.3.4 Implementation Details

The entire model employs distinct strategies for training and inference. During training, the DiffRegFormer learns to reconstruct ground-truth boxes through adaptation at state 0 from an arbitrary noisy state  $t$  (Figure 5.7(a)). Additionally, the sampling module provides class-balanced ground-truth boxes to facilitate the mask decoder’s training. Lastly, the lesion classifier operates on croppings generated from ground-truth instance masks. Each prediction head specializes in classifying lesions specific to the corresponding anatomical structure within these croppings. At the same time, during inference, the DiffRegFormer assembles multiple predictions to refine final results (Figure 5.7(b)). Moreover, the sampling module is deactivated, and all proposals are directly forwarded to the mask decoder for instance segmentation prediction. Likewise, the lesion classifier utilizes instance masks predicted by DiffRegFormer

### 5.3. Methods



**Figure 5.7:** An illustration for a diffusion model at different stages depicts a simplified process with 5 states (skip step=1), where state 4 is pure Gaussian noise and state 0 is the ground-truth. (a) During training, a diffusion model randomly selects denoising paths to train its decoder; (b) In inference, the Denoising Diffusion Implicit Model (DDIM) flexibly alternates between denoising and corrupting paths, generating multiple predictions that are then ensembled for a refined final result.

to crop structures. As a final step, each prediction head outputs probabilities for class-specific lesions within the cropped regions.

#### Training

We start by constructing noisy boxes from the ground-truth. The DiffRegFormer is then trained to recover bounding boxes and iteratively generate instance masks from an arbitrary state  $t$ . The algorithm is detailed in Algorithm 1.

**Diffusion forward.** Ground-truth boxes are first concatenated with additional random boxes to align to a fixed length across images, accounting for the variability in instance counts. Gaussian noise is then introduced as the noise scale governed by  $\bar{\alpha}$  as per Eq.(5.1). Following [492], the scale varies with state  $t$  using a monotonically decreasing *cosine* schedule. It should be noted that, for optimal performance, the signal-to-noise ratio requires a relatively high signal scaling value compared to image generation tasks [88, 260, 144, 90, 85].

**Training Loss.** The box decoder takes noisy boxes as input and predicts category labels and box coordinates. Given the one-to-many relationship between ground-truth and predicted boxes, first, a set-loss function is employed to match predictions to the ground-truth based on the lowest cost [53, 602, 870]. Furthermore, the mask decoder inputs sampled positive boxes and predicts instance masks. We use a binary cross-entropy loss function due to the one-to-one mapping. Finally, since multiple lesions can occur within a single crop, lesion classification uses a binary cross-entropy

loss function for multi-label prediction within each head.

**Sampling.** Our implementation uses 3 classes and a maximum of  $N = 500$  objects per image. We set up to  $n = \frac{500}{3} \simeq 166$  instances per class. More specifically, positive ground-truth boxes are sampled without replacement, ensuring they do not exceed the class-wise maximum. Moreover, we disregard classes that are absent in the input. For example, if there are 2 glomeruli, 0 vessels, and 200 tubuli ground-truth boxes within an ROI, then the sampling module would select 2 glomeruli, 0 vessels, and 166 random tubuli as positive samples without duplication.

---

**Algorithm 1:** DiffRegFormer Training

---

```
def train(images, gt_boxes, gt_masks):
    # images: [B,H,W,3], gt_boxes: [B,*,4], gt_masks: [B,H,W,G]
    # B: batch size, N: number of proposal boxes

    # generate multi-scale features via encoder
    feats = encoder(images)
    # generate noised boxes at state t, where t is a random integer
    # in diffusion forward; Pad boxes to N with t_boxes: [B,N,4]
    t_boxes, t = diffusion(gt_boxes, mean=0, std=1)
    # generate initial dynamic queries
    d_query = initialize_query(t_boxes, feats)
    # learn to reverse noised boxes at state t back to ground-truth
    # boxes at state 0 and return refined dynamic queries
    [pred_boxes, d_query] = box_decoder(t_boxes, feats, d_query, t)
    # obtain bbox loss via the set objective function
    loss_bbox = set_loss(pred_boxes, gt_boxes)
    # randomly select balanced positive boxes from ground-truth boxes
    pos_boxes = sampling(gt_boxes)
    # generate predicted instance masks
    pred_masks = mask_decoder(pos_boxes, feats, d_query)
    # obtain mask loss via the set objective function
    loss_mask = set_prediction_loss(pred_masks, gt_masks)
    return loss_bbox, loss_mask
```

---

## Inference

During inference, the DiffRegFormer starts from a standard Gaussian distribution corresponding to the final state  $T$  as defined in Eq.(5.1). Then, a progressive denoising operation reverses the predictions to the initial state 0. The algorithm, as depicted in Algorithm 2, outlines this process.

**Inference steps.** Each inference iteration involves two primary operations. First, the box decoder processes noisy boxes generated from the previous state  $t_{\text{now}}$  and predicts both categories and box coordinates. Subsequently, a Denoising Diffusion



## 5.4. Methods

---

Implicit Model (DDIM) [480, 583] introduces noise into the previously predicted boxes. This step generates new noisy boxes for the subsequent state  $t_{\text{next}}$ , thereby facilitating progressive denoising across states. Notably, each iteration performs a single denoising operation, transitioning from state  $t_{\text{now}}$  to  $t_{\text{next}}$ . This characteristic allows for the assembly of intermediate predictions from multiple denoising steps (i.e.,  $t_i \rightarrow 0$  where  $i \in [1, T]$ ) to refine the outcome. However, DiffRegFormer performs inference only once to improve efficiency, directly transitioning from pure Gaussian noise at state  $T$  to state 0 to obtain the final predictions.

**Box replacement.** Predicted boxes can be categorized as either *positive* (scores above a threshold, containing objects of interest) or *negative* (scores below the threshold, arbitrarily located). Applying negative boxes directly to DDIM would significantly degrade the quality of newly generated noisy boxes. This issue stems from negative boxes that originated from corrupted boxes during the training phase, which significantly deviate from the Gaussian distribution. To ensure consistency between the training and inference phases, we substitute negative boxes with random boxes sampled from a Gaussian distribution.

---

**Algorithm 2:** DiffRegFormer Inference

---

```
def infer(images, step, T):
    # images: [B,H,W,3], step: skip length of state transform
    # T: chain length, B: batch size, N: proposal box number

    # generate multi-scale features via encoder
    feats = encoder(images)
    # return Gaussian noise as noisy boxes at state T;
    # Pad boxes to N with T_boxes: [B,N,4]
    [t_boxes, _] = diffusion(mean=0, std=1)
    # generate state transform pairs skipping every step
    # [(T, T-step), (T-step, T-2*step), ..., (step, 0)]
    time_pairs = uniform(0, T, step)
    # generate initial dynamic queries
    d_query = initialize_query(t_boxes, feats)
    # iterate over stages
    for (t_now, t_next) iterate t_pairs:
        # predict boxes and dynamic query at state t_now
        [pred_boxes, d_query] = box_decoder(t_boxes, feats, d_query, t_now)
        # generate new noisy boxes from state t_now to t_next
        t_boxes = ddim(t_boxes, pred_boxes, t_now, t_next)
        # replace undesired boxes with random Gaussian noise
        t_boxes = box_repalce(t_boxes)
    # generate predicted instance masks
    pred_masks = mask_decoder(pred_boxes, feats, d_query)
    return pred_boxes, pred_masks
```

---

## 5.4 Results

This section begins with an overview of the kidney biopsy dataset used in this study. We then make a fair comparison between DiffRegFormer and established end-to-end instance segmentation models specifically designed to handle dense, multi-class, and multi-scale objects at the ROI level. In addition, to include retrospective reflection in our analysis, we have compared our model with a semantic segmentation based method [253]. The results of this comparison are presented in section 5.4.2 and the tables therein. All benchmark methods are reproduced utilizing the **mmdetection** package [75] to ensure consistency. Further, we evaluate the performance of our lesion classifier on cropped images of anatomical structures derived from the instance masks predicted by DiffRegFormer. Additionally, we have conducted a complexity analysis of DiffRegFormer to identify an optimal balance between its performance and computational burden. Finally, we present extensive ablation studies on DiffRegFormer that evaluate the impact of various training strategies.

| Source Center | Scanner      | WSIs | ROIs | Train Dataset | Eval Dataset |
|---------------|--------------|------|------|---------------|--------------|
| LUMC          | Philips UFS  | 65   | 142  | 130           | 12           |
| AUMC          | Philips UFS  | 42   | 80   | 74            | 6            |
| UMCU          | Hamamatsu XR | 25   | 49   | 45            | 4            |
| Leuven        | Philips UFS  | 16   | 32   | -             | 32           |

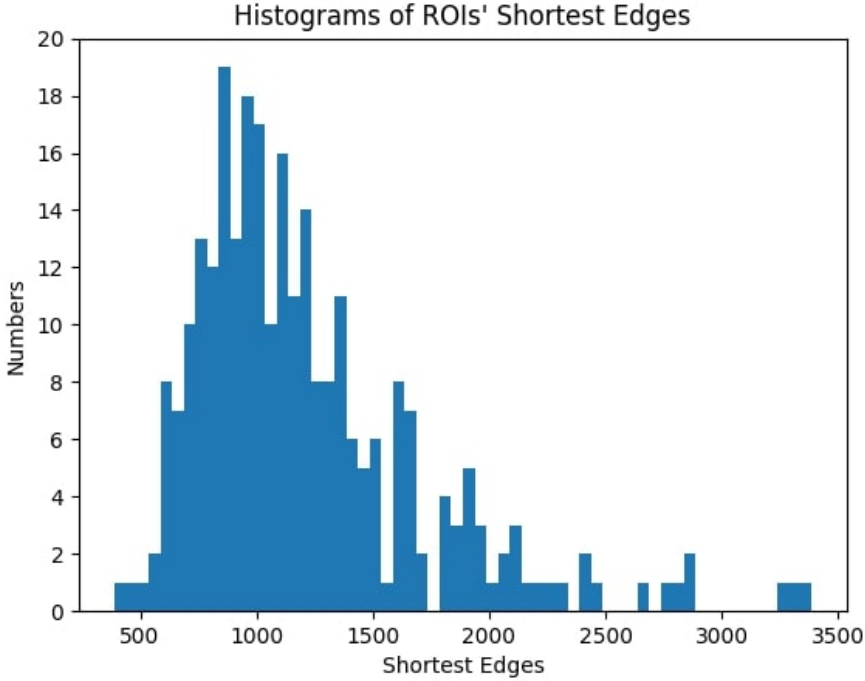
**Table 5.1:** Origin of datasets digitized from various source centers with different scanners. LUMC: Leiden University Medical Center. AUMC: Amsterdam University Medical Center. UMCU: The University Medical Center Utrecht. Leuven: KU Leuven.

### 5.4.1 Datasets

The biopsy specimens were prepared according to the Pathology Laboratory Protocol for kidney biopsies. A standard procedure of operations collected the tissues: core needle biopsy, formalin fixed, rapidly processed by a routine tissue processor, embedded in paraffin, serially sectioned at  $2-4\mu m$ , mounted onto adhesive slides, and stained with Jones silver staining. The dataset comprises 148 patient biopsy samples (both transplant and native kidneys) collected from the multi-center archives of the Departments of Pathology at LUMC, AUMC, and UMCU in the Netherlands, or from the archive of the Department of Nephrology at Leuven in Belgium. Before analysis, all biopsies were anonymized by a pathology staff member. Annotation was performed using the **ASAP**

## 5.4. Results

---



**Figure 5.8:** The histogram of the shortest edges of ROIs extracted from WSIs. In our dataset, the shortest ROI length is 387 pixels, and the largest is 4565 pixels.

version 1.9 for Windows<sup>4</sup> and our custom software **Slidescape**<sup>5</sup>. The 148 Jones-stained renal WSIs were acquired from scanners at a resolution of  $\text{PPM} = 2$  in BIG-TIFF format<sup>6</sup>. Openslide<sup>7</sup> was used to extract 303 ROIs at level 0 of the BIG-TIFF images. Table 5.1 outlines the dataset composition, and Figure 5.8 illustrates the distribution of ROI short sizes. Furthermore, only 115 LUMC PAS-stained renal WSIs were used to test stain transfer. In particular, ROIs from the same WSIs were exclusively assigned to either the training or the validation set. To train the lesion classifier, 21,889 patches of individual objects were extracted from 303 ROIs, including 916 glomeruli patches with binary lesion labels (382 with sclerosis and 534 normal) and 20,973 tubuli patches with binary labels (10,647 with atrophy and 10,326 normal), as detailed in Table 5.4.

---

<sup>4</sup><https://computationalpathologygroup.github.io/ASAP/>

<sup>5</sup><https://github.com/amspath/slidescape>

<sup>6</sup><https://www.awaresystems.be/imaging/tiff/bigtiff.html>

<sup>7</sup><https://openslide.org/>

### 5.4.2 Experiments

In our experimental analysis, we conducted a fair comparison of our proposed DiffRegFormer against well-established end-to-end instance segmentation models: Mask R-CNN [245], Cascade Mask R-CNN [50], and QueryInst [177], specifically designed to simultaneously handle dense, multi-class, multi-scale objects at the ROI level. All the models under comparison employed a ResNet-50 [248] backbone pre-trained on the ImageNet [138] and were trained on 249 ROIs over 40000 iterations. We utilized the **mmdection** data augmentation pipeline: each ROI was first subjected to *RandomFlip* with a 0.5 probability, followed by a random selection between *RandomResize* and *RandomCrop*, also with a probability of 0.5. *RandomResize* adjusted the ROI to a size with the shortest edge between 480 and 1333 pixels, while *RandomCrop* extracted patches with the shortest edge between 384 and 600 pixels. Finally, pixel values were normalized to  $(-1, 1)$ . As **mmdection** supports training with variable image sizes, all models were allowed to train on objects at multiple scales.

For feature extraction, we used 256 channels for queries. During both training and inference, DiffRegFormer employed **500** dynamic queries, while QueryInst used 300 and 500 static queries, respectively. Cascade Mask R-CNN, QueryInst, and our method each used 6 cascading stages. Table 5.2 presents performance comparisons for object detection w.r.t. bounding boxes and instance segmentation w.r.t. masks, respectively. For all experiments, DiffRegFormer did not utilize refinement during inference (i.e., iteration set to 1). Notably, Mask R-CNN [245] and cascade Mask R-CNN [50] are two-stage methods that employ RPN networks rather than dynamic queries. QueryInst is a one-stage method with static queries. In contrast, DiffRegFormer is categorized as a one-stage method with dynamic queries. We adopted the Average Precision (AP) used in MSCOCO [415] as the metric to evaluate the models' performance in detection and segmentation.

**Detection.** DiffRegFormer achieves an Average Precision (AP) of 52.1% for detection with a ResNet-50 backbone, thereby surpassing competitors such as QueryInst, Mask R-CNN, and Cascade Mask R-CNN by significant margins. We observe that iterative refinement improves detection performance for medium- to large-sized objects while degrading performance for small objects. It suggests additional operations to mitigate information loss for small objects during the refinement process. Furthermore, static queries are less effective for handling dense objects, particularly when the number of queries significantly exceeds the number of feature channels (300 or 500 compared to

## 5.4. Results

|           |               | Bounding Boxes |                  |                  |                 |                 |                 | Instances   |                  |                  |                 |                 |                 |
|-----------|---------------|----------------|------------------|------------------|-----------------|-----------------|-----------------|-------------|------------------|------------------|-----------------|-----------------|-----------------|
|           |               | AP             | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> | AP          | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
| one-stage | QueryInst_300 | 37.8           | 59.3             | 47.4             | 13.5            | 21.5            | 47.4            | 38.9        | 59.4             | 43.5             | 8.8             | 19.0            | 44.1            |
|           | QueryInst_500 | 30.1           | 31.5             | 22.8             | 0.9             | 8.7             | 24.1            | 32.5        | 31.9             | 22.7             | 0.4             | 7.8             | 27.7            |
| two-stage | Mask R-CNN    | 49.0           | 68.6             | 56.2             | <b>18.0</b>     | 24.4            | 59.2            | 45.5        | 69.0             | 53.5             | <b>16.4</b>     | 21.5            | 54.4            |
|           | CMask-RCNN    | 49.9           | 67.3             | <b>58.4</b>      | 7.6             | 24.1            | <b>61.4</b>     | 45.0        | 67.8             | <b>54.5</b>      | 6.5             | 21.6            | <b>54.5</b>     |
| our model | DiffRegFormer | <b>52.1</b>    | <b>71.1</b>      | 57.7             | 15.4            | <b>27.4</b>     | <b>61.4</b>     | <b>46.8</b> | <b>71.6</b>      | 52.8             | 14.1            | <b>23.6</b>     | 54.2            |

**Table 5.2:** Overall evaluation results. All models use ResNet-50 as a backbone network and have been trained for 40000 iterations on an NVIDIA GeForce RTX 3090 GPU with a batch size of 2. AP<sub>50</sub> and AP<sub>75</sub> denote AP with fixed thresholds for IoU ratio 50% and 75%, respectively. AP<sub>S</sub>, AP<sub>M</sub>, and AP<sub>L</sub> correspond to AP on small, medium, and large-sized objects, respectively. The best results are highlighted in bold. CMask-RCNN represents Cascade Mask R-CNN.

256), leading to a marked decline in detection performance.

**Instance segmentation.** DiffRegFormer achieves a 46.8% AP for instance segmentation and thus outperforms its counterparts. Moreover, applying iterative bounding box refinement significantly affects instance segmentation performance. This effect can be attributed to the binary segmentation process within the boxes, where the quality of the instance masks is inherently linked to the recall and precision rates of the bounding boxes.

Table 5.3 presents the performance per object class. Our model demonstrates an overall superiority in detecting all anatomical structures compared to other methods. Additionally, it performs better in glomeruli and arterial instance segmentation, while maintaining a slight advantage in tubular segmentation. These findings indicate that our model can process dense, multi-class objects with diverse scales at the ROI level. However, for each model, we observe a performance bottleneck in processing arteries, limiting clinical application due to significant size variation within the dataset. For instance, the largest arteries are two orders of magnitude larger than the smallest. The difficulty suggests that more sophisticated attention mechanisms are needed to capture global contextual information in the face of these extreme scale variations.

Figure 5.9 demonstrates DiffRegFormer’s capability to detect objects at various scales effectively. It accurately generates instance masks within bounding boxes, demonstrating precise localization of larger arteries (see the second row). In contrast, Cascade Mask R-CNN struggles to localize large objects within a single bounding box because it lacks an attention mechanism to capture long-range dependencies. Furthermore, Mask R-CNN tends to produce multiple detection results for a single object (see the third row) because it lacks cascade refinement. These observations suggest that iterative bounding box refinement during training significantly reduces multiple predictions per object. At the same time, the attention mechanism with

## Chapter 5. Instance Segmentation

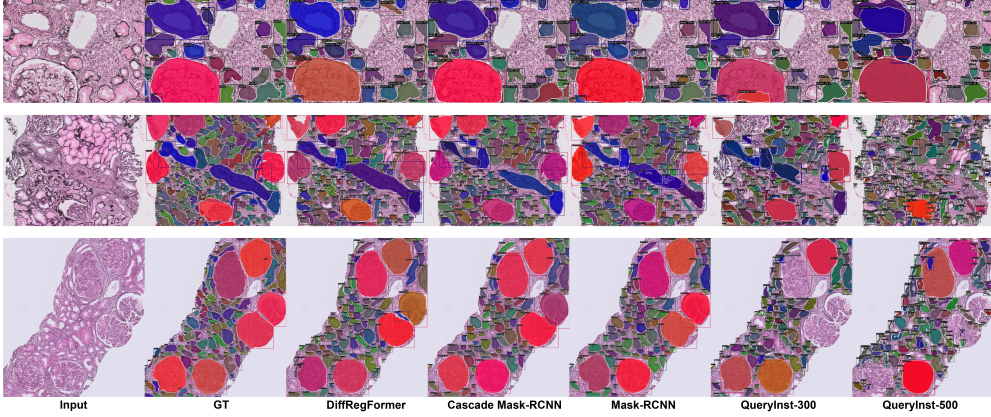
|           |               | Bounding Boxes - Glomeruli |                  |                  |                 |                 |                 | Instances - Glomeruli |                  |                  |                 |                 |                 |
|-----------|---------------|----------------------------|------------------|------------------|-----------------|-----------------|-----------------|-----------------------|------------------|------------------|-----------------|-----------------|-----------------|
|           |               | AP                         | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> | AP                    | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
| one-stage | QueryInst_300 | 62.6                       | 74.5             | 71.6             | -               | -               | 63.7            | 56.1                  | 74.5             | 68.1             | -               | -               | 57.1            |
|           | QueryInst_500 | 36.7                       | 47.5             | 44.0             | -               | -               | 37.6            | 35.9                  | 47.6             | 43.6             | -               | -               | 37.3            |
| two-stage | Mask R-CNN    | 69.4                       | 85.8             | 78.6             | -               | -               | 70.4            | 67.0                  | 85.8             | 78.8             | -               | -               | 68.1            |
|           | CMask-RCNN    | 70.7                       | 84.0             | 81.7             | -               | -               | 72.1            | 65.1                  | 84.0             | 79.8             | -               | -               | 66.4            |
| our model | DiffRegFormer | <b>80.9</b>                | <b>94.2</b>      | <b>89.8</b>      | -               | -               | <b>80.9</b>     | <b>74.3</b>           | <b>94.2</b>      | <b>80.9</b>      | -               | -               | <b>74.3</b>     |
|           |               | Bounding Boxes - Arteries  |                  |                  |                 |                 |                 | Instances - Arteries  |                  |                  |                 |                 |                 |
|           |               | AP                         | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> | AP                    | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
| one-stage | QueryInst_300 | 18.8                       | 37.0             | 19.1             | 5.2             | 20.2            | 20.6            | 17.5                  | 36.3             | 14.8             | 4.5             | 17.9            | 19.4            |
|           | QueryInst_500 | 4.5                        | 10.0             | 2.9              | 0.5             | 5.6             | 4.5             | 4.7                   | 10.5             | 3.8              | 0.1             | 5.3             | 5.3             |
| two-stage | Mask R-CNN    | 22.5                       | 40.3             | 23.5             | <b>20.2</b>     | 19.4            | 28.9            | 20.1                  | 41.4             | 18.0             | <b>20.2</b>     | 17.4            | 25.4            |
|           | CMask-RCNN    | 23.0                       | 38.0             | <b>26.9</b>      | -               | 17.8            | <b>33.3</b>     | 20.1                  | 39.4             | <b>21.9</b>      | -               | 16.9            | 27.5            |
| our model | DiffRegFormer | <b>25.7</b>                | <b>46.9</b>      | <b>25.8</b>      | 13.5            | <b>23.9</b>     | 31.6            | <b>23.4</b>           | <b>48.8</b>      | 20.2             | <b>20.2</b>     | <b>20.4</b>     | <b>30.1</b>     |
|           |               | Bounding Boxes - Tubuli    |                  |                  |                 |                 |                 | Instances - Tubuli    |                  |                  |                 |                 |                 |
|           |               | AP                         | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> | AP                    | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
| one-stage | QueryInst_300 | 45.8                       | 66.4             | 51.4             | <b>21.8</b>     | 44.3            | 57.9            | 41.2                  | 67.3             | 47.6             | 13.1            | 39.0            | 55.8            |
|           | QueryInst_500 | 21.2                       | 37.1             | 21.7             | 1.3             | 20.3            | 30.1            | 20.6                  | 37.5             | 20.7             | 0.7             | 18.1            | 40.5            |
| two-stage | Mask R-CNN    | 57.5                       | 79.6             | 66.4             | 15.8            | 53.8            | 78.2            | 51.4                  | 79.9             | <b>62.3</b>      | 12.7            | 47.3            | <b>69.8</b>     |
|           | CMask-RCNN    | 58.1                       | 79.9             | 66.5             | 15.1            | <b>54.4</b>     | <b>78.8</b>     | 51.8                  | 80.0             | 61.7             | 13.0            | <b>47.9</b>     | 69.6            |
| our model | DiffRegFormer | <b>58.5</b>                | <b>80.2</b>      | <b>67.9</b>      | 19.9            | 51.0            | 75.4            | <b>55.8</b>           | <b>80.6</b>      | <b>62.3</b>      | <b>16.6</b>     | 45.6            | 64.5            |

**Table 5.3:** Per-class evaluation results. All models use ResNet-50 as the backbone network and have been trained for 40000 iterations on a single NVIDIA GeForce RTX 3090 GPU with a batch size of 2. AP<sub>50</sub> and AP<sub>75</sub> denote AP with fixed thresholds for IoU ratio 50% and 75%, respectively. AP<sub>S</sub>, AP<sub>M</sub>, and AP<sub>L</sub> correspond to mAP on small, medium, and large-sized objects, respectively. The best results are highlighted in bold. CMask-RCNN represents Cascade Mask R-CNN.

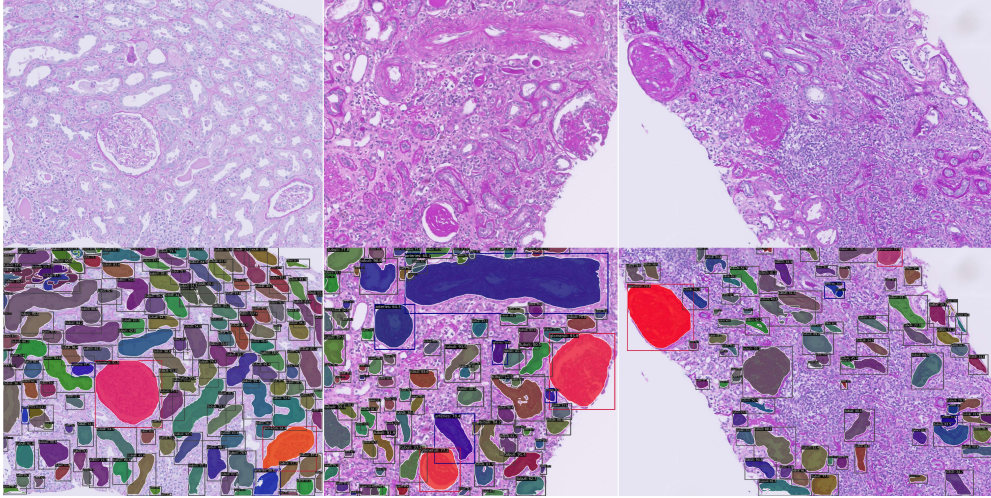
dynamic queries processes large-scale objects. Nevertheless, QueryInst, with its reliance on static queries, may struggle to effectively detect all instances in datasets with significant size and shape variations. QueryInst-300 tends to miss many structures, while QueryInst-500 can even degenerate into generating multiple false predictions for a single object. Furthermore, RCNN-based models share a limitation: a single pixel may be allocated to multiple instance masks due to pixel-wise predictions within each independent ROI. It inherently prevents enforcing the spatial exclusivity of pixels to instances. Additionally, Figure 5.10 shows DiffRegFormer’s potential for domain transfer, generating reasonable results on PAS-stained images despite being trained exclusively on Jones’ silver-stained images. Moreover, Figure 5.11 highlights that DiffRegFormer, combined with sliding window and stitching techniques, can be effectively applied to WSIs, further demonstrating its effectiveness.

**Lesion classification.** The data augmentation pipeline for training the lesion classifier is relatively straightforward. We set the batch size to 96 for the input croppings, with 32 patches per class. Each crop is first subjected to *RandomFlip* with probability 0.5, followed by resizing to  $256 \times 256$  pixels. Finally, pixel values are normalized to the  $(-1, 1)$  range. As shown in Table 5.4, the classifier achieves an overall precision of 89.2% and a recall of 64.6%. More specifically, per-class precision for atrophic tubuli

## 5.4. Results



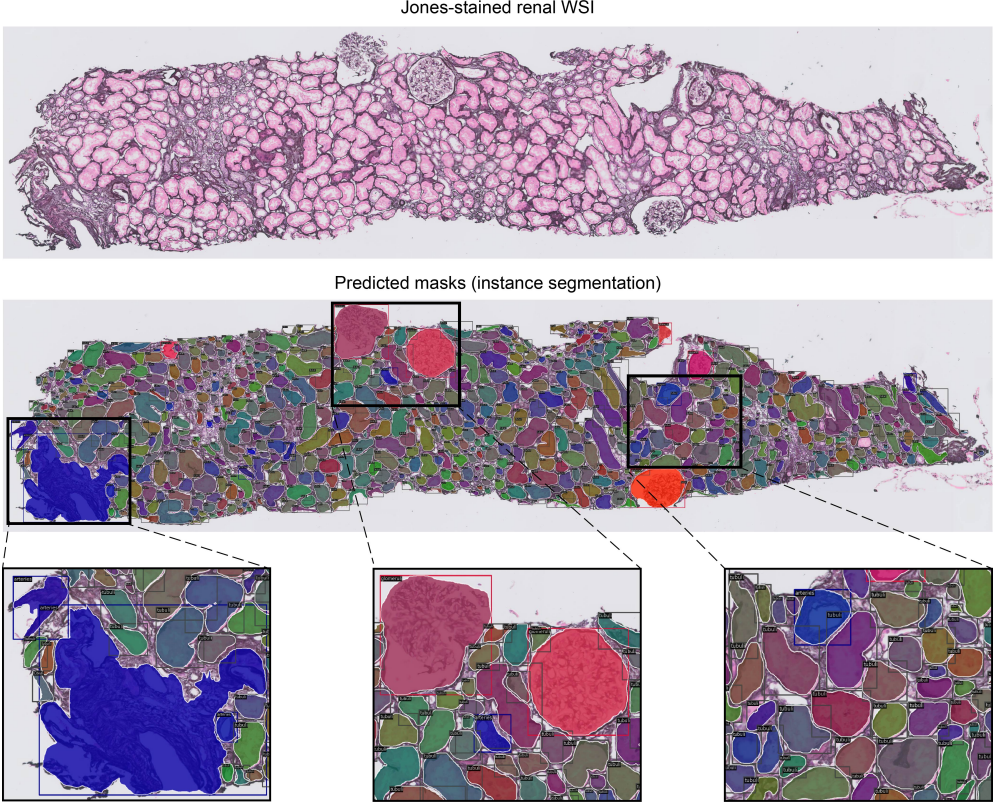
**Figure 5.9:** A visual comparison of different instance segment methods on Jones-stained kidney biopsies. It shows that our model, i.e., DiffRegFormer, outperforms in detecting objects of diverse sizes and shapes. Despite some instances overlapping, it generates precise instance masks over each region. Mask R-CNN based models struggle to localize a single large object within a single bounding box because they lack mechanisms to capture long-range dependencies. QueryInst with static queries fails to process dense objects since it fails to detect objects of each class within one ROI.



**Figure 5.10:** A visualization of the results of our model (DiffRegFormer) on the PAS-stained kidney biopsies with increasing degrees of fibrosis and inflammation (from left to right), along with variations in stain intensity.

is 85.1%, and 93.2% for sclerotic glomeruli. In addition, per-class recall is 40.5% for atrophic tubuli and 88.7% for sclerotic glomeruli. Since the lesion classifier operates on anatomical structure croppings generated by DiffRegFormer during inference, its recall





**Figure 5.11:** A representative visualization of results from DiffRegFormer stitches showing predicted dense instance objects from a WSI of a Jones-stained renal biopsy.

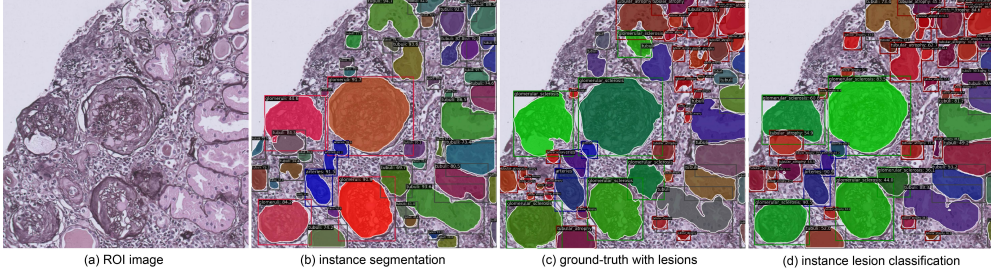
rate heavily depends on the performance of our dense instance segmentation model. However, the high precision indicates that lesion classification on correctly cropped structures is very accurate. In summary, improving the performance of DiffRegFormer is expected to further improve the lesion classifier’s recall. Figure 5.12 visualizes the mapping of lesion predictions onto the instance segmentation results. It reveals that our model can serve as a solid foundation for further clinical diagnosis.

### Metrics for Instance Segmentation

We have computed a set of commonly used metrics to assess the performance of our method in a broader scope. These are DICE, MASD (mean average surface distance), FLOPS, inference speed, and the number of parameters. In Table 5.5, we observe that our method outperforms other approaches in segmentation performance, as indicated by the DICE and MASD values. Resource efficiency, however, is moderate. The



## 5.4. Results



**Figure 5.12:** A visualization of the results of the lesion classifier predicting the probability of class-wise lesions on an ROI. Combined with dense instance segmentation, our model can generate per-instance lesion identification.

| Experimental Settings | Tubuli Lesion Classification |         | Glomeruli Lesion Classification |           |
|-----------------------|------------------------------|---------|---------------------------------|-----------|
|                       | Normal                       | Atrophy | Normal                          | Sclerosis |
| Train Dataset         | 10326                        | 10647   | 534                             | 382       |
| Eval Dataset          | 906                          | 568     | 51                              | 22        |
| Per-class Precision   | 85.1                         |         | 93.2                            |           |
| Per-class Recall      | 40.5                         |         | 88.7                            |           |
| Overall Precision     | 89.2                         |         |                                 |           |
| Overall Recall        | 64.6                         |         |                                 |           |

**Table 5.4:** Lesion dataset configuration and classification Performance.

main cause of resource cost is the attention mechanism, as seen in QueryInst and DiffRegFormer. To that end, it is beneficial to adopt more efficient attention algorithms to accelerate inference speed and reduce resource burden, tailored to the practical clinical agenda. In addition, the varying arteries across multiple scales make it difficult to obtain high-quality instance masks. All methods have illustrated a large average surface distance between the predicted and ground-truth boundaries.

### Comparison with Segmentation Based Method

To supplement our analysis in terms of completeness, we have conducted a retrospective analysis that reproduces a semantic segmentation based instance segmentation framework [253]. We denote this method as a semantic method and compare its results

|           |               | MASD        | DICE        | FLOPS<br>(Trillion) | Inference Speed<br>(Second) | # Parameters<br>(Million) |
|-----------|---------------|-------------|-------------|---------------------|-----------------------------|---------------------------|
| One-Stage | QueryInst_300 | 7.49        | 42.6        | 0.41                | 1.425                       | 173                       |
|           | QueryInst_500 | 10.95       | 35.9        | 0.55                | 1.647                       | 246                       |
| Two-Stage | Mask R-CNN    | 4.78        | 50.7        | 0.20                | <b>0.864</b>                | <b>43.98</b>              |
|           | CMask-RCNN    | 3.87        | 52.1        | <b>1.72</b>         | 1.156                       | 77.03                     |
| Our Model | DiffRegFormer | <b>3.64</b> | <b>56.6</b> | 0.55                | 1.324                       | 140                       |

**Table 5.5:** Additional metrics for instance segmentation assessment.

|                 | Bounding Boxes |                  |                  | Instance    |                  |                  | Extra Metrics |             |                     |                             |                          |
|-----------------|----------------|------------------|------------------|-------------|------------------|------------------|---------------|-------------|---------------------|-----------------------------|--------------------------|
|                 | AP             | AP <sub>50</sub> | AP <sub>75</sub> | AP          | AP <sub>50</sub> | AP <sub>75</sub> | DICE          | MASD        | FLOPS<br>(Trillion) | Inference Speed<br>(Second) | # Parameter<br>(Million) |
| DiffRegFormer   | <b>52.1</b>    | <b>71.1</b>      | <b>57.7</b>      | <b>46.8</b> | <b>71.6</b>      | <b>52.8</b>      | <b>56.6</b>   | <b>3.87</b> | 0.55                | 1.324                       | 140                      |
| Semantic Method | 44.6           | 64.8             | 46.7             | 40.3        | 65.3             | 44.6             | 45.4          | 7.86        | <b>1.24</b>         | <b>0.760</b>                | <b>32.8</b>              |

**Table 5.6:** Overall evaluation result. The semantic method is an instance segmentation framework that combines semantic segmentation with a post-processing operation. We have reproduced it following the literature [253]. *AP* denotes average precision. *AP<sub>50</sub>* and *AP<sub>75</sub>* denote AP with fixed thresholds for IoU ratio 50% and 75%, respectively.

with ours. In Table 5.6, an overall comparison is depicted. Our method outperforms the semantic method in prediction quality at the cost of increased inference time and computational resources. It should be noted that these results are specific to this dataset. In Table 5.7, the superiority of our method per class is depicted. Furthermore, Figure 5.13 illustrates that the semantic method is more vulnerable to the segmentation quality of the auxiliary boundary and interstitium class, which is insufficient in separating objects that are tightly touching. Such touching objects are typical in kidney tissue. Our method, on the contrary, can detect and segment each object.

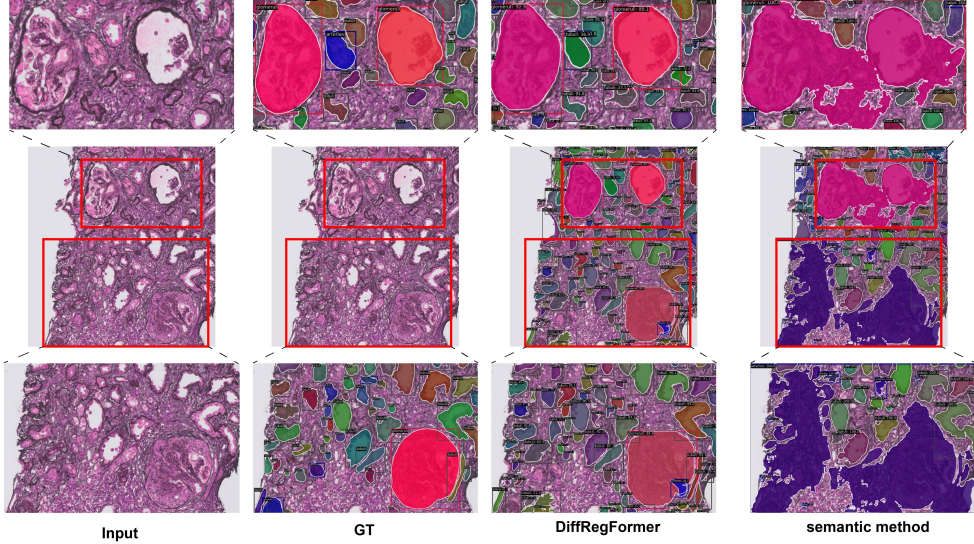
|                 | Bounding Boxes - glo |                  |                  | Instance - glo |                  |                  | Extra Metrics |       |
|-----------------|----------------------|------------------|------------------|----------------|------------------|------------------|---------------|-------|
|                 | AP                   | AP <sub>50</sub> | AP <sub>75</sub> | AP             | AP <sub>50</sub> | AP <sub>75</sub> | DICE          | MASD  |
| DiffRegFormer   | 80.9                 | 94.2             | 89.8             | 74.3           | 94.2             | 80.9             | 78.8          | 1.21  |
| semantic method | 75.6                 | 82.4             | 81.2             | 68.3           | 84.3             | 70.1             | 70.6          | 8.68  |
|                 | Bounding Boxes - art |                  |                  | Instance - art |                  |                  | Extra Metrics |       |
|                 | AP                   | AP <sub>50</sub> | AP <sub>75</sub> | AP             | AP <sub>50</sub> | AP <sub>75</sub> | DICE          | MASD  |
| DiffRegFormer   | 25.7                 | 46.9             | 25.8             | 23.4           | 48.8             | 20.2             | 30.2          | 25.94 |
| semantic method | 10.3                 | 25.8             | 13.6             | 11.5           | 24.9             | 10.3             | 16.3          | 49.14 |
|                 | Bounding Boxes - tub |                  |                  | Instance - tub |                  |                  | Extra Metrics |       |
|                 | AP                   | AP <sub>50</sub> | AP <sub>75</sub> | AP             | AP <sub>50</sub> | AP <sub>75</sub> | DICE          | MASD  |
| DiffRegFormer   | 58.5                 | 80.2             | 67.9             | 55.8           | 80.6             | 62.3             | 64.8          | 0.89  |
| semantic method | 43.6                 | 73.5             | 54.6             | 46.9           | 75.2             | 52.4             | 50.5          | 2.16  |

**Table 5.7:** Evaluation results per object class. *AP* denotes average precision. *AP<sub>50</sub>* and *AP<sub>75</sub>* denote AP with fixed thresholds for IoU ratio 50% and 75%, respectively.

### 5.4.3 Complexity Assessment of Model Training

We start by analyzing the intrinsic class imbalance within our dataset. Of the total structures (21,889 objects), 88% are tubuli, followed by 7% arteries (1679 objects), and 5% glomeruli (1226 objects). This distribution is depicted in Figure 5.14 (a), highlighting the necessity of sampling to include more rare instances, such as arteries and glomeruli, among the dominant tubuli. A violin plot, shown in Figure 5.14 (b), details the distribution of structure counts at the ROI level. On average, ROIs contain

## 5.4. Results

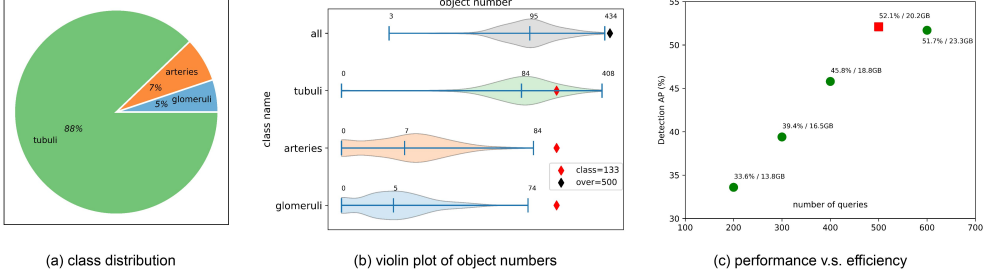


**Figure 5.13:** A visual comparison between DiffRegFormer and the semantic method. The upper row represents a zoom of the upper-box in the middle row. The lower row represents a zoom of the lower-box in the middle row. The quality of the result w.r.t. the ground truth labels can be convincingly assessed in favor of DiffRegFormer.

95 objects, with a maximum of 434. To accommodate all objects, DiffRegFormer is configured to handle more than 434 per image. Figure 5.14 (c) depicts the correlation between the increase in computational cost and performance improvement. As expected, we observe a turning point at 500 dynamic queries. This configuration yields a favorable trade-off between complexity and efficiency, achieving 52.1% AP with 20.2 GB of GPU memory. Specifically, ROIs contain an average of 5 glomeruli, 7 arteries, and 84 tubuli with corresponding maximums of 74, 84, and 408. Setting the maximum number per class to 166 is another feasible option for positive sample generation, as our sampling approach ensures comprehensive inclusion of rare instances while maintaining the rational proportion of the majority class. In conclusion, selecting a maximum of 500 objects per image and 166 per class can effectively balance performance and computational efficiency.

### 5.4.4 Ablation Study

The ablation study focuses on the effects of various training strategies and query types on model performance. By analyzing our results, we can gain insights into the specific characteristics of each strategy. This analysis highlights the significance of our



**Figure 5.14:** Complexity Analysis on Jones-stained renal biopsy dataset. Figure (a) is the sample occupation across our dataset. Figure (b) shows a violin plot for the sample number distribution per class. Figure (c) shows trade-offs between performance and efficiency for different experiment settings. The red square is the optimal setting, while the green circles are sub-optimal settings.

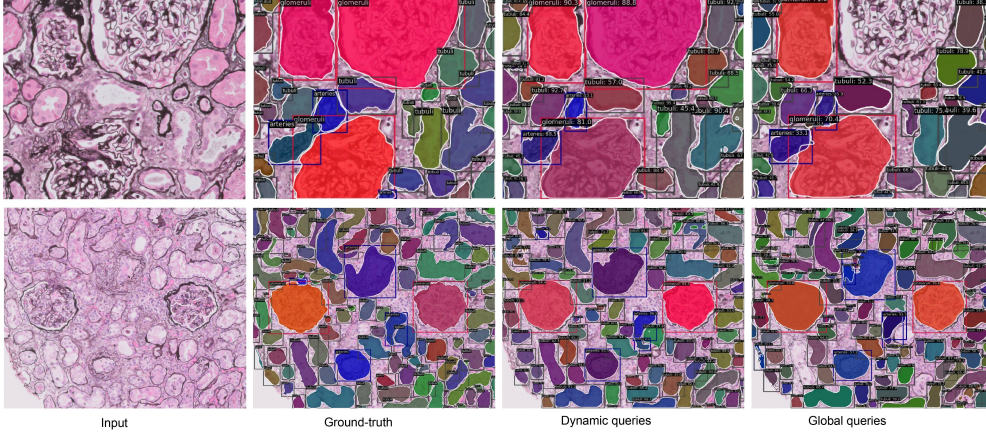
improvements as key contributions to adapting an end-to-end instance segmentation model to datasets with dense objects.

## Queries

Static queries in transformer-based models struggle to represent dense objects in large-scale datasets due to their static nature. Adequate accommodation to extensive variations in object size and shape necessitates a proportional increase in query count. Alternatively, dynamic queries, generated on-the-fly for each object in a mini-batch, demonstrate superior adaptability to dataset complexity. Theoretically, assigning more static queries could achieve performance comparable to that of dynamic queries. However, our experiments indicate that increasing the number of static queries without a corresponding increase in feature channels does not yield improvements. Consequently, employing static queries for dense object processing in transformer-based models is significantly less cost-effective than dynamic queries.

Table 5.8 delineates our comparative analysis of varying query counts for both dynamic and static queries, with the number of feature channels fixed at 256. We observe that increasing the number of dynamic queries improves performance, whereas increasing the number of static queries degrades performance. Figure 5.15 further illustrates the inherent challenge of achieving a favorable trade-off between complexity and performance with static queries. Even with 500 queries, the model with static queries fails to consistently detect structures across all classes. This limitation is not attributable to class imbalance compared to the dynamic query strategy. In conclusion, the dynamic query is a more efficient approach for datasets with dense

## 5.4. Results



**Figure 5.15:** A visual comparison from an ablation study on different query types. Dynamic queries outperform static queries on detection. It should be noted that the static queries fail to detect objects of all classes within one ROI.

objects, highlighting their superiority over static queries.

| Number<br>mAP | # 200 Query | # 300 Query | # 400 Query | # 500 Query | # 600 Query |
|---------------|-------------|-------------|-------------|-------------|-------------|
| Dynamic Query | <b>33.6</b> | <b>39.4</b> | <b>45.8</b> | <b>52.1</b> | <b>51.7</b> |
| Static Query  | 23.8        | 38.9        | 37.4        | 34.6        | 31.6        |

**Table 5.8:** Evaluation of detection performances on different types of queries. The best result of each type is highlighted in bold.

### Training strategies

| feature | boxes | sample | glo-det     | art-det     | tub-det     | all-det     | glo-ins     | art-ins     | tub-ins     | all-ins     |
|---------|-------|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| ✗       | ✗     | ✗      | -           | -           | -           | -           | -           | -           | -           | -           |
| ✗       | ✗     | ✓      | -           | -           | -           | -           | -           | -           | -           | -           |
| ✗       | ✓     | ✗      | 75.8        | 24.9        | 56.4        | 52.4        | -           | -           | 33.6        | -           |
| ✗       | ✓     | ✓      | 76.1        | <b>26.2</b> | 57.6        | 53.3        | 7.3         | 4.8         | 35.2        | 15.8        |
| ✓       | ✗     | ✗      | -           | -           | -           | -           | -           | -           | -           | -           |
| ✓       | ✗     | ✓      | 8.5         | 4.3         | 13.5        | 8.8         | 6.8         | 4.1         | 12.5        | 7.8         |
| ✓       | ✓     | ✗      | 12.6        | 6.8         | 56.7        | 25.4        | 10.4        | 5.3         | 45.8        | 20.5        |
| ✓       | ✓     | ✓      | <b>80.9</b> | 25.7        | <b>58.5</b> | <b>55.1</b> | <b>74.3</b> | <b>23.4</b> | <b>55.8</b> | <b>51.2</b> |

**Table 5.9:** Evaluation on combinations of different strategies. The symbol ✓ denotes taking one strategy, while the symbol ✗ means taking the converse strategy. The best results are highlighted in bold. *glo*: glomeruli. *art*: arteries. *tub*: tubuli. *all*: overall. *det*: detection. *ins*: instance. "-": not working. *feature*: if use separate feature maps between bbox-decoder and mask-decoder. *boxes*: if use ground-truth boxes in the sampling process. *sample*: if using class balance in the sampling process.

Training an instance segmentation model on datasets with dense objects, such as

kidney biopsies, presents distinct challenges compared to common datasets with sparser object distributions, such as MSCOCO. Instance segmentation models typically employ three strategies: (1) shared features between the bbox-decoder and mask-decoder; (2) selection of positive samples from proposal bounding boxes; and (3) class-imbalanced sampling. However, for dense object segmentation, we propose three alternative strategies: (a) separate features for the bbox and mask decoders; (b) positive sample selection directly from ground-truth boxes; and (c) class-balanced sampling. While the conventional combination of shared features, proposal boxes, and biased sampling is adequate for standard instance segmentation (see Figure 5.16a), only the combination of separate features, ground-truth boxes, and unbiased sampling proves to be successful for dense instance segmentation. Table 5.9 evaluates the combinations above, with conventional strategies marked with ✗ and our proposed strategy with ✓. Notably, only five out of eight possible combinations can train DiffRegFormer.

Figure 5.16b demonstrates that combining shared features, ground-truth boxes, and biased sampling enables object detection across all classes. However, it fails to produce accurate instance masks within these bounding boxes. The observation suggests that the bbox-decoder dominates the shared feature representation, thus hindering the mask decoder’s ability to extract pixel-wise information. Consequently, all pixels within detected boxes are classified as tubuli. Moreover, biased sampling impairs the mask decoder’s ability to learn features for rare instances (glomeruli and arteries), leading to no pixel-wise classification within those boxes.

Additionally, Figure 5.16c shows that shared features combined with ground-truth boxes and unbiased sampling can detect and segment objects across all classes, albeit with degraded mask quality. This degradation is likely due to the bbox-decoder’s dominance within the shared feature representation.

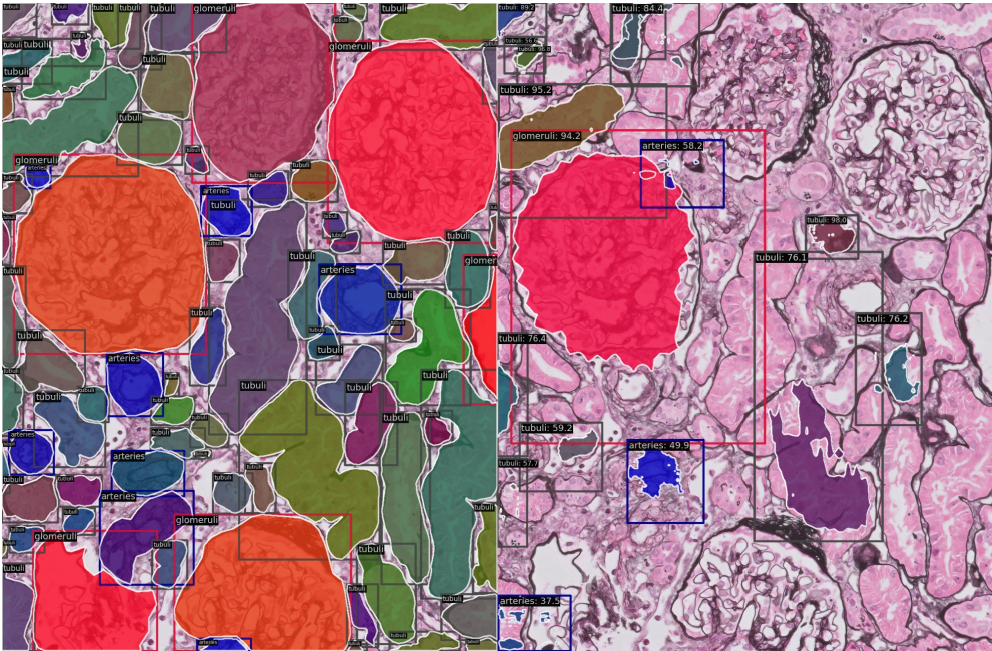
In contrast, Figure 5.16d reveals that separate features, proposal boxes, and unbiased sampling lead to inaccurate object detection, probably due to the accumulation of errors introduced by proposals during the early stages of diffusion model training.

Lastly, Figure 5.16e demonstrates that separate features, ground-truth boxes, and biased sampling result in correct detection and segmentation for only a small proportion of tubuli. This finding further highlights the challenges caused by biased sampling in learning features for rare instances.

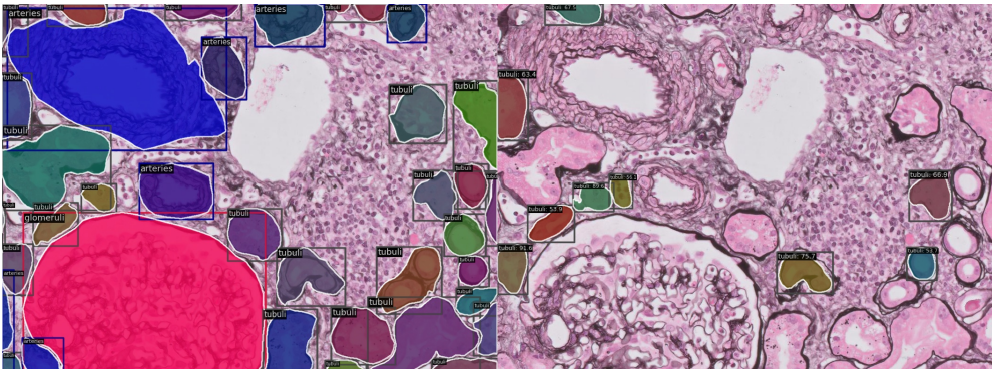
In summary, shared feature maps lead the bbox-decoder to dominate feature representation learning, impairing the mask-decoder’s ability to extract pixel-wise information necessary for accurate binary instance masks. Biased sampling further hinders the mask-decoder from learning information to segment rare instances (glomeruli







(d) separate feature + proposals + unbiased sampling on a ROI



(e) separate feature + ground-truth + biased sampling on a ROI

**Figure 5.16:** Ablation study on combinations of different strategies. The images on the left are ground-truth. The images on the right are our predictions.

and arteries), causing it to be overwhelmed by the majority class (tubuli). Finally, using proposals for sampling rather than ground-truth boxes introduces cumulative errors that disrupt training for both decoders, preventing the learning of contextual information necessary for long-range correlations via attention mechanisms.



## 5.5 Conclusion and Future Work

Our research introduces a novel framework for processing large-scale renal WSI datasets that accounts for potential changes in lesion combinations. Our model performs better in predicting lesions within dense objects, effectively handling multi-class and multi-scale challenges at the ROI level. We present the first dense instance segmentation module, DiffRegFormer, that seamlessly integrates a diffusion model with a transformer in an RCNN-style architecture. This approach leverages the diffusion model to generate refined object boxes, upon which a regional transformer creates dynamic queries that are ultimately transformed into accurate instance masks within the boxes. Furthermore, our lesion classifier accurately predicts class-specific lesions for each cropped structure. The model demonstrates several advantages in handling dense object datasets: (1) its iterative noise-to-box approach eliminates the dependence on prior knowledge about object size and shape, facilitating rapid object localization; (2) dynamic queries, generated on-the-fly for each object in a mini-batch style, ensuring scalability to large datasets; (3) the modular design of the model, with independent plug-in prediction heads, allows for replacement and promotes adaptability. This flexibility enables the efficient reuse of trained components when upscaling to large-scale datasets. Experimental results on Jones' stained renal biopsies demonstrate that our model surpasses existing benchmarks. Comparing results between methods based on the data is tricky. Measures like DICE behave differently with different datasets. The comparison should be based on the same dataset (cf. 5.4.2). In that case, results can be well understood and objectively compared.

The current model needs further exploration to meet the rigorous requirements for clinical applications. Still, it establishes a robust framework for dense, multi-class, multi-scale object recognition at the ROI level, setting a solid foundation for future computational improvements. Potential areas for future research include: (1) further enhancing the compatibility of diffusion models with transformers like DETR to avoid NMS post-processing. (2) developing innovative regional feature extractors to generate more informative dynamic queries and mitigate the loss of information for small objects during iterative refinements. (3) integrating spatial constraints into regional mask inference to prevent erroneous assignment of single pixels to multiple instances. (4) designing lightweight attention mechanisms to lessen the computational burden. In conclusion, the approach presented paves the way for efficient and reliable automated assessment as a tool for the workbench of renal pathologists.