Boltzmann Attention Sampling for Image Analysis with Small Objects

Supplementary Material

A. Implementation details

A.1. Image encoder

We used Hiera or FocalNet as the image backbone. Both models outputs four scales of features with strides 4, 8, 16 and 32. The pixel decoder takes the multiscale backbone features to output multiscale visual features of resolution 32×32, 64×64, 128×128 and 256×256. The multiscale visual features convolute to semantic map in resolution 256×256. *BoltzFormer* attends to multiscale visual features of resolution 32×32, 64×64 and 128×128 in loops. We repeated the loop of the three scales in this order for three times, resulting in nine layers in total. The query vectors after the ninth layer was used for prediction.

A.2. Language encoder

We used UniCL as the language encoder for text prompts. The context length is 77. We input the full token embedding sequence to *BoltzFormer*.

A.3. Boltzmann sampling

In our main experiments, we used a sampling ratio of 10%, which means at each layer the number of independent trials is 10% of the total number of visual features of that scale. We used default base temperature $\tau_0 = 1$.

A.4. Training specifics

We used 32 NVIDIA A100 80GB GPUs for *BoltzFormer* training. We used batch size 12×32 for Hiera-S and Hiera-BP backbones, and batch size 8×32 for Focal-L backbone. During training we split out 20% from the training set for validation, and monitored the validation loss. We trained with early stopping based on validation loss for a maximum of 40 epochs. We used AdamW [16] as the optimizer with equal weighted Dice loss and pixel-wise binary cross-entropy loss. Selected training hyper-parameters based on validation loss for learning rates $2 \times 10^{-5}, 4 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}$, and weight decays $10^{-4}, 10^{-3}, 10^{-2}$. We used learning rate 2×10^{-5} and weight decay 10^{-2} for training *BoltzFormer*.

A.5. Data augmentation

We implemented random transformation for the images and ground truth masks. Each training example had 50% probability to be transformed. We randomly rotated the image and corresponding ground truth mask by $0°, 90°, 180°, 270°$. We randomly cropped the image and the corresponding ground truth mask by shifting the center by $\pm 10\%$ horizontally and vertically, and scaling by $\pm 10\%$. All random augmentation were uniformly sampled.

B. Dataset details

We listed the object size statistics for all the benchmark datasets we used for evaluation in Table 9. The benchmark suite used in our study covered a wide range of object sizes, from 0.002% to 20% of the image area, crossing four orders of magnitudes. The majority of the object types in each dataset have mean object size less than 1% of the image area.

C. Visualizations

C.1. Boltzmann sampling

We visualize the examples of Boltzmann attention sampling through the layers in *BoltzFormer*. In each layer, each query vector only attend to the sampled visual features visualized in the figures. We looped through the three scales of visual features for three times, resulting in nine layers in total. The shaded regions are completely invisible to the query in that layer. We show the Boltzmann sampling for the first query in the ensemble in these examples.

From Fig. 5, 6, 7 and 8 we can see that the sampling is wide spread during the early layers. The sampling began to concentrate towards the target region during the middle layers, while still exploring the other regions. At the final layers the sampling was highly concentrated on the target regions.

In Fig. 8 we show the Boltzmann sampling example for an object with size greater than 20% of the total image area. Because the sampling ratio is just 10%, it is impossible to cover all the visual features in the target region. *BoltzFormer* still works for objects with large sizes as the sampled visual features are enough to summarize the semantics of the object.

C.2. Segmentation result comparison

In Fig. 9 we visualize the segmentation results from the baseline models and *BoltzFormer*. All transformer decoders took in text prompts through the UniCL language encoder. The backbone was fixed as Hiera-S for all the models. We visualize the segmantation masks in green, and outline the boundary of the ground truth target in red for reference.

We can see that *BoltzFormer* consistently delivered accurate segmentation, while the baseline models could miss the target completely when the object is very small (lung nodule and tumor, pancreas tumor). When the object is very large, *BoltzFormer* still output accurate segmentation result.

Table 9. Statistics about the benchmark datasets. We listed all the object types for segmentation in each of the datasets, along with the mean, median, min and max of the object size as a ratio to the total image area.

Dataset	Object type	Area mean (%)	Area median (%)	Area min (%)	Area max (%)	Support
LIDC-IDRI	nodule	0.059	0.029	0.002	0.802	1733
MSD-Lung	tumor	0.283	0.200	0.011	1.062	242
MSD-Pancreas	tumor	0.629	0.295	0.018	5.106	575
MSD-Pancreas	pancreas	0.567	0.465	0.029	2.431	1635
MSD-HepaticVessel	tumor	0.535	0.281	0.005	3.652	918
MSD-HepaticVessel	vessel	0.211	0.178	0.017	0.878	2204
MSD-Colon	tumor	0.472	0.341	0.036	2.287	245
AMOS22-CT	adrenal gland	0.185	0.169	0.037	0.503	420
AMOS22-CT	esophagus	0.142	0.102	0.018	1.189	1964
AMOS22-CT	postcava	0.237	0.218	0.034	0.751	4105
AMOS22-CT	spleen	1.952	1.752	0.142	6.587	1587
AMOS22-CT	right adrenal gland	0.087	0.073	0.010	0.304	571
AMOS22-CT	left kidney	1.194	1.206	0.065	2.649	1776
AMOS22-CT	kidney	2.509	2.473	0.334	4.798	1465
AMOS22-CT	duodenum	0.497	0.425	0.036	2.816	1677
AMOS22-CT	bladder	2.216	1.780	0.161	7.851	864
AMOS22-CT	gallbladder	0.664	0.490	0.035	2.946	712
AMOS22-CT	liver	7.896	7.285	0.694	21.498	4648
AMOS22-CT	right kidney	1.171	1.191	0.052	2.295	1649
AMOS22-CT	left adrenal gland	0.091	0.082	0.004	0.279	635
AMOS22-CT	pancreas	0.818	0.653	0.071	3.016	1345
AMOS22-CT	stomach	3.222	2.724	0.165	11.275	1837
AMOS22-CT	aorta	0.433	0.281	0.051	3.957	4409
AMOS22-MRI	adrenal gland	0.115	0.111	0.048	0.216	49
AMOS22-MRI	esophagus	0.064	0.058	0.022	0.167	134
AMOS22-MRI	postcava	0.135	0.116	0.033	0.351	463
AMOS22-MRI	spleen	1.353	1.296	0.130	2.707	197
AMOS22-MRI	right adrenal gland	0.056	0.053	0.011	0.157	56
AMOS22-MRI	left kidney	0.917	0.955	0.151	1.774	237
AMOS22-MRI	kidney	1.909	1.913	0.637	3.027	198
AMOS22-MRI	duodenum	0.280	0.264	0.075	0.860	201
AMOS22-MRI	gallbladder	0.407	0.366	0.046	0.914	75
AMOS22-MRI	liver	6.060	5.950	0.891	11.907	304
AMOS22-MRI	right kidney	0.882	0.935	0.146	1.467	224
AMOS22-MRI	left adrenal gland	0.051	0.045	0.020	0.109	84
AMOS22-MRI	pancreas	0.583	0.522	0.106	1.626	195
AMOS22-MRI	stomach	1.054	0.975	0.224	2.687	216
AMOS22-MRI	aorta	0.179	0.173	0.075	0.355	490

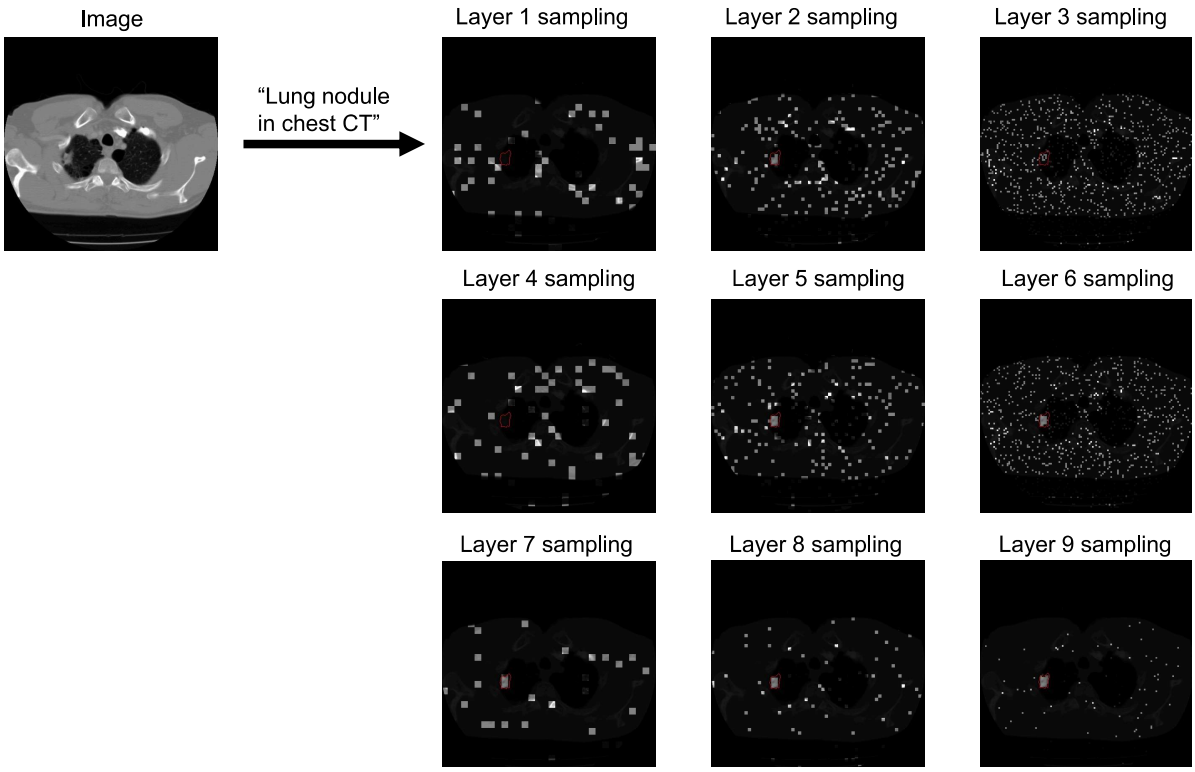

Figure 5. Boltzmann sampling example for lung nodule in chest CT. The sample patches are bright with target region circled in red.

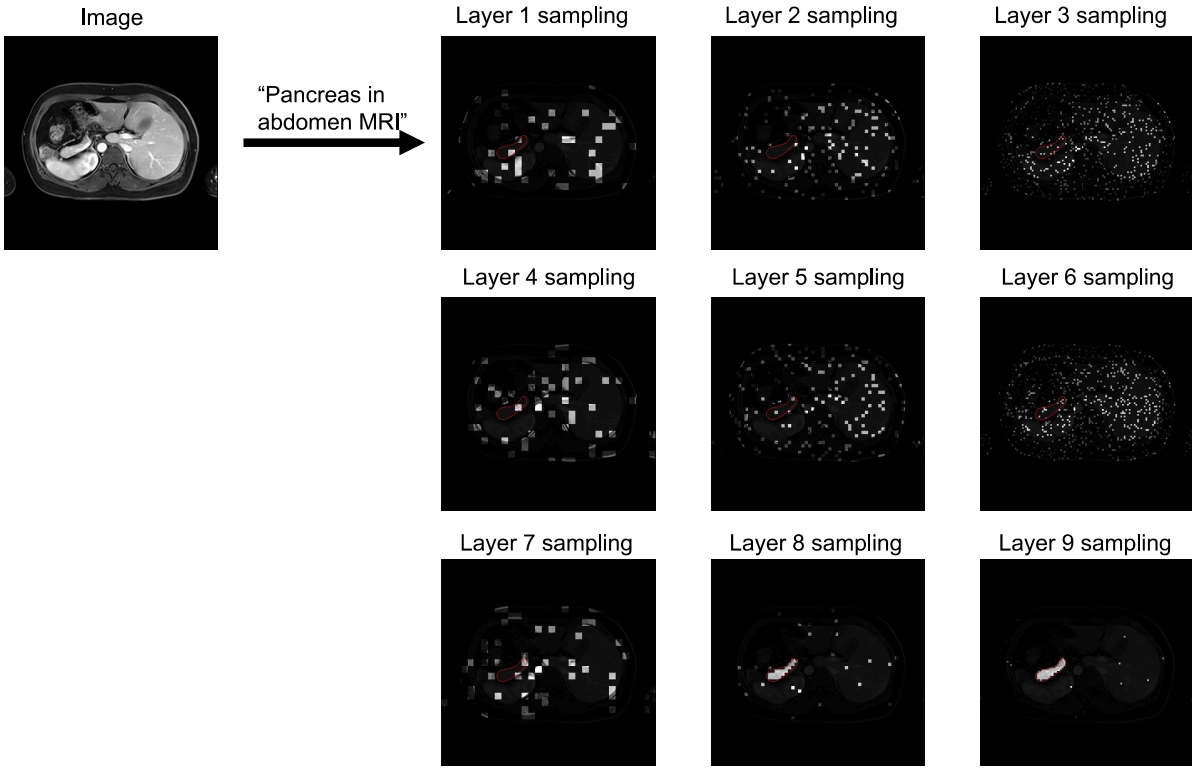

Figure 6. Boltzmann sampling example for pancreas in abdomen MRI. The sample patches are bright with target region circled in red.

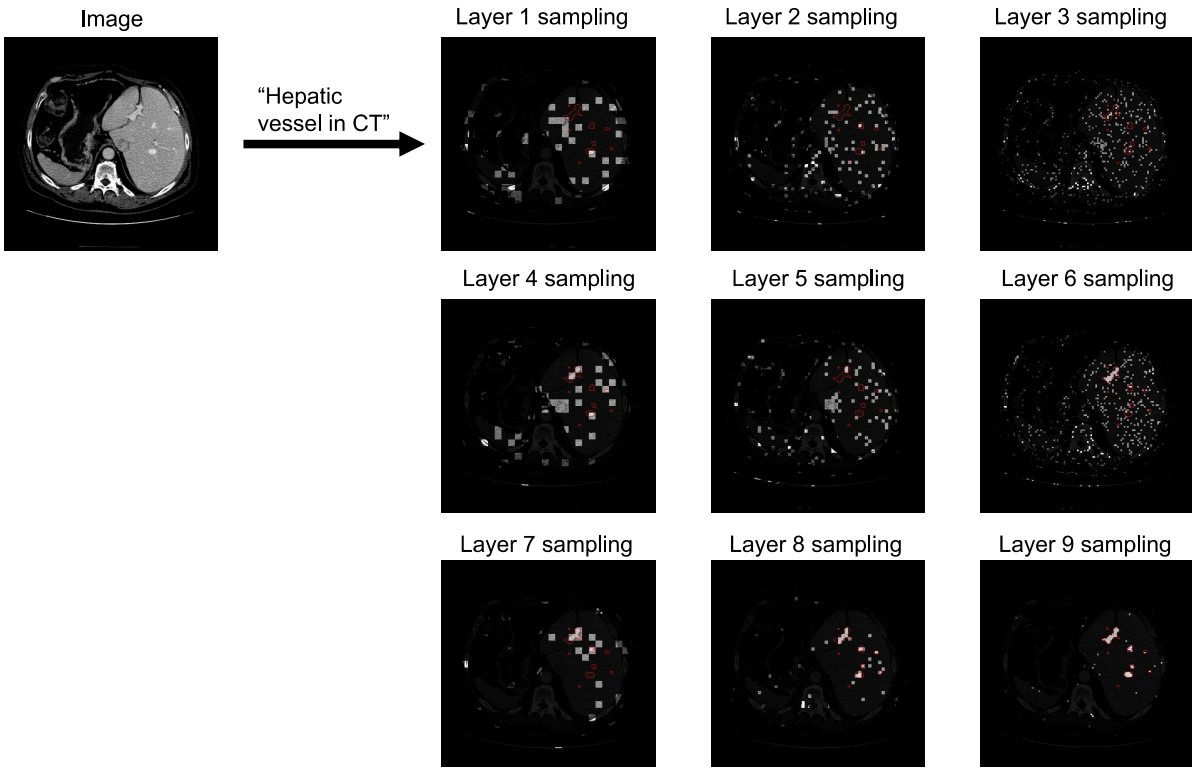

Figure 7. Boltzmann sampling example for hepatic vessel in CT. The sample patches are bright with target region circled in red.

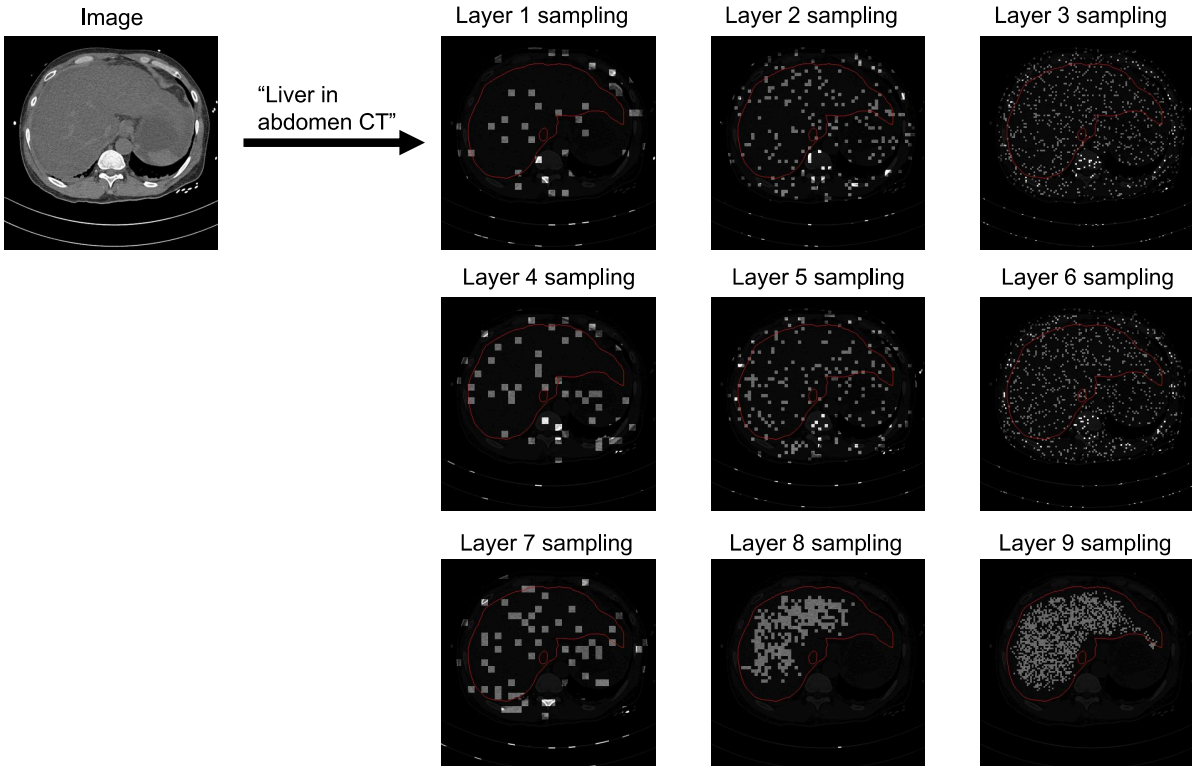

Figure 8. Boltzmann sampling example for liver in abdomen CT. The sample patches are bright with target region circled in red.

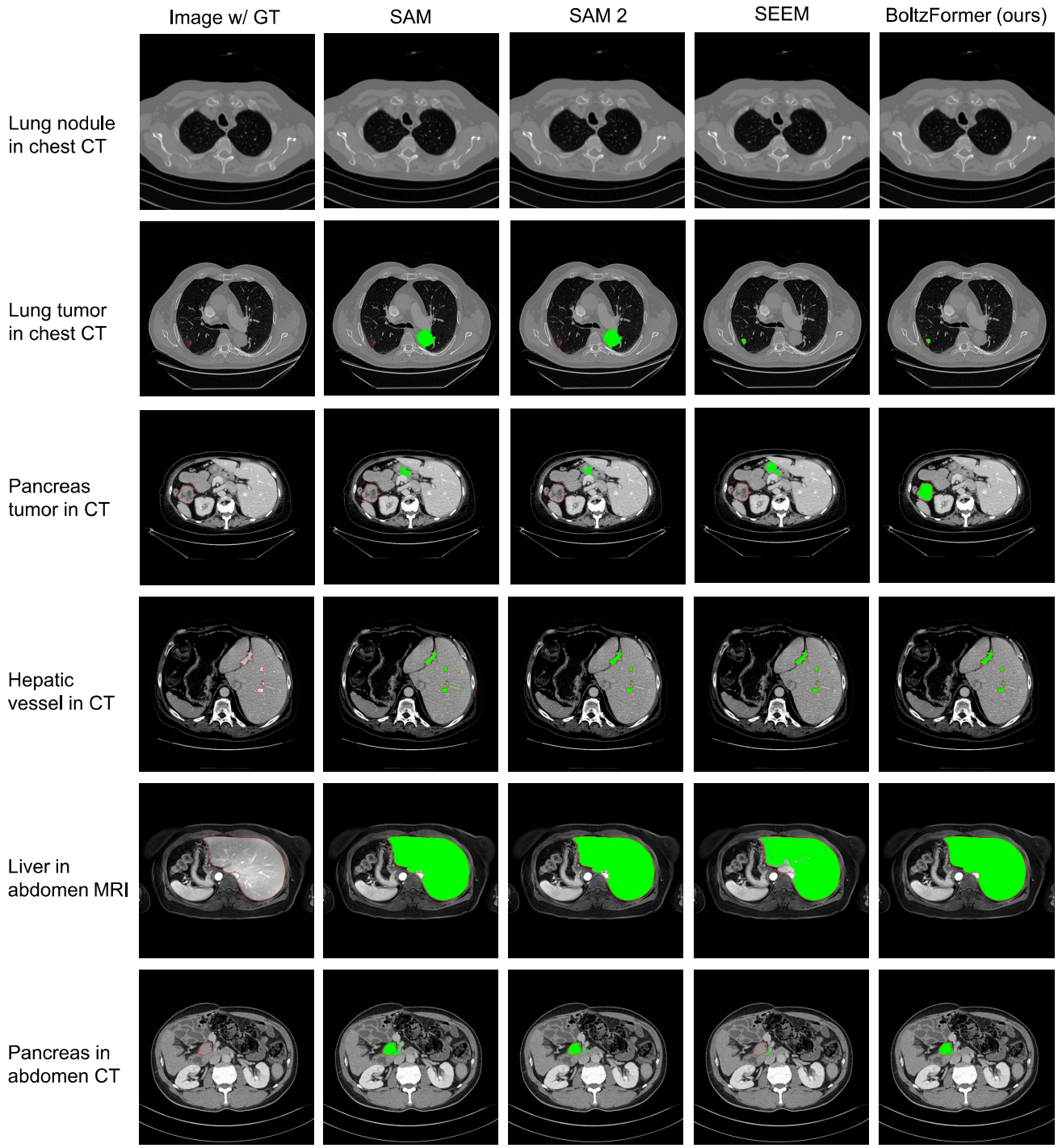

Figure 9. Segmentation prediction examples for baseline decoders and *BoltzFormer*. Target region is circled in red. Predicted segmentation masks are covered by green. We used the text prompts for all the models. The image backbone is fixed to Hiera-S for all the decoder models.