A Stability Evaluation

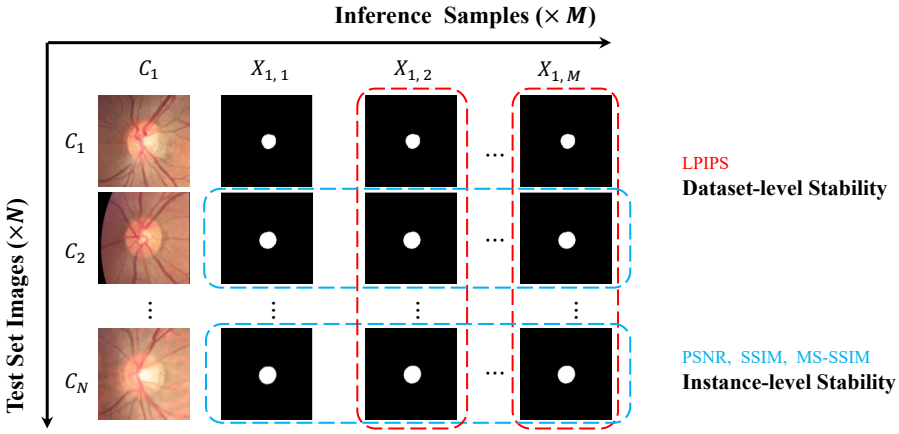

Fig. 1. Illustration of our Stability Evaluation on REF. We first conduct M times of inference process to prepare for the evaluation. Then, **Dataset-level Stability** is evaluated on every two sets of the inference results; **Instance-level Stability** is estimated on every two segmentation maps of each image conditioning.

B Qualitative Analysis

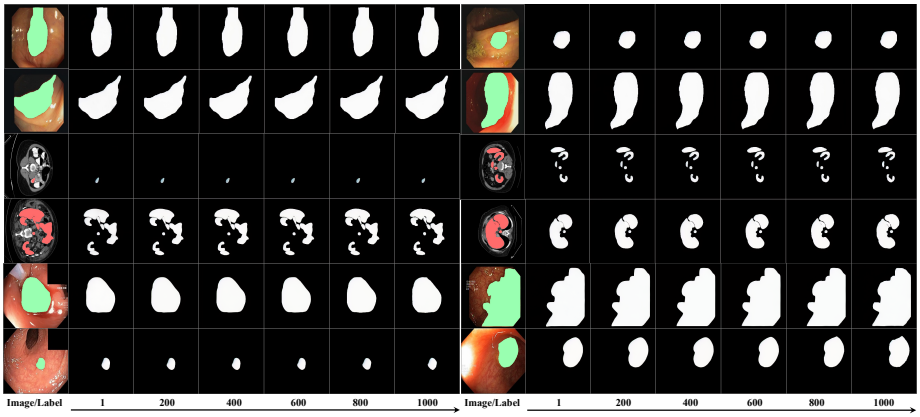

Fig. 2. From top to bottom: Visualization of the predicted probability maps in reverse process on CVC, BTCV, and KSEG (SDSeg trained for 50,000 steps). The horizontal axis denotes DDIM sampling steps. DDIM sampler generates fine and stable results during the entire reverse process. This demonstrates that SDSeg can generate great results under limited steps of the reverse process.

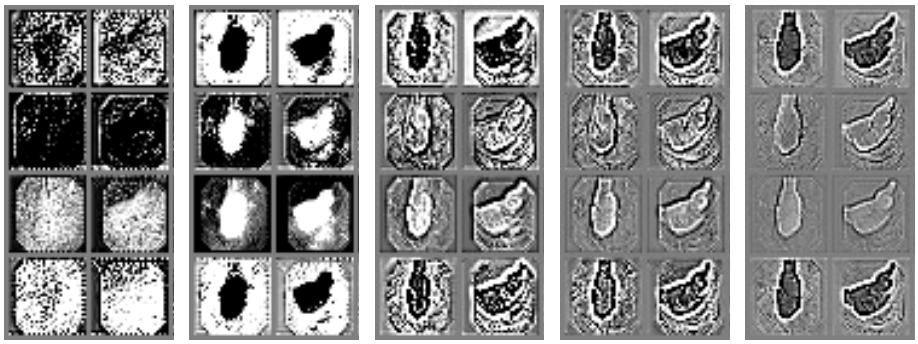

Fig. 3. Visualization of the latent representations of medical images from the trainable vision encoder, on CVC. At iteration 0, the encoder pre-trained on natural images couldn't capture enough meaningful semantic features for segmentation. During training, the conditioning encoder gradually learns to focus on segmentation targets.

C The architecture of the trainable vision encoder

We use a KL-regularized autoencoder model with the downsampling rate $r = \frac{H}{h} = \frac{W}{w} = 8$. The proposed trainable vision encoder has the same network architecture as the autoencoder model's encoder. Specifically, the trainable vision encoder's architecture can be separated into three blocks: the Downsampling block (Table. 1), the Mid block (2) and the Out block (3).

In Table. 1, 'Conv 3×3' denotes convolution block with kernel size 3, 'ResBlock' represents the building block in ResNet, and 'Down' corresponds to downsampling. In Table. 2, 'Attention' denotes self-attention block.

Table 1. The architecture of the **Downsampling** block.

input	$\mathbb{R}^{H\times W\times 3}$
Conv 3×3	$\mathbb{R}^{H\times W\times C}$
ResBlock×2+Down	$\mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$
ResBlock×2+Down	$\mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 2C}$
ResBlock×2+Down	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4C}$
ResBlock×2	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4C}$

Table 2. The architecture of the **Mid** block.

input	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4C}$
ResBlock	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4C}$
Attention	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4C}$
ResBlock	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4C}$

Table 3. The architecture of the **Out** block.

input	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4C}$
GroupNorm	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4C}$
Conv 3×3	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 2Z}$
Conv 1×1	$\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times Z}$

The input segmentation map $X \in \mathbb{R}^{H\times W\times 3}$ successively goes through these three blocks to get its corresponding latent representation $z \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times Z}$, where $C = 128$ is the channel dimension of the vision encoder, and $Z = 4$ is the channel dimension of the latent representation.