A Architectures

We evaluate two different segmentation networks proposed by [6], i.e., a fully connected neural network (FCN) and a convolutional neural network (CNN).

The FCN performs its segmentation pixel-wise, with an input of shape $\mathbb{R}^{(n_x \cdot n_y) \times n_c}$, whereby n_c contains RGB values and further pixel information, based on the selected input feature setting; $n_c = 5$, for RGB and spatial information ω, the same for RGB and semantic information ξ [1], and further $n_c = 7$, for RGB with spatial as well as semantic information. The FCN consists of 5 linear layers, with ReLU activations, respectively, and a width of 16.

On the other hand, the CNN operates on the full image of size $\mathbb{R}^{n_x \times n_y \times n_c}$, consisting of 4 convolutional layers with kernel size 3, width 16 and Leaky ReLU activation function.

Our input convex neural network $\mathcal{G}_\nu$ [2] is defined by 3 linear layers with a width of 130 and ReLU activations, and using pixel-wise spatial coordinates as input $\omega \in \mathbb{R}^2$. We additionally incorporate linear layer skip-connections (without bias), whose outputs are added in a point-wise manner to the respective output of the $\mathcal{G}_\nu$ layer outputs. Lastly, to ensure positivity of the output of each layer, which is crucial to form a convex region, the weights of the layers are altered using a ReLU function after each optimization step.

B Training Variants

B.1 Joint Training

For the training of our networks in a joint fashion, we use the energy minimization approach (3) with a binary cross entropy as data term and our proposed convexity regularizer (4). As data is scarce, e.g. the set of scribbled pixels $s \in [0, 1]$ in one image (f), whereby 0 denotes foreground and 1 background, one can evaluate the data term only within these. Yet, this could lead to an optimum of the smallest convex region around the foreground scribbles. To prevent this, we evaluate our regularizer on random pixels r in the case of FCN and on all pixels of the image when using the CNN. Preliminary work [6] has shown that an additional regularization with respect to the RGB color channels (c), or spatial inputs can be beneficial. Therefore, we consider for the CNN a mean gradient regularization of these input parts with respect to the sum of our segmentation network output (e.g. segmentation u), resulting in a combined cost function (6), with f concatenating all information, i.e., c, ω, ξ.

$$\begin{aligned}\mathcal{L}_{\text{joint}} = \underset{\theta,\nu}{\operatorname{argmin}}\ & \operatorname{BCE}(\mathcal{N}_\theta(f_s), s) + \operatorname{BCE}(\mathcal{G}_\nu(\omega_s), s) \\ & + \alpha \left\| \mathcal{N}_\theta(f_r) - \sigma(\mathcal{G}_\nu(\omega_r)) \right\| + \beta \sum_{i=0}^{P} \left\| \frac{\partial \mathcal{N}_\theta}{\partial c_i}(c_{s,r}, \omega_{s,r}, \xi_{s,r}) \right\| \\ & + \gamma \sum_{i=0}^{P} \left\| \frac{\partial \mathcal{N}_\theta}{\partial \omega_i}(c_{s,r}, \omega_{s,r}, \xi_{s,r}) \right\|,\end{aligned} \tag{6}$$

with α being an additional hyperparameter for our proposed convexity regularization method. The additional hyperparameter β influences the decision boundary for the RGB information, γ is the penalizer for the spatial decision boundaries. The networks were trained using Adam optimizer with a learning rate of 0.02 for 3000 optimization steps per image. For the first 200 steps, we set $\alpha = 0$, to train both networks, $\mathcal{N}_\theta, \mathcal{G}_\nu$, individually without regularization effects. This allows the networks to first individually predict stable segmentation before further joint optimization. Also with the start of joint training, we decrease the learning rate to 0.002. We set β and γ to 0.01. The joint loss function (6), is defined with the training for RGB, spatial and semantic features [1]. Following [6], we also performed training runs with RGB and spatial, as well as RGB and semantic features, respectively. If spatial features are not used for optimization, the corresponding regularization term is omitted ($\gamma = 0$). For the FCN, we only use the joint regularization term but no gradient regularization ($\beta = 0, \gamma = 0$); the other parameters are also consistent.

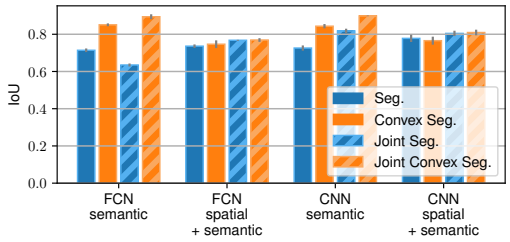

Figure 4: Experimental results w.r.t different input schemes. The convex prior segmentation clearly outperforms the normal segmentation when using semantic features. In all cases, joint training with the convex segmentation (dashed orange bar) works best. All numerical results of these experiments are also listed in Table 3.

B.2 Sequential Training

For the sequential training, we use the same models and parameters as for joint training. The difference to the previously mentioned *joint training* is that the segmentation model is trained first and afterward its output is projected onto the convex set by fitting the convex network. We use $\mathcal{L}_{seq}$ (7) and $\mathcal{L}_{cvx}$ (8) as loss functions for the respective networks, similar to the one stated in (6). The training procedure is also the same, except that we are keeping the learning rate constant over the whole training time.

$$\begin{aligned}\mathcal{L}_{\text{seq}} = \underset{\theta}{\operatorname{argmin}}\ \mathrm{BCE}(\mathcal{N}_\theta(f_s), s) + \beta\frac{1}{p}\sum_{i=0}^{P}\left\|\frac{\partial \mathcal{N}_\theta}{\partial c_i}(c_{s,r}, \omega_{s,r}, \xi_{s,r})\right\| \\ +\gamma\sum_{i=0}^{P}\left\|\frac{\partial \mathcal{N}_\theta}{\partial \omega_i}(c_{s,r}, \omega_{s,r}, \xi_{s,r})\right\|,\end{aligned} \tag{7}$$

$$\mathcal{L}_{cvx} = \underset{\nu}{\operatorname{argmin}}\ \mathrm{BCE}(\mathcal{G}_\nu(\omega_s), s) + \alpha\left\|\mathcal{N}_\theta(f_r) - \sigma(\mathcal{G}_\nu(\omega_r))\right\|. \tag{8}$$

We have visualized the training results in Figure 4.

B.3 Sequential Training with SAM

In case of SAM [11], we project its outputs similar to (8) on the convex set. Therefore, we evaluate the sigmoid of logits of the SAM model on the convexity dataset and sample an equal number of foreground and background points. The spatial information of these samples is then the input to the convex network.

Table 3: Segmentation results using different network types and their convex projection, as well as joint training of segmentation and input convex neural network. We report the mean, as well as the standard deviation, over three differently seeded runs. The intersection over union (IoU) as well as the pixel accuracy (Acc.) are calculated as the mean over each foreground object within the images in the used dataset. See also Figure 4 for a bar plot.

Training Type	Segmentation	Model	additional Input	IoU ↑	Acc. ↑
Individually Trained	Predicted Seg.	CNN	spatial	0.697 ± 0.066	0.903 ± 0.028
			semantic	0.726 ± 0.014	0.929 ± 0.006
			spatial and semantic	0.778 ± 0.019	0.937 ± 0.005
		FCN	spatial	0.732 ± 0.010	0.928 ± 0.003
			semantic	0.714 ± 0.010	0.929 ± 0.003
			spatial and semantic	0.736 ± 0.009	0.928 ± 0.003
	Convex Seg.	CNN	spatial	0.763 ± 0.013	0.934 ± 0.006
			semantic	0.843 ± 0.012	0.965 ± 0.004
			spatial and semantic	0.766 ± 0.022	0.935 ± 0.007
		FCN	spatial	0.711 ± 0.019	0.921 ± 0.006
			semantic	0.851 ± 0.008	0.967 ± 0.002
			spatial and semantic	0.746 ± 0.021	0.931 ± 0.005
Jointly Trained	Predicted Seg.	CNN	spatial	0.798 ± 0.004	0.943 ± 0.001
			semantic	0.818 ± 0.012	0.957 ± 0.003
			spatial and semantic	0.805 ± 0.014	0.946 ± 0.003
		FCN	spatial	0.755 ± 0.013	0.931 ± 0.004
			semantic	0.635 ± 0.008	0.898 ± 0.006
			spatial and semantic	0.768 ± 0.003	0.935 ± 0.002
	Convex Seg.	CNN	spatial	0.799 ± 0.005	0.944 ± 0.001
			semantic	0.899 ± 0.002	0.978 ± 0.000
			spatial and semantic	0.809 ± 0.015	0.948 ± 0.004
		FCN	spatial	0.756 ± 0.006	0.932 ± 0.001
			semantic	0.894 ± 0.013	0.977 ± 0.002
			spatial and semantic	0.769 ± 0.010	0.936 ± 0.003