Implicit Representations for Constrained Image Segmentation

Jan Philipp Schneider [1] **Mishal Fatima** [2] **Jovita Lukasik** [1]
Andreas Kolb [1] **Margret Keuper** [2 3] **Michael Moeller** [1]

Abstract

Implicit representations allow to use a parametric function that maps (spatial) coordinates to the value that is traditionally stored in each pixel, e.g. RGB values, instead of a discrete grid. This has recently proven quite advantageous as an internal representation for images or scenes for deep learning models. Yet, its potential to ensure certain properties of the solution has not yet been fully explored. In this work, we demonstrate that implicit representations are a powerful tool for enforcing a variety of different geometric constraints in image segmentation. While convexity, star-shape, path-connectedness, periodicity, or symmetry of the (spatial or space-time) region to be segmented are very challenging to enforce for pixel-wise discretizations, a suitable parametrization of an implicit representation, mapping spatial or spatio-temporal coordinates to the likeliness of a pixel belonging to the fore- or background, allows to **provably** ensure such constraints. Several numerical examples demonstrate that challenging segmentation scenarios can benefit from the inclusion of application-specific constraints, e.g. when occlusions prevent a faithful segmentation with classical approaches.

1. Introduction

The past decade has led to tremendous advances in the field of image segmentation via data-driven techniques, including seminal works such as the U-net architecture (Ronneberger et al., 2015) and, more recently, large foundation image segmentation models such as SAM (Kirillov et al., 2023). Yet, when training data are scarce or objects of interest are occluded, *constraints* to obey certain *modeling assumptions* can become crucial.

[1]University of Siegen [2]University of Mannheim [3]Max-Planck-Institute for Informatics, Saarland Informatics Campus. Correspondence to: Jan Philipp Schneider <Jan.Schneider@uni-siegen.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

The inclusion of modeling assumptions or constraints has been extremely successful in common (classical) energy minimization methods (Mumford & Shah, 1989; Vese & Chan, 2002), and lately been combined with learning-based approaches, e.g. by interpreting the (softmax-)output of a neural network as the solution to a minimization problem to which additional regularizers can be added (Liu et al., 2022). Yet, some seemingly simple constraints, such as the region to be segmented being *convex* or *path-connected*, can lead to highly complex regularizers. As segmentations of an image $f \in \mathbb{R}^{n_x \times n_y \times 3}$ are commonly represented as masks $m \in [0,1]^{n_x \times n_y}$, graphs with $n_x \cdot n_y$ many nodes or (discretized) level set functions $\phi \in \mathbb{R}^{n_x \times n_y}$ to identify the segmented region with $\{(i,j) \mid \phi_{i,j} \leq 0\}$, enforcing geometric properties often leads to constraints that (naïvely) grow quadratically in the number of pixels: For instance, *convexity* means that for any two points in the segmentation the line segment in between them is part of the segmentation; *path-connectedness* means that for any two points in the segmentation there exists a path between them within the segmentation. Thus, even relaxations of common segmentation problems face great difficulties in enforcing such properties and binary optimization approaches are typically NP-hard (see e.g. Royer et al. (2016)). We will provide a more detailed overview of related work on enforcing geometric constraints in image segmentation in Sec. 5.

In this work, we consider a significantly simpler option for still *provably* satisfying constraints: Instead of a discrete representation of the segmentation as, for example, a binary image (mask), we parameterize a segmentation function mapping from the image domain $\Omega \subset \mathbb{R}^2$ to the real numbers *implicitly* via a neural network: With seminal works such as Sitzmann et al. (2020) and Tancik et al. (2020b) demonstrating that fully connected networks (possibly including the computation of *Fourier Features* in the first layer) are well suited for representing natural images, we exploit such implicit representations for image segmentation. By choosing a parametric function $\mathcal{G}_\nu \colon \mathbb{R}^2 \to \mathbb{R}$ to map suitable coordinates to a segmentation likelihood such that $\{x \in \mathbb{R}^2 \mid \mathcal{G}_\nu(x) \leq 0.5\}$ represents the segmented region, and restricting the architecture in suitable ways, we can provably ensure a variety of constraints, see Fig. 1.

We summarize our main contributions as follows:

- We **prove** how (star-) convexity, mirror and rotation symmetry, periodicity and path-connectedness w.r.t. the object shape can be implemented with implicit representations.
- We propose enforcing these constraints in common deep learning pipelines and variational approaches.
- We show the segmentation improvements in cases where the assumed geometric properties of the segmented regions are valid.

We detail our methodology leading to the results shown in Fig. 1 in Sec. 2, showing the benefit of implicit representations for enforcing convexity and path-connectedness constraints quantitatively in Sec. 3, providing limitations in Sec. 4, before we summarize related work on constraining segmentations in Sec. 5. Lastly, we draw conclusions on the idea of implicit representations for image segmentation in Sec. 6. Our code is available at `https://github.com/jp-schneider/awesome`.

2. Implicit Representations for Segmentation

We first detail different parametrizations for representing segmentations as (coordinate-based) functions, before discussing different ways to benefit from such representations in variational as well as learning-based approaches to image segmentation.

2.1. Convexity

In case of high noise levels, little training data, or in the presence of occlusions, the assumption that the main object to be segmented is convex can significantly stabilize the segmentation as shown in Fig. 1a and 1b. Although the segmentation of a red tomato on a green plant is almost trivial, autonomous tasks like measuring the size of the tomato require a faithful segmentation of the whole tomato with occlusions. While approaches working with classical representations have to turn to computationally expensive approaches such as orientation-based lifting (Chen et al., 2021; 2023) or curvature penalties requiring the solution of a fourth-order differential equation in every step of an alternating direction method of multipliers (Luo et al., 2019), the use of *input convex neural networks* (Amos et al., 2017) for an implicit representation of the segmentation makes such a constraint easy to include. By choosing

$$\begin{aligned} &\mathcal{G}_\nu(x) = z^K, \quad z_{i+1} = \mathrm{ReLU}(\nu_i^z z_i + \nu_i^x x + b_i), \\ &\nu_i^z \geq 0 \quad \forall i \in \{1, \dots, K-1\}, \end{aligned} \tag{1}$$

as a network architecture, one can assure $\mathcal{G}_\nu$ to be convex in x (Amos et al., 2017) such that any level set, e.g.

$$\{x \in \mathbb{R}^2 \mid \mathcal{G}_\nu(x) \leq 0\} \tag{2}$$

can be used to represent a convex segmentation[1]. In (1), z_i denote the activations of the i-th layer, v_i^z are the weights of the i-th layer that are multiplied with the output of the previous layer, v_i^x are the weights of the i-th layer that are used in a skip-connection from the input, and b_i are the i-th layer's bias. As we can see in Fig. 1b, the occlusions consequently become part of the segmentation.

2.2. Star-Shape

If convexity is a too-restrictive assumption for the desired object, an assumption of a star-shape (c.f. Fig. 1c and 1d might still hold. A set S is called *star-shaped*, if there exists an $x_0 \in S$ such that for every $x \in S$, it holds that the line segment between x_0 and x is part of S, i.e., $\alpha x_0 + (1-\alpha)x \in S \ \forall\, \alpha \in [0, 1]$.

A simple network architecture for an implicit representation to ensure this property is to compute

$$\begin{aligned} x_{\text{new}}(x) &= x - \theta_{\text{off}} \\ r &= \|x_{\text{new}}\| \\ v &= \begin{cases} \frac{x_{\text{new}}}{r} & \text{if } r \neq 0, \\ (0,0)^T & \text{otherwise.} \end{cases} \\ \text{output}(x) &= \mathcal{G}_\nu(v, r) \quad \text{with } \mathcal{G}_\nu(v, 0) = -1 \ \forall v, \end{aligned} \tag{3}$$

for a learnable center $\theta_{\text{off}} \in \mathbb{R}^2$ and a partially input convex neural network $\mathcal{G}_\nu$, which is convex in r for every given v (see Amos et al. (2017) for a suitable construction). Constraining the network's output to be negative for a radius of 0, i.e., for $x = \theta_{\text{off}}$, can ensure a star-shape.

Proposition 2.1. *If the network $\mathcal{G}_\nu$ in (3) is convex in r, then the set*

$$S = \{x \in \mathbb{R}^2 \mid \textit{output}(x) \leq 0\}$$

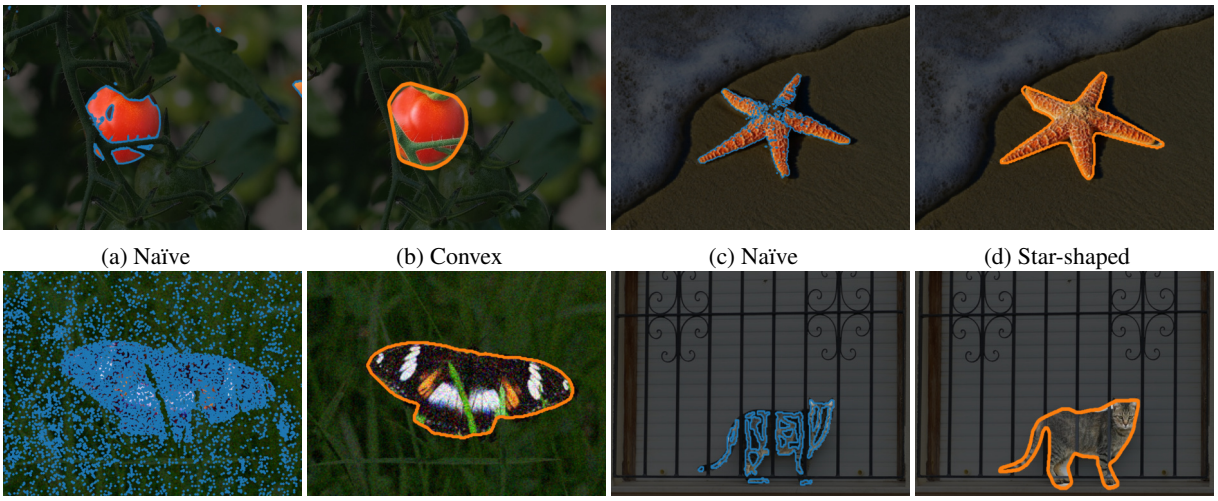
is star-shaped with center $\theta_{\textit{off}}$.

Proof. In a nutshell, the function (3) operates in polar coordinates while being convex along the radial direction. More formally, consider an arbitrary point $x \in S$, i.e., $\text{output}(x) < 0$. It holds that $\text{output}(\theta_{\text{off}}) = -1$ by definition. Now for any $\alpha \in]0, 1[$ it holds that

$$\begin{aligned} &\text{output}(\alpha x + (1-\alpha)\theta_{\text{off}}) \\ &= \mathcal{G}_\nu\left(\frac{x - \theta_{\text{off}}}{\|x - \theta_{\text{off}}\|}, \alpha\|x - \theta_{\text{off}}\|\right) \\ &\leq \alpha \mathcal{G}_\nu\left(\frac{x - \theta_{\text{off}}}{\|x - \theta_{\text{off}}\|}, \|x - \theta_{\text{off}}\|\right) + (1-\alpha)(-1) < 0. \end{aligned}$$

Thus, $(\alpha x + (1-\alpha)\theta_{\text{off}}) \in S$. □

[1] Amos et al. (2017) have shown that any network architecture as presented in (1) is input convex as long as a convex and non-decreasing activation function is used. For the sake of simplicity, we have used a ReLU function.



(a) Naïve (b) Convex (c) Naïve (d) Star-shaped

(e) Naïve (f) Mirror Symmetric (g) Naïve (h) Path-connected

Figure 1. Illustration of different constraints imposed via a suitable parametrization of the segmentation using an implicit representation. In blue, naïve *unaries* are illustrated, i.e., pointwise fore- and background likelihoods at each pixel. For simplicity, these were generated by color thresholding.

Note that the same network architecture without the constraint $\mathcal{G}_\nu(v, 0) = -1\ \forall v$ can also be of interest, leading to a connected interval in every radial direction, while, however, not necessarily being connected anymore.

2.3. Mirror and Rotational Symmetry

Using the same representation as in (3) (without any convexity of $\mathcal{G}_\nu$) easily allows the inclusion of rotational or mirroring symmetries: Thanks to the representation using polar coordinates, more precisely, the radius r and the vector v that corresponds to the sine and cosine of the polar coordinate angle, one can easily obtain rotational symmetry by applying a periodic function (with a suitable known or learnable period) to v. Similarly, by manipulating the sign of (components of) v, one obtains mirroring symmetries. Fig. 1f shows a fitting of a butterfly with (3) and $\mathcal{G}_\nu$ being a fully connected network that v after the mappings

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \leftarrow \begin{pmatrix} \cos(\theta_{\text{angle}}) & \sin(\theta_{\text{angle}}) \\ -\sin(\theta_{\text{angle}}) & \cos(\theta_{\text{angle}}) \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{4}$$

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \leftarrow \begin{pmatrix} |v_1| \\ v_2 \end{pmatrix} \tag{5}$$

instead of v itself as an input. The absolute value in Eq. 1(f) ensures the mirror symmetry, and the rotation matrix parameterized by θ_{angle} in (4) makes the symmetry axis learnable. As we can see in Fig. 1e and 1f, this allows to faithfully segment the symmetrized shape of the butterfly despite rather challenging likelihoods, occlusions, or smaller missing parts.

2.4. Path-Connected Regions with Genus Zero

Even more general than star-shaped regions are path-connected (PC) regions of genus zero (i.e., connected regions without holes) as illustrated in Fig. 1h. To realize such a constraint via an implicit representation, consider the composition $\mathcal{G}_\nu \circ \mathcal{D}_\phi$ of an input convex neural network $\mathcal{G}_\nu$ parameterized by ν and a diffeomorphism $\mathcal{D}_\phi$ parameterized by ϕ as e.g. frequently arising in the field of normalizing flows. Recall that a diffeomorphism is a continuous, differentiable, invertible transformation whose inverse is continuously differentiable as well. Again, we represent the set of points we consider to be the foreground as

$$S = \{x \in \mathbb{R}^2 \mid (\mathcal{G}_\nu \circ \mathcal{D}_\phi)(x) \leq t\} \tag{6}$$

for a suitable threshold t, e.g. $t = 0$.

Definition 2.2 (Path-connectedness)**.** Recall that a set S is called *path-connected* if for every two points $v, w \in S$ there exists a continuous map $p\colon [0, 1] \to \mathbb{R}^2$ such that $p(0) = v$, $p(1) = w$ and $p(s) \in S$ for all $s \in [0, 1]$.

Proposition 2.3. *The set S given by Eq. 6 is path-connected*[2]*.*

Proof. Let $v, w \in S$ be arbitrary. Define

$$p(s) = \mathcal{D}_\phi^{-1}(s\mathcal{D}_\phi(w) + (1 - s)\mathcal{D}_\phi(v)).$$

Then p is continuous because $\mathcal{D}_\phi$ and $\mathcal{D}_\phi^{-1}$ are, and it clearly holds that $p(0) = v$ and $p(1) = w$. Moreover, for any

[2]Note that while the same result also holds for $\mathcal{G}_\nu$ following (3), we focus on input convex networks here as this allows us to smoothly transfer between our experimental settings, see Sec. 4.

$s \in [0, 1]$:

$$\begin{aligned}(\mathcal{G}_\nu \circ \mathcal{D}_\phi)(p(s)) &= \mathcal{G}_\nu(\mathcal{D}_\phi(\mathcal{D}_\phi^{-1}(s\mathcal{D}_\phi(w) + (1-s)\mathcal{D}_\phi(v)))),\\ &= \mathcal{G}_\nu(s\mathcal{D}_\phi(w) + (1-s)\mathcal{D}_\phi(v)),\\ &\overset{\mathcal{G}_\nu \text{convex}}{\leq} s\mathcal{G}_\nu(\mathcal{D}_\phi(w)) + (1-s)\mathcal{G}_\nu(\mathcal{D}_\phi(v)),\\ &\overset{v,w \in S}{\leq} st + (1-s)t\\ &= t,\end{aligned} \tag{7}$$

which shows that $p(t) \in S$. □

Remark 2.4. Any smoothly path-connected set with genus zero can be represented as the zero level set of the composition of a diffeomorphism and an input convex function.

As we can see in Fig. 1h, even complex (highly non-convex) shapes such as the shape of a cat can be fitted via a suitable deformation of a convex set, resulting in a connected segmentation of the cat behind the fence of the balcony.

2.5. Union of Constrained Regions

The constraints discussed above are mostly useful for segmenting a single foreground object from the background. Yet, the minimum of multiple implicit representations of any of the above forms allows forming the union of the respective geometric constraints in the set S formed as a lower level set of the resulting function, thus further loosening restrictions on the particular segmentation. For multiple objects, the use of multiple implicit representations (possibly along with a penalty that reduces their overlap) might, however, be more advisable.

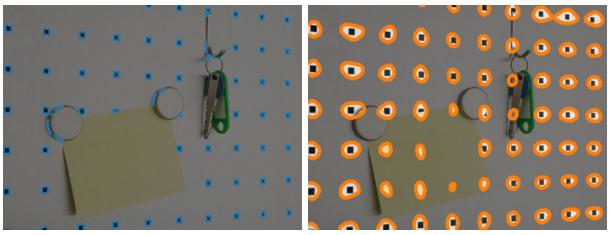
2.6. Periodicity

To illustrate an interesting constraint for image segmentation that is easy to realize via implicit representations, consider

$$\text{output}(x) = \theta_o + \sum_i \theta_f^{(i)} \sin(\theta_w^{(i)} x + \theta_b^{(i)}). \tag{8}$$

Due to the periodicity of the sine (and consequently also of the superposition of sine waves), the above results in a periodic function leading to the level sets also having a periodic structure.

Shown in Fig. 2a is a perforated metal sheet and a simple thresholding of the likelihood of holes in the metal sheet based on brightness. The resulting segmentation is very accurate but does not allow for any estimate of occluded holes, e.g. under the yellow paper, or behind the magnet or the keys. Using an implicit function parameterized via (8) that was optimized for matching the result of Fig. 2a only, results - for a suitable threshold - in the segmentation shown in Fig. 2b. While the contour around each hole is less precise, one can see that all but one hole has been identified despite the complete occlusions of several neighboring holes - a result that is very challenging to obtain if the frequency and orientation of the periodic pattern is unknown (i.e., learnable) and not perfectly regular (due to the perspective distortion in our example). An additional sparsity penalty on the $\theta_f^{(i)}$ decreases the expressiveness, but increases the extrapolation capabilities of the resulting segmentation.



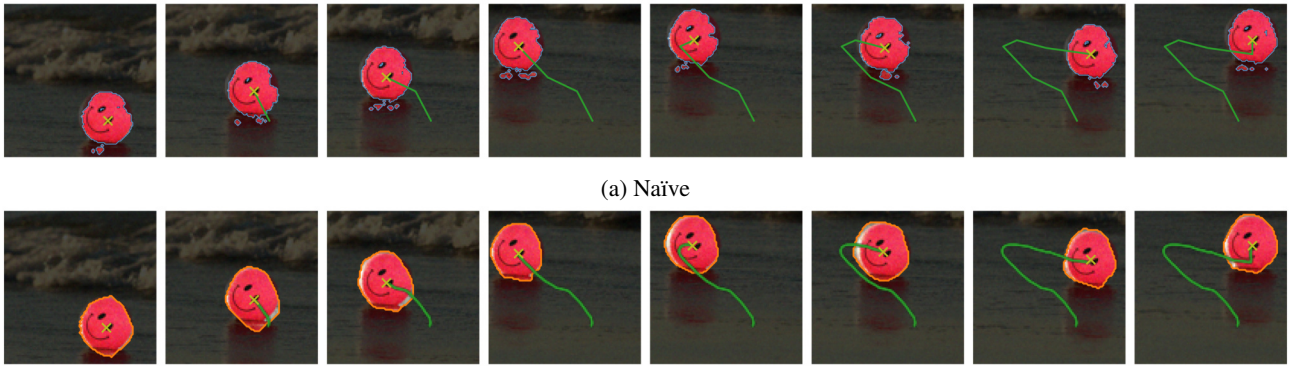
(a) Likelihood Thresholding (b) Periodic

Figure 2. Illustrating periodicity in implicit representations.

2.7. Extensions to Higher Dimensions

All implicit representations for constraining segmentations that we discussed above extend beyond the usual two spatial dimensions straightforwardly. While applications are obvious for the segmentation of 3D objects, e.g. in computerized tomography images, similar constraints can also be useful for tracking objects utilizing a spatio-temporal implicit representation for the segmentation of objects in a video: Fig. 3 demonstrates the use of the path-connectedness prior over several video frames of a ball at the beach. Interestingly, the spatio-temporal representation of the segmentation allows evaluating the segmentation even for times, for which no RGB-frame is available. Fig. 3b illustrates the movement of the object's centroid super-resolved in time as opposed to a pixel- and framewise segmentation 3a, which shows a significantly less smooth behavior.

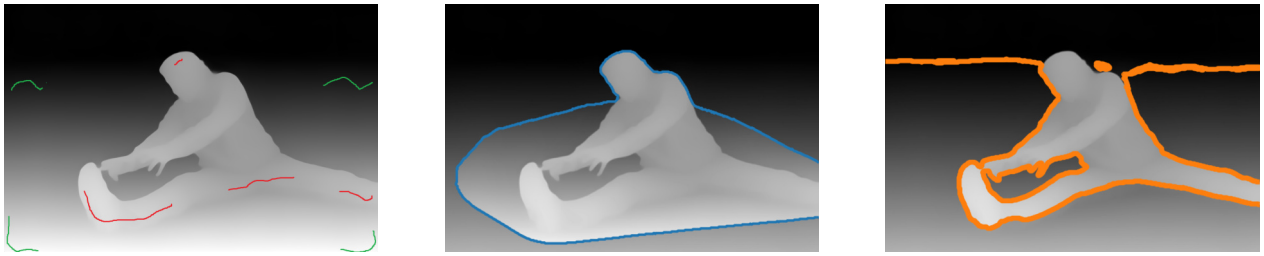
Furthermore, particular applications could make the use of convexity constraints in different coordinate spaces useful, e.g. the segmentation of a plane in a depth image: The depth values should change linearly with changing (x, y)-coordinates (leading to a convex object in 3D but not necessarily to a convex region in x and y). Fig. 4 illustrates the results one gets when training an implicit representation on predicting the foreground and background values, i.e., 0 and 1, on the given scribbles (marked as green and red lines for fore- and background, respectively, in the left image) only. We compare an unconstrained segmentation (similar to Dröge & Moeller (2021)) with an input convex network in $(x,\ y,\ \text{depth})$-space. As we can see, suitable constraints can help to improve the results significantly.



(a) Naïve

(b) Spatio-temporally Path-connected

Figure 3. Illustration of a spatio-temporal segmentation for the path of the segmentation's centroid with a naive approach (a), and a temporally super-resolved version (b) enabled by evaluating the implicit representation at intermediate time steps, leading to a significantly smoother and more realistic path.



(a) Input Scribbles (b) Without Convexity Constraint (c) With Convexity Constraint

Figure 4. Illustrating the results of the scribble-based segmentation with input scribbles shown on the left, with (right) and without (middle) convexity constraints in $(x,\ y,\ \text{depth})$-space. In this case, we segment the ground plane to get quite an accurate segmentation.

2.8. Inclusion in Segmentation Pipelines

2.8.1. Variational Methods and Postprocessing

Classical methods for image segmentation design a model-based function that computes the likelihood of each pixel belonging to the foreground, see e.g. Vese & Chan (2002) for a seminal classical work with very simple likelihood or Nieuwenhuis & Cremers (2013) for an extension to spatially varying color histograms for determining the likelihood. A suitably scaled version of such likelihoods is subsequently used as a data term in optimization-based approaches to image segmentation along with a regularizer such as the total variation (TV) to penalize the contour length. Similarly, researchers have used the prediction of machine learning methods to construct a data term in optimization-based methods with additional regularizers, see Dröge & Moeller (2021) for exemplary work on scribble-based segmentation.

Denoting the function of pointwise unaries by g, a suitable measure of similarity by d, and a suitable desired regularizer (such as the TV) by R, implicit representations can easily be incorporated into variational approaches by phrasing

$$\min_{\nu} \int_{\Omega} d(\sigma(\mathcal{F}_\nu(x)), g(x))\, dx + R(\mathcal{F}_\nu) \tag{9}$$

for $\mathcal{F}$ denoting any of the geometrically constraining implicit functions from above, ν denoting its learnable parameters, and σ denoting a suitable function to map the function's predictions to $[0, 1]$, e.g. a sigmoid (although small additional regularization such as weight decay might be required to guarantee the existence of minimizers of (9)).

We propose to handle (9) via stochastic optimization, i.e., using random points to (Monte-Carlo) approximate the integral in (9) in every step of the optimization, which we typically conduct using an ℓ^2-squared loss for d, no additional regularization R and an Adam optimizer (Kingma & Ba, 2015) for updating ν. Depending on the particular application, a weighting between fore- and background data consistency is used to account for occlusions or a systematically missing foreground likeliness in certain regions of g.

Naturally, a simple way of enforcing constraints on a learning-based approach as a post-processing step is to op-

timize (9) for g being the prediction of an (independently trained) segmentation network.

2.8.2. Neural Networks and Deep Learning

While the inclusion of an implicit representation in variational methods (or as post-processing) is straightforward via (9), there are several options for directly including them in a learning-based approach: First, one can design and train networks on predicting the parameters of an implicit representation (as, for instance, successfully done in (Chen & Wang, 2022)). Yet, the parameters of implicit representations are rather unstructured and one can not directly benefit from the large body of research on network architectures for image segmentation that have worked with pixel-based representations. Second, one can consider the post-processing (9) as a final layer of the network and differentiate through the corresponding optimization. Yet, the training becomes a bi-level optimization problem that is computationally challenging to handle. Therefore, if geometric information ought to be incorporated into a segmentation network's training procedure, we propose to use implicit representations as a soft penalty instead.

Note that any prediction of a network $\mathcal{N}_\theta$ working on the pixel grid of a given image $f \in \mathbb{R}^{n_y \times n_x \times 3}$ can be interpreted as a (piecewise constant) function $\mathcal{N}_\theta(f)\colon \Omega \to \mathbb{R}$. We propose to phrase the training process on data consisting of pairs of images f^i and corresponding ground truth segmentation u^i mathematically as

$$\min_\theta \sum_i \left(\text{loss}(\mathcal{N}_\theta(f^i), u^i) + \beta \cdot \text{dist}(\mathcal{N}_\theta(f^i), M)\right), \quad (10)$$

where M denotes the *set of all functions* whose zero level sets have a desired geometrically constraining property, e.g. the set of all convex functions. dist denotes the distance of an element ($z := \mathcal{N}_\theta(f)$) to a set, defined as

$$\text{dist}(z, M) = \min_{g \in M} \|z - g\|, \quad (11)$$

which consequently assures a projection of z to M. By approximating the set of all functions with the desired properties with the implicit representations $\mathcal{F}_\nu$ parameterized by ν as discussed above, we arrive at an overall training problem of

$$\min_{\theta, \{\nu^i\}} \sum_i \left(\text{loss}(\mathcal{N}_\theta(f^i), u^i) + \beta \cdot \text{dist}(\mathcal{N}_\theta(f^i), \mathcal{F}_{\nu^i})\right). \quad (12)$$

Although the above approach requires keeping track of one implicit representation, i.e., one set of parameters ν^i per training example, the storage overhead is almost negligible in comparison to the images themselves. In our examples on PC constraints below, our implicit representation consists of the composition of a parameterized diffeomorphism with 3,936 parameters and an input convex network with 35,103 parameters, resulting in a total number of 39,043 parameters versus an RGB-image with a lateral length of 1,000 pixels, where three million values have to be stored.[3]

3. Numerical Experiments

We analyze the possible advantages of the proposed framework for two of the above constraints, namely convexity and path-connectedness, more extensively in the numerical experiments presented below. All details of our numerical experiments can be found in the appendix (B, C).

3.1. Convexity Constraints

To investigate the influence of implicit convex representations numerically, we exploit the scribble-based convexity dataset (Gorelick et al., 2014). It consists of 51 images with user scribbles, and (approximately) convex foreground objects to be segmented.

We adopt the experimental setup of Dröge & Moeller (2021) for single image scribble-based segmentation with neural networks and train simple convolutional neural networks (CNNs) or pixel-wise fully connected networks (FCNs) to predict the correct label for the labeled (scribbled) pixels. The input to both types of architectures are RGB values along with spatial coordinates, semantic features from Aksoy et al. (2018), or a combination of both. We test both, the sequential approach (*seq.*) of first training a CNN or FCN on the scribbles, letting it predict a likelihood for each pixel and then fitting our implicit convex representation (prior) to the fixed likelihoods, as well as the *joint* approach (12) (with a single (scribbled) image only). Table 1 summarizes the intersection over union (IoU) obtained with the three different types of inputs to the two different types of segmentation networks (CNN and FCN) in the two cases of sequential and joint training. The IoU is stated for both, the original segmentation network as well as for our fitted, provably convex implicit representation (denoted by 'convex' in Tab. 1).

We can see that the implicitly enforced convexity assumption can improve the results, with a joint unification being superior to a sequential one. Interestingly, the impact of the convex projection method is significantly larger when the segmentation network does not receive any spatial information, which leads to substantial improvements and is the best-performing approach by far.

[3] For simplicity, neglecting metadata and compression, we are considering an image with one-megapixel resolution, RGB color channels of bit depth 8 ($\approx 1,000^2 \cdot 3$ bytes), which is still 4.8 times more than storing 39,043 parameters as single-precision floating point numbers with 4 bytes each.

Table 1. Intersection over Union (IoU) of the foreground object w.r.t the ground truth. We report the result for sequentially fitting an implicit convex representation to a fixed prediction of a trained network (first row) as well as the use of the implicit convexity prior as a penalty during training using (12) (second row). All provided numbers are averages over three runs as well as all images of the convexity dataset. For standard deviation and pixel accuracy values we refer to Tab. 3.

	RGB+spatial		RGB+semantic		RGB+spatial+semantic	
	CNN / convex	FCN / convex	CNN / convex	FCN / convex	CNN / convex	FCN / convex
seq.	0.697 / **0.763**	**0.732** / 0.711	0.726 / **0.843**	0.714 / **0.851**	**0.778** / 0.766	0.736 / **0.746**
joint	0.798 / **0.799**	0.755 / **0.756**	0.818 / **0.899**	0.635 / **0.894**	0.805 / **0.809**	0.768 / **0.769**

We exemplify the effect of the projection as well as the joint training qualitatively in Fig. 5: While the original segmentation $\mathcal{N}_\theta$ is highly scattered (b), an implicit input convex projection yields the segmentation of the main convex object (c). Joint training allows both representations to find an agreement leading to even more accurate contours (d).

3.2. Path-Connected Constraints

For the numerical evaluation of our path-connected (PC) constraint or prior, we consider the problem of motion segmentation and adopt the experimental setup by Kardoost & Keuper (2021) in which sparse labels are created via a multicut approach (Keuper et al., 2015a) on the estimated motion of objects in a video. The resulting clusters are subsequently densified using a UNet (Ronneberger et al., 2015) on the previously found labels. We evaluate the use of implicit representations for our PC prior on the sequences (18 in total) from the FBMS-59 dataset (Brox & Malik, 2010; Ochs et al., 2013) where the baseline implementation (Kardoost & Keuper, 2021) segmented single objects. Similar to the above, we study the use of a sequential fitting of the implicit representation to precomputed labels as well as a joint training via (12). Averaging the IoU over all frames of all 18 videos, we find that the sequential enforcement of PC yields a gain of about 1% in IoU, while the joint training gains about 2% IoU over the baseline. Looking at the results for the different videos in more detail (see Tab. 4 in the supplementary material for all 18 sequences), it is insightful to see that in some cases, where the PC segmentations have a worse IoU are due to the ground truth not annotating occlusions as part of the object, c.f. Fig. 6.

4. Limitations and Trade-Offs

One trade-off of the continuously defined implicit representations and geometric constraints they enforce is that properties like path-connectedness can be satisfied via infinitely thin connections that get lost when visualizing results by evaluating the implicit representation on a discrete grid (see Fig. 14 in the appendix). Closely related, minimizers of the fitting costs might not exist as the minimization converges to infinitely thin connections between regions that the costs prefer to be unconnected. To prevent such behavior, stabilizing mechanisms like the use of ℓ^2-squared regularization (weight decay) are necessary. Interestingly, a parameterized diffeomorphism with all parameters being equal to zero represents the identity, such that the corresponding weight decay acts as a trade-off parameter between fitting even thin, highly non-convex regions (such as legs of a horse) and preventing unwanted thin connections between essentially disconnected regions: Fig. 7 demonstrates that an increasing weight decay parameter allows to smoothly transfer from an almost arbitrary connected (expressive but fragile) to a convex (less expressive but highly robust) region.

We proposed to use our constraints in applications where data is scarce such that usual data-driven approaches tend to produce undesired results. Nevertheless, a discussion on using our approach with SOTA data-driven models on non-binary segmentation tasks and large datasets might be insightful. In general, our constraints can be used with any unaries or model $\mathcal{N}_\theta$ as described in 2.8.2. Above, we were limiting ourselves to one object per image, e.g. one set of parameters ν^i per image i, while an extension to multiple objects is in principle straightforward. The problem of segmenting an image into j-many (not necessarily disjoint) regions can be decomposed into j binary segmentation problems. Further, (12) needs to be altered by iterating over all j binary segmentations using per object weights $\nu^{i,j}$ for the implicit representation $\mathcal{F}_{\nu^{i,j}}$ in the regularizer. If one wants to evaluate on large segmentation datasets, the choice of a suitable constraint is of great importance. Our proposed constraints differ significantly in their expressivity and the amount of objects they can adequately represent. Unfortunately, the PC constraint which is the weakest but most expressive one, can lead to less optimal results when the ground truth in datasets is often not even connected (Fig. 6). Thus, one can either accept this potential performance loss, e.g. if the unaries are already scattered and thus the constraint still adds value (Fig. 5, 15, 18, 19), or one must assure the constraint is valid within the examples.

5. Related Work

Discrete Perspectives on Image Segmentation In parallel to the continuous relaxation approaches described in Sec. 2.8.1, discrete perspectives (typically exploiting min-cut/max-flow algorithms (Boykov & Kolmogorov, 2004))

(a) Scribbled Image (b) Segmentation (c) Sequential Convex Fitting (d) Joint Training

Figure 5. Qualitative results of a FCN segmentation, trained on scribbles (a) with RGB input and semantic features. In (b), the learned segmentation is given, which is very scattered, due to the lack of spatial information, while its convex projection (c) fits the balloon quite well. We get an even better segmentation using the joint training approach (d).

Figure 6. Ground truth annotations, baseline network prediction, and implicitly connected segmentation of one frame of the "cats05" sequence. In this case, the ground truth does not indicate a connected object - which contradicts our PC constraint. Accordingly, the prior with 57.8% IoU is inferior to the segmentation network (62.3%). Despite the IoU drop of 4.5%, it is application-dependent which of the segmentations are preferable.

were developed such as the seminal graph cuts (Boykov & Jolly, 2001) or grab cuts (Rother et al., 2004). Their underlying costs are similar to classical work (Vese & Chan, 2002) with a stronger perspective on images as graphs and the total variation prior typically being anisotropic. Extensions to an unknown number of regions, for instance, involve multicut approaches, c.f. Andres et al. (2011); Keuper et al. (2015a).

Implicit Representations Implicit representations already have many applications in computer vision after (Sitzmann et al., 2020; Tancik et al., 2020a) showed their effectiveness. These were initially realized as fully connected neural networks, while later works substituted these for performance reasons, for example (Fridovich-Keil et al., 2022; Kerbl et al., 2023). Notable is OmniMotion (Wang et al., 2023) as they also use an implicit volume representation with a diffeomorphism for their tracking, as we do for the realization of our PC prior, but without the guarantee of path-connectedness. Furthermore, Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) are also similar due to their volume representation of the scene, but their area of application is different and they do not guarantee any constraints.

There are only a few works that combine implicit representations with segmentation. Two notable exceptions exist in the medical context for resolution-independent representations of segmentations (Stolt-Ansó et al., 2023; Khan & Fang, 2022), but none of them use nor guarantee geometric constraints.

Geometric Constraints on Segmentation Geometric constraints such as path-connectedness or convexity of the shape to be segmented have received significant attention: Early works using segmentation representations via a polygon representing the boundary of the objects such as *snakes* (Marcos et al., 2018; Cheng et al., 2019) implicitly satisfy the PC constraint, but are prone to get stuck in bad local optima or using multicuts and being in an NP-hard formulation (Royer et al., 2016). Yet, similar representations have gained recent attention in the context of convexity priors exploiting orientation-based lifting (Chen et al., 2021; 2023). Level-set functions have been made topology-preserving by preventing local changes that would alter the topology (Han et al., 2003), yet consequently also have difficulties overcoming local minima. Alternative approaches for the convexity of a level-set-based segmentation are curvature-based penalties that are incorporated into a minimization based on the alternating direction method of multipliers (ADMM) (Luo et al., 2019).

More recent approaches work on discretizations of the variational method (Vese & Chan, 2002), e.g. using a heuristic for projecting onto convex sets characterized via a certain condition about the convolution with circular kernels (Liu et al., 2020), or characterizing the numerical computability of convex shape priors (Luo et al., 2023).

The property of an object being star-convex is frequently discussed in cell detection, segmentation, and accurate volume calculation, e.g. the StarDist (Schmidt et al., 2018; Weigert et al., 2020), and further enhanced by addressing overlapping objects (Walter et al., 2021). To overcome the lack of labeled samples in the microscopy domain, (Dey

(a) $\alpha = 1 \cdot 10^{-6}$ (b) $\alpha = 5 \cdot 10^{-4}$ (c) $\alpha = 5 \cdot 10^{-3}$

Figure 7. PC prior with different weight-decay penalties α on the diffeomorphism.

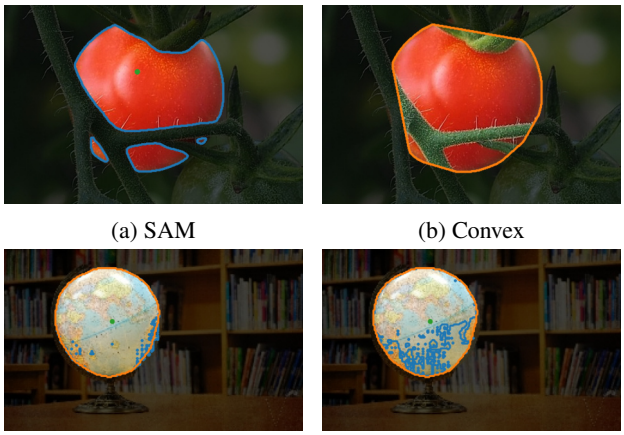
et al., 2024) proposed a framework to generate synthetic data for star-convex 3D instance segmentation networks.

The mirror symmetry constraint has been used in combination with segmentation models, for example, in the recent SymmNeRF (Li et al., 2022) or SymmSLIC (Nagar & Raman, 2017) which is an extension of the superpixel segmentation approach SLIC (Achanta et al., 2012). Both approaches were able to achieve improvements compared to their baseline insofar as their symmetry condition applies.

The use of a PC constraint in image segmentation has been studied in various domains - like medical (Rempfler et al., 2016) or robotics (Milioto et al., 2019) - and in combination with various models. Early works as (Andres et al., 2011), later extended in (Keuper et al., 2015b; Levinkov et al., 2017) using probabilistic models or rely on graph cut segmentations, e.g. in Isack et al. (2016; 2018), which were also proven to be NP-hard.

Naturally, PC constraints also occur in the area of object tracking, often addressed with disjoint path models (Hornakova et al., 2020). The constraint proves to be beneficial, as it prevents label flipping and thus enables more stable tracking. In the area of scribble-based segmentation (Shen et al., 2020) and panoptic segmentation (Shen et al., 2022), the path-connectedness combined with a superpixel approach was implemented, leading to a more robust approach.

Constraints in the Era of Foundation Models Recent foundation models such as SAM (Kirillov et al., 2023) or SEEM (Zou et al., 2024) have considerably improved the image segmentation task due to their data-driven approaches. In some cases, however, they may fail or not be the appropriate method of choice[4]. Suppose one considers a volume-preserving segmentation, as in Fig. 8a where we try to segment a tomato occluded by its leaf petioles with a single point prompt (green) using SAM. In such case, occlusions should be part of the segmentation, while SAM divides it. Opposingly, the convex projection (Fig. 8b) of the SAM unaries leads to a better segmentation. From a robustness perspective, SAM may yield a scattered segmentation. For illustration, we prompted SAM on the clean image of a globe (Fig. 8c) producing already a scattered segmentation (blue), which becomes even worse when we apply a small brightness corruption of severity 1 (Hendrycks & Dietterich, 2018) as shown in Fig. 8d. The convex projection (orange) yields a better, connected segmentation in both cases. While both problems may be solved by human supervision and suitable prompt selection, geometric constraints can be essential in their absence.



(a) SAM (b) Convex

(c) Clean Image (d) Corrupted Image

Figure 8. Segmentation of SAM and its convex projection in an occlusion and corruption scenario.

6. Conclusions

This work is the first to study the use of implicit representations in image segmentation to (provably) enforce geometric constraints. We highlighted the versatility of the approach, provided different architecture templates for different constraints, and evaluated the (small but systematic) benefits of enforcing constraints in several numeric experiments, which suggest that the direct inclusion of constraints into the networks' training has advantages over a sequential fitting.

[4]Two further challenging scenarios are given in appendix A.

Impact Statement

This paper presents work whose goal is to advance the field of image segmentation using implicit representations. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgment

The authors acknowledge support by the DFG research unit 5336 - Learning to Sense.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. ISSN 0162-8828.

Aksoy, Y., Oh, T.-H., Paris, S., Pollefeys, M., and Matusik, W. Semantic soft segmentation. *ACM Transactions on Graphics*, 37(4):1–13, 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201275.

Amos, B., Xu, L., and Kolter, J. Z. Input convex neural networks. In *International Conference on Machine Learning*, pp. 146–155, 2017.

Andres, B., Kappes, J. H., Beier, T., Köthe, U., and Hamprecht, F. A. Probabilistic image segmentation with closedness constraints. In *2011 International Conference on Computer Vision*, pp. 2611–2618. IEEE, 2011.

Boykov, Y. and Jolly, M.-P. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pp. 105–112 vol.1, 2001.

Boykov, Y. and Kolmogorov, V. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

Brox, T. and Malik, J. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pp. 282–295, 2010.

Chen, D., Cohen, L. D., Mirebeau, J.-M., and Tai, X.-C. An elastica geodesic approach with convexity shape prior. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021. doi: 10.1109/iccv48922.2021.00682.

Chen, D., Mirebeau, J.-M., Shu, M., Tai, X., and Cohen, L. D. Geodesic models with convexity shape prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8433–8452, 2023. doi: 10.1109/TPAMI.2022.3225192.

Chen, Y. and Wang, X. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pp. 170–187. Springer, 2022.

Cheng, D., Liao, R., Fidler, S., and Urtasun, R. Darnet: Deep active ray network for building segmentation. In *CVPR*, 2019.

Dey, N., Abulnaga, M., Billot, B., Turk, E. A., Grant, E., Dalca, A. V., and Golland, P. Anystar: Domain randomized universal star-convex 3d instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7593–7603, 2024.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017.

Dröge, H. and Moeller, M. Learning or Modelling? An analysis of single image segmentation based on scribble information. In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 2274–2278, 2021.

Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., and Kanazawa, A. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5501–5510, 2022.

Gorelick, L., Veksler, O., Boykov, Y., and Nieuwenhuis, C. Convexity shape prior for segmentation. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 675–690, 2014.

Han, X., Xu, C., and Prince, J. L. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

Hornakova, A., Henschel, R., Rosenhahn, B., and Swoboda, P. Lifted disjoint paths with application in multiple object tracking. In *International Conference on Machine Learning*, pp. 4364–4375, 2020.

Isack, H., Gorelick, L., Ng, K., Veksler, O., and kov, Y. K-convexity shape priors for segmentation. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.), *Computer vision – ECCV 2018. Part XI*, LNCS

sublibrary. SL 6, Image processing, computer vision, pattern recognition, and graphics, pp. 38–54. Springer, Cham, Switzerland, 2018. ISBN 9783030012519. doi: 10.1007/978-3-030-01252-6_3.

Isack, H. N., Veksler, O., Sonka, M., and Boykov, Y. Hedgehog shape priors for multi-object segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.

Kardoost, A. and Keuper, M. Uncertainty in minimum cost multicuts for image and motion segmentation. In *Uncertainty in Artificial Intelligence*, pp. 2029–2038, 2021.

Kerbl, B., Kopanas, G., Leimkühler, T., and Drettakis, G. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Keuper, M., Andres, B., and Brox, T. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3271–3279, 2015a.

Keuper, M., Levinkov, E., Bonneel, N., Lavoue, G., Brox, T., and Andres, B. Efficient decomposition of image and mesh graphs by lifted multicuts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015b.

Khan, M. O. and Fang, Y. Implicit neural representations for medical imaging segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 433–443. Springer, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2015.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *Advances in Neural Information Processing Systems*, 31, 2018.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W., Dollár, P., and Girshick, R. B. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Levinkov, E., Uhrig, J., Tang, S., Omran, M., Insafutdinov, E., Kirillov, A., Rother, C., Brox, T., Schiele, B., and Andres, B. Joint graph decomposition & node labeling: Problem, algorithms, applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6012–6020, 2017.

Li, X., Hong, C., Wang, Y., Cao, Z., Xian, K., and Lin, G. SymmNeRF: Learning to explore symmetry prior for single-view view synthesis. In *Proceedings of the Asian Conference on Computer Vision*, pp. 1726–1742, 2022.

Liu, J., Tai, X.-C., and Luo, S. Convex shape prior for deep neural convolution network based eye fundus images segmentation. *arXiv preprint arXiv:2005.07476*, 2020.

Liu, J., Wang, X., and Tai, X.-C. Deep convolutional neural networks with spatial regularization, volume and star-shape priors for image segmentation. *Journal of Mathematical Imaging and Vision*, 2022. doi: 10.1007/s10851-022-01087-x.

Luo, S., Tai, X.-C., Huo, L., Wang, Y., and Glowinski, R. Convex shape prior for multi-object segmentation using a single level set function. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Luo, S., Chen, J., Xiao, Y., and Tai, X.-C. A binary characterization method for shape convexity and applications. *Applied Mathematical Modelling*, 2023.

Marcos, D., Tuia, D., Kellenberger, B., Zhang, L., Bai, M., Liao, R., and Urtasun, R. Learning deep structured active contours end-to-end. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Milioto, A., Mandtler, L., and Stachniss, C. Fast instance and semantic segmentation exploiting local connectivity, metric learning, and one-shot detection for robotics. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5481–5487. IEEE, 2019.

Mumford, D. and Shah, J. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

Nagar, R. and Raman, S. Symmslic: Symmetry aware superpixel segmentation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1764–1773, 2017.

Nieuwenhuis, C. and Cremers, D. Spatially varying color distributions for interactive multilabel segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1234–1247, 2013.

Ochs, P., Malik, J., and Brox, T. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2013. ISSN 0162-8828.

Rempfler, M., Andres, B., and Menze, B. H. The minimum cost connected subgraph problem in medical image analysis. In *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III 19*, pp. 397–405. Springer, 2016.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, 2015.

Rother, C., Kolmogorov, V., and Blake, A. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

Royer, L. A., Richmond, D. L., Rother, C., Andres, B., and Kainmueller, D. Convexity shape constraints for image segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. doi: 10.1109/cvpr.2016.50.

Schmidt, U., Weigert, M., Broaddus, C., and Myers, G. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pp. 265–273, 2018.

Shen, R., Tang, B., Lodi, A., Tramontani, A., and Ayed, I. B. An ILP model for multi-label MRFs with connectivity constraints. *IEEE Transactions on Image Processing*, 29: 6909–6917, 2020.

Shen, R., Tang, B., Lodi, A., Ayed, I. B., and Guthier, T. Connectivity-constrained interactive panoptic segmentation. *arXiv preprint arXiv:2212.06756*, 2022.

Sitzmann, V., Martel, J. N. P., Bergman, A. W., Lindell, D. B., and Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

Stimper, V., Liu, D., Campbell, A., Berenz, V., Ryll, L., Schölkopf, B., and Hernández-Lobato, J. M. normflows: A pytorch package for normalizing flows. *Journal of Open Source Software*, 8(86):5361, 2023. doi: 10.21105/joss.05361.

Stolt-Ansó, N., McGinnis, J., Pan, J., Hammernik, K., and Rueckert, D. Nisf: Neural implicit segmentation functions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 734–744. Springer, 2023.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33: 7537–7547, 2020a.

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, 2020b.

Vese, L. A. and Chan, T. F. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50(3): 271–293, 2002.

Walter, F. C., Damrich, S., and Hamprecht, F. A. Multistar: Instance segmentation of overlapping objects with star-convex polygons. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 295–298, 2021.

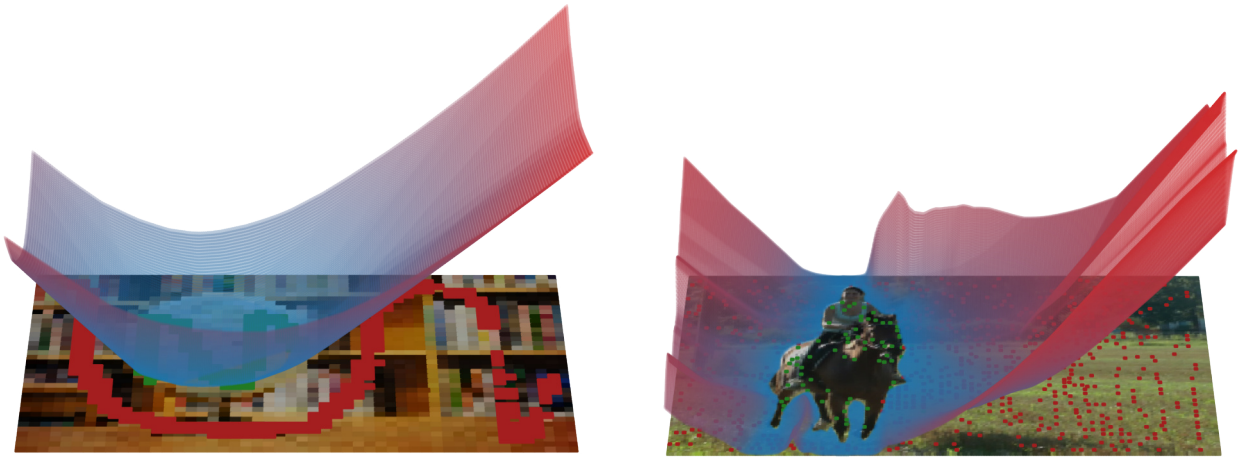
Wang, Q., Chang, Y.-Y., Cai, R., Li, Z., Hariharan, B., Holynski, A., and Snavely, N. Tracking everything everywhere all at once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19795–19806, 2023.

Weigert, M., Schmidt, U., Haase, R., Sugawara, K., and Myers, G. Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2020. doi: 10.1109/wacv45572.2020.9093435.

Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J., and Lee, Y. J. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36, 2024.

Supplementary Material

The supplementary material will go into more detail about our Motivation A and the experiments we conducted. As we mainly carried out our experiments concerning the convexity and path-connectedness, we will have a detailed discussion in the sections B and C respectively. The code used for all experiments is provided at https://github.com/jp-schneider/awesome.



(a) Convex Representation

(b) Path-connected Representation

Figure 9. Super-sampled implicit convex and path-connected representation. The output of $\mathcal{F}_\nu$ without sigmoid is displayed as a surface, the intersection with the image represents the segmented foreground.

A. Challenging Scenarios for SOTA Segmentation Models

As discussed in Sec. 5, foundation models like SAM may produce unwanted results or fail in occlusion and corruption scenarios. Further problems are out-of-distribution prompts and prompt ambiguity. If one has user scribbles at hand, like the ones proposed in the convexity dataset (Gorelick et al., 2014) (Fig. 10a) they can not be used to prompt SAM directly, as SAM will produce a scattered result (Fig. 10b). This is most likely due to an out-of-distribution prompt.

The automatic selection of one or more points from the scribble set is also difficult due to the ambiguity of the point prompts. Clicking on a country on the globe may mean only one country in conjunction with the entire globe (Fig. 10c), whereby two clicks without human supervision do not necessarily lead to a segmentation of the whole globe (Fig. 10d).
A prior known constraint as the globe is round (or convex) that is enforced on the segmentation may lead to a reduction in ambiguity and an increase in segmentation performance.

We have already introduced the problem of SAM in a corruption setting in Sec. 5. To analyze this quantitatively, we query SAM with a random foreground-scribbled point as a prompt, due to its failure when using full scribbles. Yet, even using the corruptions proposed in (Hendrycks & Dietterich, 2018) on the convexity dataset yields challenging cases for SAM. Projecting the results onto our implicit convex representation yields a small but systematic improvement as shown in Tab. 2. Interestingly, even without corruptions, using a foreground-scribbled point as a prompt in SAM cannot compete with a joint training of a simple network with semantic features and our implicit convex representation (Tab. 1).

(a) Globe image and user scribbles

(b) SAM result, when prompted with scribbles in (a)

(c) SAM result with one positive click-prompt (green)

(d) SAM result with two positive click-prompts (green)

Figure 10. Out-of-distribution and non-well-defined prompts lead to unintended results in SAM.

Table 2. IoU for SAM (first row) and the convex projection (second row) in the case of additional corruptions (Hendrycks & Dietterich, 2018). A severity value of 5 is used to corrupt the images. Results are averaged across three random seeds.

Model	Clean	Spatter	Contrast	Brightness	Impulse	Shot Noise	Gaussian Noise	Defocus Blur	Glass Blur
SAM	0.7275	0.5627	0.6489	0.6456	0.5330	0.6298	0.6246	0.7333	0.7187
proj_S(SAM)	**0.7407**	**0.5817**	**0.6597**	**0.6516**	**0.5504**	**0.6371**	**0.6357**	**0.7426**	**0.7321**

B. Convexity Prior

We conducted the numerical experiments mainly on the convexity dataset (Gorelick et al., 2014)[5], where we trained the non-convex predictor $\mathcal{N}_\theta$ using inference learning, e.g. having a separate set of parameters ν for every training example also for our $\mathcal{N}_\theta$. For a training example, we can refer to Fig. 9a, which also contains the implicit representation of the prior network $\mathcal{G}_\nu$. The architectures used, including our training scheme, are described below.

B.1. Architectures

We evaluate two different segmentation networks $\mathcal{N}_\theta$ proposed by Dröge & Moeller (2021), 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 (Aksoy et al., 2018), 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.

[5]The convexity dataset is available at https://vision.cs.uwaterloo.ca/data/.

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 $\mathcal{G}_\nu$ (Amos et al., 2017) 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 the 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.2. Training Variants

We divide the training schemes into two sections, the joint training in which we train $\mathcal{N}_\theta$ and $\mathcal{G}_\nu$ simultaneously and the sequential training in which $\mathcal{N}_\theta$ is projected onto $\mathcal{G}_\nu$ by fitting $\mathcal{G}_\nu$ to unaries given by $\mathcal{N}_\theta$.

Joint Training For the training of our networks in a joint fashion, we use a binary cross entropy loss as a data term for $\mathcal{N}_\theta$ and our proposed convexity regularizer (12). As data in the convexity dataset (Gorelick et al., 2014) 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 (Dröge & Moeller, 2021) has shown that an additional regularization concerning 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 for the sum of our segmentation network output (e.g. segmentation u), resulting in a combined cost function (13), with f concatenating all information, i.e., c, ω, ξ. Note: We will omit writing $\sigma(\mathcal{N}_\theta)$ and $\sigma(\mathcal{G}_\nu)$, for σ being the sigmoid function, for all occurrences of $\mathcal{N}_\theta$ and $\mathcal{G}_\nu$ for better readability.

$$\begin{aligned}\mathcal{L}_{\text{joint}} = \underset{\theta,\nu}{\operatorname{argmin}}\ & \text{BCE}(\mathcal{N}_\theta(f_s), s) + \text{BCE}(\mathcal{G}_\nu(\omega_s), s) + \alpha \cdot \text{MSE}(\mathcal{G}_\nu(\omega_r), \mathcal{N}_\theta(f_r)) \\ & + \beta \sum_{i=0}^{P} \left\| \frac{\partial \mathcal{N}_\theta}{\partial c_i}(c_{s,r}, \omega_{s,r}, \xi_{s,r}) \right\| + \gamma \sum_{i=0}^{P} \left\| \frac{\partial \mathcal{N}_\theta}{\partial \omega_i}(c_{s,r}, \omega_{s,r}, \xi_{s,r}) \right\|,\end{aligned} \tag{13}$$

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 3,000 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 (13), is defined with the training for RGB, spatial and semantic features (Aksoy et al., 2018). Following Dröge & Moeller (2021), 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.

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 $\mathcal{L}_{seq}$ (14) and $\mathcal{L}_{cvx}$ (15) as loss functions for the respective networks, similar to the one stated in (13). The training procedure is also the same, except that we are keeping the learning rate constant over the whole training time.

$$\begin{aligned}\mathcal{L}_{\text{seq}} = \underset{\theta}{\operatorname{argmin}}\ & \text{BCE}(\mathcal{N}_\theta(f_s), s) + \beta \frac{1}{p} \sum_{i=0}^{P} \left\| \frac{\partial \mathcal{N}_\theta}{\partial c_i}(c_{s,r}, \omega_{s,r}, \xi_{s,r}) \right\| \\ & + \gamma \sum_{i=0}^{P} \left\| \frac{\partial \mathcal{N}_\theta}{\partial \omega_i}(c_{s,r}, \omega_{s,r}, \xi_{s,r}) \right\|,\end{aligned} \tag{14}$$

$$\mathcal{L}_{cvx} = \underset{\nu}{\operatorname{argmin}} \, \mathrm{BCE}(\mathcal{G}_\nu(\omega_s), s) + \alpha \cdot \mathrm{MSE}(\mathcal{G}_\nu(\omega_r), \mathcal{N}_\theta(f_r)). \tag{15}$$

B.3. Extended Experimental Results

In addition, to the results presented in Tab. 1, we report these again in an extended form. Tab. 3 lists also the standard deviation and the pixel accuracy for the various input types and networks. Furthermore, we have visualized the training results in Fig. 11.

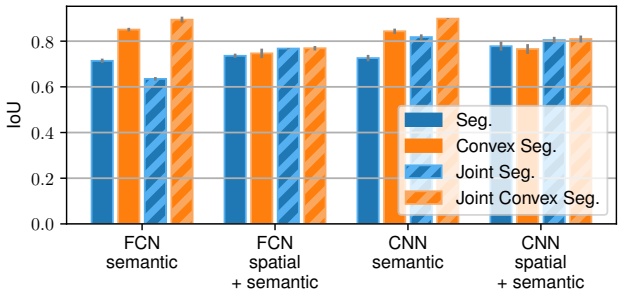


Figure 11. Experimental results w.r.t different input schemes. The convex segmentation outperforms the normal segmentation when using semantic features. In all cases, joint training with the convex segmentation (dashed orange bar) works best. All numerical results of these experiments are also listed in Table 3.

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 Fig. 11 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

B.4. Convexity Comparison Method

In the main part of our manuscript, we did not mention a quantitative comparison method for the convexity prior. For a comparable method in getting convex segmentations that made us their code available (Chen et al., 2021), we retrieved partially non-convex results. The authors also report this behavior in a subsequent manuscript (Chen et al., 2023). Accordingly, we did not include this method in the comparison with our provably convex method. In Fig. 12 we state some non-convex results, generated by the method of Chen et al. (2021) (right) in comparison to our convexity projection method (left). The non-convex parts are indicated by a blue dashed line whereas our method achieves the expected provable convex result.

Figure 12. Comparison of our convex segmentation prior (left) vs. the convex segmentation proposed by (Chen et al., 2021) (right). Within these images, we illustrate paths that are violating the convexity in blue.

C. Path-Connected Prior

As already mentioned in section 3.2, we evaluate the path-connected (PC) prior on a subset of the FBMS-59 dataset, on sequences for which we have received both the multicut trajectories with uncertainties and pretrained UNet models from Kardoost & Keuper (2021) An illustration of the PC prior on one image of the *horses01* sequence can be seen in Fig. 9b.

C.1. Architectures

Since we use the implementation from Kardoost & Keuper (2021) as the segmentation network $\mathcal{N}_\theta$, we focus on describing the details of the diffeomorphism $\mathcal{D}_\phi$ in combination with the convex network $\mathcal{G}_\nu$ resulting in the path-connected prior $\mathcal{P}_{\phi,\nu}$. In addition, we use a learnable and linear transformation $\mathcal{T}_\phi$, which allows us to easily shift the implicit representation. Accordingly, since the linear transformation does not affect (7), the PC prior can be expressed as a composition $\mathcal{P}_{\phi,\nu}(\omega) = \mathcal{G}_\nu \circ \mathcal{D}_\phi \circ \mathcal{T}_\phi$. For $\mathcal{G}_\nu$ we stick to the implementation in B.1 and $\mathcal{T}_\phi$ can be implemented using an invertible 1 x 1 convolution. The input $\omega \in \mathbb{R}^2$ are the normalized spatial coordinates of the image pixels.

As diffeomorphisms are used within the area of normalizing flows, Real NVP (Dinh et al., 2017; Stimper et al., 2023) is a straightforward architecture choice. The multi-layer perceptrons for scale and translation mapping are configured with a width of 32, a hidden layer, leaky-relu activation, tanh as an output function, and zero initialization. Other settings were kept to the library defaults (Stimper et al., 2023). After every flow block, we use the data-dependent ActNorm layer, as described in Glow (Kingma & Dhariwal, 2018), to normalize the block output. We repeat this combination of Real NVP block and ActNorm 12 times, alternately transforming our input dimensions ω, respectively the x and y pixel coordinates.

C.2. Training Variants

Similar to B.2, we also specify the two training variants below. However, we do not train the segmentation UNet $\mathcal{N}_\theta$ for the sequential fitting ourselves but use the pretrained models provided. But, unlike to B.1, Kardoost & Keuper (2021) used an RGB image in combination with an edge mask as input $f \in \mathbb{R}^4$.

Sequential Training The training of the PC prior for the first image within a sequence can be divided into three steps. Firstly, we need to ensure that $\mathcal{D}_\phi$ approximately represents the identity transformation.

Secondly, we optimize $\mathcal{P}_{\phi,\nu}$ only for ν using the Adam optimizer with a learning rate of $1 \cdot 10^{-3}$ and 1,000 optimization

steps on $\mathcal{L}_{\text{pc\_seqc}}$. This results in a $\mathcal{P}_{\phi,\nu}$ with an approximately convex fit on the unaries.

Thirdly, we optimize $\mathcal{P}_{\phi,\nu}$ for ν and ϕ together, but now with an Adamax optimizer, a learning rate of $1 \cdot 10^{-3}$ and a weight decay of $1 \cdot 10^{-5}$ on $\mathcal{D}_\phi$ for 4,000 optimization steps on $\mathcal{L}_{\text{pc\_seqp}}$[6]. This results in a good fit of the prior to the first frame of a sequence Fig. 19. We can now simplify the creation of priors for the remaining images in the sequence.

The difference between two consecutive images is usually very small and predominantly a rigid transformation of the object with minor small, non-rigid transformations. Therefore, it is useful to initialize with the previously generated weights (ϕ, ν) for every new image. We then train only 400 optimization steps on $\mathcal{L}_{\text{pc\_seqp}}$ for each subsequent image, which is sufficient to learn the rigid motion and encode also slight non-rigid transformations, and generate a good fit over the entire sequence. Likewise, this approach allows us to control how many non-rigid transformations we tolerate within each image, as the linear transformation $\mathcal{T}_\phi$ of the coordinate system (rigid transformation) is much faster to learn than a non-rigid transformation that has to be represented by $\mathcal{D}_\phi$.

$$\begin{aligned} \mathcal{L}_{\text{pc\_seqc}} &= \underset{\nu}{\operatorname{argmin}}\ \text{MSE}(\mathcal{P}_{\phi,\nu}(\omega), \mathcal{N}_\theta(f)) \\ \mathcal{L}_{\text{pc\_seqp}} &= \underset{\phi,\,\nu}{\operatorname{argmin}}\ \text{MSE}(\mathcal{P}_{\phi,\nu}(\omega), \mathcal{N}_\theta(f)) \end{aligned} \tag{16}$$

Note: We also omitted and will further omit writing $\sigma(\mathcal{N}_\theta)$ and $\sigma(\mathcal{P}_{\phi,\nu})$, for σ being the sigmoid function, for all occurrences of $\mathcal{N}_\theta$ and $\mathcal{P}_{\phi,\nu}$ for better readability.

Joint Training We assume for the joint training that our models $\mathcal{P}_{\phi,\nu}$ and $\mathcal{N}_\theta$ have already performed the sequential training or have pretrained weights. Therefore, we assume that a representation of the object already exists in both networks and that we only need to fine-tune the representations. To address this, we define a loss based on a weighted combination of our path-connected projection, and the loss function in Kardoost & Keuper (2021) which we denote now as $\mathcal{L}_{\text{multicut}}$[7] (17).

$$\begin{aligned} \mathcal{L}_{\text{pc\_joint}} &= \underset{\phi,\,\nu,\,\theta}{\operatorname{argmin}}\ \mathcal{L}_{\text{multicut}}(\mathcal{N}_\theta(f_s), s) + \alpha \cdot \text{MSE}(\mathcal{P}_{\phi,\nu}(\omega), \mathcal{N}_\theta(f)) \\ \alpha &= \begin{cases} \frac{\mathcal{L}_{\text{multicut}}(\dots)}{\text{MSE}(\dots)} & \text{if } (\text{MSE}(\dots) > \mathcal{L}_{\text{multicut}}(\dots)) \\ 1 & \text{else} \end{cases} \end{aligned} \tag{17}$$

Whereby the α parameter is used to implement soft clipping, e.g. dynamically weighting the MSE projection loss to a maximum of the actual weak label loss, so that the latter will be the dominant factor. Using (17) we finetuned the UNet and our prior using the Adam optimizer for 15 epochs with a learning rate of $1 \cdot 10^{-4}$.

It should be noted that for training a network jointly over several images, as usual in normal deep learning pipelines and frameworks, the prior weights must be adjusted in each case. Before each forward pass of the model, the corresponding implicit representation must therefore be loaded and saved again after an optimization step so that the next image can be evaluated. This creates an overhead whose effects must be evaluated in future work. We also limit ourselves to a batch size of 1, although a representation of several implicit representations is technically possible.

C.3. Extended Experimental Results

C.3.1. Qualitative and Quantitative Examples

As we have seen in Fig. 6 the ground truth annotations of a partly-occluded object can result in a lower IoU using our proposed PC prior on image segmentation, which by definition includes the occlusion into the segmentation. Therefore, it is important to evaluate and analyze the resulting segmentations individually.

[6] The use of the Adamax optimizer with infinity norm and weight decay proved to be suitably robust to us and still were able to fit complex shapes reasonably well. Without the infinity norm and weight decay, the training was much less stable (Fig. 7).

[7] $\mathcal{L}_{\text{multicut}}$ can be described as a specially weighted BCE loss, which only evaluates on pixels that are covered by a multicut trajectory $s \in [0, 1]$ denoting foreground or background. These are acting as weak labels in this setting, similar to the scribbles in B.2.

Table 4. Results on all 18 sequences from the FBMS-59 dataset with and without our proposed PC prior in both, a sequential and joint training approach. We report the mean over 3 training runs.

	Sequential				Joint			
	IoU ↑		Acc. ↑		IoU ↑		Acc. ↑	
Sequence	UNet	Prior	UNet	Prior	UNet	Prior	UNet	Prior
bear01	**0.840**	0.834	0.988	0.988	0.838	0.820	0.988	0.987
bear02	0.803	0.802	0.968	0.968	0.758	**0.804**	0.951	0.968
cars2	0.559	0.572	0.964	0.967	0.606	**0.617**	0.970	0.973
cars3	0.779	0.880	0.986	0.992	0.885	**0.887**	0.992	0.993
cars6	0.718	0.766	0.990	0.992	0.791	**0.808**	0.993	0.994
cars7	0.691	0.693	0.988	0.988	**0.775**	0.751	0.992	0.991
cars8	0.771	0.770	0.977	0.977	**0.793**	0.784	0.980	0.979
cats04	0.714	0.763	0.991	0.992	**0.769**	0.760	0.995	0.993
cats05	**0.607**	0.571	0.971	0.969	0.406	0.534	0.964	0.969
horses01	0.872	0.849	0.990	0.989	**0.856**	0.845	0.988	0.988
horses03	0.737	0.744	0.941	0.941	0.728	**0.761**	0.937	0.947
marple1	0.709	0.710	0.873	0.872	0.705	**0.733**	0.868	0.887
marple10	0.014	0.020	0.261	0.253	**0.023**	0.020	0.452	0.222
marple11	0.732	**0.800**	0.987	0.990	0.742	0.774	0.989	0.990
marple5	0.615	0.615	0.945	0.945	**0.688**	0.622	0.962	0.946
meerkats01	0.616	0.617	0.980	0.980	0.546	**0.656**	0.978	0.983
people04	0.843	0.843	0.992	0.992	0.834	**0.845**	0.993	0.992
rabbits01	**0.816**	0.806	0.991	0.990	**0.816**	0.791	0.993	0.988

On the other hand, it is possible that our PC prior indeed segments the non-occluded parts but also generates a thin line between these segments due to the path-connectedness assumption. Figure 13 shows a frame of the *cats05* sequence in which this behavior is visualized. Further, we state in Fig. 14 that the diffeomorphisms can become unconnected in discrete $\mathbb{R}^2$.

Figure 13. Ground truth annotations, baseline network prediction, and implicitly connected segmentation of one frame of the *cats05* sequence, after a sequential fit (left) and joint training (right). In this case, the ground truth does not indicate a connected object - which contradicts our path-connected constraint. Accordingly, the prior with 72.3% IoU is inferior to the segmentation network (73.8%).

We further exemplify in Fig. 15 the importance of a reasonable prior for a good segmentation. The left plot shows the sequential fit, where the car in the background is also segmented, though having a background label. The right plot shows the improvement in segmenting the as foreground labeled car, i.e., the white car, when we jointly train the UNet and our PC prior. Also note here, that we improve the segmentation noticeably ($\approx 11\%$ over baseline), especially in the car sequences (see Tab. 4 *cars* sequences). One reason is that the segmentation network produces a non-path-connected segmentation in some cases of several objects (here cars), although only one car needs to be segmented. This is illustrated in Fig. 19.

Figure 14. As our PC prior is only path-connected in continuous $\mathbb{R}^2$ this may not hold for discrete $\mathbb{R}^2$, visible as the orange prior outline is unconnected.

Figure 15. Example of positive impact of the PC prior on the *cars3* sequence, showing the sequential fit (left) and the joint training result (right).

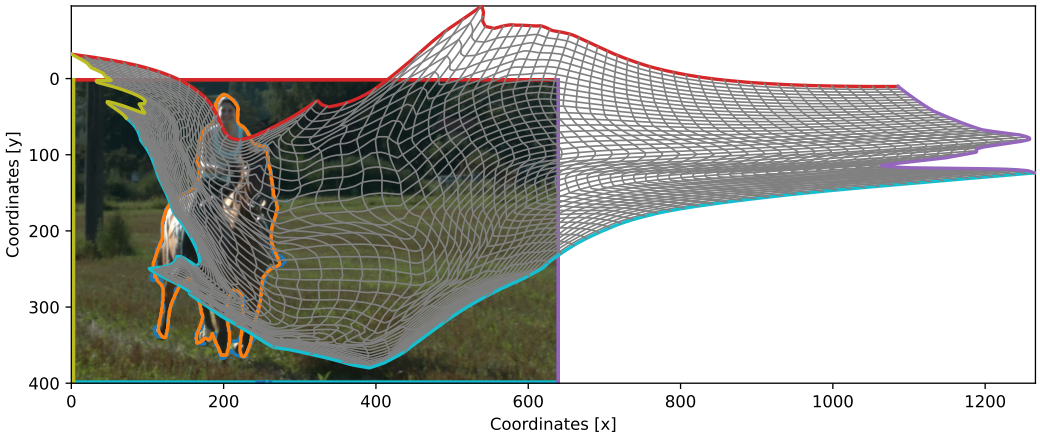
C.3.2. Visualizing Diffeomorphism Deformations

Since the composition of $\mathcal{D}_\phi$, representing a deformation of the coordinate system, and the convex network $\mathcal{G}_\nu$, presents a path-connected region, we will also visualize what this transformation looks like. In Fig. 16 we present several visualizations of a sequentially fitted frame of the *horses01* sequence.

In (a) we evaluated the convex prior on a regular grid, without shifting or deforming it through the diffeomorphism. Accordingly, this is what the $\mathcal{G}_\nu$ learns to represent; a convex shape of the horse. We furthermore plotted grey gridlines every 10 pixels and colored these on the image edges, to visualize the relation of the image dimensions and the underlying regular coordinate system. Using these gridlines, we can further illustrate how the $\mathcal{D}_\phi$ in (b) stretches the regular coordinate system and deforms it both globally and locally, after joint training. To better illustrate which regions in the convex output with regular grid belong to the output of $\mathcal{P}_{\phi,\nu}$, we have colored the foreground according to semantic parts (c) and displayed these regions by in-painting in (d)[8].

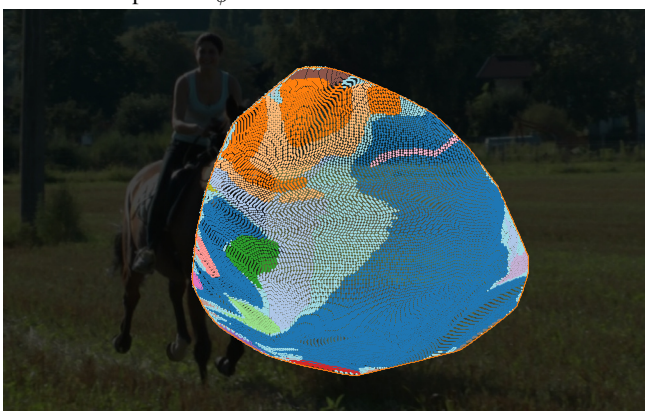
[8]Since the diffeomorphism can scale the output, the displayed mapping may be scattered. Although we could overcome this since our model is resolution invariant, we preserve the deformations in this visualization since they also represent the applied deformations.

(a) Output of $\mathcal{G}_\nu$ on a non-deformed grid



(b) Grid deformation of the diffeomorphism $\mathcal{D}_\phi$

(c) Semantically colorized foreground



(d) Semantically colorized foreground inpainted in the non-deformed output of $\mathcal{G}_\nu$

Figure 16. The diffeomorphism deforms the regular input grid (a) intensely (b) to fit the given explicit segmentation. For a better illustration of the deformation, we generated semantic masks of the predicted foreground in (c) and inpainted their corresponding location within the output of $\mathcal{G}_\nu$ in (d).

C.3.3. Spatio-Temporal Path-Connected Prior

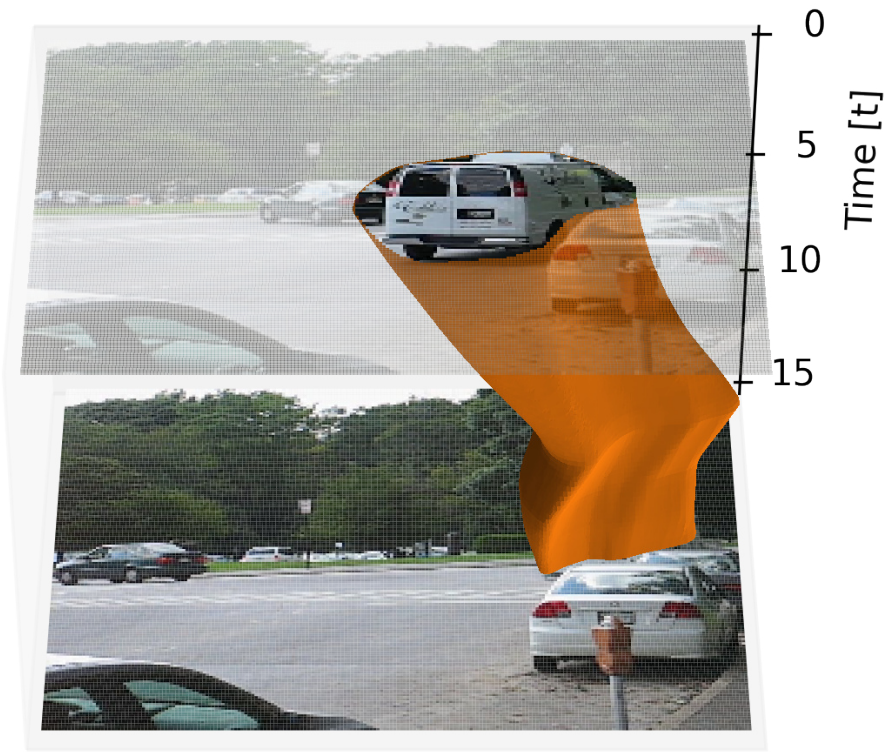


Figure 17. Spatio-temporal PC prior along a sequence of images, when super-resolving the prior in time. t on the z-axis is annotated as the frame index of existing images within the sequence.

We illustrated the ability to use the PC prior over several frames in Sec. 2.7, Fig. 3b. The prior can also be applied in the context of the FBMS-59 motion segmentation dataset, to represent the spatio-temporal connectedness of an object like a car from the sequence *cars3*. We use the sequential approach as described in C.2, and add the (normalized) time t to our input $\omega \in \mathbb{R}^3$. Correspondingly, the parameter ν usage changes from one set of parameters per image to one set of parameters for an image sequence.

We also change the combination of masks for the scale and translation mapping of each Real NVP block (within our diffeomorphism network $\mathcal{D}_\phi$) in order to transform each combination of the three dimensions the same number of times. This will enable the PC prior to learn rigid and non-rigid motion, in space and time. Further, we can super-resolve time frames with the prior, by running inference on intermediate (or past and future) time stamps for unseen t.

We visualized this prior-based interpolation in Fig. 17. The prior was trained on the UNet unaries for the 19 images within the sequence, the intermediate steps are super-resolved. As time passes, the rigid motion of the car to the right of the image is clearly visible, ending at frame 11. Afterwards, the camera tilts significantly to the right, following the car, which is visualized as a slight movement to the left. Also mentionable is the non-rigid transformation caused by the increasing occlusion of the truck by another car.

C.3.4. Benefits of a Spatio-Temporal Path-Connected Prior

The question may arise: What kind of benefit lies in using a PC prior which is also spatio-temporal?
One assumption for a spatio-temporal prior could be, that it is more robust to noisy frames on some training examples. To this end, we experimented on the *cars3* sequence to investigate the extent to which such a prior can be influenced by replacing some unaries with complete noise. Accordingly, we created 6 scenarios in which we replaced 0 to 60 % of the unaries, i.e., synthetic outputs of $\mathcal{N}_\theta$, with random noise and trained a prior as described above.

The results are stated in Tab. 5. If the amount of noisy frames is rather small, up to 20 % corresponding to 4 frames within

the 19 frames sequence, the prior can still keep the shape on noisy frames well, resulting in a decrease of just 3 % IoU. With a larger amount of noise, the IoU drops significantly.

Table 5. Spatio-temporal PC prior trained on scenarios with different amounts of labels replaced by pure noise. We state the IoU against the actual non-noisy unaries to indicate the fitting as a mean over three runs.

Label Noise	IoU ↑
0 %	0.826
10 %	0.799
20 %	0.796
30 %	0.527
40 %	0.514
50 %	0.186
60 %	0.189

In Fig. 18 we visualize the 20 % noisy setting of the actual frames 10 to 15 and also super-resolve the frames in between. The noisy frames are in this case 5, 12, 13, and 16. Accordingly, 12 and 13 are displayed in the figure. Yet, we can see that the prior is still able to perform a reasonable fit to the car, despite being fitted to the noisy unaries, and also performing a reasonable super-resolution in time in between.

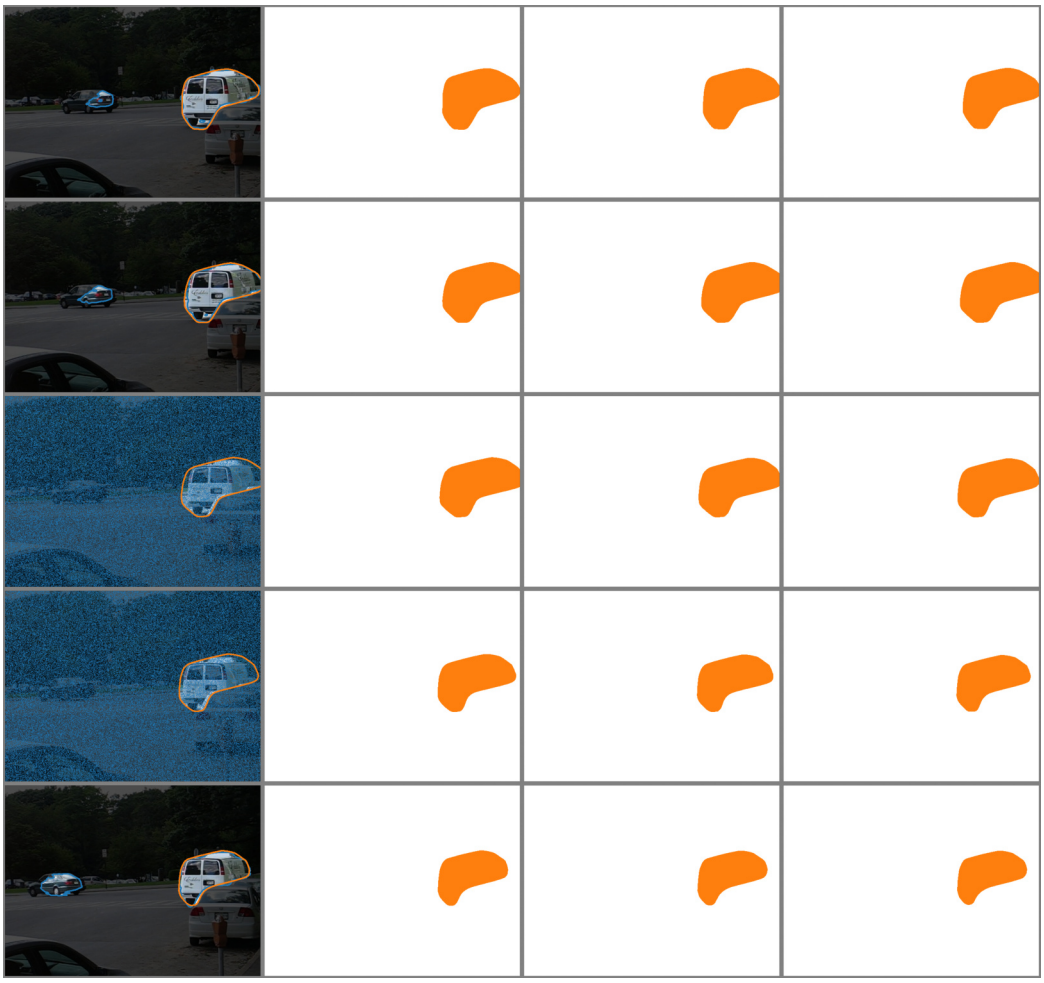


Figure 18. Spatio-temporal PC prior, when trained on unaries which partially (20 %) replaced by pure noise. The frames shown are index 10 to 15 of the *cars3* sequence, with the white images showing a temporal super-resolution based on the prior between those frames.

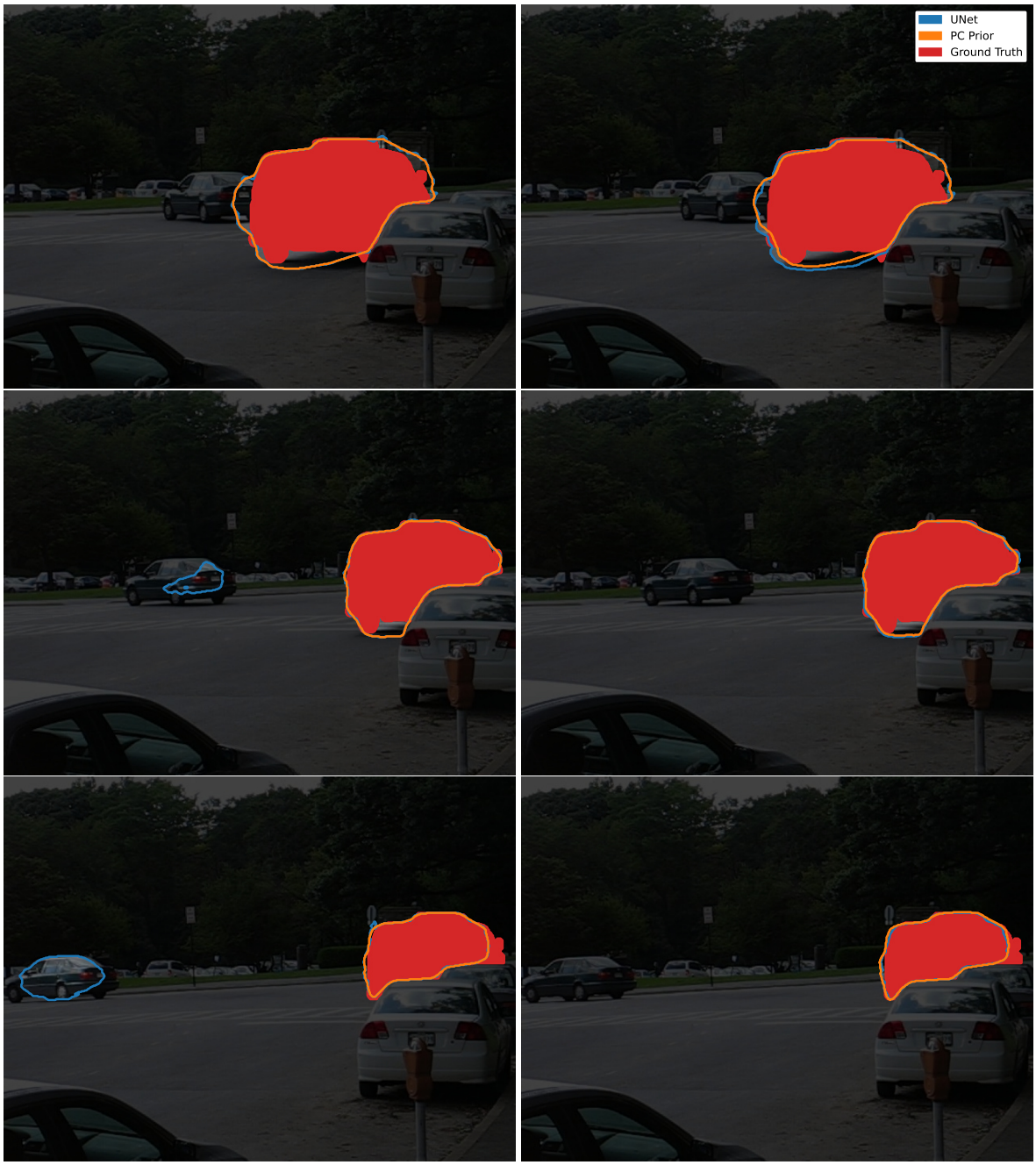


Figure 19. All ground truth frames of the cars3 sequence with UNet segmentation and PC prior.