Conditional Diffusion for Interactive Segmentation

Xi Chen, Zhiyan Zhao, Feiwu Yu, Yilei Zhang, Manni Duan*
Alibaba Group
{xizhi.cx, zhiyan.zzy, feiwu.yfw, yilei.z}@alibaba-inc.com, manyi@taobao.com

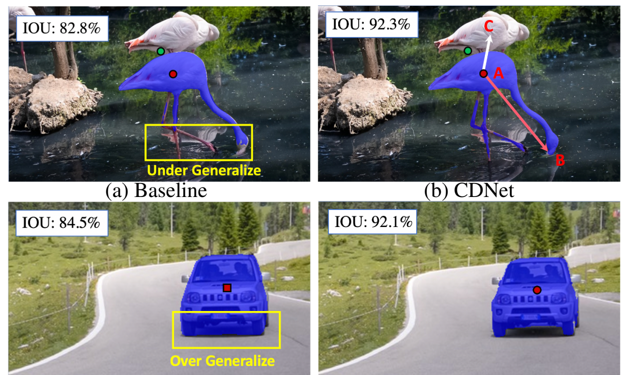
Abstract

In click-based interactive segmentation, the mask extraction process is dictated by positive/negative user clicks; however, most existing methods do not fully exploit the user cues, requiring excessive numbers of clicks for satisfactory results. We propose Conditional Diffusion Network (CDNet), which propagates labeled representations from clicks to conditioned destinations with two levels of affinities: Feature Diffusion Module (FDM) spreads features from clicks to potential target regions with global similarity; Pixel Diffusion Module (PDM) diffuses the predicted logits of clicks within locally connected regions. Thus, the information inferred by user clicks could be generalized to proper destinations. In addition, we put forward Diversified Training (DT), which reduces the optimization ambiguity caused by click simulation. With FDM,PDM and DT, CDNet could better understand user's intentions and make better predictions with limited interactions. CDNet achieves state-of-the-art performance on several benchmarks.

1. Introduction

Interactive segmentation has been a topic of research for a long while; various forms of interactions have been explored. Human could provide bounding boxes [26, 30, 15], scribbles [16, 6, 1], or clicks [27, 20, 32, 11] to express the segmentation intentions, which guide the algorithm for the mask extraction process. The segmentation target could be anything that users want, which requires interactive segmentation to be a flexible tool and makes it a challenging vision task. In this work, we address click-based interactive segmentation, and we aim to improve upon existing works by better understanding user's intentions.

For click-based interactive segmentation, users places positive/negative clicks (red/ green points in Fig. 1) to indicate foreground/ background regions. In general, a user's click contains two layers of information: the first layer is spatial – the location of the foreground/background could be indicated by the distribution of clicks; the second layer is visual – the label of regions around the clicks could be inferred by visual similarity. In a standard pipeline, the first step is encoding clicks into distance maps [32, 27, 11, 31], Gaussian maps [20, 23, 21, 14], or super-pixels [22]; next, encoded maps is concatenated with the original image and fed into a segmentation network to make predictions. This kind of method exploits good use of spatial information but ignores the visual hint. Consequently, we see two problems: 1) like the example in Fig. 1 (a), the label of a click often fails to be generalized to target regions further away from clicks, even if parts of the target region has almost the same appearance as the click. 2) the labels could spill over to the wrong regions, even if the target has clear boundaries, as demonstrated in Fig. 1 (c).



(a) Baseline (b) CDNet
(c) Baseline (d) CDNet

Figure 1. Comparison of baseline method and CDNet. Positive /negative clicks are marked in red and green. Diffusion flows are visualized in colored arrows. As the red arrow denotes, CDNet propagates information from clicks to target destinations, while suppressing the invalid flow denoted by the white arrow.

To address the aforementioned issues, we explored to model the affinity between different locations and diffuse the representations from clicks to unlabeled regions. A straightforward solution is modeling the affinity simply based on feature similarity. However, it would lead to over-generalization: in Fig. 1 (b), the information should not propagate from A to C (under the assumption that the target is the bottom flamingo), even though they are visually simi-

*Corresponding Author

lar. To deal with over-generalization, constraining the diffusion within a instance/ semantic- level range around clicks is an option that can be thought of directly. Again, since the target was flexibly defined and could be either part, instance, or stuff, a fixed constrain would not be a cure-all. In general, it is a dilemma that we expect to enlarge the diffusion range and avoid over-generalization simultaneously.

Facing this challenge, we conduct an in-depth study of information diffusion and propose Conditional Diffusion Network (CDNet), which diffuses information from clicks and dynamically constrains the diffusion destination. Specifically, two components are designed:

- **Feature Diffusion Module (FDM)** generates a conditional affinity map. It firstly considers feature similarity to diffuse the representations from clicks to all unlabeled regions. We then constrain the diffusion destination by a primitive prediction of foreground/ background per the distribution of clicks and the content of the image. FDM propagates features in full-image with a global perspective.

- **Pixel Diffusion Module (PDM)** constructs a conditional diffusion source on the predicted logits, and leverages color similarity to diffuse the labels of clicks. It constrains the diffusion in local regions with uniform color to avoid over-generalization. Meanwhile, the diffusion is carried out iteratively to enlarge the range of destination. PDM mainly refines the details.

FDM and PDM work in synergy to extract representations with both high-level and low-level consistency in a coarse-to-fine manner. To better train CDNet, we develop a practical training regime called Diversified Training (DT), which eases the optimization ambiguity caused by click simulation. Combing FDM/PDM/DT together, CDNet better exploits the information contained in clicks by diffusing the representations of clicks to correct destinations and makes better predictions with fewer clicks.

Large amounts of experiments have been conducted on GrabCut [26], Berkeley [24], SBD [8], and DAVIS [25] datasets. Results prove the effectiveness of our method and show that our CDNet achieves state-of-the-art performance. Our contributions could be summarized as follows: 1) We formulate click-based interactive segmentation as a process of conditional diffusion and propose CDNet, which predicts better segmentation results with fewer clicks. 2) We design FDM and PDM, which propagate the labeled information of clicks to enhance unlabeled regions on different levels. 3) To better train CDNet, we develop Diversified Training, which reduces the optimization ambiguity caused by click simulation.

2. Related Works

Classical Methods. Before the era of deep learning, researchers take interactive segmentation as an optimization problem. GrabCut [26] uses the Gaussian mixture model to solve the problem of max-flow in color space. Geodesics [7] calculates geodesic between clicked points and other pixels to predict segments with minimum energy cost. [6] applied random walk algorithm to predict the labels of unseeded pixels. [12] proposes a high order method with the constrain of label consistency. These classical methods model the relationship between pixels according to low-level similarity, which enables them to predict segmentation results with local consistency. However, lacking high-level semantic information limits the performance of classical methods.

Deep Learning Methods. The first deep learning-based method [32] embeds clicked points into distance maps and uses a fully convolutional network to predict the mask of foreground and background. RIS-Net [18] adds a local branch to refine the predicted result around human clicks. [22] uses super-pixels to embed clicked points to provide guidance with local consistency. [17] predicts multiple potential results and train another network to choose from them. FCANet [20] underlines the importance of the first click and proposes first click attention to get better results. BRS [11] uses backward propagation to finetune the guidance map in an online manner. f-BRS [27] refines the intermediate feature to get more precise masks with faster speed compared with BRS. Most of these learning-based methods only use clicks to generate the guidance map to indicate the rough location of the target object. BRS and f-BRS use the given labels of clicked points to fine-tune the network, but online learning brings extra computation during inference and makes them hard to deploy. Compared with classical methods, deep learning models get better performance. However, they are not utilizing the user inputs to the full potential.

3. Method

3.1. Pipeline Overview

The pipeline of Conditional Diffusion Network (CDNet) is shown in Fig.2. The blocks in blue demonstrate a commonly used baseline, on which we add two diffusion modules in red. First, we embed positive and negative clicks as two Gaussian maps and concatenate them with the original RGB image to get the 5-channel input. Second, the input is fed into a segmentation network to extract high-level features. In this work we use DeeplabV3+ [3] with ResNet-50 [9] backbone. Then, the stride-8 high-level feature and the Gaussian maps are sent into Feature Diffusion Module (FDM). In this block, the labeled features around clicks could be propagated to prospective unlabeled regions. Next, following DeeplabV3+, we make feature fusion with

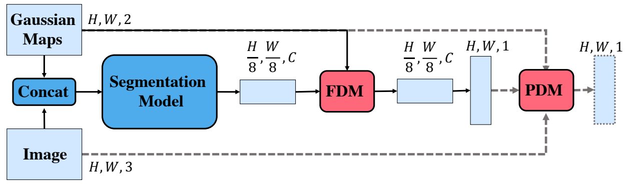


Figure 2. An overview of our Conditional Diffusion Network. FDM denotes Feature Diffusion Module. PDM denotes Pixel Diffusion Module. The dotted lines mean PDM only exists during inference.

low-level features and upsample the predicted logits to the size of the original image. Afterward, Pixel Diffusion Module (PDM) takes the original image, Gaussian Maps, and predicted logits to propagates the logits from clicks to their neighbors iteratively, which refines the prediction with low-level consistency. PDM only exists during inference. Training supervision is applied on the logits before PDM.

Both FDM and PDM are easily extensible and could be simple plugins for different segmentation models. With FDM and PDM, the effect of clicks could be amplified as their labeled representations are propagated to wider ranges, which enables the model to make better predictions.

3.2. Feature Diffusion Module

Feature Diffusion Module (FDM) propagates the labeled high-level features from clicks to conditional destinations. It enables unlabeled regions to be better represented via matching labeled templates and helps the labels of clicks to be generalized to both wide and accurate destinations.

The implementation of FDM is inspired by self-attention series [5, 10, 29, 33]. We first revisit the formulation of self-attention. Then we elaborate on the architecture of FDM.

3.2.1 Revisiting Self-Attention

Non-local Network [29] proposes a standard formula for self-attention, its variants [5, 10, 29, 33] are widely applied in the task of semantic segmentation and proven to be effective. The formulation could be summarized as Eq. (1) (2), where $\mathbf{A}$ denotes the affinity matrix which measures the dependency between features of each two positions. $\mathbf{x}$ stands for the input feature. $\mathbf{g}, \theta, \phi$ are transformation functions which are implemented with 1×1 Convs.

$$\mathbf{y}^{HW\times C} = \mathbf{Softmax}(A^{HW\times HW}) \times \mathbf{g}(\mathbf{x})^{HW\times C} \quad (1)$$

$$\mathbf{A} = \theta(\mathbf{x})^{HW\times C} \times \phi(\mathbf{x})^{C\times HW} \quad (2)$$

With self-attention, the information in $\mathbf{x}$ could be propagated cross long-distance between every two positions, which helps to build more unified feature representations with the global context.

3.2.2 FDM Overview

Regarding Eq. (1) as a process of information diffusion, the affinity matrix in Eq. (2) assigns equal chances for each location to act as the diffusion source; the diffusion destination is decided by only considering semantic similarity. It works for semantic segmentation, but it is not an optimal solution for interactive segmentation for two reasons: 1) as the labels of positive/ negative clicks are given, features around clicks are more informative and should be prioritized for diffusion; 2) since the foreground/ background are dynamically defined by clicks, we could not constrain the diffusion destination statistically using instance or semantic similarity.

To address the aforementioned problem, FDM introduces two additional features via dynamically re-weighting the affinity matrix: it highlights the diffusion flows from clicks, and constrains the diffusion destinations in the meantime. Formulated as Eq. (3), FDM uses two conditional affinity matrix $\mathbf{CA}_{F/B}$ to model the diffusion flow for foreground/ background information, and add the diffusion results together. The pipeline of FDM is demonstrated in Fig. 3, we first calculate the raw affinity matrix $\mathbf{A}$ following Eq. (2). Then, we generate conditional affinity matrices by re-weighting. The details of the conditional affinity would be introduced in the next paragraph.

$$\mathbf{y} = \mathbf{CA}_F \times \mathbf{g}(\mathbf{x}) + \mathbf{CA}_B \times \mathbf{g}(\mathbf{x}) \quad (3)$$

3.2.3 Conditional Affinity

We re-weight the raw affinity map by setting source constraint and destination constraint. Source constraint highlights the diffusion flows starting from clicks; Destination constraint defines a rough range for the diffusion, which prevents the label of clicks to be over-generalized. With these constraints, in Fig.1 (b), features from A could be diffused to B, but would not be propagated to C.

Concretely, we generate source constrain maps $\mathbf{S}_{F/B}$ by placing Gaussian kernels on foreground/ background clicks, with the amplitude and the standard derivation both set as 1. Besides, we calculate destination constrain maps $\mathbf{D}_{F/B}$ through adding an auxiliary head on the input feature of FDM, which makes a primitive prediction for the foreground/ background regions. $\mathbf{D}_{F/B}$ is not expected to be accurate, it is used to control the probability for each position to collect information from the foreground/ background source. From another angle, FDM could be considered as a further refinement based on $\mathbf{D}_{F/B}$.

Since $\mathbf{S}_{F/B}$, $\mathbf{D}_{F/B}$ are normalized to $[0, 1]$, we directly re-weight the raw affinity matrix $\mathbf{A}$ according to Eq. (4). As denoted in the right block of Fig. 2, we first reshape $\mathbf{A}$ to shape $\mathbb{R}^{HW\times H\times W}$ and do element-wise multiplication with $\mathbf{S} \in \mathbb{R}^{1\times W\times H}$ to highlight the flows starting from

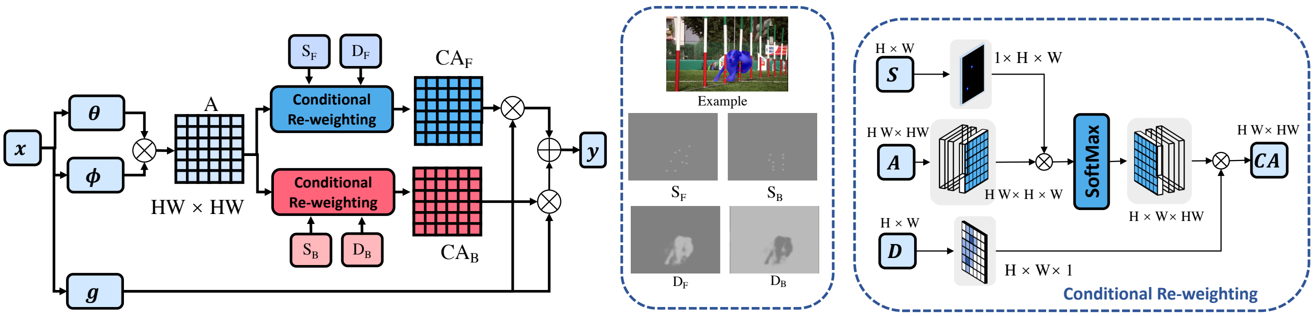


Figure 3. Demonstration of Feature Diffusion Module. We first measure the dependency between features and generate a raw affinity matrix $\mathbf{A}$; Then, we apply source constrain and destination constrain in Conditional Re-weighting Block to get a Conditional Affinity Matrix $\mathbf{CA}$. Guided by the Conditional Affinity Map, features are propagated from source (clicked points) to target destinations to get the enhanced representation. The concrete structure of Conditional Re-weighting Block is demonstrated in the right part.

clicks. Then, softmax function normalizes the summation of source features that could be aggregated for each target. Afterward, the modified affinity map is reshaped to $\mathbb{R}^{H\times W\times HW}$ and multiplies the $\mathbf{D} \in \mathbb{R}^{H\times W\times 1}$ to suppress the flows across foreground/background splits. Thus, we get two conditional affinity matrices $\mathbf{CA}_{F/B}$ which control the information flow from foreground/ background clicks to corresponding target regions.

$$\mathbf{CA}_{F/B} = \text{Softmax}(\mathbf{S}_{F/B} \odot \mathbf{A}) \odot \mathbf{D}_{F/B} \tag{4}$$

3.2.4 Function Analysis

FDM propagates features from positive/ negative clicks to corresponding destinations. It could also be analyzed from the perspective of information gathering. For features predicted as foreground/ background with high confidence, they only gather information from foreground/ background clicks, which assists to make more unified representations for the target. For features with uncertain predictions, they have equal chances to gather information from the foreground and background sources. Thus, it could make a more reliable prediction by matching the foreground and background templates according to feature similarity.

From the perspective of optimization, FDM enforces each unlabeled feature to get closer to features of clicks with the same label, while enlarging the distance with clicks with opposite labels, which contributes to more unified and more discriminative representations.

3.3. Pixel Diffusion Module

Pixel Diffusion Module (PDM) is designed to complement FDM and focus on the details that could not be refined on high-level features. It follows the formula of FDM and propagates information from clicks to unlabeled regions according to affinity. Requiring the representations with rich details, PDM is performed on the full-resolution predicted logits.

3.3.1 PDM Overview

Limited by computing resources, the formulation of FDM could not be directly applied to the full-resolution logits. Therefore, we constrain the pixel diffusion in local regions and implement the diffusion iteratively. Following the basic form of FDM, we formulate PDM as Eq. (5), where $\mathbf{A}_{ij}$ represents the affinity between logit i and its neighbor j; $\mathbf{y}^0$ denotes the pixel diffusion source. Information is propagated from each logit i to its neighboring regions N_i iteratively.

$$\mathbf{y}_i^{t+1} = \sum_{j\in N_i} \mathbf{Softmax}(\mathbf{A}_{ij}) \cdot \mathbf{y}_j^t \tag{5}$$

Given the pipeline of PDM, there are still two key areas for consideration: 1) How to highlight the diffusion flows starting from clicks. 2) How to enlarge the diffusion destination while avoiding over-generalization.

First, we highlight the flows from clicks by constructing a conditional diffusion source, in which the information concentration around clicks is augmented. Thus, the labels of clicks get higher priorities to be propagated out; Second, we calculate the affinity using color similarity, and constrain each iteration of diffusion within a small range. In this way, the diffusion would be truncated when it meets boundaries or a sharp color change. At the same time, we conduct the pixel diffusion iteratively, thus the enhanced logits could diffuse further step by step in regions with uniform colors.

The Pipeline of PDM is demonstrated in Fig. 4. Given the input image, the predicted logits, and the clicks, we construct the conditional diffusion source and propagate the information of the source in the local neighborhood iteratively. Finally, we add the original logits as a residual on the diffusion result and set the threshold as 0 to get the binary prediction mask. Noticing that the whole pipeline of PDM could be implemented with Conv layers. PDM runs on GPUs with high efficiency.

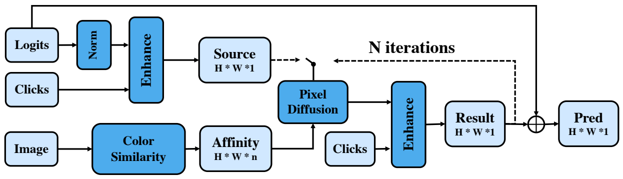


Figure 4. Pipeline of Pixel Diffusion Module (PDM). PDM takes the original image, Gaussian maps of clicks, and the predicted logits as input. It propagates information from clicks to neighbors iteratively to get the refined prediction map.

3.3.2 Conditional Diffusion Source

We construct the conditional diffusion source by augmenting the information concentration around clicks. Considering the network is trained with Sigmoid function and Cross-entropy loss, the absolute value of logits could be large. To manipulate the logits, we first normalize the values of logits into a controllable range. Concretely, we normalize the logits into [-1, 1] according to Eq. (6). Then, we enhance the normalized logits $\mathbf{y}_{norm}$ via placing Gaussian kernels at clicked positions as in Eq. (7), with amplitude and standard derivative set to 1 and 10.

$$\mathbf{y}_{norm} = \mathbf{Tanh}(\mathbf{y}_{raw}) \times 2 \tag{6}$$

$$\mathbf{y}^0 = \mathbf{Enhance}(\mathbf{y}_{norm}) = \mathbf{y}_{norm} + \mathbf{G}_F - \mathbf{G}_B \tag{7}$$

3.3.3 Dynamic Diffusion Range

We leverage color similarity to measure the affinity, which is a robust low-level feature used in some traditional methods [26, 13]. The affinity between pixel i and pixel j as in Eq.(8). σ_i denotes the standard derivation.

$$\mathbf{A}_{ij} = -|\mathcal{I}_i - \mathcal{I}_j| \; / \; \sigma_i^2 \tag{8}$$

Each diffusion iteration is constrained within the local range of n neighbors; the affinity is also calculated locally. In this work, we use four 3×3 convolution filters with dilation $\{1,2,4,8\}$ to sample the neighbors, so $n = 4 \times 8 = 32$. Then, we enlarge the diffusion range by applying the diffusion iteratively. Thus, information flows could go further in regions with uniform color, and would be truncated when encountering edges.

3.3.4 Function Analysis

For regions around clicks, PDM propagates logits from clicks to visually similar neighbors, which guarantees the correct prediction around clicks could be generalized in local regions. For regions far from clicks, PDM also makes refinement with local-consistency, it enforces adjacent pixels with similar colors to predict similar labels.

3.4. Diversified Training

Ambiguity is a common problem for interactive segmentation; divergence between segmentation result and user's true intention frequently happens. For example, in Fig. 5, when only one click on the leg of the rider was placed as foreground, there could be many possible and reasonable targets: the leg, the rider, or the entirety of rider/motorcycle.

Some works [17, 19] focus on the inference procedure to tackle the ambiguity. They propose to predict multiple masks and require the user or a selection network to pick one of them. However, our analysis concludes that if the model is well-trained, the ambiguity can be naturally reduced during inference when more clicks are sequentially placed; the real challenge lies in training, where the ambiguity is difficult to reduce even with bigger numbers of clicks. Many previous works [27, 11, 32, 22] simply simulate interactions during training by randomly sampling positive/ negative clicks inside/ outside the given ground truth mask. There is no guarantee that clicks could clarify the outline of the given ground truth. Consequently, it is hard to train the model well, as the optimization target varies.

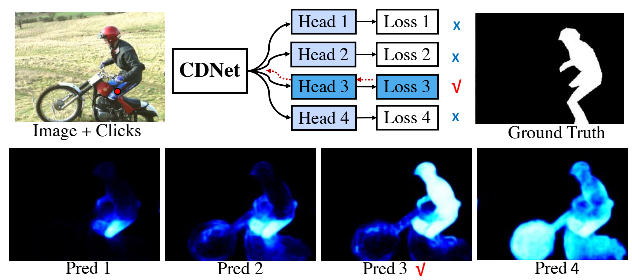


Figure 5. Demonstration of Diversity Training. We predict N more potential results and chose the one most similar to the ground truth to propagation the gradients. The original head is not shown for simplicity.

In this work, we reduce the ambiguity by developing Diversified Training (DT). Inspired by [17, 19], we explore the latent diversity; differently, we focus on the training process instead of inference. Fig. 5 depicts the pipeline of DT. We add another N latent heads during training and remove them during inference. In this work, we set N = 4. We supervise these latent heads with diversity loss and click loss. As Eq. (9), diversity loss calculates the cross-entropy of each latent head and chooses the minimum one to initiate backward propagation. With this setup, the ambiguity is eased by permitting all reasonable predictions. We also design a click loss to enforce each latent head to make correct predictions around labeled clicks. As Eq. (10), M_i denotes the Gaussian mask of clicks, which only keeps the gradients in clicked regions. The total loss is the sum of the original binary cross-entropy loss, the diversity loss, and the click loss as Eq. (11). In this work, we set a_0, a_1, a_2 as 0.5,1,1.

$$\mathcal{L}_{div} = min(\mathcal{L}_{bce}(\mathbf{P}_i, \mathbf{gt}) \mid i \in [1, n]) \tag{9}$$

$$\mathcal{L}_{click} = \sum_{i \in [1,n]} \mathcal{M}_i \, \mathcal{L}_{bce}(\mathbf{P}_i, \mathbf{gt}) \tag{10}$$

$$\mathcal{L}_{total} = a_0 \, \mathcal{L}_{bce}(\mathbf{P}_0, \mathbf{gt}) + a_1 \, \mathcal{L}_{div} + a_2 \, \mathcal{L}_{click} \tag{11}$$

We only keep the original head during inference. Although removed for prediction, latent heads make contributions during training; they assist the model to learn better representations. Based on the representations, the original head learns the projection relation between the distribution of input clicks and the prospective mask, which enables the original head to make better predictions during inference when given enough clicks.

4. Experiments

4.1. Experiment Configurations

Implementation Details. For the 5-channel input, we embed user clicks into two Gaussian Maps with the amplitude as 1, the standard derivation as 10. Following f-BRS [27], we use the same Map Fusion block to adjust the 5-channel input to 3-channel tensor using 1×1 convolutions and LeakyReLU. So that the 3-channel tensor could be fed into a ResNet-50 [9] backbone pretrained on ImageNet [4]. During inference, we applied the same cropping strategy as f-BRS. Starting from the third click, we calculate a minimum box around the predicted mask and expand the box with 40% along sides. Then, we crop the image according to the box and apply interactive segmentation only on the Zoom-In region.

Training Hyper-Parameters. We train our CDNet on SBD [8] train set with 8498 images. We crop training images with 320×480. For data augmentation, we applied random rotation, flip, random resize (0.75 1.25), random brightness (-0.25 0.25), random contrast (-0.15 0.4), and RGB shift (shift limit = 10). We use Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ for 120 epochs. The first two epochs are the warm-up stage in which the learning rate increases linearly from 0 to 5×10^{-4}. For the remaining epochs, cosine annealing learning rate is applied. We train our model on 4 GPUs with batch size 32 using asynchronous BatchNorm.

Training Click Simulation. Positive/ negative clicks are simulated by sampling points inside/outside the ground truth mask following [32]. The number of foreground and background points is randomly chosen in [1,10] and [0,10] with a probability decay rate of 0.7.

Evaluation Protocol. For fair comparisons, we follow the protocol of previous works [32, 18, 18, 22, 27, 11], and generate clicks automatically: The first click is placed on the center of the ground truth mask. Following clicks are placed at the center of the largest error region iteratively until reaching the targeted Intersection over Union (IoU) or the max click number.

Evaluation Metrics. We report the average Number of Click (NoC) required to reach the target IoU and set the target IoU as 85% and 90%. We set the default max number of clicks as 20 and report the Number of Failure (NoF) that could not reach the target IoU with 20 clicks. Since response time is important for industrial applications, we also report Second Per Click (SPC) to measure the speed of our method on a single 1080 Ti GPU.

4.2. Comparison to state-of-the-art

We compare our Conditional Diffusion Network with other state-of-the-art click-based methods on four benchmarks. Comparison results could be found in Tab. 1.

- **GrabCut** [26] : GrabCut dataset contains 50 images. It is commonly used to evaluate the performance of interactive segmentation models.
- **Berkeley** [24] : Berkeley dataset contains 96 images with 100 instance masks for testing.
- **SBD** [8] : SBD is a relatively larger dataset with 2,802 test images with 6,671 instance masks.
- **DAVIS** [25] : DAVIS dataset is annotated for the task of video object segmentation, which contains 50 videos. We sample the same 345 frames as BRS [11].

Result Analysis. As shown in Tab. 1, CDNet outperforms other models on all four datasets with large margins. We do not include FCANet [20] because it uses more training data than other works.

Speed Analysis. Interactive segmentation is often used in annotation tools that need immediate feedback. Hence, inference speed is an important factor. FDM models the affinity across the full image, but it is applied on features with low resolution. PDM is applied on full-resolution logits, but it diffuses the information in the local neighborhood. Therefore, the budgets brought by FDM and PDM are affordable. In Tab. 2, we compare the running speed of our method on DAVIS dataset, with f-BRS [27] and BRS [11], the previous SOTA methods. f-BRS [27] and BRS [11] apply online learning to fine-tune the parameters of the network, which enables them to make accurate predictions with few clicks.

Method	GrabCut		Berkeley	SBD		DAVIS	
	NoC@85	NoC@90	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
Graph cut [2]	7.98	10.00	14.22	13.6	15.96	15.13	17.41
Geodesic matting [7]	13.32	14.57	15.96	15.36	17.60	18.59	19.50
Random walker [6]	11.36	13.77	14.02	12.22	15.04	16.71	18.31
Euclidean star convexity [7]	7.24	9.20	12.11	12.21	14.86	15.41	17.70
Geodesic star convexity [7]	7.10	9.12	12.57	12.69	15.31	15.35	17.52
Growcut [28]	–	16.74	18.25	–	–	–	–
DOS w/o GC [32]	8.02	12.59	–	14.30	16.79	12.52	17.11
DOS with GC [32]	5.08	6.08	–	9.22	12.80	9.03	12.58
Latent diversity [17]	3.20	4.79	–	7.41	10.78	**5.05**	9.57
RIS-Net [18]	–	5.00	–	6.03	–	–	–
CM guidance [22]	–	3.58	5.60	–	–	–	–
BRS [11]	2.60	3.60	5.08	6.59	9.78	5.58	8.24
f-BRS-B-50 [27]	2.50	2.98	4.34	5.06	8.08	5.39	7.81
Ours	**2.22**	**2.64**	**3.69**	**4.37**	**7.87**	5.17	**6.66**

Table 1. Evaluation results on GrabCut, Berkeley, SBD and DAVIS datasets. NoC@85/ 90 denotes the average Number of Clicks required the get IoU of 85/ 90%.

However, online learning is time-consuming and hard to deploy. Results show that CDNet not only surpasses BRS and f-BRS on accuracy, but also is evidently faster.

Method	baseline	Ours	BRS[27]	f-BRS [27]
SPC (s)	0.20	0.23	1.47	0.32
NoC@90	8.42	**6.66**	7.93	7.81

Table 2. Comparison for inference speed on DAVIS dataset. Speeds are measured with the same hardware settings.

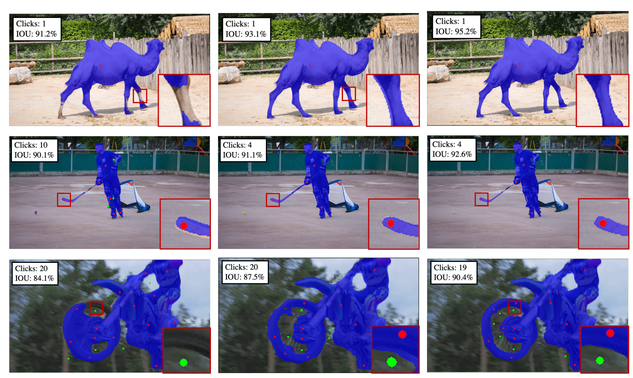
Method	NoF_{20}@90	NoF_{100}@90	NoC_{100}@90
Baseline	84	64	24.03
BRS [11]	77	51	20.89
f-BRS [27]	78	50	20.70
Ours	**65**	**48**	**18.59**

Table 3. Experiment for NoC_{100} on DAVIS dataset. NoF_{100}@90 denotes the Number of Failure images that could not reach IoU 0.9 under 100 clicks. NoC_{100}@90 means the average Number of Clicks required to reach IoU 0.9 under 100 clicks.

Analysis for 100 Clicks. Following f-BRS [27], in Tab. 3, we also report the metric under 100 clicks on DAVIS dataset. The motivation is that the traditional NoC_{20} evaluates images requiring 20 clicks and images requiring 200 clicks with the same NoC results. However, many images need more than 20 clicks, which makes NoC_{20} not distinguishing for difficult images. Results show that our method gets better performance than f-BRS with a clear margin under the metric of NoC_{100}.

4.3. Ablation Studies

We carry out plenty of ablation studies to verify the effectiveness of our method. We choose DAVIS [25] dataset to evaluate the performance. Since the masks in DAVIS are annotated with high quality, and images in DAVIS cover various scenarios, the result on DAVIS is more convincing. We first prove the effectiveness of our three core components: FDM, PDM, and DT. Then we dive into details to give an in-depth analysis for FDM and PDM.



(a) Baseline (b) With FDM (c) With FDM+PDM

Figure 6. Qualitative results on DAVIS dataset. Three columns demonstrate the result of the baseline method, our method with FDM, and our method with FDM+PDM.

Effectiveness of Core Components. In Tab. 4, we verify the effectiveness of FDM, PDM, and DT. The baseline listed in the first row is a DeeplabV3+ [3] with 5-channel input. Results demonstrate that three new modules all bring steady improvements. Baseline$^+$ with DT could serve as a stronger baseline without extra computation during inference. FDM brings clear improvement with only 0.01 seconds of inference time. PDM requires affordable 0.02 seconds, while brings significant improvement for the NoC. Equipped with all three proposed components, CDNet posts remarkable performance gains with reasonable computation

Version	DT	FDM	PDM	NoC@80	NoC@85	NoC@90	NoF@85	NoF@90	SPC (s)
Baseline				4.27	5.60	8.42	52	84	0.198
Baseline^{+}	✓			4.24	5.47	8.14	51	84	0.198
CDNet		✓		4.10	5.40	7.64	51	72	0.208
CDNet			✓	4.07	5.39	7.06	49	63	0.221
CDNet	✓	✓	✓	**3.89**	**5.17**	**6.66**	**46**	**61**	0.230

Table 4. Ablation studies for core components of CDNet on DAVIS dataset. **FDM** denotes Feature Diffusion Module. **PDM** denotes Pixel Diffusion Module. **DT** denotes Diversified Training. For the metrics, NoC stands for the average Number of Clicks required to get target IoU. NoF denotes Numbers of Failure cases that could not reach the target IoU in 20 clicks. SPC means Second Per Click.

Version	S	D	NoC@85	NoC@90	SPC (s)
Baseline			5.60	8.42	0.198
Auxiliary loss			5.60	8.61	0.198
Non-local [29]			5.87	8.81	0.202
Source2Full	✓		5.56	8.49	0.203
Des2Des	✓	✓	5.62	8.10	0.208
Source2Des^{-}	✓	✓	5.53	8.04	0.208
FDM	✓	✓	**5.40**	**7.64**	0.208

Table 5. Ablation studies for Feature Diffusion Module on DAVIS dataset. S, D denotes Source Constrain, Destination Constrain.

overhead.

Qualitative comparisons for FDM and PDM are shown in Fig. 6. Three columns demonstrate the results of baseline, CDNet with FDM, and CDNet with FDM+PDM. FDM enables the features of clicks to be generalized to wider regions, which assists to correct large regions of false prediction. PDM refines the boundary with low-level consistency, which helps to capture fine details as shown in zoomed-in patches. The positions of clicks are not exactly the same for each column, because the clicks are generated according to the evaluation protocol introduced in 4.1.

Experiments for FDM. In Tab. 5, we prove the effectiveness of our proposed conditional affinity. First, as FDM introduces the auxiliary supervision for the destination constraint, we add an auxiliary loss on the baseline without FDM to make the comparison in the second row. Then, we develop four variants of FDM with different settings of source and destination constraints: **Non-local** could be regarded as a variant of FDM without any constraint; **Source2Full** denotes the version only with source constraint; **Des2Des** uses the destination constrain map in FDM to constrain both the diffusion source and destination; **Source2Des^{-}** applies both source /destination constraints, but remove the supervision for destination constraints and make the model learn it in an end-to-end manner.

Results show that: 1) A single auxiliary loss could not bring improvement. 2) Non-local exert a negative effect. We analyze that it is hard to directly learn a conditional affinity map with fixed convolutional filters for interactive segmentation. 3) Single source /destination constrain do bring improvements compared with the raw non-local layer, but it is still sub-optimal compared with FDM. Thus, the conditional affinity in FDM is proven to be effective.

Amplitude	0	0.2	0.5	1	2
NoC@90	6.89	6.87	6.86	**6.66**	6.72

Table 6. Experiment for the enhancement amplitude for PDM on DAVIS dataset with 10 times of iteration.

Iterations	0	1	5	10	20
SPC (s)	0.208	0.216	0.224	0.230	0.243
NoC@90	7.54	7.15	6.74	**6.66**	6.66

Table 7. Experiment for the diffusion iteration of PDM on DAVIS dataset.

Experiments for PDM. PDM constructs a conditional diffusion source and diffuses the information in local regions iteratively. The conditional diffusion source is constructed by enhancing the logits around clicks using Gaussian kernels as Eq. (7). In Tab. 6, we prove the effectiveness of the conditional source by changing the amplitude of Gaussian kernels. It can be observed that source enhancement brings steady improvements, and we simply set the amplitude to 1.

In Tab. 7, we explore the trade-off between speed and accuracy for different diffusion iterations. The performance reaches saturation at about 10 times refinement. Considering the trade-off between accuracy and speed, we set the iteration number as 10 in this work.

5. Conclusion

In this paper, we formulate click-based interactive segmentation as a process of information diffusion and propose Conditional Diffusion Network. We design a Feature Diffusion Module and a Pixel Diffusion Module to propagate information from clicked points to target regions. Experiments show that our method is effective on four benchmarks and sets new state-of-the-art.

References

[1] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 392–399, 2014. 1

[2] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pages 105–112. IEEE, 2001. 7

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2, 7

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[5] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 3

[6] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006. 1, 2, 7

[7] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3129–3136. IEEE, 2010. 2, 7

[8] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE, 2011. 2, 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 6

[10] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 603–612, 2019. 3

[11] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5306, 2019. 1, 2, 5, 6, 7

[12] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Nonparametric higher-order learning for interactive segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3201–3208. IEEE, 2010. 2

[13] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. 5

[14] Hoang Le, Long Mai, Brian Price, Scott Cohen, Hailin Jin, and Feng Liu. Interactive boundary prediction for object selection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 18–33, 2018. 1

[15] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *2009 IEEE 12th international conference on computer vision*, pages 277–284. IEEE, 2009. 1

[16] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Transactions on Graphics (ToG)*, 23(3):303–308, 2004. 1

[17] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 577–585, 2018. 2, 5, 7

[18] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2746–2754. IEEE, 2017. 2, 6, 7

[19] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, Sim-Heng Ong, and Jiashi Feng. Multiseg: Semantically meaningful, scale-diverse segmentations from minimal user input. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 662–670, 2019. 5

[20] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13339–13348, 2020. 1, 2, 6

[21] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. *arXiv preprint arXiv:1805.04398*, 2018. 1

[22] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11602–11611, 2019. 1, 2, 5, 6, 7

[23] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–625, 2018. 1

[24] Kevin McGuinness and Noel E O'connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010. 2, 6

[25] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 2, 6, 7

[26] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004. 1, 2, 5, 6

[27] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 8623–8632, 2020. 1, 2, 5, 6, 7

[28] Vladimir Vezhnevets and Vadim Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. In *proc. of Graphicon*, volume 1, pages 150–156. Citeseer, 2005. 7

[29] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3, 8

[30] Jiajun Wu, Yibiao Zhao, Jun-Yan Zhu, Siwei Luo, and Zhuowen Tu. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 256–263, 2014. 1

[31] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. *arXiv preprint arXiv:1707.00243*, 2017. 1

[32] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381, 2016. 1, 2, 5, 6, 7

[33] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 3