Supplementary Materials for "Interactive Segmentation as Gaussian Process Classification"

Abstract

In the supplementary materials, we first provide more implementation details, including training strategy, network architectures, and model hyperparameters. Then, we list the algorithm flowcharts of the training procedure and forward process, respectively. Furthermore, we provide detailed derivations about Gaussian process posterior approximation and efficient sampling in Sec. 4.2 of the main text. Finally, more experimental results are given, including quantitative evaluation on more backbone segmentors and visual comparisons on more diverse images selected from different datasets. Besides, we also analyze how the inference time changes as the number of clicks increases.

1. Implementation Details

Training strategy. For click simulation during training, following RITM [16], we use the iterative training strategy [10] with a maximum of 3 iterations, and the maximum number of clicks is set as 24 with a probability decay of 0.8. Following [3,15,16], we adopt Target crop by cropping the minimum external box of the previous mask and the newly added click and expanding the box with a ratio of 1.4. Then, the target area is resized as 256×256 pixels for subsequent processing. Randomly cropping and scaling are adopted for data augmentation, which follows the configuration in [3]. The entire framework is trained based on Adam optimizer [8] with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is 5×10^{-3} for SegFormerB0-S2 and ResNet50, and 5×10^{-4} for HRNet18s-S2. Following [3, 16], the initialization of the three backbones utilizes weights pre-trained on ImageNet, with the learning rate of the backbone weights reduced by a factor of 0.1.

Network architectures. In experiments, we adopt three backbone segmentors without the last-layer classifier for extracting deep features. For the three backbones, *i.e.*, SegFormerB0-S2 [3, 20], HRNet18s-S2 [3, 17], and DeepLabv3+ [2] with ResNet50 [6], we set the number of feature channels d to 96, 48, and 48, respectively, and adopt the weights pre-trained on ImageNet [4] as initialization. To feed the click information into the network, we follow [3, 16] and encode clicks as disks with a fixed radius. Then, we follow the "Conv1S" architecture in [16] to encode the click maps along with the previous mask into auxiliary features and add them to the image branch.

Model hyperparameters. For the backbone segmentors SegFormerB0-S2, HRNet18s-S2, and ResNet50 [6], the number of basis functions l is set to 128, 256, and 256, respectively. For function-space deep kernel learning, the hyperparameter η_0 is initialized as 1 and η_t $(t > 0)$ as e^{-1}. In our experiments, we adopt the log computation manner to learn the kernel hyperparameter η_t for guaranteeing its positiveness. Equivalently, the initialization is 0 for $\log \eta_0$, and -1 for $\log \eta_t$ $(t > 0)$. For weight-space deep kernel learning, we initialize the related parameters as $\boldsymbol{\theta}_r \sim \mathcal{N}(0, \mathbf{I}_d)$, $\tau_r \sim U(0, 2\pi)$, $\boldsymbol{\mu}_{\mathbf{w}} \sim \mathcal{N}(0, 0.25\mathbf{I}_d)$, and $\sigma^2_{\mathbf{w}} = 0.025$.

2. Algorithm Flowchart

For the proposed GPCIS, we present the corresponding training pipeline and the forward process, as listed in Alg. S1 and Alg. S2, respectively.

Algorithm S1 GPCIS Training Pipeline for an Epoch

Input: Training dataset $\mathcal{D} = \{(\mathcal{I}_i, \mathbf{y}_{gt,i})\}_{i=1}^{N}$, initialized trainable parameters $\Omega = \{\psi, \xi, \boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\tau}, \boldsymbol{\mu}_{\mathbf{w}}, \sigma_{\mathbf{w}}\}$, $\sigma^2 = \epsilon^2 = 0.01$.
Output: Trained parameters Ω.

for $(\mathcal{I}_i, \mathbf{y}_{gt,i}) \sim \mathcal{D}$ **do**
  $\mathbf{y}_{prev} \leftarrow zeros\_like(\mathbf{y}_{gt,i})$ ▷ Initialize the previous mask with a zero-map
  $L_{click} \leftarrow point\_sampler(\mathcal{I}_i, \mathbf{y}_{gt,i})$ ▷ Simulate clicks for training
  $N_{iter} \leftarrow random(0, 3)$ ▷ Number of iterations for iterative training strategy
  for n_{iter} in $range(N_{iter})$ **do**
    $(c, y_c) \leftarrow get\_click(\mathbf{y}_{prev}, \mathbf{y}_{gt})$
    $L_{click} \leftarrow L_{click} \cup (c, y_c)$
    $\mathbf{y}_{prev}, \_ \leftarrow GPCIS(\mathcal{I}_i, L_{click}, \mathbf{y}_{prev}; \Omega, \sigma^2, \epsilon^2)$ ▷ GPCIS forward process in Alg. S2
  end for
  $\tilde{\mathbf{y}}, \mathcal{L}_{VI} \leftarrow GPCIS(\mathcal{I}_i, L_{click}, \mathbf{y}_{prev}; \Omega, \sigma^2, \epsilon^2)$
  $\mathcal{L} \leftarrow \mathcal{L}_{NFL}(\tilde{\mathbf{y}}, \mathbf{y}_{gt,i}) + \alpha \mathcal{L}_{VI}$
  $\Omega \leftarrow adam\_opt(\mathcal{L}, \Omega)$
end for

Algorithm S2 GPCIS Forward Algorithm $GPCIS(\cdot)$

Input: Input image $\mathcal{I}$, click list L_{click}, previous mask $\mathbf{y}_{prev}$, parameters $\Omega, \sigma^2, \epsilon^2$.
Output: Prediction $\tilde{\mathbf{y}}$, amortized variational inference loss $\mathcal{L}_{VI}$.

$S_{click} \leftarrow encode\_click(L_{click})$ ▷ Encode the clicks into disk maps
$\mathbf{X} \leftarrow g_{\psi}(\mathcal{I}, S_{click}, \mathbf{y}_{prev})$ ▷ Backbone segmentor
$\bar{\mathbf{X}} \leftarrow normalize(\mathbf{X}) \cup \mathcal{I}$
$\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}}, \sigma_{\mathbf{w}}^2 \mathbf{I}_l)$
$\mathbf{f}_{prior} \leftarrow \boldsymbol{\Phi}(\bar{\mathbf{X}})\mathbf{w} = \sqrt{2/l}\cos(\boldsymbol{\Theta}^T \bar{\mathbf{X}} + \boldsymbol{\tau})\mathbf{w}$ ▷ Weight-space prior
$\bar{\mathbf{X}}_n, \boldsymbol{\Phi}(\bar{\mathbf{X}}_n) \leftarrow locate(\bar{\mathbf{X}}, L_{click}), locate(\mathbf{f}_{prior}, L_{click})$ ▷ Features at clicked positions
$\mathbf{m}_{\xi} \leftarrow \mathrm{Softplus}(\mathrm{MLP}(\bar{\mathbf{X}}_n)) * \mathbf{y}_n$
$\mathbf{f}_n \sim \mathcal{N}(\mathbf{m}_{\xi}, \sigma^2 \mathbf{I}_n)$
$\mathbf{K}_{n,n}, \mathbf{K}_{m,n} \leftarrow k_{\boldsymbol{\eta}}(\bar{\mathbf{X}}_n, \bar{\mathbf{X}}_n), k_{\boldsymbol{\eta}}(\bar{\mathbf{X}}, \bar{\mathbf{X}}_n)$
$\mathbf{f}_{update} \leftarrow \mathbf{K}_{m,n}(\mathbf{K}_{m,n} + \epsilon^2 \mathbf{I}_n)^{-1}(\mathbf{f}_n - \boldsymbol{\Phi}(\bar{\mathbf{X}}_n))$ ▷ Function-space update
$\tilde{\mathbf{y}} \leftarrow s(\mathbf{f}_{prior} + \mathbf{f}_{update})$
$\mathcal{L}_{VI} \leftarrow -\sum_{c=1}^{n}\left[y_c \log s(f_c) + (1 - y_c)\log(1 - s(f_c))\right] + \frac{1}{2}\mathbf{m}_{\xi}^T(\mathbf{K}_{n,n} + \epsilon^2 \mathbf{I}_n)^{-1}\mathbf{m}_{\xi}$.

3. Derivations

3.1. KL Divergence in Eq. (8)

For the variational distribution $q(\mathbf{f}_n|\mathbf{X}_n, \mathbf{y}_n) = \mathcal{N}(\mathbf{m}_\xi, \sigma^2\mathbf{I}_n)$, the KL divergence in Eq. (5) can be written as:

$$
\begin{aligned}
&\min_\xi D_{KL}(q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)||p(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n))\\
\Rightarrow &\min_\xi -\int q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\log\frac{p(\mathbf{y}_n|\mathbf{X}_n,\mathbf{f}_n)p(\mathbf{f}_n|\mathbf{X}_n)}{q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)}\mathrm{d}\mathbf{f}_n\\
\Rightarrow &\min_\xi -\int \mathcal{N}(\mathbf{m}_\xi,\sigma^2\mathbf{I}_n)\log(\Pi_{c=1}^n s(y_c f_c))\mathrm{d}\mathbf{f}_n\\
&\qquad -\int \mathcal{N}(\mathbf{m}_\xi,\sigma^2\mathbf{I}_n)\log\mathcal{N}(0,\mathbf{K}_{n,n})\mathrm{d}\mathbf{f}_n\\
&\qquad +\int \mathcal{N}(\mathbf{m}_\xi,\sigma^2\mathbf{I}_n)\log\mathcal{N}(\mathbf{m}_\xi,\sigma^2\mathbf{I}_n)\mathrm{d}\mathbf{f}_n\\
\Rightarrow &\min_\xi -\int \mathcal{N}(\mathbf{m}_\xi,\sigma^2\mathbf{I}_n)\sum_{c=1}^n[\mathbb{1}_{\{y_c=1\}}\log s(f_c)+\mathbb{1}_{\{y_c=-1\}}\log(1-s(f_c))]\mathrm{d}\mathbf{f}_n\\
&\qquad -\frac{1}{2}(-\mathbf{m}_\xi^T\mathbf{K}_{n,n}^{-1}\mathbf{m}_\xi-\log\sigma^2-\log|\mathbf{K}_{n,n}|-\sigma^2\mathrm{Tr}(\mathbf{K}_{n,n}^{-1}))\\
&\qquad +\frac{1}{2}(-\log\sigma^2-n-n\log 2\pi)\\
\Rightarrow &\min_\xi -\int \mathcal{N}(\mathbf{m}_\xi,\sigma^2\mathbf{I}_n)\sum_{c=1}^n[\mathbb{1}_{\{y_c=1\}}\log s(f_c)+\mathbb{1}_{\{y_c=-1\}}\log(1-s(f_c))]\mathrm{d}\mathbf{f}_n+\frac{1}{2}\mathbf{m}_\xi^T\mathbf{K}_{n,n}^{-1}\mathbf{m}_\xi
\end{aligned}
\tag{S1}
$$

where we have used $p(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n) \propto p(\mathbf{y}_n|\mathbf{X}_n,\mathbf{f}_n)p(\mathbf{f}_n|\mathbf{X}_n)$, $p(\mathbf{f}_n|\mathbf{X}_n) = \mathcal{N}(\boldsymbol{\mu}_n, \mathbf{K}_{n,n})$, and $p(\mathbf{y}_n|\mathbf{X}_n,\mathbf{f}_n) = \Pi_{c=1}^n s(y_c f_c)$, as analyzed in Sec. 4.2 of the main text. Then, Eq. (S1) can be rearranged into Eq. (8) of the main text.

3.2. GP Posterior in Eqs. (9) (10)

After obtaining the Gaussian variational distribution $q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n) = \mathcal{N}(\mathbf{m}_\xi,\sigma^2\mathbf{I}_n)$, with $\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n \sim \mathcal{N}(\boldsymbol{\mu}_{*|n},\mathbf{K}_{*,*|n})$ as defined in Eq. (1) of the main text, we can easily know that $p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n) = \int p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n)q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n$ is Gaussian. Next, we aim to compute the mean and variance of $p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n)$. The mean can be computed as:

$$
\begin{aligned}
&\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n]\\
=&\int \mathbf{f}_*\ p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_*\\
=&\int \mathbf{f}_*(\int p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n)q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n)\mathrm{d}\mathbf{f}_*\\
=&\int \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n]q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n\\
=&\int \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{f}_n q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n\\
=&\mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbb{E}_q[\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n]\\
=&\mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{m}_\xi
\end{aligned}
\tag{S2}
$$

where we have used $\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] = \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{f}_n$ and $\mathbb{E}_q[\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n] = \mathbf{m}_\xi$.

The variance can be computed as:

$$
\begin{aligned}
&\mathit{Var}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n] \\
=&\int (\mathbf{f}_* - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n])^2 p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_* \\
=&\int [(\mathbf{f}_* - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n]) + (\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n])]^2 p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_* \\
=&\int (\mathbf{f}_* - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n])^2 \int p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n) q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n\mathrm{d}\mathbf{f}_* \\
&+ \int (\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n])^2 \int p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n) q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n\mathrm{d}\mathbf{f}_* \\
&+ \int 2(\mathbf{f}_* - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n])(\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n]) \int p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n) q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n\mathrm{d}\mathbf{f}_* \\
=&\int \left[\int (\mathbf{f}_* - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n])^2 p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n)\mathrm{d}\mathbf{f}_*\right] q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n \quad (\star) \\
&+ \int \left[\int (\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n])^2 q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n\right] p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n)\mathrm{d}\mathbf{f}_* \quad (\dagger) \\
&+ \int 2(\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n])(\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n]) q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)\mathrm{d}\mathbf{f}_n \\
=&\mathbb{E}_{p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n)}[(\mathbf{f}_* - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n])^2] + \mathbb{E}_{q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)}[(\mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] - \mathbb{E}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n])^2] + 0 \\
=&\mathit{Var}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] + \mathbb{E}_{q(\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n)}[(\mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{f}_n - \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{m}_\xi)^2] \quad (\ddagger) \\
=&\mathbf{K}_{*,*} - \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{K}_{n,*} + \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\sigma^2\mathbf{I}_n\mathbf{K}_{n,n}^{-1}\mathbf{K}_{n,*} \\
=&\mathbf{K}_{*,*} - \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}(\mathbf{I}_n - \sigma^2\mathbf{K}_{n,n}^{-1})\mathbf{K}_{n,*}
\end{aligned}
\tag{S3}
$$

For the term $(\star)$, we have acknowledged that the integral in the brackets is equal to $\mathit{Var}[\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{f}_n] = \mathbf{K}_{*,*} - \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{K}_{n,*}$ and is irrelevant to the distribution of $\mathbf{f}_n$. For the term $(\dagger)$, we also use the fact that the integral in the brackets is irrelevant to $\mathbf{f}_*$. For the second term of $(\ddagger)$, we have used the fact that $\mathbb{E}_q[\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n] = \mathbf{m}_\xi$ and $\mathit{Var}_q[\mathbf{f}_n|\mathbf{X}_n,\mathbf{y}_n] = \sigma^2\mathbf{I}_n$.

Based on Eqs. (S2)(S3), we can get:

$$
p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n) \sim \mathcal{N}(\boldsymbol{\mu}_{*|n}, \mathbf{K}_{*,*|n}), \tag{S4}
$$

where

$$
\begin{aligned}
\boldsymbol{\mu}_{*|n} &= \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{m}_\xi, \\
\mathbf{K}_{*,*|n} &= \mathbf{K}_{*,*} - \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}(\mathbf{I}_n - \sigma^2\mathbf{K}_{n,n}^{-1})\mathbf{K}_{n,*}.
\end{aligned}
\tag{S5}
$$

Here Eqs. (S4)(S5) correspond to Eqs. (9)(10) in the main text.

3.3. Decoupled GP Posterior

In this subsection, we briefly review the background of the decoupled GP posterior for efficient sampling [18, 19] and provide the derivations of Eq. (11) in the main text.

3.3.1 Double-Space Views of GP

Function-Space View of GP. In the main text, we have introduced the *function-space view* of GP in Sec. 3, *i.e.*, reasoning about the prior and posterior distribution of f evaluated at data points. Specifically, for a GP $f \sim \mathcal{GP}(\mu, k)$, we denote the marginal $\mathbf{f}_n = f(\mathbf{X}_n)$. Given n noisy observations $\mathbf{y}_n \sim \mathcal{N}(\mathbf{f}_n, \sigma^2\mathbf{I}_n)$ at training data $\mathbf{X}_n$, the GP posterior at testing data $\mathbf{X}_*$ is written as:

$$
\mathbf{f}_*|\mathbf{X}_*,\mathbf{X}_n,\mathbf{y}_n \sim \mathcal{N}(\boldsymbol{\mu}_{*|n}, \mathbf{K}_{*,*|n}), \tag{S6}
$$

where

$$
\begin{aligned}
\boldsymbol{\mu}_{*|n} &= \mathbf{K}_{*,n}(\mathbf{K}_{n,n} + \sigma^2\mathbf{I}_n)^{-1}\mathbf{f}_n, \\
\mathbf{K}_{*,*|n} &= \mathbf{K}_{*,*} - \mathbf{K}_{*,n}(\mathbf{K}_{n,n} + \sigma^2\mathbf{I}_n)^{-1}\mathbf{K}_{n,*}.
\end{aligned}
\tag{S7}
$$

Weight-Space View of GP. As an alternative view, the *weight-space view* of GP is to view f as a weighted sum of basis functions with weights $\mathbf{w}$ and to reason about the prior and posterior distribution of $\mathbf{w}$. Specifically, for a GP with stationary covariance function $k(\cdot,\cdot)$, *e.g.* RBF kernels, we can find the corresponding basis functions for approximation in the weight-space, *i.e.*, the random Fourier features [13] $\mathbf{\Phi}$ given by Eq. (12) in the main text. Then, the GP can be expressed as:

$$f(\cdot) = \mathbf{\Phi}(\cdot)\mathbf{w} = \sum_{r=1}^{l} w_r \phi_r(\cdot), \tag{S8}$$

where l is the number of basis functions, $\phi_r(\mathbf{x}) = \sqrt{2/l}\cos(\boldsymbol{\theta}_r^T \mathbf{x} + \tau_r)$, $\tau_r \sim U(0, 2\pi)$, and $\boldsymbol{\theta}_r \in \mathbb{R}^d$ is sampled from the spectral density of the kernel $k(\cdot,\cdot)$. Given the observations $\mathbf{y}_n \sim \mathcal{N}(\mathbf{\Phi}(\mathbf{X}_n)\mathbf{w}, \sigma^2\mathbf{I}_n)$, the posterior distribution of $\mathbf{w}$ is $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}|n}, \mathbf{\Sigma}_{\mathbf{w}|n})$, where

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{w}|n} &= (\mathbf{\Phi}^T\mathbf{\Phi} + \sigma^2\mathbf{I}_n)^{-1}\mathbf{\Phi}^T\mathbf{y}_n, \\ \mathbf{\Sigma}_{\mathbf{w}|n} &= (\mathbf{\Phi}^T\mathbf{\Phi} + \sigma^2\mathbf{I}_n)^{-1}\sigma^2, \end{aligned} \tag{S9}$$

where $\mathbf{\Phi}$ is short for $\mathbf{\Phi}(\mathbf{X}_n)$.

3.3.2 Pathwise Updates for Efficient Sampling of GP Posterior

To improve the sampling efficiency of the GP posterior, [18, 19] utilize the pathwise updates, *i.e.*, first sampling from the GP prior and then updating it using training data. The idea comes from Matheron's rule for Gaussian random variables [7]:

Theorem 1 (Matheron's rule)**.** *Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be jointly Gaussian random variables. Then, the distribution of $\boldsymbol{a}$ conditioned on $\boldsymbol{b} = \boldsymbol{\beta}$ satisfies*

$$(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}) \overset{d}{=} \boldsymbol{a} + \mathrm{Cov}(\boldsymbol{a}, \boldsymbol{b})\mathrm{Cov}(\boldsymbol{b}, \boldsymbol{b})^{-1}(\boldsymbol{\beta} - \boldsymbol{b}), \tag{S10}$$

where $\overset{d}{=}$ means equal in distribution and $\mathrm{Cov}(\cdot,\cdot)$ is the covariance operation.

Matheron's rule can be easily extended to the GP case:

Corollary 1. *For a Gaussian process $f \sim \mathcal{GP}(0, k)$ with marginal $\mathbf{f}_n = f(\mathbf{X}_n)$, the process conditioned on $\mathbf{f}_n = \boldsymbol{u}_n$ satisfies*

$$\underbrace{(f \mid \mathbf{f}_n = \boldsymbol{u}_n)(\cdot)}_{\textit{posterior}} \overset{d}{=} \underbrace{f(\cdot)}_{\textit{prior}} + \underbrace{k(\cdot, \mathbf{X}_n)\mathbf{K}_{n,n}^{-1}(\boldsymbol{u}_n - \mathbf{f}_n)}_{\textit{update}}. \tag{S11}$$

Given the observations $\mathbf{y}_n \sim \mathcal{N}(f(\mathbf{X}_n), \sigma^2\mathbf{I}_n)$, we can apply Corollary 1 to both function-space and weight-space of GP [18, 19]:

$$\text{Function-space:}\ \underbrace{\mathbf{f}_* \mid \mathbf{y}_n}_{\text{posterior}} \overset{d}{=} \underbrace{\mathbf{f}_*}_{\text{prior}} + \underbrace{\mathbf{K}_{*,n}(\mathbf{K}_{n,n} + \sigma^2\mathbf{I}_n)^{-1}(\mathbf{y}_n - \mathbf{f}_n - \boldsymbol{\varepsilon})}_{\text{update}}, \tag{S12}$$

$$\text{Weight-space:}\ \underbrace{\mathbf{w} \mid \mathbf{y}_n}_{\text{posterior}} \overset{d}{=} \underbrace{\mathbf{w}}_{\text{prior}} + \underbrace{\mathbf{\Phi}^T(\mathbf{\Phi}^T\mathbf{\Phi} + \sigma^2\mathbf{I}_n)^{-1}(\mathbf{y}_n - \mathbf{\Phi}\mathbf{w} - \boldsymbol{\varepsilon})}_{\text{update}}, \tag{S13}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_n)$. With the decoupled form, sampling from the function-space prior requires computing $\mathbf{K}_{*,*}^{1/2}$ and still has the computational cost of $\mathcal{O}(*^3)$, while sampling from the weight-space prior only requires sampling $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I}_l)$, whose cost is $\mathcal{O}(l)$. Besides, the update term from the function-space view can better utilize the representations of data with the canonical basis [1]. Therefore, [18, 19] propose to sample from the constructed GP posterior with a weight-space prior term and a function-space update term, written as:

$$\underbrace{\mathbf{f}_* \mid \mathbf{y}_n}_{\text{posterior}} \overset{d}{\approx} \underbrace{\mathbf{\Phi}(\mathbf{X}_*)\mathbf{w}}_{\text{weight-space prior}} + \underbrace{\mathbf{K}_{*,n}(\mathbf{K}_{n,n} + \sigma^2\mathbf{I}_n)^{-1}(\mathbf{y}_n - \mathbf{\Phi}(\mathbf{X}_n)\mathbf{w} - \boldsymbol{\varepsilon})}_{\text{function-space update}}. \tag{S14}$$

3.3.3 Derivation of Eq. (11)

Suppose we have obtained the gaussian variational distribution $q(\mathbf{f}_n|\mathbf{y}_n) = \mathcal{N}(\mathbf{m}_\xi, \sigma^2\mathbf{I}_n)$, we aim to draw a sample $\tilde{\mathbf{f}}_*$ from the posterior distribution $p(\mathbf{f}_*|\mathbf{y}_n)$. An equivalent approach is to sample from the joint distribution $p(\mathbf{f}_*, \mathbf{f}_n|\mathbf{y}_n) = p(\mathbf{f}_*|\mathbf{f}_n)q(\mathbf{f}_n|\mathbf{y}_n)$ and only take the sampled $\mathbf{f}_*$ as a sample from $p(\mathbf{f}_*|\mathbf{y}_n)$. Hence, following the ancestral sampling approach, we can first sample $\mathbf{f}_n \sim q(\mathbf{f}_n|\mathbf{y}_n)$ and then we can easily sample from $p(\mathbf{f}_*|\mathbf{f}_n)$ using Matheron's rule and obtain a drawn sample $\tilde{\mathbf{f}}_*$. Different from the case with noisy observations in Sec. 3.3.2, we have $\mathrm{Cov}(\mathbf{f}_n, \mathbf{f}_n) = \mathbf{K}_{n,n}$ and ε disappears. Therefore, we can easily derive the drawn sample from the decoupled GP posterior as:

$$\tilde{\mathbf{f}}_* = \underbrace{\mathbf{\Phi}(\mathbf{X}_*)\mathbf{w}}_{\text{weight-space prior}} + \underbrace{\mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}(\mathbf{f}_n - \mathbf{\Phi}(\mathbf{X}_n)\mathbf{w})}_{\text{function-space update}}, \tag{S15}$$

where $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I}_l)$ and $\mathbf{f}_n \sim q(\mathbf{f}_n|\mathbf{y}_n)$. Computing Eq. (S15) requires the cost of $\mathcal{O}(*)$. Specifically, for the interactive segmentation (IS) task, we have $n \ll *$, $l \ll *$, and $d \ll *$. Sampling and computing the weight-space prior term has the cost of $\mathcal{O}(l + d)$ and the cost of computing the function-space update term mainly stems from $\mathbf{K}_{*,n}$, which is $\mathcal{O}(*)$. Although computing $\mathbf{K}_{n,n}^{-1}$ has the cost of $\mathcal{O}(n^3)$, we will demonstrate in Fig. S1 that in practice the inference speed would not increase much when the number of clicks $n \leq 20$.

4. More Experimental Results

4.1. Quantitative Results

In Table 3 of the main text, due to the limited space, we have only provided the quantitative evaluations under the backbone segmentor ResNet50, on Berkeley and DAVIS datasets. Here, we provide more comparisons under all three backbones, *i.e.*, ResNet50, SegFormerB0-S2, and HRNet18s-S2 on all four datasets, *i.e.*, GrabCut, Berkeley, SBD, and DAVIS. From Table S1, we can see that our proposed GPCIS almost achieves the best or the second-best performance in all conditions, showing its good generality.

Table S1. Complete experimental results with three backbones on four datasets. NoC_{100}@90 and NoF_{100}@90 for FocusCut on SBD are not reported due to time-consuming inference process on the large dataset.

Backbone	Method	NoC_{100}@90	NoF_{100}@90	IoU&1	IoU&5	NoIC	NoC_{100}@90	NoF_{100}@90	IoU&1	IoU&5	NoIC
SegFormerB0-S2 [3, 20]		Berkeley [11]					DAVIS [12]				
	RITM [16]	4.44	**1**	79.06%	**94.92%**	**0**	18.38	49	71.32%	89.32%	86
	FocalClick [3]	4.41	**1**	77.67%	94.49%	**0**	17.25	45	70.20%	89.19%	75
	GPCIS *(Ours)*	**3.50**	**1**	**80.14%**	94.84%	**0**	**16.92**	**42**	**73.03%**	**89.58%**	**5**
		GrabCut [14]					SBD [5]				
	RITM [16]	1.82	**0**	83.71%	96.73%	**0**	12.23	299	63.91%	88.67%	881
	FocalClick [3]	1.86	**0**	83.32%	95.65%	**0**	11.84	292	**64.43%**	87.95%	485
	GPCIS *(Ours)*	**1.76**	**0**	**85.50%**	**97.21%**	**0**	**11.72**	**279**	64.11%	**88.78%**	**12**
HRNet18s-S2 [3, 17]		Berkeley [11]					DAVIS [12]				
	RITM [16]	3.99	**1**	78.55%	93.60%	1	18.67	50	71.83%	88.53%	108
	FocalClick [3]	4.48	2	**80.34%**	94.96%	1	**17.14**	46	72.61%	**89.36%**	79
	GPCIS *(Ours)*	**3.45**	**1**	77.45%	**95.07%**	**0**	17.45	**44**	**73.67%**	89.02%	**0**
		GrabCut [14]					SBD [5]				
	RITM [16]	2.24	**0**	82.79%	93.34%	**0**	12.95	425	63.88%	88.01%	1282
	FocalClick [3]	2.04	**0**	82.59%	94.59%	1	13.01	366	**64.62%**	87.18%	703
	GPCIS *(Ours)*	**1.94**	**0**	**82.80%**	**96.42%**	**0**	**11.83**	**317**	63.81%	**88.43%**	**54**
ResNet50 [6]		Berkeley [11]					DAVIS [12]				
	f-BRS-B [15]	6.21	2	77.06%	85.00%	1	22.62	57	70.97%	83.87%	**0**
	RITM [16]	3.75	**1**	76.88%	94.66%	2	18.09	51	72.89%	89.14%	74
	FocusCut [9]	4.63	**1**	78.89%	92.89%	1	19.00	45	72.71%	87.58%	6
	FocalClick [3]	4.46	2	75.59%	94.90%	**0**	17.74	49	70.76%	88.90%	42
	GPCIS *(Ours)*	**3.36**	**1**	**79.43%**	**95.11%**	**0**	**17.03**	**44**	**75.67%**	**89.60%**	2
		GrabCut [14]					SBD [5]				
	f-BRS-B [15]	4.18	1	80.79%	89.72%	**0**	16.61	479	74.60%	81.69%	954
	RITM [16]	2.40	**0**	79.86%	95.15%	14	13.16	523	**69.66%**	89.02%	1250
	FocusCut [9]	**1.78**	**0**	**86.30%**	94.99%	1	-	-	69.32%	88.86%	150
	FocalClick [3]	2.14	**0**	80.15%	95.50%	**0**	12.52	503	66.84%	89.30%	745
	GPCIS *(Ours)*	1.82	**0**	84.44%	**96.82%**	**0**	**11.23**	**331**	67.51%	**89.60%**	**51**

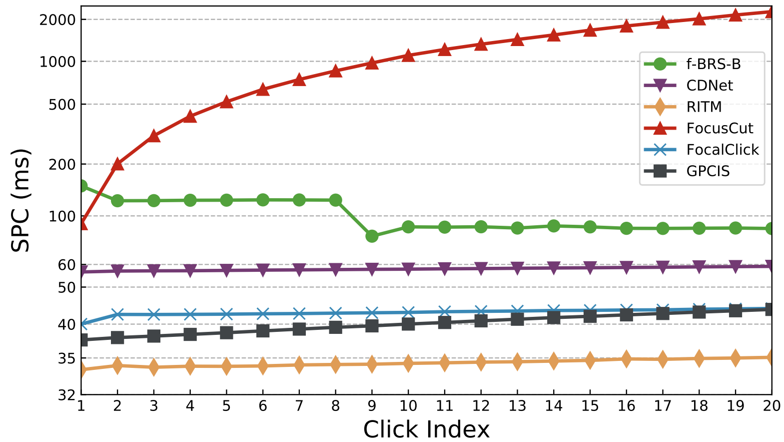

Figure S1. Change of inference time with the click index, with Resnet50 as the backbone segmentor. The y-axis is in the log scale.

Table S2. Model complexity analysis about different components of RITM and GPCIS. FLOPs are computed using input images with size of 384×384.

Method	FLOPs (GB)			Params (MB)		
	Total	Backbone	Classifier/GP inference	Total	Backbone	Classifier/GP inference
RITM	98.87	98.37	0.50	39.48	39.43	0.05
GPCIS	99.81	97.88	1.93	39.39	39.37	0.02

In Fig. S1, we report the relationship between the inference speed and the click index, *i.e.*, the number of clicks n. From the results, we can easily observe that although the computational cost of the efficient sampling framework Eq. (S15) is cubic w.r.t. n, in real interactive segmentation situations where n is usually no larger than 20 and $n \ll *$, the inference speed would not increase much as the number of clicks increases.

Based on the ResNet50 backbone, we analyze the model complexity of different model parts for the baseline RITM and our GPICS. The difference between RITM and GPICS is that we replace the last-layer classifier in RITM with our proposed GP inference module. Besides, for the backbone segmentor, in our GPCIS, we have reduced the number of channels of the last-layer features extracted by the backbone. Table S2 reports the FLOPs and the number of parameters of different model parts. As seen, GPCIS has smaller FLOPs and fewer parameters in the backbone, and the GP inference module has fewer parameters and comparable FLOPs than the classifier of RITM.

4.2. Qualitative Results

We provide more visualizations of the output probability maps and prediction masks of different methods. The images shown in Figs. S2, S3, S4, S5 are from GrabCut, Berkeley, SBD, and DAVIS datasets, respectively. It can be seen that our method achieves better segmentation results mainly attributed to the powerful GP classification framework which explicitly models the relations between pixels.

References

[1] David Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Rates of convergence for sparse variational gaussian process regression. In *International Conference on Machine Learning*, pages 862–871. PMLR, 2019. 5

[2] 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*, pages 801–818, 2018. 1

[3] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. FocalClick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1300–1309, 2022. 1, 6

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 248–255, 2009. 1

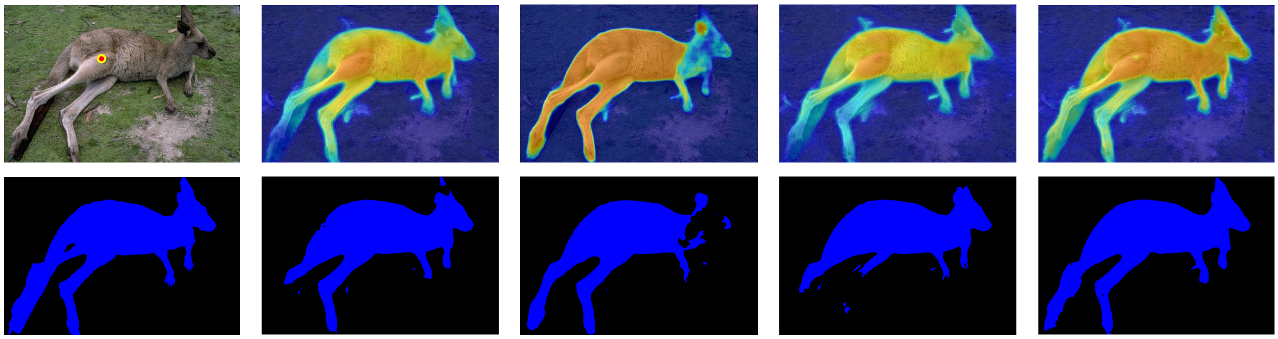

Image & GT Mask | RITM (IoU 84.81%) | FocusCut (IoU 78.57%) | FocalClick (IoU 79.36%) | GPCIS (IoU **94.62%**)

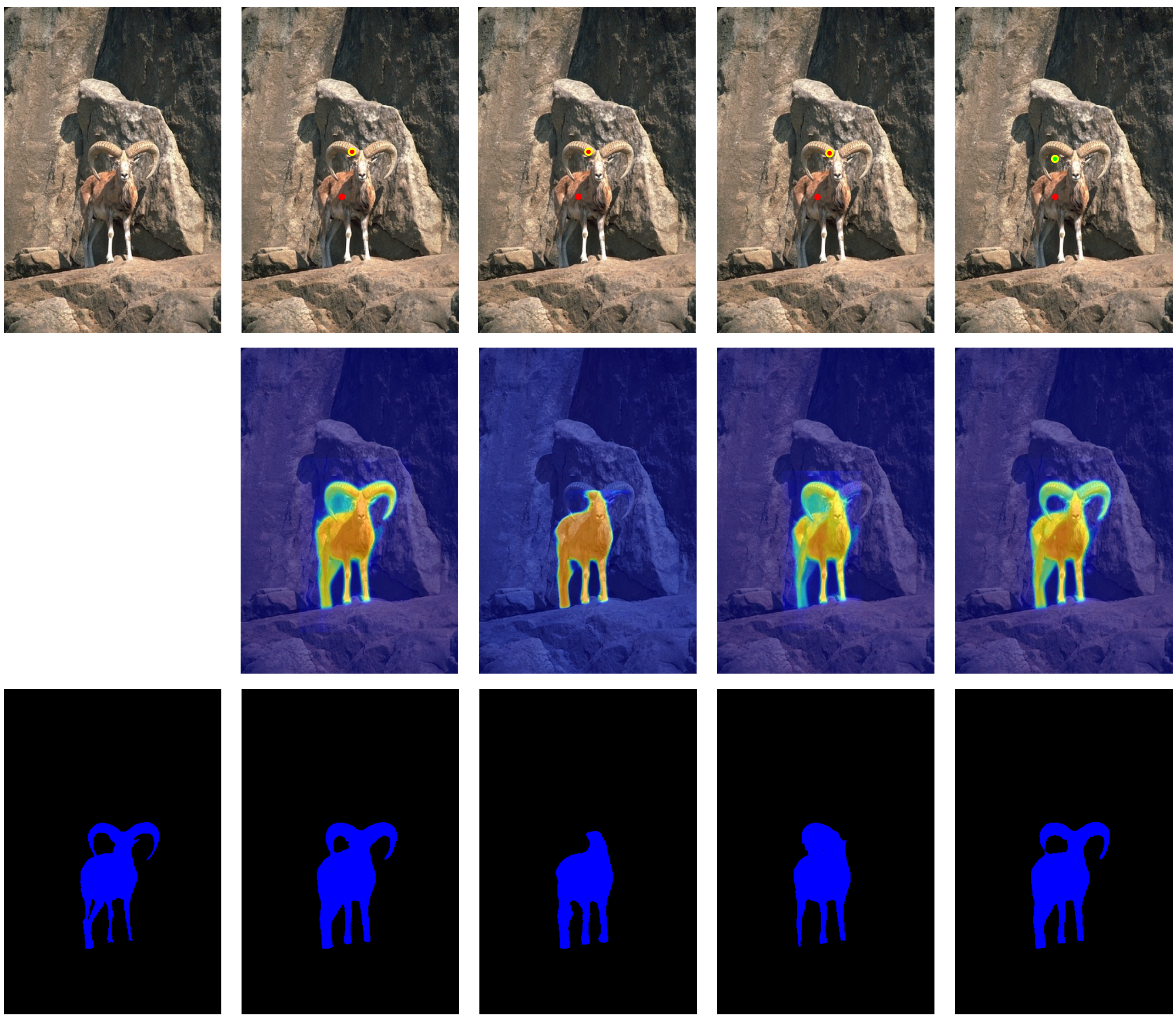

Image & GT Mask | RITM (IoU 93.42%) | FocusCut (IoU 77.61%) | FocalClick (IoU 76.09%) | GPCIS (IoU **94.03%**)

Figure S2. Visual comparisons of different competing methods on exemplar images from the GrabCut dataset.

[5] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors.

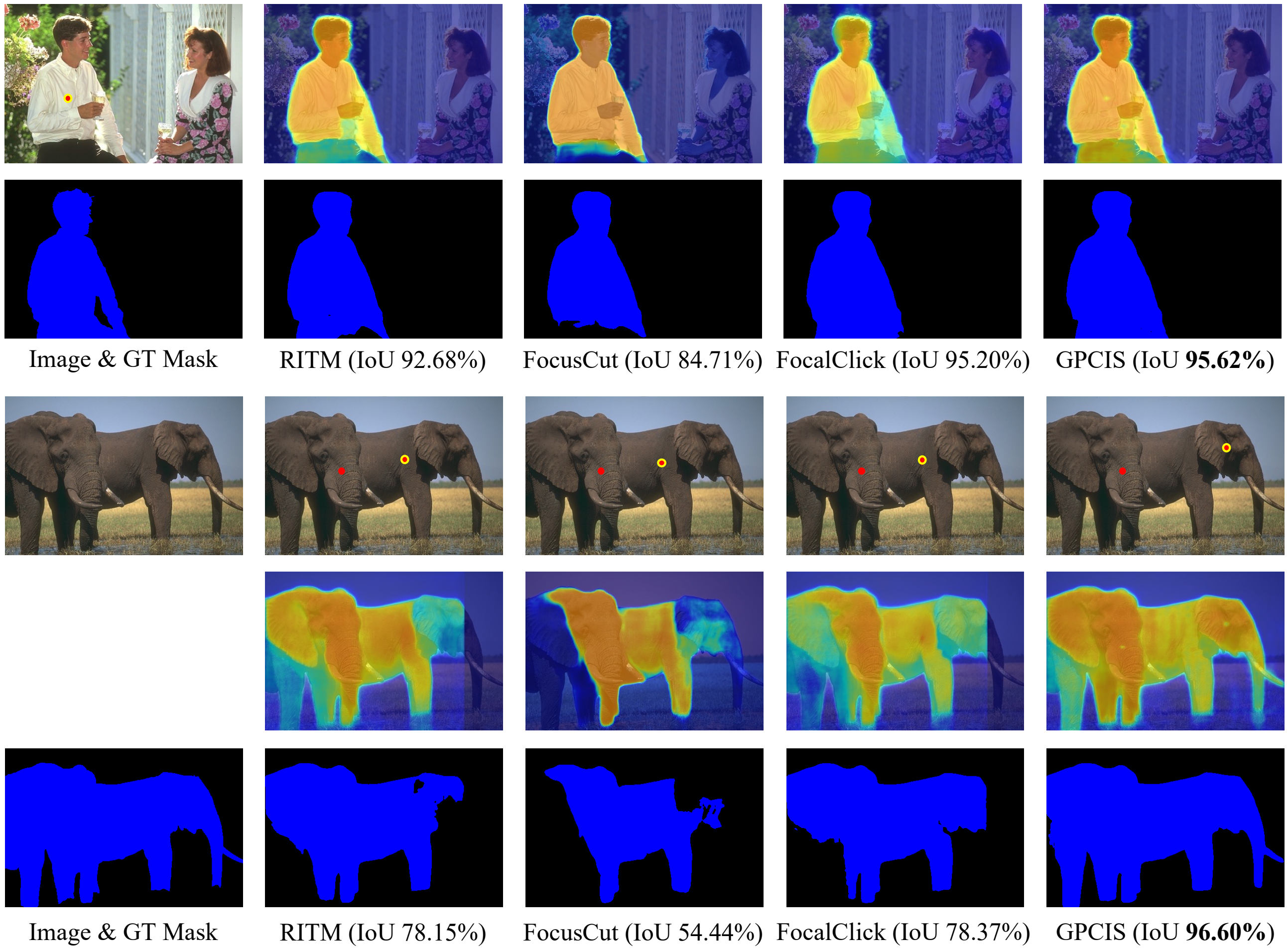

Figure S3. Visual comparisons of different competing methods on exemplar images from the Berkeley dataset.

In *Proceedings of IEEE International Conference on Computer Vision*, pages 991–998, 2011. 6

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 770–778, 2016. 1, 6

[7] Andre G Journel and Charles J Huijbregts. *Mining geostatistics*. Academic Press London, 1976. 5

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[9] Zheng Lin, Zheng-Peng Duan, Zhao Zhang, Chun-Le Guo, and Ming-Ming Cheng. FocusCut: Diving into a focus view in interactive segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2637–2646, 2022. 6

[10] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. In *British Machine Vision Conference*, 2018. 1

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

[12] 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/CVF International Conference on Computer Vision*, pages 724–732, 2016. 6

[13] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007. 5

[14] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. 6

[15] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-BRS: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8623–8632, 2020. 1, 6

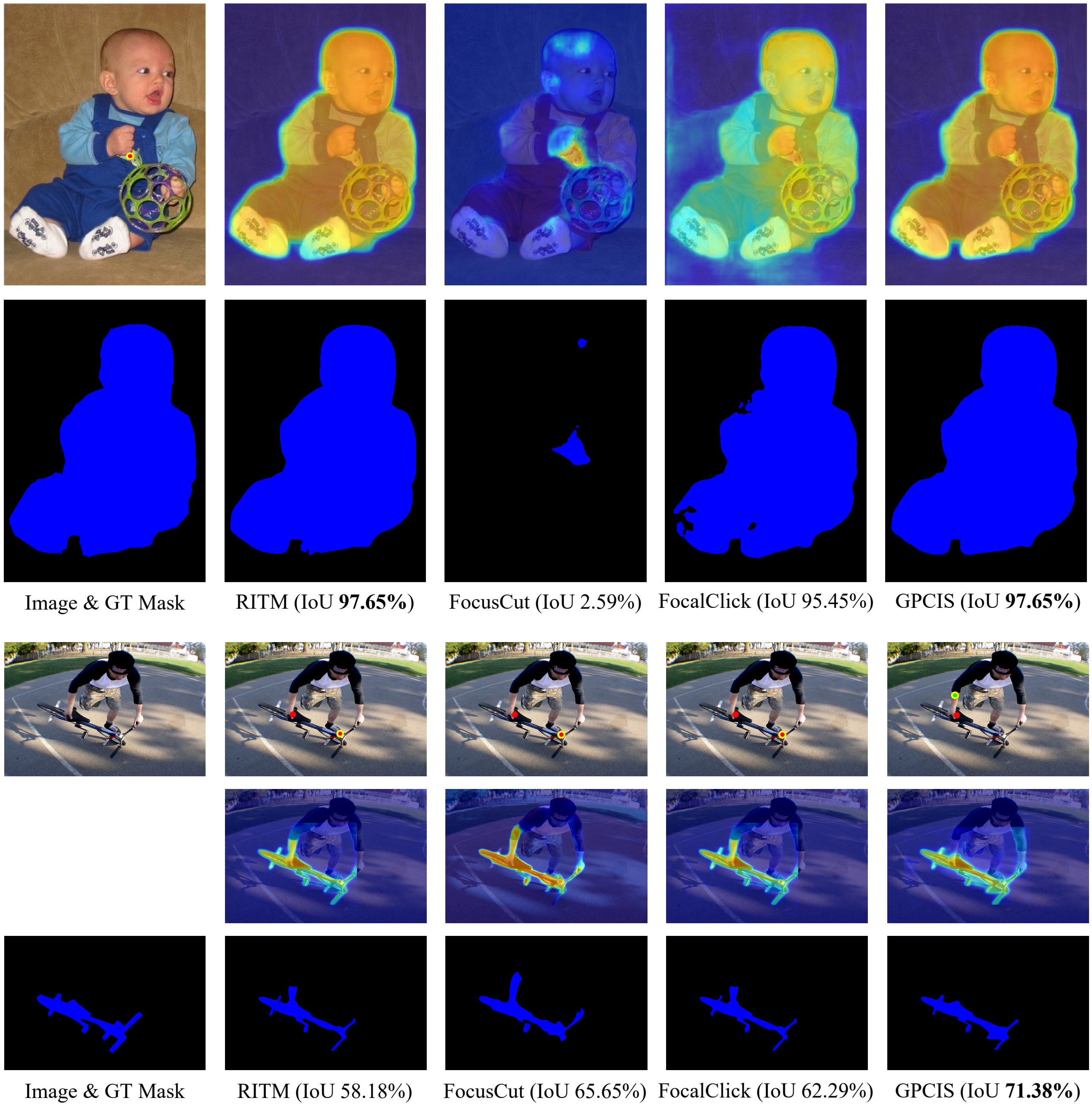

Figure S4. Visual comparisons of different competing methods on exemplar images from the SBD dataset.

[16] Konstantin Sofiiuk, Ilia A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. *arXiv preprint arXiv:2102.06583*, 2021. 1, 6

[17] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3349–3364, 2020. 1, 6

[18] James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*, pages 10292–10302. PMLR, 2020. 4, 5

[19] James T Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Pathwise conditioning of Gaussian processes. *Journal of Machine Learning Research*, 22:105–1, 2021. 4, 5

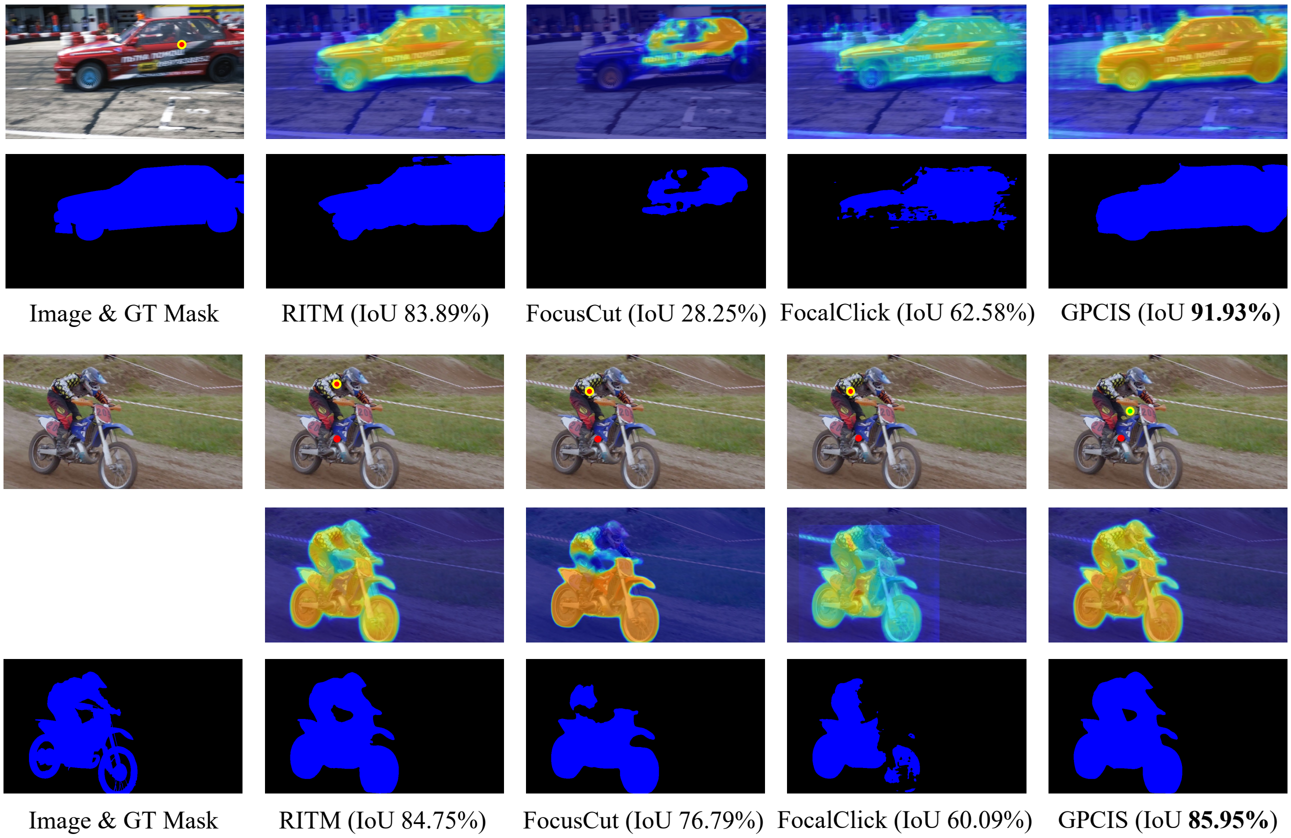

Figure S5. Visual comparisons of different competing methods on exemplar images from the DAVIS dataset.

[20] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1, 6