ALPHACLICK: SCALING INTERACTIVE MEDICAL IMAGE SEGMENTATION AT TEST-TIME VIA MONTE CARLO TREE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Powered by large-scale training data, promptable segmentation models such as the Segment Anything Model (SAM) have shown impressive generalization in medical image segmentation, where a few corrective clicks can turn a coarse mask into an accurate one. However, these clicks still have to be provided by experts, which is costly and limits the scalability of interactive segmentation in clinical practice. Existing automatic prompting methods, built on heuristics or reinforcement learning (RL) policies, decide clicks without looking ahead and cannot turn additional test-time computation into better segmentation. In this paper, we propose AlphaClick, a novel RL framework that scales interactive medical image segmentation at test time through Monte Carlo tree search (MCTS). Specifically, AlphaClick formulates corrective clicking as a sequential decision process and trains a policy-value network with the Dice improvement of each click as the reward. The policy proposes promising clicks, and the value predicts the final segmentation quality reachable from the current state. At inference, MCTS uses the policy to expand candidate clicks and the value to evaluate simulated click sequences, so that more simulations yield better click decisions without any ground truth. Extensive experiments on 80 public datasets covering seven imaging modalities show that AlphaClick outperforms existing automatic prompting methods by a large margin, reaching a Dice of 0.69 after ten corrective clicks compared with 0.43 for the strongest baseline, and that its accuracy improves steadily as the search grows from one to sixteen simulations per click. Results on three segmentation backbones and twelve unseen datasets further confirm its generality. In conclusion, AlphaClick makes test-time computation a new dimension for improving interactive segmentation and provides a promising solution for reducing the annotation burden in medical imaging. The code will be made publicly available.

1 INTRODUCTION

Segmenting anatomical structures and lesions is a fundamental step in medical image analysis. Training a dedicated network for each task (Ronneberger et al., 2015; Isensee et al., 2021) requires many expert annotations and transfers poorly across modalities, devices, and targets (Tajbakhsh et al., 2020). Interactive segmentation reduces this burden by letting a user correct the prediction with a few clicks (Xu et al., 2016; Sofiiuk et al., 2022; Liu et al., 2023), and has long been studied for medical images (Wang et al., 2019; Luo et al., 2021).

Powered by large-scale training data, promptable foundation models such as the Segment Anything Model (SAM) (Kirillov et al., 2023) and SAM 2 (Ravi et al., 2025) generalize to images and targets unseen during training. Medical variants such as MedSAM (Ma et al., 2024), SAM-Med2D (Cheng et al., 2023), MedSAM2 (Ma et al., 2025), and IMIS-Net (Cheng et al., 2025) bring this capability to clinical imaging, where a few well-placed corrective clicks can turn a coarse mask into an accurate one. However, the first prediction of these models on medical images is often far from the target (Mazurowski et al., 2023; Huang et al., 2024c), and the corrective clicks still have to be placed by an expert, one after another. Consequently, placing corrective clicks automatically, without ground truth and without a human at test time, is crucial for bringing promptable segmentation to the volume of clinical data.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Several recent efforts have explored automatic prompting for SAM. One line of work selects clicks with heuristics, such as the uncertainty of the current prediction (Lewis & Gale, 1994), or simulates clicks and aggregates the resulting masks, as in SimSAM (Towle et al., 2024). Another line trains reinforcement learning (RL) agents. AlignSAM (Huang et al., 2024a) learns an actor-critic policy that chooses prompt positions for a frozen SAM, while the Plug-and-Play Point Prompt Optimizer (PPO) (Liu et al., 2025) and Point Prompt Defender (PPD) (Liu et al., 2026) refine a set of point prompts on a feature graph with deep Q-learning. These methods automate prompting, but each decides its clicks from the current state without looking ahead at the masks that later clicks will produce, even though the value of a click depends on the clicks that follow it. As a result, existing methods struggle to improve the initial mask on medical images, and none of them is designed to turn additional test-time computation into better segmentation, as the right panel of Figure 1 shows.

We observe that choosing corrective clicks is a planning problem of the kind that game-playing agents solve well. The promptable model computes the mask that results from any click, so a planner can try click sequences before committing to one. What is missing is a way to judge where a sequence ends without the ground truth. In AlphaGo and its successors, a learned value network plays this role and lets tree search turn more simulations into stronger decisions (Silver et al., 2016; 2017; 2018; Schrittwieser et al., 2020). The same principle underlies the recent test-time scaling of language models (Yao et al., 2023; Snell et al., 2025). Lookahead search with a learned value has not yet been brought to interactive segmentation.

In this paper, we propose AlphaClick, a novel RL framework that scales interactive medical image segmentation at test time through Monte Carlo tree search (MCTS). As shown in Figure 1, unlike existing methods that decide clicks from the current state alone, AlphaClick plans each click by searching over click sequences. Specifically, AlphaClick formulates corrective clicking as a sequential decision process whose reward is the Dice improvement of each click. A policy-value network, trained on trajectories guided by the ground truth of training images, proposes promising clicks and predicts the final segmentation quality reachable from the current state. At test time, MCTS uses the policy to expand candidate clicks and the value to evaluate simulated click sequences, so more simulations yield better clicks without any ground truth. On 80 public datasets covering seven imaging modalities, AlphaClick raises the Dice after ten corrective clicks from 0.40 to 0.69, while the strongest existing method reaches 0.43, and its accuracy grows steadily from one to sixteen simulations per click on three segmentation backbones. Our main contributions are summarized as follows.

- We propose AlphaClick, an RL framework that formulates corrective clicking as a sequential decision process and plans every click with Monte Carlo tree search guided by a policy-value network, without ground truth or a human at test time.
- We show that AlphaClick converts test-time computation into accuracy on every segmentation backbone we test, and that a more accurate value estimate extends this scaling further.
- Extensive experiments on 80 public datasets across seven imaging modalities, including datasets unseen in training, show that AlphaClick outperforms existing automatic prompting methods by a large margin.

2 Related Work

Interactive and promptable segmentation. Early interactive methods turned user strokes into constraints of an energy minimization, as in graph cuts (Boykov & Jolly, 2001), GrabCut (Rother et al., 2004), and random walks (Grady, 2006). Deep interactive segmentation encodes clicks as additional input channels (Xu et al., 2016) and was improved by inference-time refinement (Sofiiuk et al., 2020), previous-mask feedback (Sofiiuk et al., 2022), local refinement (Chen et al., 2022), and plain vision transformers (Liu et al., 2023). In medical imaging, DeepIGeoS (Wang et al., 2019) and MIDeepSeg (Luo et al., 2021) encode user interactions as geodesic distance maps. SAM (Kirillov et al., 2023) made promptable segmentation a general capability, later improved in mask quality (Ke et al., 2023) and extended to videos (Ravi et al., 2025), and medical counterparts include MedSAM (Ma et al., 2024), SAM-Med2D (Cheng et al., 2023), SAM-Med3D (Wang et al., 2025a), MedSAM2 (Ma et al., 2025), and IMIS-Net with its IMed-361M benchmark (Cheng et al., 2025). These models respond to clicks and leave the choice of clicks to the user, which is the decision AlphaClick makes.

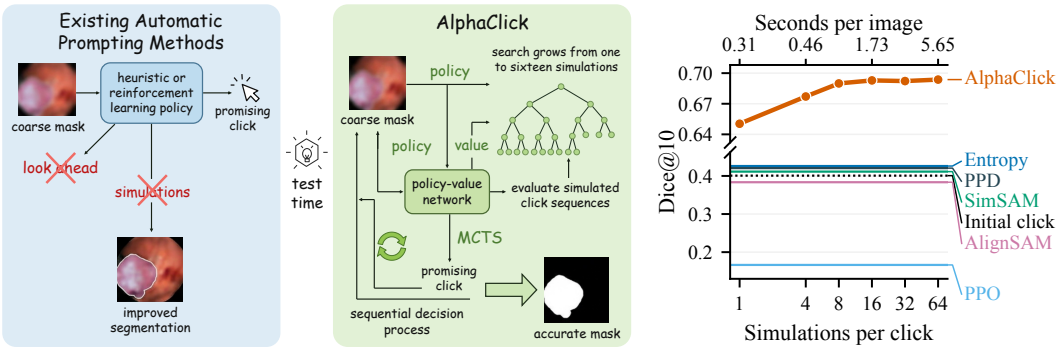

Figure 1: Left, existing methods choose clicks from the current mask, whereas AlphaClick searches over click sequences with a policy-value network. Right, Dice after ten corrective clicks (Dice@10) on 80 datasets. AlphaClick improves with more simulations per click (top axis, seconds per image), while the baselines stay close to the initial click.

Automatic prompting and reinforcement learning for segmentation. Choosing where to click was cast as a sequential decision problem before promptable models. SeedNet (Song et al., 2018) adds seed points with a deep Q-network, and IteR-MRL (Liao et al., 2020) refines 3D medical segmentations with multi-agent RL. For SAM, one family generates prompts without training, from a reference example as in PerSAM (Zhang et al., 2024) and Matcher (Liu et al., 2024), from the uncertainty of the current prediction (Lewis & Gale, 1994; Settles, 2009), or by simulating clicks and aggregating the resulting masks as in SimSAM (Towle et al., 2024). Another family learns prompting policies with RL. AlignSAM (Huang et al., 2024a) trains an actor-critic agent that supplies prompts to a frozen SAM, PPO (Liu et al., 2025) refines point prompts on a graph built from feature and physical distances, and PPD (Liu et al., 2026) trains an attacker and a defender with deep Q-networks and keeps the defender at inference to prune disruptive prompts. In medical imaging, RL has also been used to choose the form of the next prompt and when to stop (Shen et al., 2023; Huang et al., 2024b) and to select prompt points on lesion datasets (Wang et al., 2025b). SegAgent (Zhu et al., 2025) lets a multimodal language model emit clicks and ranks sampled candidates by a process reward on the next mask, a one-step selection learned from human annotations on natural images, whereas AlphaClick backs up values of multi-step click sequences in a search tree and improves with more simulations.

Planning and test-time computation. Monte Carlo tree search grows a tree from simulated trajectories and balances exploration with bandit rules (Coulom, 2006; Kocsis & Szepesvári, 2006; Browne et al., 2012). AlphaGo (Silver et al., 2016) combined it with policy and value networks, AlphaGo Zero and AlphaZero (Silver et al., 2017; 2018) trained a single policy-value network whose predictions guide the search, and MuZero (Schrittwieser et al., 2020) learned the dynamics model as well. Gumbel AlphaZero (Danihelka et al., 2022) replaces the heuristic root selection with a policy improvement step that remains effective with few simulations. In language models, more samples, tree-structured reasoning, and search against learned reward models convert test-time computation into accuracy (Brown et al., 2024; Yao et al., 2023; Hao et al., 2023; Snell et al., 2025). In segmentation, SAM2Long (Ding et al., 2025) searches over memory pathways of SAM 2 to reduce error accumulation in long videos. AlphaClick brings value-guided MCTS to corrective clicking, where the promptable model computes the transitions and a learned value replaces the missing ground truth.

3 Method

AlphaClick places corrective clicks on a promptable segmentation model after a single initial click, with no ground truth and no human in the loop (Figure 2). Corrective clicking is formulated as a Markov decision process (Section 3.1), a policy-value network proposes clicks and predicts the reachable segmentation quality (Section 3.2), it is trained on ground-truth-guided trajectories (Section 3.3), and Monte Carlo tree search plans every click at test time (Section 3.4).

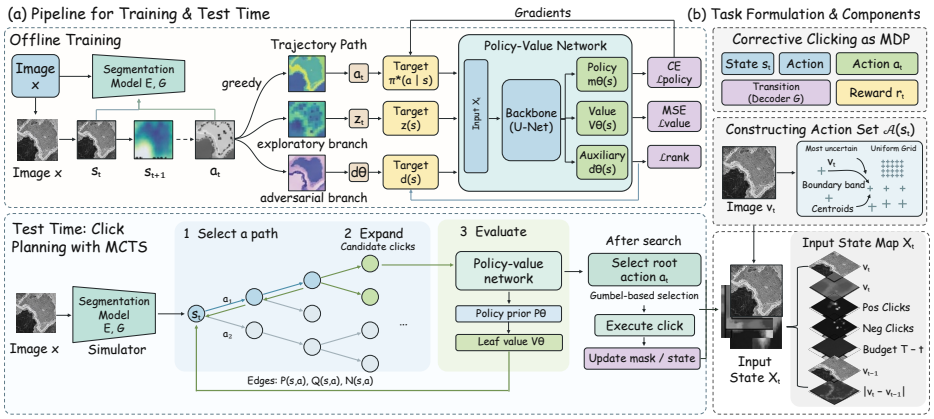

Figure 2: Overview of AlphaClick. Offline, ground-truth-guided trajectories provide policy, value, and current-Dice targets. At test time, Monte Carlo tree search expands candidates with the policy, decodes each new node, evaluates it with the value network, and backs up the value before committing a click.

3.1 Corrective clicking as a sequential decision process

A click changes the mask that every later click has to correct, so whether a click is worth placing depends on the clicks that follow it. We formulate the task as a sequential decision process so that a planner can compare sequences of clicks before committing to one.

Let $\boldsymbol{x}$ be an image and $\boldsymbol{y} \in \{0,1\}^{H\times W}$ the mask of the target structure, which is available only for training images. A promptable segmentation model consists of an image encoder E and a prompt-conditioned mask decoder G (Kirillov et al., 2023). The image embedding $\boldsymbol{e} = E(\boldsymbol{x})$ is computed once per image, and every transition below reuses it. A click $c = (u, v, l)$ has a position (u, v) and a polarity $l \in \{+1, -1\}$ that marks foreground or background. Given a click set C, the decoder returns a probability map $\boldsymbol{p} = \mathrm{sigmoid}(G(\boldsymbol{e}, C))$ and the binary mask $\boldsymbol{m} = \mathbb{1}[\boldsymbol{p} \geq 0.5]$. Following the standard click simulation protocol (Xu et al., 2016; Sofiiuk et al., 2022), the initial click c_0 is a positive click at the center of the largest connected component of $\boldsymbol{y}$. It specifies the structure to segment and is shared by all methods, and none of the T corrective clicks that follow may use $\boldsymbol{y}$.

After t corrective clicks the state is $s_t = (\boldsymbol{e}, \boldsymbol{p}_t, \boldsymbol{p}_{t-1}, C_t, T - t)$, where $C_t = \{c_0, \ldots, c_t\}$ and the previous probability map $\boldsymbol{p}_{t-1}$ records how the last click changed the mask. An action a_t is a click from a candidate set $\mathcal{A}(s_t)$ defined below. The transition is deterministic and is computed by the decoder, $C_{t+1} = C_t \cup \{a_t\}$ and $\boldsymbol{p}_{t+1} = \mathrm{sigmoid}(G(\boldsymbol{e}, C_{t+1}))$. The reward of a click is the Dice improvement it produces,

$$r_t = \mathrm{Dice}(\boldsymbol{m}_{t+1}, \boldsymbol{y}) - \mathrm{Dice}(\boldsymbol{m}_t, \boldsymbol{y}), \tag{1}$$

so the return from state s_t telescopes to the Dice after the last click,

$$R_t = \sum_{k=t}^{T-1} r_k = \mathrm{Dice}(\boldsymbol{m}_T, \boldsymbol{y}) - \mathrm{Dice}(\boldsymbol{m}_t, \boldsymbol{y}). \tag{2}$$

Maximizing the return is thus the same as maximizing the final segmentation quality. At test time $\boldsymbol{y}$ is unknown, so AlphaClick learns to estimate the return and plans with this estimate.

Since one action per pixel and polarity is far too many for search, we build a compact candidate set from the current probability map alone. Positive candidates lie outside the mask, where foreground may be missing, and negative candidates lie inside it, where the mask may leak into the background. They are drawn from the most uncertain locations under $\boldsymbol{u}_t = 1 - 2\,|\boldsymbol{p}_t - 0.5|$, a uniform grid, the boundary band of the mask, and small disconnected mask components (Appendix A). Polarity follows from the side of the mask, so $\mathcal{A}(s_t)$, with about 55 candidates, never uses $\boldsymbol{y}$. The action set can also contain a stop action that ends the episode, which we use for adaptive stopping (Section 4.5).

3.2 POLICY-VALUE NETWORK

Search needs two estimates at every state it visits, which candidates deserve attention and how good the state is without the ground truth. A single network provides both, in the spirit of AlphaGo Zero (Silver et al., 2017). Both can be read from the segmentation model's own output, where uncertain regions point to likely errors, the change caused by the last click shows where the mask is unstable, and earlier clicks show what has been corrected.

The network therefore takes a state map $\boldsymbol{X}_t \in \mathbb{R}^{7\times256\times256}$ that stacks the probability map $\boldsymbol{p}_t$, the uncertainty $\boldsymbol{u}_t$, Gaussian maps of the positive and of the negative clicks, the remaining budget $(T-t)/T$ as a constant map, the previous probability map $\boldsymbol{p}_{t-1}$, and the change $|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}|$. The image embedding $\boldsymbol{e}$ is projected to 32 channels by a 1×1 convolution, resized to 256×256, and concatenated with $\boldsymbol{X}_t$. A four-level U-shaped convolutional network (Ronneberger et al., 2015) processes the input from 256^2 down to 32^2 with group normalization (Wu & He, 2018) and SiLU activations (Elfwing et al., 2018), and its decoder returns to full resolution through skip connections (Appendix A). Three heads share this backbone. The policy head outputs a dense logit map $\boldsymbol{L}_\theta(s) \in \mathbb{R}^{256\times256}$, and a stop logit $\ell_\theta^{\text{stop}}(s)$ is computed from the pooled bottleneck features. The prior over candidates reads the logit map at the candidate locations,

$$P_\theta(a \mid s) = \frac{\exp\big(\boldsymbol{L}_\theta(s)[a]\big)}{\sum_{a'\in\mathcal{A}(s)} \exp\big(\boldsymbol{L}_\theta(s)[a']\big) + \exp\big(\ell_\theta^{\text{stop}}(s)\big)}. \tag{3}$$

The value head $V_\theta(s) \in [0,1]$ applies a two-layer perceptron and a sigmoid to the globally pooled bottleneck features and predicts the final Dice $\text{Dice}(\boldsymbol{m}_T, \boldsymbol{y})$ reachable from s. By Equation 2, this is the return R_t plus the Dice of the current mask, which is itself unknown at test time. An auxiliary head of the same form predicts the current Dice, $\hat{d}_\theta(s_t) \approx \text{Dice}(\boldsymbol{m}_t, \boldsymbol{y})$, and scores the stop action when adaptive stopping is enabled.

The dense map handles any number of candidates, and only the 1×1 input projection of the 3.38M-parameter network depends on the segmentation model.

3.3 LEARNING FROM GROUND-TRUTH-GUIDED TRAJECTORIES

A value network helps search only if it can tell apart states that differ by a single click, including poor states that search will visit. The training data therefore need several continuations per image, good and bad, each labeled with the final Dice it reaches. On training images, every candidate can be scored exactly by the Dice gain it produces, since each transition costs one decoder call, and these gains drive a strong behavior policy and dense targets.

For each training image we generate seven trajectories within the budget of T clicks. The main trajectory decodes all candidates at every state, takes the one with the largest Dice gain, and ends with the stop action once no candidate improves the Dice. Three exploratory branches leave it at a random step with a random non-greedy candidate, and three adversarial branches leave it with a harmful click, a positive click deep in the background or a negative click deep inside the target. All branches then continue greedily. When the main trajectory stops, the state reached by forcing its best remaining click is also stored. Branch states differ from their siblings on the main trajectory by one click and usually end lower, which gives the value head the within-image contrast that search relies on.

Each state s on a trajectory is labeled with three targets. Let $\Delta(a) = \text{Dice}(\boldsymbol{m}^a, \boldsymbol{y}) - \text{Dice}(\boldsymbol{m}, \boldsymbol{y})$ be the Dice gain of candidate a, where $\boldsymbol{m}^a$ is the mask after adding a. The policy target is a softmax over these gains,

$$\pi^*(a \mid s) \propto \exp\big(\Delta(a)/\tau\big), \qquad a \in \mathcal{A}(s) \cup \{\text{stop}\}, \tag{4}$$

where the stop action has gain zero by definition and $\tau = 0.05$. The target spreads probability over every candidate that helps and puts most of the mass on the stop action when every candidate hurts. The value target $z(s)$ is the Dice at the end of the trajectory that contains s, which is a Monte Carlo sample of the return under the behavior policy plus the current Dice. The auxiliary target is the current Dice $d(s) = \text{Dice}(\boldsymbol{m}, \boldsymbol{y})$. The training loss is

$$\mathcal{L} = -\sum_a \pi^*(a \mid s) \log P_\theta(a \mid s) + \big(V_\theta(s) - z(s)\big)^2 + \lambda\, \mathcal{L}_{\text{rank}} + \big(\hat{d}_\theta(s) - d(s)\big)^2. \tag{5}$$

The ranking term trains the ability that search uses most, which is ordering sibling states. For pairs of states (s_i, s_j) from the same image with the same number of clicks and $|z(s_i) - z(s_j)| \geq \epsilon$, it applies $\mathrm{softplus}\big(-\kappa \,\mathrm{sign}(z(s_i) - z(s_j))\,(V_\theta(s_i) - V_\theta(s_j))\big)$ and averages over pairs, with $\kappa = 10$, $\epsilon = 0.01$, and $\lambda = 1$.

The trajectories cover $35{,}551$ training images from 80 datasets and contain about 1.27 million states.

3.4 Test-time Monte Carlo tree search

A policy commits to each click without seeing the masks it leads to, whereas tree search tries click sequences first. The decoder computes every transition exactly, the value network estimates where a sequence ends, and more simulations explore more sequences, so the committed click improves as the search budget grows.

Each node of the search tree corresponds to a click set together with its decoded mask, and each edge (s, a) stores a prior $P(s, a) = P_\theta(a \mid s)$, a visit count $N(s, a)$, and a mean value $Q(s, a)$. When a node is expanded, it keeps the $m = 4$ candidates with the highest prior and renormalizes their priors. A simulation descends from the root, choosing below the root the action

$$\arg\max_a \; Q(s, a) + c_{\mathrm{puct}}\, P(s, a)\, \frac{\sqrt{\sum_b N(s, b)}}{1 + N(s, a)}, \tag{6}$$

with $c_{\mathrm{puct}} = 1.5$ (Rosin, 2011; Silver et al., 2018), until it reaches an unexpanded node. That node is decoded with one decoder call, the network evaluates it, and its value V_θ is backed up along the path. As in AlphaZero, there are no rollouts to the end of the episode. Leaves are processed in batches with virtual loss (Chaslot et al., 2008), so decoder calls and network passes run in parallel on the GPU.

At the root, simulations are allocated as in Gumbel AlphaZero (Danihelka et al., 2022), which improves the policy even with a handful of simulations. It draws $g(a) \sim \mathrm{Gumbel}(0, 1)$ for every root action, keeps the m actions with the largest $g(a) + \log P(s_0, a)$, and splits the n simulations of the click among them by sequential halving (Karnin et al., 2013). Each round keeps the half with the largest score $g(a) + \log P(s_0, a) + \sigma(\tilde{Q}(a))$, and the committed click is the remaining action with the largest score, where

$$\sigma(q) = \big(c_{\mathrm{visit}} + \max_b N(s_0, b)\big)\, c_{\mathrm{scale}}\, q \tag{7}$$

and $\tilde{Q}$ holds the completed values of the root actions rescaled to $[0, 1]$, which keeps the small Dice differences among siblings on the scale of the prior logits (Appendix A).

After a click is committed, its subtree becomes the new root and keeps its statistics, and the next click again runs n new simulations. We place a fixed number of T clicks so that search budgets are compared at equal numbers of clicks. Each simulation adds one decoder call and one network pass, so the cost of a click grows linearly with n. For SAM ViT-B a batched decoder call takes 1.46 ms, about $1/70$ of the image encoder, which makes n a direct and affordable control of test-time computation (Algorithm 1).

4 Experiments

4.1 Experimental setup

Datasets and protocol. We use all 80 datasets of the IMed-361M collection (Cheng et al., 2025) that provide an official test split, covering magnetic resonance (MR), computed tomography (CT), endoscopy, fundus photography, dermoscopy, ultrasound, and X-ray. Up to 200 test targets per dataset are drawn in a fixed random order, $12{,}448$ in total, and training uses the official training splits with up to 500 images per dataset. After the shared initial click, each method places ten corrective clicks without any access to the ground truth, including click polarity. We report Dice@k, the Dice after k corrective clicks counted from the second click, so Dice@10 is the Dice after the initial click and all ten corrective clicks. Dice is averaged over the targets of each dataset and then over datasets, and we also report the percentage of targets whose final Dice reaches 0.75 (Appendix B).

Table 1: Main results on 80 datasets (12,448 targets). All methods share the initial click and place ten corrective clicks without ground truth. Dice@k is the macro Dice after k corrective clicks, counted from the second click, and Reached is the percentage of targets with a final Dice of at least 0.75. PPO and PPD place all clicks at once. Gray rows use the ground truth at test time.

Method	Trained with GT	Dice@2	Dice@4	Dice@10	Reached (%)
Initial click only	–	0.4008	0.4008	0.4008	25.6
Random	✗	0.3149	0.2615	0.1751	5.5
Entropy	✗	0.4274	0.4269	0.4260	26.8
SimSAM	✗	0.4193	0.4181	0.4114	25.0
AlignSAM	✓	0.4165	0.4182	0.3835	19.0
PPO[†]	✓	–	–	0.1664	3.9
PPD	✓	–	–	0.4205	27.0
AlphaClick, 1 simulation	✓	0.6427	0.6596	0.6504	46.8
AlphaClick, 16 simulations	✓	0.6621	0.6909	0.6929	53.0
AlphaClick, 64 simulations	✓	**0.6648**	**0.6915**	**0.6937**	**53.4**
Candidate ceiling	–	0.7609	0.8154	0.8306	78.3
Oracle clicks	–	0.7002	0.8032	0.8833	87.9

[†]PPO builds no prompt graph for 24 of the targets; its entries average the remaining 12,424.

Backbones. The main segmentation model is SAM ViT-B (Kirillov et al., 2023), and we also test SAM2.1 with a Hiera-L encoder (Ravi et al., 2025) and MedSAM2 (Ma et al., 2025), a SAM2 model fine-tuned on medical images. All of them stay frozen.

Baselines. Random picks a candidate uniformly, and Entropy clicks the candidate of highest predictive entropy (Lewis & Gale, 1994; Settles, 2009), both from the candidate set of AlphaClick. SimSAM (Towle et al., 2024) simulates clicks and aggregates the resulting masks without training. AlignSAM (Huang et al., 2024a) trains an actor-critic agent that selects prompt positions. PPO (Liu et al., 2025) and PPD (Liu et al., 2026) refine a set of point prompts on a feature graph with deep reinforcement learning and decode the set once, so they are evaluated after the last click only, with the same budget of ten corrective clicks. All baselines use the same frozen segmentation model and no test-time ground truth, AlignSAM and PPD are trained on our training splits, and PPO refines its prompts on each test image (Appendix C). As references that use the ground truth, oracle clicks go to the center of the largest error region, and the candidate ceiling takes the candidate with the largest true Dice gain at every step.

Implementation. The policy-value network is trained for 20 epochs with AdamW (Loshchilov & Hutter, 2019) at a learning rate of 3×10^{-4} (Appendix A). At test time we use $m = 4$ and vary the number of simulations per click n. Every configuration is run once, and differences are assessed with a paired bootstrap over datasets with 2,000 resamples (Efron & Tibshirani, 1993).

4.2 Main results

With ten corrective clicks (Table 1), AlphaClick raises the macro Dice from 0.4008 after the initial click to 0.6937 with 64 simulations per click, while the strongest baseline, Entropy, reaches 0.4260. The margin of 0.268 holds with a bootstrap interval of $[0.243, 0.293]$. AlphaClick brings more than half of the targets above a Dice of 0.75, about twice the share of any baseline, and reaches 83.5% of the candidate ceiling, which uses the ground truth to choose every click. Every search budget of AlphaClick, from a single simulation per click upward, exceeds every baseline by a wide margin.

The gap between AlphaClick and the baselines opens early (Figure 3(a)). The baselines stay close to the initial mask or make it worse, whereas AlphaClick improves it steeply over the first few clicks and keeps the gain as clicks accumulate. In the cases of Figure 4, the baselines leave the initial error in place or add clicks that cause leakage, whereas AlphaClick places positive clicks on missed parts of the target and negative clicks on leaked regions.

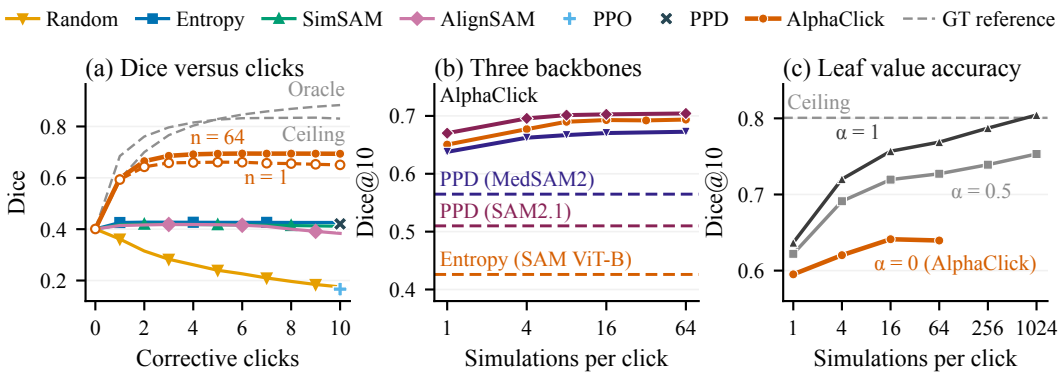

Figure 3: (a) Dice versus the number of corrective clicks on 80 datasets. PPO and PPD appear at the last click only, and oracle and ceiling use the ground truth. (b) Scaling on three backbones, with dashed lines for the strongest baseline on each backbone. (c) Diagnostic on 24 datasets in which each leaf value is mixed with the true Dice of the leaf mask at weight α. $\alpha = 0$ is AlphaClick at the budgets of the main experiments, and $\alpha > 0$ uses the ground truth.

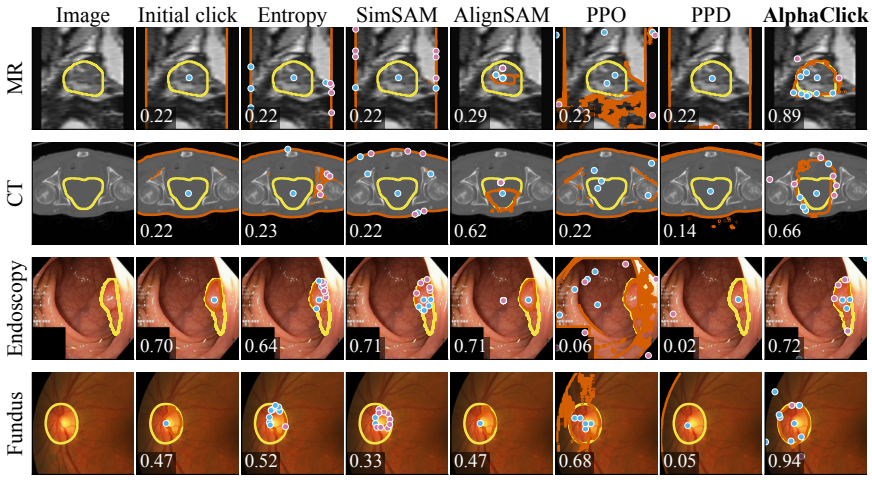

Figure 4: Results after ten corrective clicks for the median case of each modality under the mean Dice of the five baselines ($n = 16$). Yellow and orange contours mark the ground truth and the prediction, blue and pink dots positive and negative clicks, and numbers the Dice (more in Appendix E).

4.3 Test-time scaling

AlphaClick converts test-time computation into accuracy. On SAM ViT-B, the Dice after the last click rises from 0.6504 with one simulation per click to 0.6771 with four and 0.6929 with sixteen (Figure 1), and both steps are significant under the paired bootstrap. Sixteen simulations run at 1.7 seconds per image on one H100 GPU, fast enough for interactive use, while the baselines have no such control.

Table 2 reports the wall-clock time per image for ten corrective clicks, measured on one H100 GPU with a single process on 150 test images. The image encoder runs once per image, and the time grows with the number of simulations as expected from Section 3.4.

The same behavior appears on every backbone (Figure 3(b)). On SAM ViT-B, SAM2.1, and MedSAM2, each with its own policy-value network, the Dice rises from one to sixteen simulations, and every point lies far above the strongest baseline on the same backbone.

The accuracy of the value estimate sets the height of the curve (Figure 3(c)). In a diagnostic on 24 datasets, we mix each leaf value with the true Dice of its mask at weight α. As α grows, the curve rises at every budget and keeps rising with more simulations. With exact leaf values, the search

Table 2: Seconds per image for ten corrective clicks on SAM ViT-B, measured on one H100 GPU with a single process.

	Entropy	AlphaClick, simulations per click			
		1	4	16	64
Seconds per image	0.16	0.31	0.46	1.73	5.65

Table 3: Dice after the last click on three backbones, each with its own policy-value network. Initial is the Dice of the initial click.

				AlphaClick, n	
Backbone	Initial	Entropy	PPD	1	64
SAM ViT-B	0.4008	0.4260	0.4205	0.6504	**0.6937**
SAM2.1	0.4808	0.4768	0.5100	0.6702	**0.7043**
MedSAM2	0.5738	0.5342	0.5648	0.6379	**0.6726**

Table 4: Ablations with 16 simulations per click. Δ is the paired difference to the full model.

Variant	Dice@10	Δ
Full model	0.6929	–
No search (policy only)	0.6366	-0.0562
No tree reuse	0.6884	-0.0044
No history channels	0.6746	-0.0183
No negative candidates	0.3589	-0.3341

surpasses the candidate ceiling at 1024 simulations. The search itself has ample room to scale, and the benefit of more simulations grows with the accuracy of the value estimate, which makes a better value network a direct path to stronger test-time scaling.

4.4 Generality across backbones, modalities, and unseen datasets

On all three backbones, AlphaClick far exceeds the strongest baseline (Table 3). With SAM2.1 it reaches 0.7043, the highest Dice of any method without test-time ground truth in our experiments, and on MedSAM2, where the baselines fall below the initial click, it still adds about 0.10 Dice.

Split by imaging modality, AlphaClick leads the strongest baseline in every one of the seven modalities, with gains from 0.18 to 0.32 Dice (Table 7 in Appendix D). To test generalization to datasets never seen in training, we train the policy-value network on 68 datasets and evaluate it on the remaining 12, which span all seven modalities. AlphaClick reaches 0.6648 on them against 0.5420 for the strongest baseline and exceeds every baseline on 11 of the 12 datasets.

4.5 Ablation study

Table 4 removes one component at a time with 16 simulations per click. Committing the top-prior candidate, which is the policy without search, lowers the Dice by 0.056, so a substantial part of the result comes from planning. Tree reuse adds 0.004 at no extra cost, and the history channels, which show how the last click changed the mask, add 0.018. Negative candidates are essential, since without them the agent cannot remove leaked regions and the Dice drops to 0.3589. With the stop action enabled, AlphaClick also decides on its own when to end (Appendix D).

5 Conclusion

We presented AlphaClick, a reinforcement learning framework that formulates corrective clicking as a sequential decision process, learns a policy-value network from ground-truth-guided trajectories, and plans every click with Monte Carlo tree search, without ground truth or a human at test time. On 80 datasets covering seven modalities, it raises the Dice after ten corrective clicks from 0.40 to 0.69 where existing methods stay near the initial mask, and it improves with more simulations on all three backbones and generalizes to unseen datasets. Test-time computation thus becomes a new dimension for improving interactive segmentation, and more accurate value estimates will carry it further in reducing the annotation burden of medical imaging.

REFERENCES

Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 1, pp. 105–112, 2001. doi: 10.1109/iccv.2001.937505.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024. URL https://arxiv.org/abs/2407.21787.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/tciaig.2012.2186810.

Guillaume M. J.-B. Chaslot, Mark H. M. Winands, and H. Jaap van den Herik. Parallel Monte-Carlo tree search. In *Proceedings of the International Conference on Computers and Games (CG)*, pp. 60–71. Springer, 2008. doi: 10.1007/978-3-540-87608-3_6.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1290–1299, 2022. doi: 10.1109/cvpr52688.2022.00136.

Junlong Cheng, Jin Ye, Zhongying Deng, Jianpin Chen, Tianbin Li, Haoyu Wang, Yanzhou Su, Ziyan Huang, Jilong Chen, Lei Jiang, Hui Sun, Junjun He, Shaoting Zhang, Min Zhu, and Yu Qiao. SAM-Med2D. *arXiv preprint arXiv:2308.16184*, 2023. URL https://arxiv.org/abs/2308.16184.

Junlong Cheng, Bin Fu, Jin Ye, Guoan Wang, Tianbin Li, Haoyu Wang, Ruoyu Li, He Yao, Junren Cheng, Jinwen Li, Yanzhou Su, Min Zhu, and Junjun He. Interactive medical image segmentation: A benchmark dataset and baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20841–20851, 2025. doi: 10.1109/cvpr52734.2025.01941.

Rémi Coulom. Efficient selectivity and backup operators in Monte-Carlo tree search. In *Proceedings of the International Conference on Computers and Games (CG)*, pp. 72–83. Springer, 2006. doi: 10.1007/978-3-540-75538-8_7.

Ivo Danihelka, Arthur Guez, Julian Schrittwieser, and David Silver. Policy improvement by planning with Gumbel. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=bERaNdoegnO.

Shuangrui Ding, Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Yuwei Guo, Dahua Lin, and Jiaqi Wang. SAM2Long: Enhancing SAM 2 for long video segmentation with a training-free memory tree. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13614–13624, 2025. doi: 10.1109/iccv51701.2025.01264.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, New York, 1993. doi: 10.1007/978-1-4899-4541-9.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. doi: 10.1016/j.neunet.2017.12.012.

Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006. doi: 10.1109/tpami.2006.233.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8154–8173. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.507.

Duojun Huang, Xinyu Xiong, Jie Ma, Jichang Li, Zequn Jie, Lin Ma, and Guanbin Li. AlignSAM: Aligning segment anything model to open context via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3205–3215, 2024a. doi: 10.1109/cvpr52733.2024.00309.

Yifei Huang, Chuyun Shen, Wenhao Li, Xiangfeng Wang, Bo Jin, and Haibin Cai. Optimizing efficiency and effectiveness in sequential prompt strategy for SAM using reinforcement learning. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 478–488. Springer, 2024b. doi: 10.1007/978-3-031-72111-3_45.

Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, Sijing Liu, Haozhe Chi, Xindi Hu, Kejuan Yue, Lei Li, Vicente Grau, Deng-Ping Fan, Fajin Dong, and Dong Ni. Segment anything model for medical images? *Medical Image Analysis*, 92:103061, 2024c. doi: 10.1016/j.media.2023.103061.

Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021. doi: 10.1038/s41592-020-01008-z.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1238–1246. PMLR, 2013. URL https://proceedings.mlr.press/v28/karnin13.html.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 29914–29934, 2023. doi: 10.52202/075280-1303.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003, 2023. doi: 10.1109/iccv51070.2023.00371.

Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *Proceedings of the European Conference on Machine Learning (ECML)*, pp. 282–293. Springer, 2006. doi: 10.1007/11871842_29.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 3–12. Springer, 1994. doi: 10.1007/978-1-4471-2099-5_1.

Xuan Liao, Wenhao Li, Qisen Xu, Xiangfeng Wang, Bo Jin, Xiaoyun Zhang, Yanfeng Wang, and Ya Zhang. Iteratively-refined interactive 3D medical image segmentation with multi-agent reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9391–9399, 2020. doi: 10.1109/cvpr42600.2020.00941.

Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. SimpleClick: Interactive image segmentation with simple vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22233–22243, 2023. doi: 10.1109/iccv51070.2023.02037.

Xueyu Liu, Rui Wang, Yexin Lai, Guangze Shi, Feixue Shao, Fang Hao, Jianan Zhang, Jia Shen, Yongfei Wu, and Wen Zheng. Plug-and-play PPO: An adaptive point prompt optimizer making SAM greater. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4332–4342, 2025. doi: 10.1109/cvpr52734.2025.00409.

Xueyu Liu, Xiaoyi Zhang, Meilin Liu, Guangze Shi, Jia Shen, Yujie Wang, Cai Zhao, Ziyuan He, Yongfei Wu, Mingqiang Wei, and Yongle Chen. Attack for defense: Adversarial agents for point prompt optimization empowering segment anything model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6591–6600, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liu_Attack_for_Defense_Adversarial_Agents_for_Point_Prompt_Optimization_Empowering_CVPR_2026_paper.html.

Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=yzRXdhk2he.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Xiangde Luo, Guotai Wang, Tao Song, Jingyang Zhang, Michael Aertsen, Jan Deprest, Sebastien Ourselin, Tom Vercauteren, and Shaoting Zhang. MIDeepSeg: Minimally interactive segmentation of unseen objects from medical images using deep learning. *Medical Image Analysis*, 72: 102102, 2021. doi: 10.1016/j.media.2021.102102.

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024. doi: 10.1038/s41467-024-44824-z.

Jun Ma, Zongxin Yang, Sumin Kim, Bihui Chen, Mohammed Baharoon, Adibvafa Fallahpour, Reza Asakereh, Hongwei Lyu, and Bo Wang. MedSAM2: Segment anything in 3D medical images and videos. *arXiv preprint arXiv:2504.03600*, 2025. URL https://arxiv.org/abs/2504.03600.

Maciej A. Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: An experimental study. *Medical Image Analysis*, 89:102918, 2023. doi: 10.1016/j.media.2023.102918.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=Ha6RTeWMd0.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28.

Christopher D. Rosin. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230, 2011. doi: 10.1007/s10472-011-9258-6.

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. doi: 10.1145/1015706.1015720.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison, Department of Computer Sciences, 2009. URL http://digital.library.wisc.edu/1793/60660.

Chuyun Shen, Wenhao Li, Ya Zhang, Yanfeng Wang, and Xiangfeng Wang. Temporally-extended prompts optimization for SAM in interactive medical image segmentation. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 3550–3557, 2023. doi: 10.1109/bibm58861.2023.10385291.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. doi: 10.1038/nature16961.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017. doi: 10.1038/nature24270.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=2NlGyqNjns.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

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 (CVPR)*, pp. 8620–8629, 2020. doi: 10.1109/cvpr42600.2020.00865.

Konstantin Sofiiuk, Ilya A. Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 3141–3145, 2022. doi: 10.1109/icip46576.2022.9897365.

Gwangmo Song, Heesoo Myeong, and Kyoung Mu Lee. SeedNet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1760–1768, 2018. doi: 10.1109/cvpr.2018.00189.

Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey N. Chiang, Zhihao Wu, and Xiaowei Ding. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63:101693, 2020. doi: 10.1016/j.media.2020.101693.

Benjamin Towle, Xin Chen, and Ke Zhou. SimSAM: Zero-shot medical image segmentation via simulated interaction. In *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, 2024. doi: 10.1109/isbi56570.2024.10635227.

Guotai Wang, Maria A. Zuluaga, Wenqi Li, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Anna L. David, Jan Deprest, Sebastien Ourselin, and Tom Vercauteren. DeepIGeoS: A deep interactive geodesic framework for medical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1559–1572, 2019. doi: 10.1109/tpami.2018.2840695.

Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, Bin Fu, Shaoting Zhang, and Junjun He. SAM-Med3D: A vision foundation model for general-purpose segmentation on volumetric medical images. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):17599–17612, 2025a. doi: 10.1109/tnnls.2025.3586694.

Yuli Wang, Victoria Shi, Wen-Chi Hsu, Yuwei Dai, Sophie Yao, Zhusi Zhong, Zishu Zhang, Jing Wu, Aaron Maxwell, Scott Collins, Zhicheng Jiao, and Harrison X. Bai. Optimizing prompt strategies for SAM: advancing lesion segmentation across diverse medical imaging modalities. *Physics in Medicine & Biology*, 70(17):175014, 2025b. doi: 10.1088/1361-6560/adfc20.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19. Springer, 2018. doi: 10.1007/978-3-030-01261-8_1.

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 (CVPR)*, pp. 373–381, 2016. doi: 10.1109/cvpr.2016.47.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 11809–11822, 2023. doi: 10.52202/075280-0517.

Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. In *International Conference on Learning Representations (ICLR)*, 2024. URL `https://openreview.net/forum?id=6Gzkhoc6YS`.

Muzhi Zhu, Yuzhuo Tian, Hao Chen, Chunluan Zhou, Qingpei Guo, Yang Liu, Ming Yang, and Chunhua Shen. SegAgent: Exploring pixel understanding capabilities in MLLMs by imitating human annotator trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3686–3696, 2025. doi: 10.1109/cvpr52734.2025.00349.

A Implementation details

Candidate clicks. Candidates are computed on the 256×256 probability map returned by the mask decoder and mapped back to image coordinates. Each polarity has 32 slots. For positive candidates, which lie outside the current mask, 16 slots take the most uncertain locations under $\boldsymbol{u}_t$ with at most one location per 8×8 cell, 10 slots come from a uniform grid over the region outside the mask, and the remaining slots sample the boundary band where $\boldsymbol{p}_t$ crosses 0.5. Negative candidates, which lie inside the mask, use the same three sources with 16, 10, and 2 slots, and the centroids of up to four disconnected mask components other than the largest one are added. Locations within 16 pixels of an earlier click at the input resolution of 1024 are excluded, and duplicates are removed, which leaves about 55 candidates per state.

Network. The click maps are Gaussians with a standard deviation of 8 pixels at 256×256. The encoder levels have 64, 128, 256, and 256 channels at 256^2, 128^2, 64^2, and 32^2. Each level consists of two 3×3 convolutions, each followed by group normalization with eight groups and a SiLU activation, and the three downsampling levels use a stride of two in the first convolution. The decoder upsamples the 32^2 features, concatenates them with the 64^2 encoder features, and applies one block, then repeats this at 128^2 and resizes the result bilinearly to 256^2. The policy logit map is a 1×1 convolution on the 256^2 features. The stop logit, the value, and the current-Dice estimate are two-layer perceptrons with 128 hidden units on the globally pooled 32^2 features, and the value and the current-Dice estimate end with a sigmoid. The network has $3{,}369{,}220$ parameters and the input projection of the SAM ViT-B embedding adds $8{,}224$.

Algorithm 1 AlphaClick at test time

Require: image $\boldsymbol{x}$, initial click c_0, budget T, simulations per click n, policy-value network θ

$\boldsymbol{e} \leftarrow E(\boldsymbol{x}), \quad C \leftarrow \{c_0\}, \quad \boldsymbol{p} \leftarrow \mathrm{sigmoid}(G(\boldsymbol{e}, C))$
create the root node for C and expand it with $\mathcal{A}$, P_θ, and V_θ
for $t = 1, \dots, T$ **do**
  draw $g(a) \sim \mathrm{Gumbel}(0, 1)$ and keep the m root actions with the largest $g(a) + \log P(s_0, a)$
  for each round of sequential halving **do**
    for each kept root action, for its share of the n simulations **do**
      descend below the root by Equation 6 to an unexpanded node
      decode the node with G, evaluate V_θ, expand its top m candidates, and back up V_θ
    end for
    keep the half of the root actions with the largest $g(a) + \log P(s_0, a) + \sigma(\tilde{Q}(a))$
  end for
  commit a_t, the remaining action with the largest score, and set $C \leftarrow C \cup \{a_t\}$
  make the child of a_t the new root and keep its subtree
end for
return the mask $\boldsymbol{m}_T = \mathbb{1}[\mathrm{sigmoid}(G(\boldsymbol{e}, C)) \geq 0.5]$

Trajectory generation. To shorten data generation, a preliminary network trained on a small pilot set ranks the candidates at every state, and only the 24 highest ranked ones are decoded when computing Dice gains. On pilot images from the training datasets, this keeps 98% of the Dice gain reachable with the full candidate set. The seven trajectories per image and the targets follow Section 3.3. Generation on the 80 training splits yields 1.27 million states from 35,551 images.

Training. We train for 20 epochs with AdamW, a learning rate of 3×10^{-4}, a weight decay of 10^{-4}, gradient clipping at 1.0, and mini-batches of four groups of four states. A tenth of the training images is held out, and the epoch whose value predictions have the highest correlation with the value targets within each held-out image is kept, which is epoch 16 with a correlation of 0.738. The same recipe is used for SAM2.1 and MedSAM2, with trajectories generated on each backbone.

Search. Below the root, the selection rule of Equation 6 uses $c_{\mathrm{puct}} = 1.5$. Leaves are expanded in batches of 16 with virtual loss, which counts every pending simulation as one visit with a value of -1 until its result is backed up. At the root, the completed value of an unvisited action is

$$v_{\mathrm{mix}} = \frac{1}{1 + \sum_b N(s_0, b)} \Big(V_\theta(s_0) + \frac{\sum_b N(s_0, b)}{\sum_{b \in \mathcal{V}} P(s_0, b)} \sum_{b \in \mathcal{V}} P(s_0, b)\, Q(s_0, b) \Big), \tag{8}$$

where $\mathcal{V}$ is the set of visited root actions, following Danihelka et al. (2022). The completed values are rescaled to $[0, 1]$ over the root actions before Equation 7 is applied. We set $c_{\mathrm{visit}} = 50$ as in Gumbel AlphaZero and choose c_{scale} so that an unvisited value range spans as many logits as the prior. The median spread of the prior logits over training states is 5.30, which gives $c_{\mathrm{scale}} = 5.30/50 = 0.106$. Algorithm 1 summarizes one episode.

B Datasets and evaluation protocol

Table 5 lists the 80 datasets. Test targets are the first targets of each official test split in a random order fixed by seed 42, at most 200 per dataset, and every method sees the same targets. The macro Dice weights every dataset equally, so the cap only affects the precision of each per-dataset mean. On complete test splits evaluated before the main experiments, drawing 200 targets per dataset changed the macro Dice of three reference methods by at most 0.0011, with a standard deviation over five draws of at most 0.0013, far below the differences reported in this paper.

C Baselines

All baselines use the frozen SAM ViT-B, the shared initial click, and a budget of ten corrective clicks after the initial click, and none of them reads the ground truth of a test image.

Table 5: The 80 datasets of IMed-361M with an official test split, grouped by modality. Datasets marked with an asterisk form the 12 unseen datasets of Table 7, which are excluded from training for that experiment only.

Modality	Count	Datasets
MR	37	ACDC*, ADAM2020, AMOS2022_MR, ATLAS2023, braimMRI, Brain-MRI, BraTS-TCGA-GBM, BraTS-TCGA-LGG, BraTS2013, BraTS2015, BraTS2018, BraTS2019, BraTS2020, BraTS2021, BraTS2023_GLI, BraTS2023_MEN, BraTS2023_MET, BraTS2023_PED, BraTS2023_SSA, CHAOS_Task_4_t1*, CHAOS_Task_4_t2, CMRxMotions, EMIDEC, Heart_Seg_MRI, ISBI-MR-Prostate-2013, Learn2Reg2022_L2R_task1_AbdomenMRCT_seg_MR, MMWHS_MR, mnms2, MSD_Heart*, MSD_Prostate, Myops2020_t2, PROMISE12, Prostate_MRI_Segmentation_Dataset, PROSTATEx-Seg-HiRes, PROSTATEx-Seg-Zones, SPPIN2023, μ-RegPro_t2
CT	24	AbdomenCT1K, AMOS2022_CT, BTCV*, Colorectal_Liver_Metastases, Continuous_Registration_task1, COVID19CTscans, finding-lungs-in-ct-data_2d, FLARE21, FLARE22, HCC-TACE-Seg, KiTS, KiTS2021, KiTS2023, Learn2Reg2022_L2R_task1_AbdomenCTCT, Learn2Reg2022_L2R_task1_AbdomenMRCT_seg_CT, MMWHS_CT, MSD_Liver, MSD_Spleen*, PleThora, Prostate-Anatomical-Edge-Cases, SegRap2023_ct, SegRap2023_cta, VESSEL2012, WORD
Endoscopy	6	cvc_clinicdb*, endovis15, hyper-kvasir-segmented-images, kvasir_seg, kvasir_seg_aliyun*, kvasircapsule_seg
Fundus	6	drishti_gs_cup*, drishti_gs_od, gamma*, ichallenge_adam_task2, PAPILA, rimonedl
Dermoscopy	3	isic2016_task1, isic2017_task1*, isic2018_task1
Ultrasound	3	CETUS2014, TDSC-ABUS2023*, μ-RegPro_us
X-ray	1	sz_cxr*

Entropy and SimSAM. Entropy uses the candidate set of AlphaClick and clicks the candidate with the highest binary entropy of the current probability map. SimSAM is implemented on the same segmentation interface as all other methods. At each step it simulates ten clicks, decodes the resulting masks, and aggregates them with its consensus rule.

AlignSAM. We run AlignSAM from a public implementation with SAM ViT-B, episodes of the initial click plus ten corrective clicks, a vocabulary for its contrastive language-image pretraining (CLIP) (Radford et al., 2021) text encoder built from the class names of the 80 datasets, and the same training splits. Its reward includes the Dice term, and no step reads the ground truth of a test image.

PPO. We use the official code with SAM ViT-B, the segmentation model shared by all methods. For each dataset, an annotated training image serves as the reference for DINOv2 (Oquab et al., 2024) feature matching, which produces the initial prompt set that PPO refines with per-image deep Q-learning. The refined eleven prompts are decoded once. On 24 targets no prompt graph can be built from the matches, and these targets are excluded from its average.

PPD. The official code is available but the weights are not released, so we trained the attack and defense agents on the same training splits and selected the checkpoint on 400 validation images from 40 datasets. PPD uses DINOv3 (Siméoni et al., 2026) matching with an annotated training image as reference and removes prompts with its defense agent until its own stopping rule ends the episode. On 11 datasets no reference image produces a usable prompt set, and PPD keeps the initial click there.

D Additional results

Table 6 lists the Dice at every search budget.

Table 6: Dice after the last click on all 80 datasets for more budgets of simulations per click.

Simulations per click	1	4	8	16	32	64
Dice@10	0.6504	0.6771	0.6899	0.6929	0.6922	0.6937

Modalities and unseen datasets. Table 7 lists the results by imaging modality and on the 12 datasets excluded from training. AlphaClick leads in every modality and exceeds every baseline on 11 of the 12 unseen datasets.

Table 7: Dice after the last click with 64 simulations per click, by modality (upper block) and on 12 datasets excluded from training (lower block), against the strongest baseline of each group. On the unseen datasets AlphaClick exceeds every baseline on 11 of the 12 datasets.

Group	Datasets	Initial click	Strongest baseline	AlphaClick	Gain
MR	37	0.3621	0.3824 (Entropy)	**0.6484**	+0.2660
CT	24	0.4224	0.4373 (Entropy)	**0.7049**	+0.2675
Endoscopy	6	0.4611	0.5055 (Entropy)	**0.8293**	+0.3238
Fundus	6	0.3720	0.4625 (Entropy)	**0.6866**	+0.2241
Dermoscopy	3	0.6218	0.6172 (SimSAM)	**0.7950**	+0.1778
Ultrasound	3	0.3963	0.4124 (Entropy)	**0.7336**	+0.3212
X-ray	1	0.4739	0.6774 (PPO)	**0.9064**	+0.2290
Unseen datasets	12	0.5088	0.5420 (Entropy)	**0.6648**	+0.1228

Adaptive stopping. With the stop action enabled and 256 simulations per click, AlphaClick ends on its own after 8.96 corrective clicks on average and stops early on 17% of the targets. It reaches a Dice of 0.6896, compared with 0.6915 when all ten clicks are placed on the same $12{,}379$ targets, so it saves a tenth of the clicks while keeping the accuracy.

E Qualitative results

Figures 5 and 6 show one case per modality, selected as the case whose mean Dice over the five baselines is closest to the median and to the lower quartile of its modality. The rule only uses baseline results and only considers images with a single target.

F Ethics statement

This work uses only publicly available, de-identified medical imaging datasets released for research, and follows their licenses. No new data were collected from patients. AlphaClick is a research method for reducing annotation effort and is not intended for clinical decision making without expert review.

G Reproducibility statement

Section 3 and Appendix A specify the decision process, the network, the training targets and loss, and the search with all constants. Appendix B lists the datasets and the rule that selects the test targets, and Appendix C describes how every baseline was run. The code, the trained networks, and the scripts that produce every table and figure from the raw evaluation results will be released.

H Use of AI assistance

AI tools assisted us in two ways, polishing the language of the paper and discovering related work. We checked each suggestion and each retrieved paper before using it. All content of the paper is our responsibility.

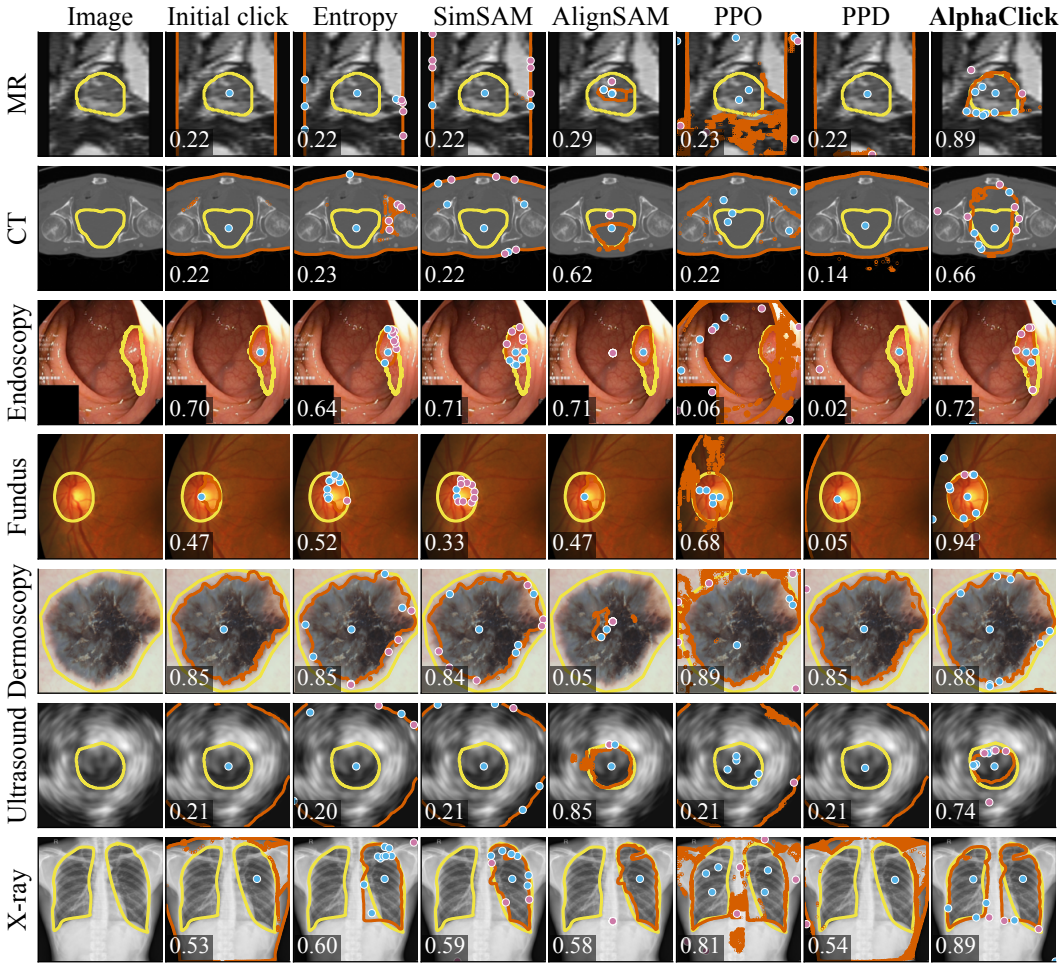

Figure 5: Median cases of all seven modalities, drawn as in Figure 4.

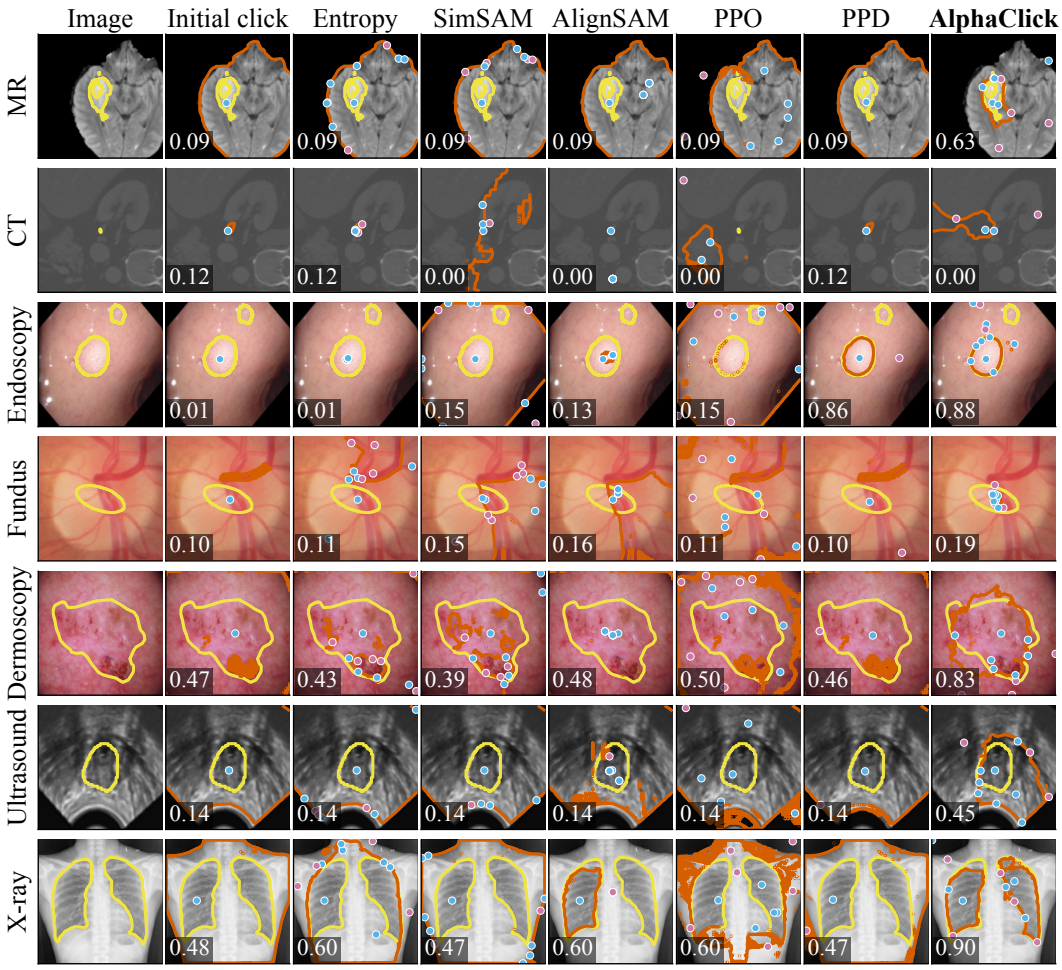

Figure 6: Lower-quartile cases of all seven modalities, drawn as in Figure 4.