Karlsruhe Institute of Technology

cv:hci @ KIT
Computer Vision for Human-Computer Interaction

Deep Learning for Computer Vision II: Advanced Topics

Interactive Segmentation (held by Zdravko Marinov)

Motivation

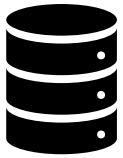
Cityscapes image: 1.5h annotation time [5]
(1024 x 2048 pixels)

5000 train + test images to annotate in total

How much time?

Motivation

Cityscapes image: 1.5h annotation time [5]
(1024 x 2048 pixels)



5000 train + test images to annotate in total

312.5 days 24/7
Or 4 work years!!!

Motivation

Do we really need to click all 1024x2048 pixels?

Cityscapes image: 1.5h annotation time [5]
(1024 x 2048 pixels)

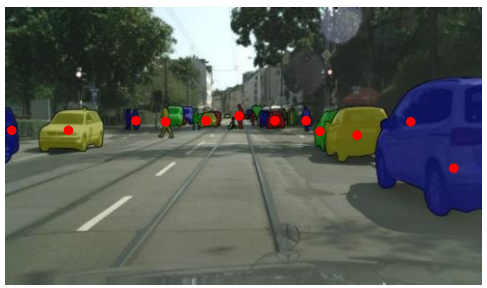
5000 train + test images to annotate in total

312.5 days 24/7
Or 4 work years!!!

Motivation

Do we really need to click all 1024x2048 pixels?
→ No! Solution: Interactive Segmentation
→ Only 2-3 clicks per object are needed!

Cityscapes image: 1.5h annotation time [5]
(1024 x 2048 pixels)



5000 train + test images to annotate in total

312.5 days 24/7
Or 4 work years!!!

Annotation time is reduced significantly!

Outline

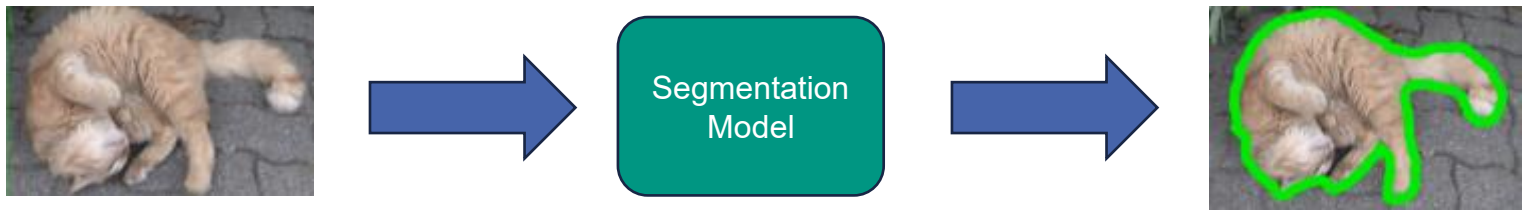
- Definition of Interactive Segmentation
- Components in Deep Interactive Segmentation
  - Type of Interaction
  - Guidance Signal
  - Robot User
  - Active Learning
- Evaluation Metrics
- Applications
  - Natural Images
  - Medical Image Analysis
- Segment Anything Model (SAM)

Definition of Interactive Segmentation [1]

- Definition:
  - *Interactive segmentation describes an **iterative feedback loop**, where user-provided corrections to the model's output inform subsequent predictions, leading to **updated predictions**. User guidance is provided in the form of, e.g., **clicks, scribbles, or other interactions**.*

Definition of Interactive Segmentation [1]

- Definition:
  - *Interactive segmentation describes an **iterative feedback loop**, where user-provided corrections to the model's output inform subsequent predictions, leading to **updated predictions**. User guidance is provided in the form of, e.g., **clicks, scribbles, or other interactions**.*



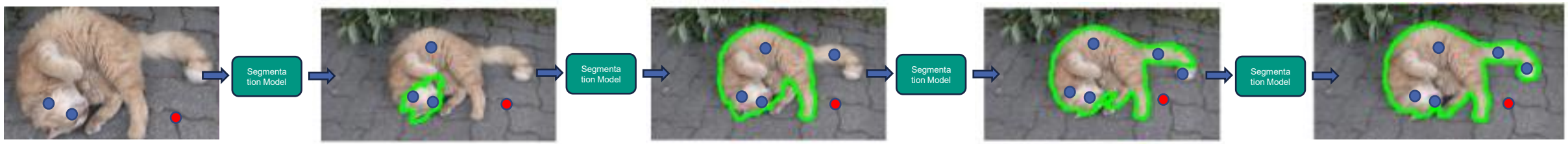
Examples:
CNN, ViT, K-means clustering

Non-interactive Segmentation

Image adapted from: Jain, Suyog Dutt, and Kristen Grauman. "Click carving: Interactive object segmentation in images and videos with point clicks." *International Journal of Computer Vision* 127 (2019): 1321-1344.

Definition of Interactive Segmentation [1]

- Definition:
  - *Interactive segmentation describes an **iterative feedback loop**, where user-provided corrections to the model's output inform subsequent predictions, leading to **updated predictions**. User guidance is provided in the form of, e.g., **clicks, scribbles, or other interactions**.*



Click Initialization | Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4

Interactive Segmentation

- "Cat" Click
- "Background" Click

Image adapted from: Jain, Suyog Dutt, and Kristen Grauman. "Click carving: Interactive object segmentation in images and videos with point clicks." *International Journal of Computer Vision* 127 (2019): 1321-1344.

Domains of Interactive Segmentation

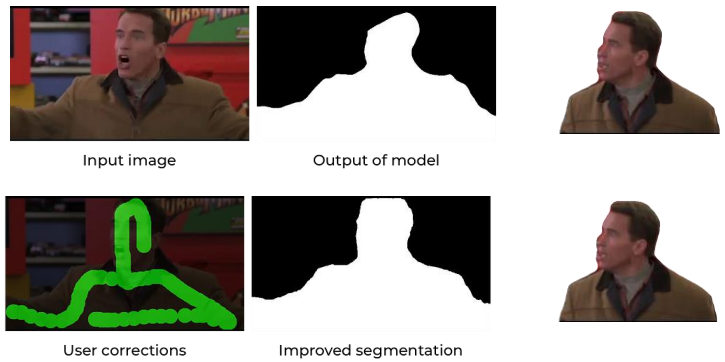
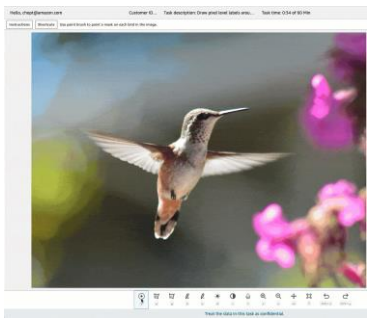


Image & Video Editing

Source: https://medium.com/gifs-ai/interactive-segmentation-with-convolutional-neural-networks-2e171a85df82



Source: https://docs.aws.amazon.com/sagemaker/latest/dg/sms-auto-segmentation.html

Rapid Image Annotation [6]

Object Instance Retrieval [9]

Tumour Segmentation in PET/CT [7]
(Previous Master Thesis at CV:HCI, 2023)

Full-body Anatomy Segmentation [8]

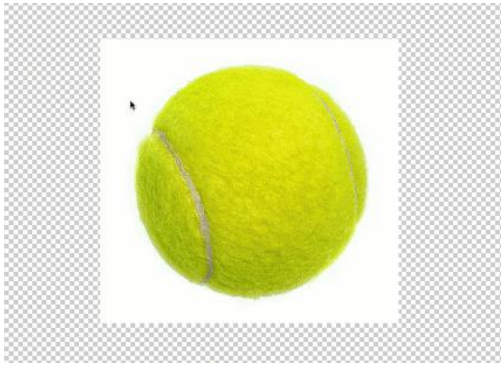
History

- Classical approaches
  - Active Contours
  - Graph Cut
- Deep Learning-based approaches

Classical Approaches

- Active Contours [3]
  - Magnetic Lasso (Adobe Photoshop)
  - User-drawn scribble (red curve) is *moved* along the gradient vector field
    - Scribble “snaps” to the object boundary
- Contour (also called a snake) minimizes the energy functional
  - E_{int} → Internal energy: enforces the contour to be smooth and without sharp edges
  - E_{image} → Image Gradient: attracts the contour towards local minimums in the gradient field (edges)
  - E_{con} → Users can drag the edges of the contour → Additional constraint to force the curve to obey the user interactions

$$E^*_{\text{snake}} = \int_0^1 E_{\text{snake}}(\mathbf{v}(s))\, ds$$
$$= \int_0^1 E_{\text{int}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s))$$
$$+ E_{\text{con}}(\mathbf{v}(s))\, ds$$



GIF Source: https://www.clydepixel.com/blog/photoshop-tips-tricks-beginners-magnetic-lasso-tool/

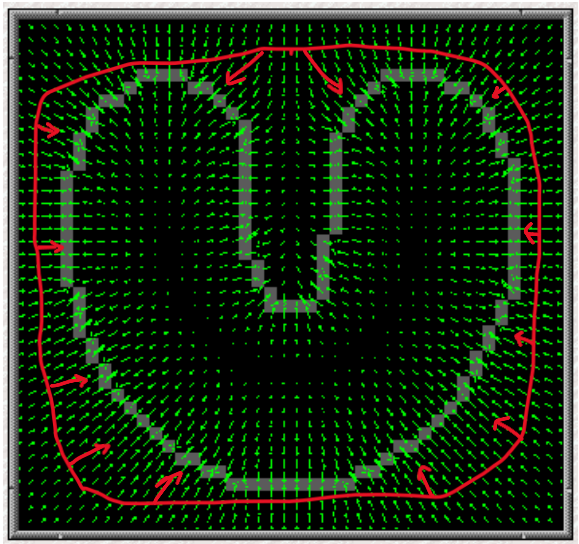


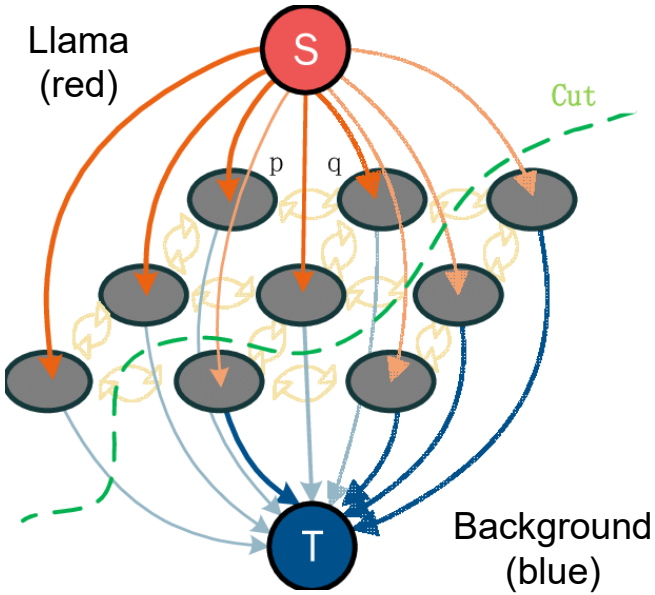
Image Source: https://iacl.ece.jhu.edu/Projects/gvf/

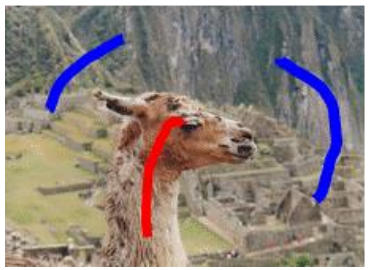
Classical Approaches

- Graph Cut [4]
  - Represent image + scribbles as graph
- Each pixel in the image is a vertex
  - Additional **S** vertex for the object and **T** for the background
- Edges are defined for all pairs of pixels
  - All marked “Llama” pixels have an infinite weight to S
  - Same for all “Background” pixels to T
  - Other edges:
    - Small weight if colour difference or distance to other pixel is large

$$B_{\{p,q\}} \propto exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}$$

- Segmentation boils down to computing the minimum cut





Images Sources: https://www.datasciencecentral.com/interactive-image-segmentation-with-graph-cut-in-python/

Deep Learning-based Approaches

Classical approaches:

- Rely solely on appearance features
  - Do not incorporate any semantic meaning
  - Struggle with weak boundaries
    - Or multiple similar objects next to each other (herd of llamas)
- Difficult to encode prior knowledge such as shapes and textures of segmentation targets
- No redundancies in the representations
  - Small variations in appearance lead to large prediction variations

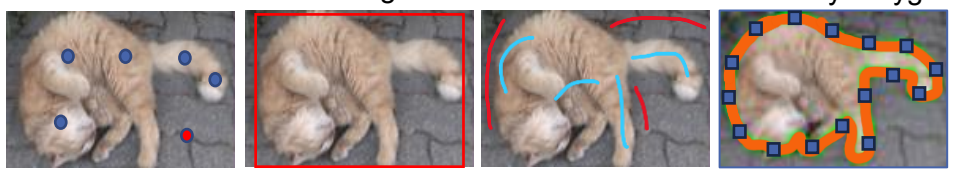
Solution:

- Deep Learning-based approaches

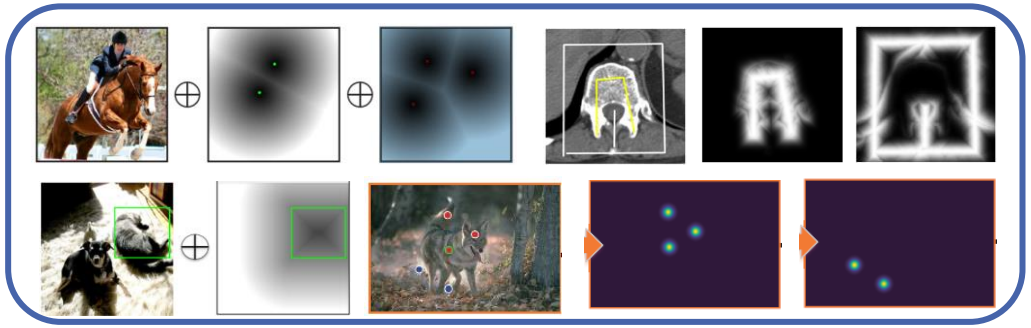
Deep Learning-based Interactive Segmentation

All components are here… Let's build an interactive model together!

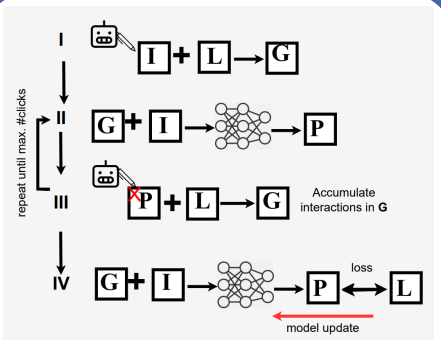
Types of Interactions



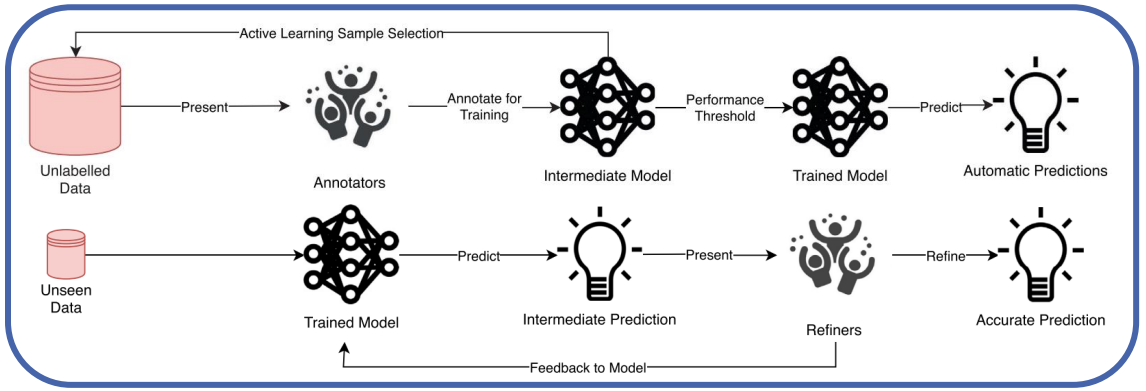
Guidance Signal



Robot User



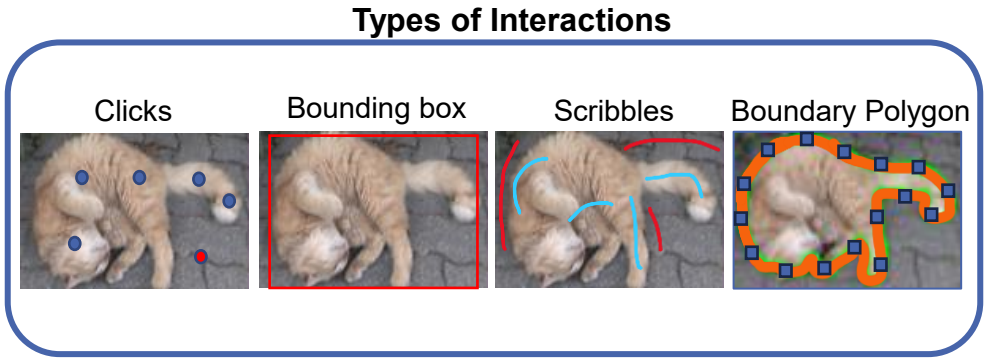
Active Learning



TYPES OF INTERACTIONS

Types of Interactions

- The way the user communicates with the model
- **Click**
  - A point* $c_i = (x_i, y_i)$
- **Scribble**
  - Set of n points $S = \{c_1, c_2, \ldots, c_n\}$
- **Bounding box**
  - Points representing a rectangular region
    - Can be parametrized by two points (top left, bottom right), initialized by user
- **Boundary polygon**
  - Sequence of m vertices $P = \{v_1, v_2, \ldots, v_m\}$ lying on the boundary of the object
  - User interactions are to **put** and **drag** them to correct positions
- **Other interactions (rarely used)**
  - Examples: Eye gaze, text prompts



*2D or 3D depending on image dimension

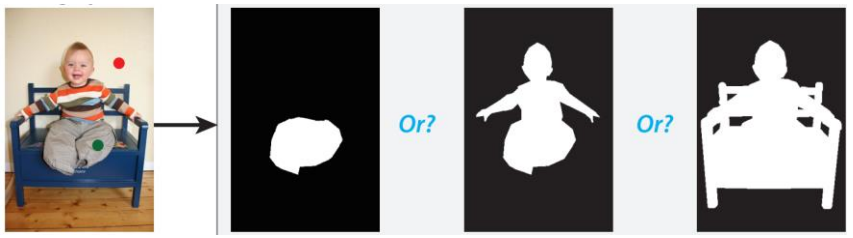
Clicks

- Click
  - A point* $c_i = (x_i, y_i)$
  - **Pros**
    - Quick
    - Precise
      - Can be placed in tight spots to correct small errors
    - Easy to simulate during training
      - Center of largest object / error
      - Extreme points
      - Random click
  - **Cons**
    - Ambiguous
      - Not always clear what the user intends
      - Precise, but low amount of information
    - May require many clicks for complex objects

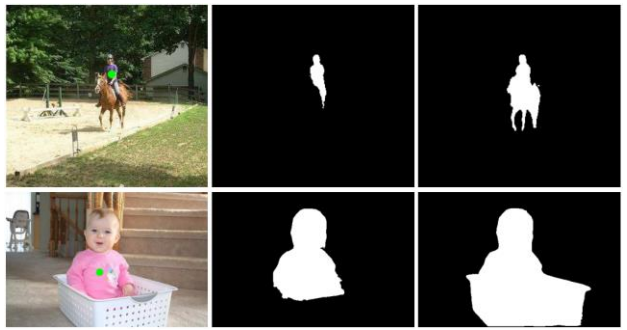
Click annotation example

Source: SimpleClick [10], CVPR 2023

Ambiguity in clicks



Source: PhraseClick [11], ECCV 2020



Source: Latent Diversity [12], CVPR 2018

Scribbles

- Scribble
  - Set of n points $S = \{c_1, c_2, \ldots, c_n\}$
  - **Pros**
    - Flexible and precise
      - Can approximate any shape
    - Low ambiguity due to its expressiveness
  - **Cons**
    - Simulations are possible but introduce a “user shift”
      - **User shift**: Discrepancy between simulated interactions during training and real interactions during evaluation
      - Occurs due to the larger flexibility and “infinite” ways to simulate it
    - Takes slightly more time to draw

Scribble annotation example

Source: MiVOS [13], CVPR 2021

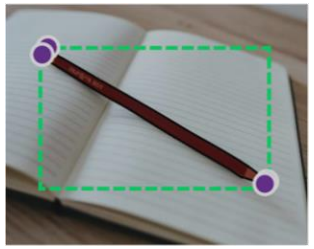
Bounding Boxes

- Bounding Box
  - Points representing a rectangular region
  - **Pros**
    - Quick
      - Requires 2 – 4 clicks
    - Localizes the context
      - Model can ignore everything outside the box
    - Easy to simulate
    - Can be represented in many ways
      - Top-left, bottom-right points
      - Extreme points: the farthest left, right, top, and bottom points.
  - **Cons**
    - Low precision
      - Bounding box contains background information as well

Bounding box annotation example

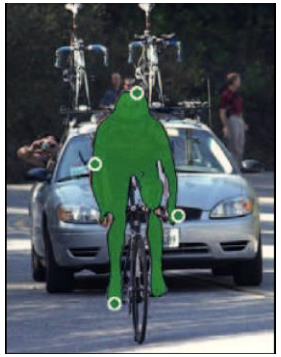
Source: https://blog.redbrickai.com/blog-posts/fast-meta-sam-for-medical-imaging

Top-left, bottom-right



Source: Inside-Outside Guidance [15], CVPR 2020

Extreme Points



Source: Deep Extreme Cut [14], CVPR 2028

Boundary Polygons

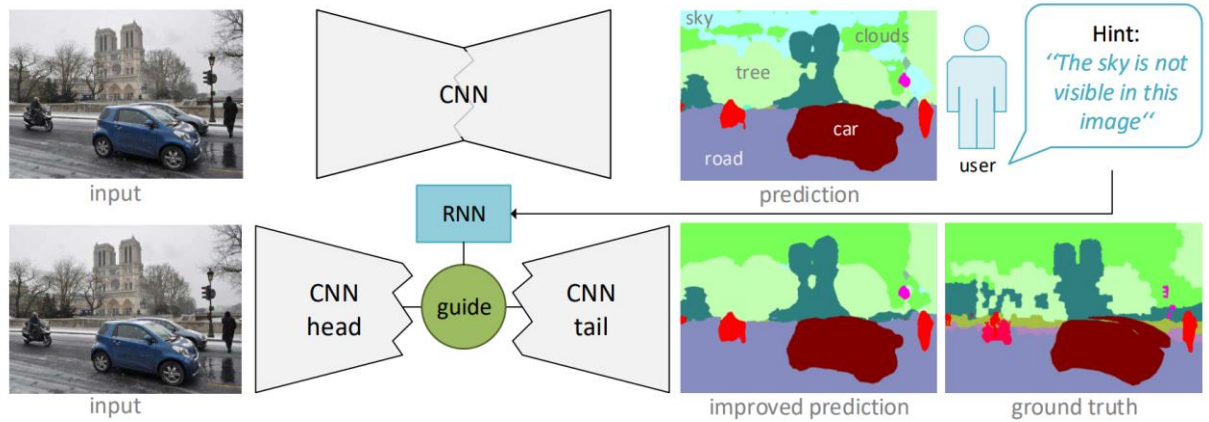
- Boundary Polygon
  - Sequence of m vertices P = {v_1, v_2,…, v_m} lying on the boundary of the object
  - **Pros**
    - Gives control to the user to **exactly** fit the boundary and correct the initial prediction
  - **Cons**
    - Takes more time to drag all vertices to the correct position
    - Difficult to simulate
      - Many possible ways to correct a vertex

Boundary polygon annotation example

Source: Polygon RNN+ [16], CVPR 2018

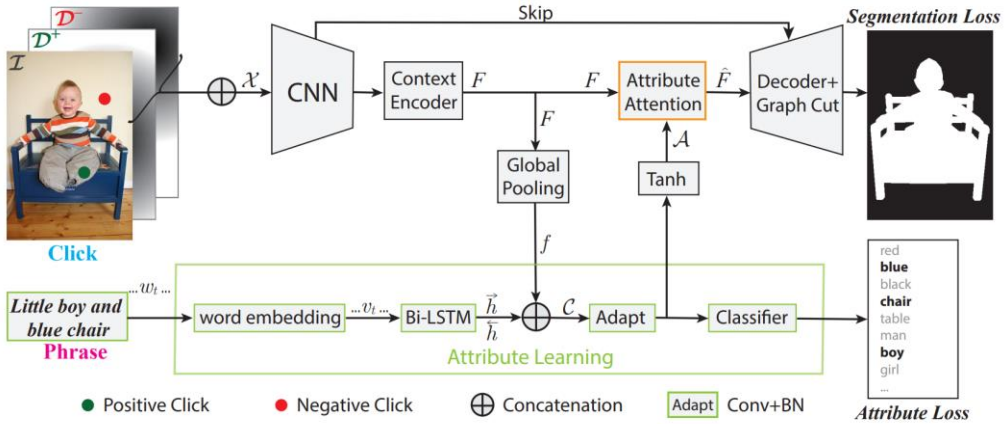
Other Types of Interactions - Text

- Text prompts:
  - Intuitive
  - Can eliminate ambiguity of clicks when combined



Source: Guide Me [17]

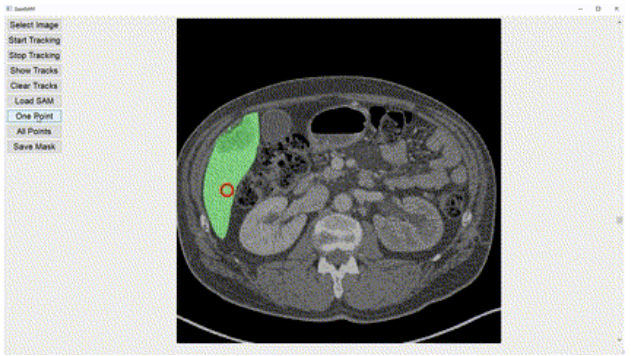
Source: SAM [9]



Source: PhraseClick [11]

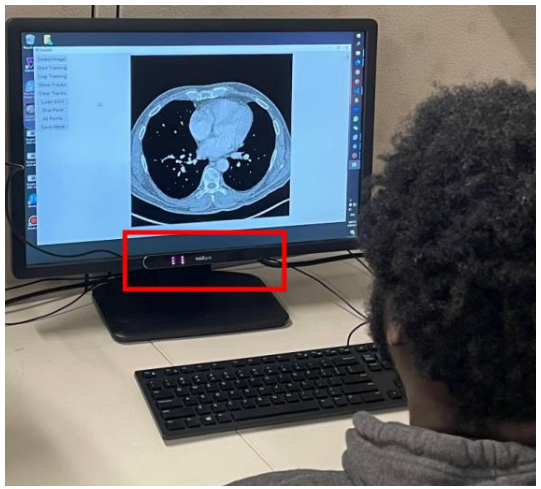
Other Types of Interactions – Eye Gaze

- Gaze := Eye Tracking
  - Follow gaze direction and segment that area
- Intuitive
- Very Quick
- Removes the need of touch
  - E.g. touching a screen in a surgery room
  - Or splitting attention while driving



Source: GazeSAM [18[

Source: GazeSAM [18[



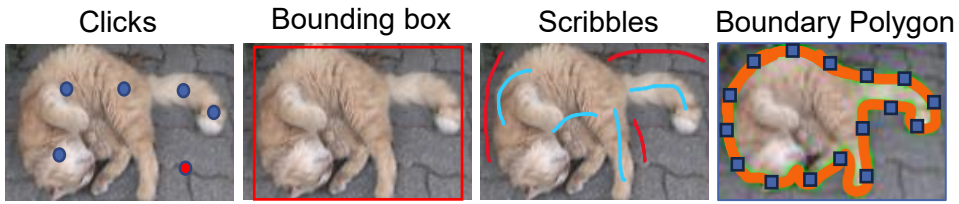
Source: GazeSAM [18[

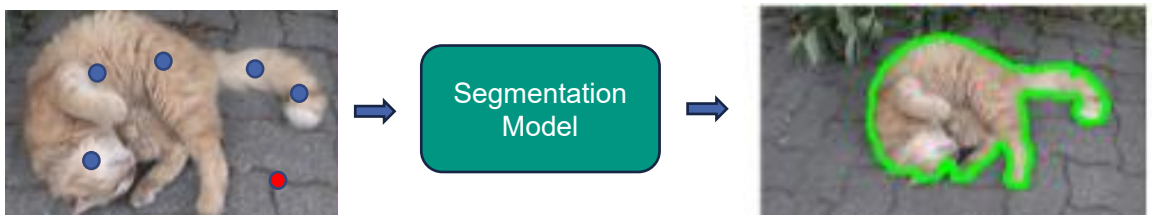
Source: https://www.tobii.com/learn-and-support/get-started/what-is-eye-tracking

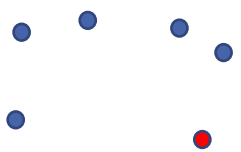
GUIDANCE SIGNAL

Guidance Signal

- Now that we know which interactions we have
  - How do we integrate that into the segmentation model?







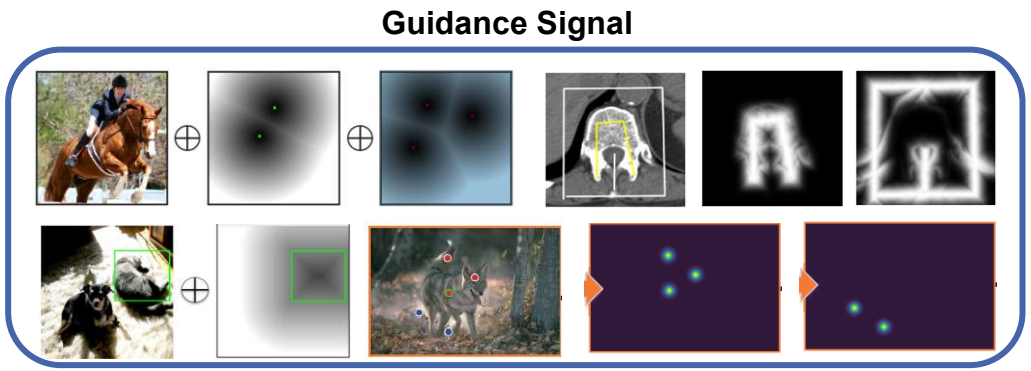
How are these clicks “presented” to the model?

Guidance Signal Definition [1]

- Interactions as Guidance Signals
  - "*A guidance signal is a representation of the user interactions **in a form in which the model can process it**. This can be an **explicit** representation that involves transforming the user interaction into an additional **structured input** for the model to process and learn from [...] or **implicit**, where user interaction information is subtly integrated into the model's learning process without the provision of explicit structured input.*"

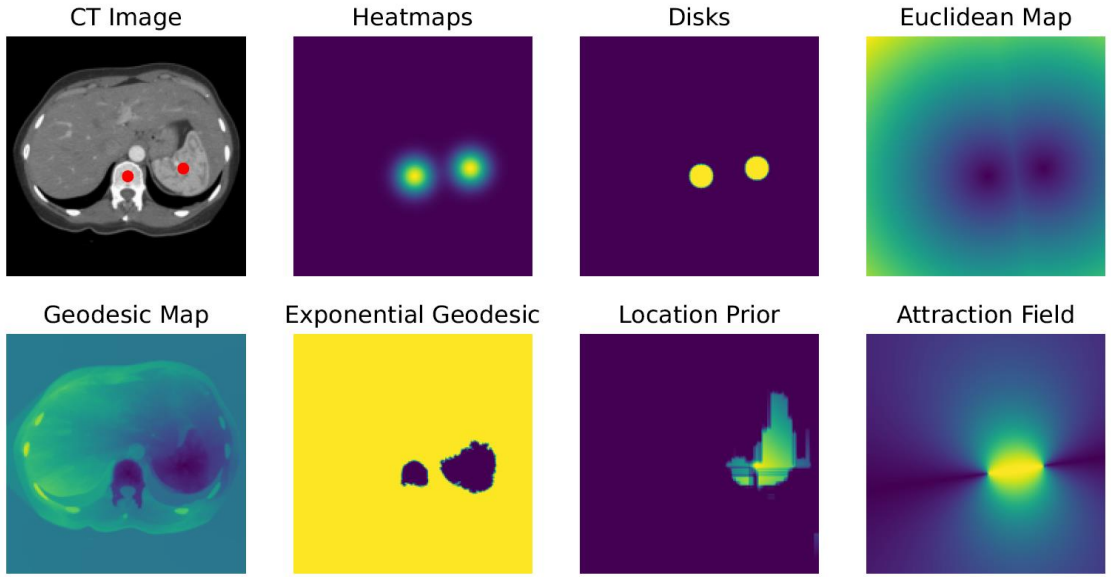
Guidance Signals Examples (Explicit)

- Explicit Guidance Signals
  - Additional structured inputs to the model
  - Deterministically computed from the interactions or interactions + image



Guidance Signals Examples (Explicit) [1]

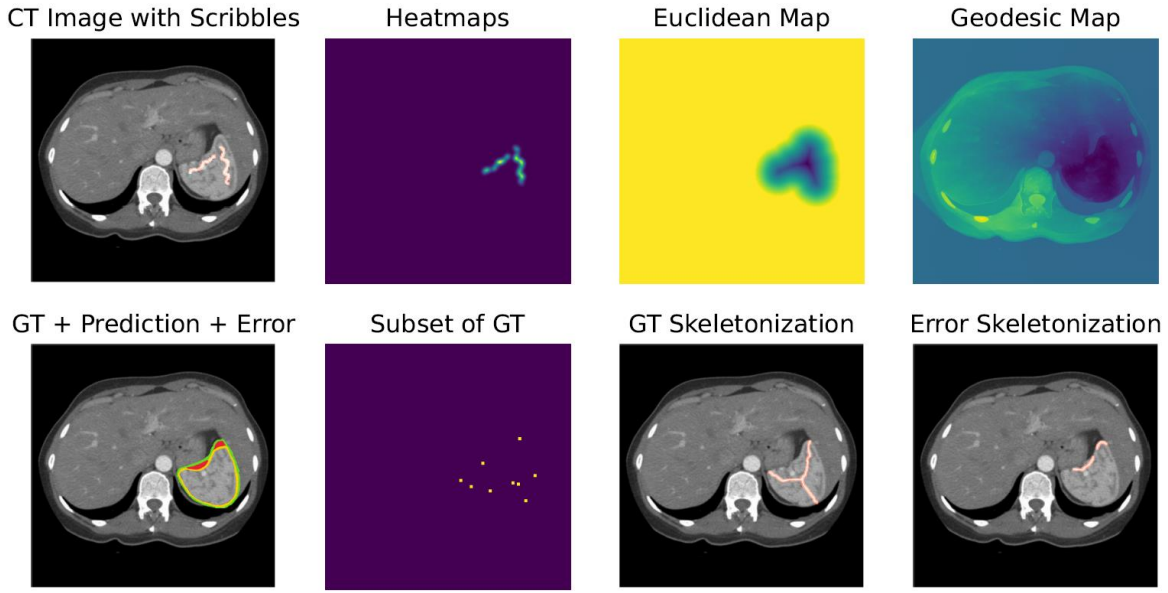
- Click-based explicit Guidance Signals
  - Gaussian Heatmaps and Disks
    - Locally encode the clicks
    - New clicks do not change the previous signal but are just pasted on top
  - Euclidean or Geodesic Map
    - Encode the minimum Euclidean or Geodesic distance to the set of clicks
    - New clicks require to recompute the whole guidance signal
    - Geodesic Distance := Euclidean + “Appearance” distance
      - Larger distance if there are large intensity changes
      - Similar to the Graph Cut [4] idea
  - Location Prior [20]
    - Start from 255
    - Reduce by 10 if you cross and edge
  - Attraction Field [19]
    - Model attraction field of punctual electric charges
  - Many others…



Guidance Signals Examples (Explicit)

- Scribble-based explicit Guidance Signals
  - Same as with clicks but applied over set of points (scribbles)
  - Scribbles are usually simulated by skeletonizing the ground-truth mask
    - Replicate brush strokes

Source: Wong, Hallee E., et al. "ScribblePrompt: Fast and Flexible Interactive Segmentation for Any Medical Image." *arXiv preprint arXiv:2312.07381* (2023).



Source: Marinov, Zdravko et al. "Deep Medical Interactive Segmenation: A Systematic Review and Taxonomy" [1]

Guidance Signals Examples (Implicit)

- Implicit Guidance Signals
  - Encoded without a tangible entity
    - Analogy: Explicit signals are “things”, Implicit signals are “actions”
  - Examples:
    - In the loss function
      - Higher loss in missegmentations near interactions → Force the model to “listen” to the interactions
    - In the input pre-processing
      - Use bounding box to crop the image and feed only crop to the model

Guidance Signals Examples (Implicit)

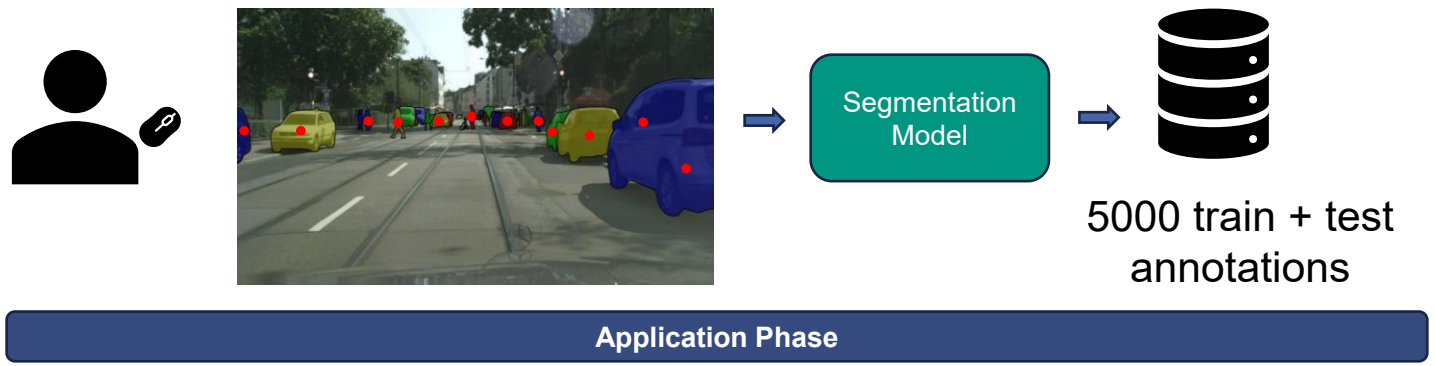
- Implicit Guidance Signals
  - Encoded without a tangible entity
    - Analogy: Explicit signals are “things”, Implicit signals are “actions”
  - Examples:
    - In the loss fun
      - Higher lo
    - In the input pre
      - Use bou

Takeaway: Guidance Signals are “things” or “actions” that transform the user interactions in a way that the model can process them

ROBOT USER

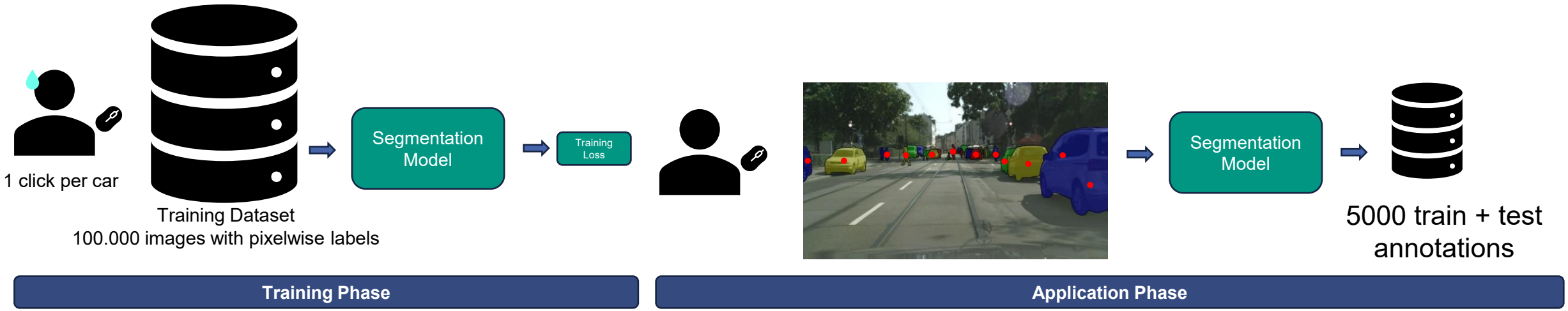
Robot User - Motivation

- How was the model trained **before** annotating the 5000 images?
- Deep neural networks are data hungry so it requires a much larger training set



Robot User - Motivation

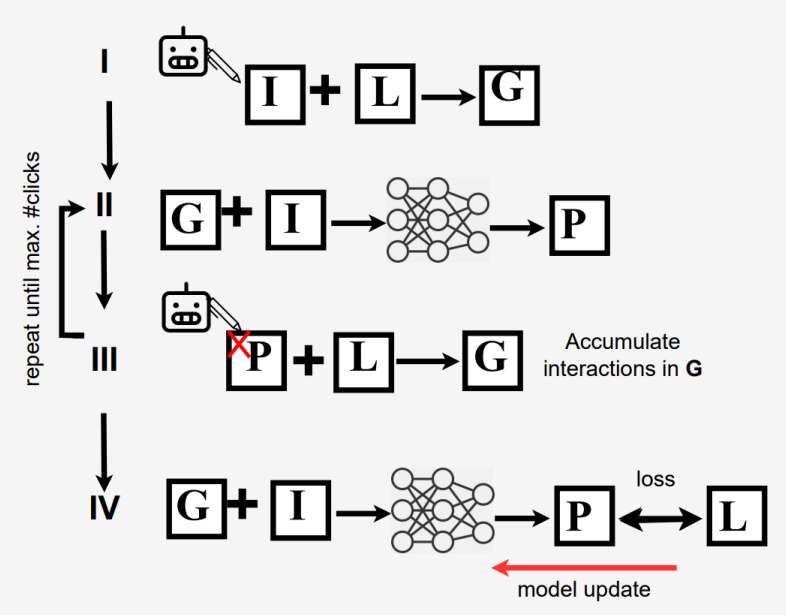
- How was the model trained **before** annotating the 5000 images?
- Deep neural networks are data hungry so it requires a much larger training set



- Minimum 100.000 clicks → Still a lot and quite expensive
- Solution: Simulate the clicks (**Robot User**)!

Robot User

- Definition [1]: *"A **simulated** model that mimics the behaviour of a real human annotator. The robot user **leverages ground-truth labels** to simulate user interactions at **plausible locations**."*



I Input image

L Ground-truth label

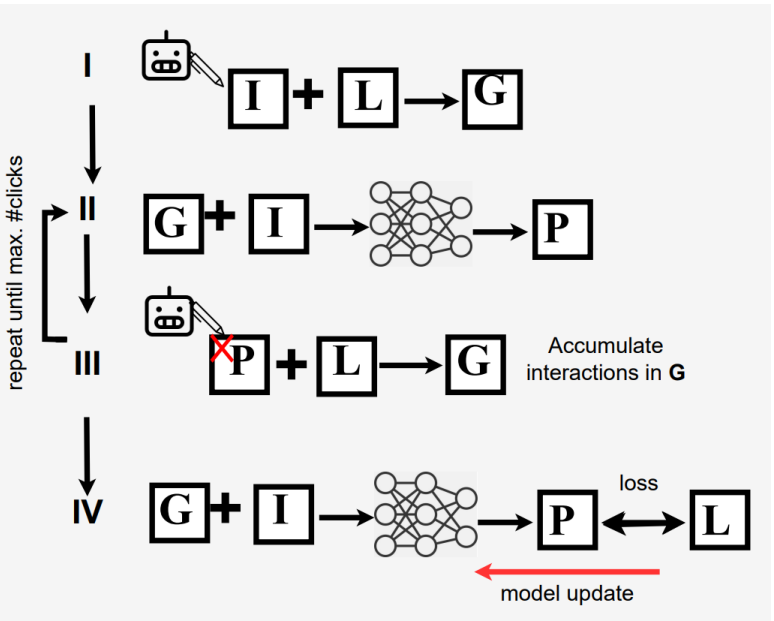
G Guidance Signal

P Prediction

Robot User I

I

Robot user generates an initial click



I Input image

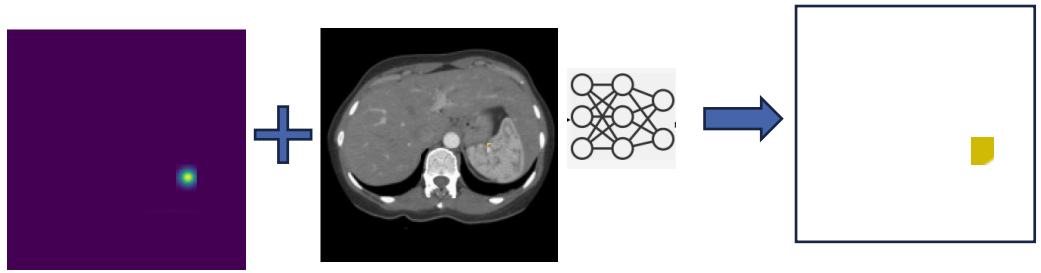
L Ground-truth label

G Guidance Signal

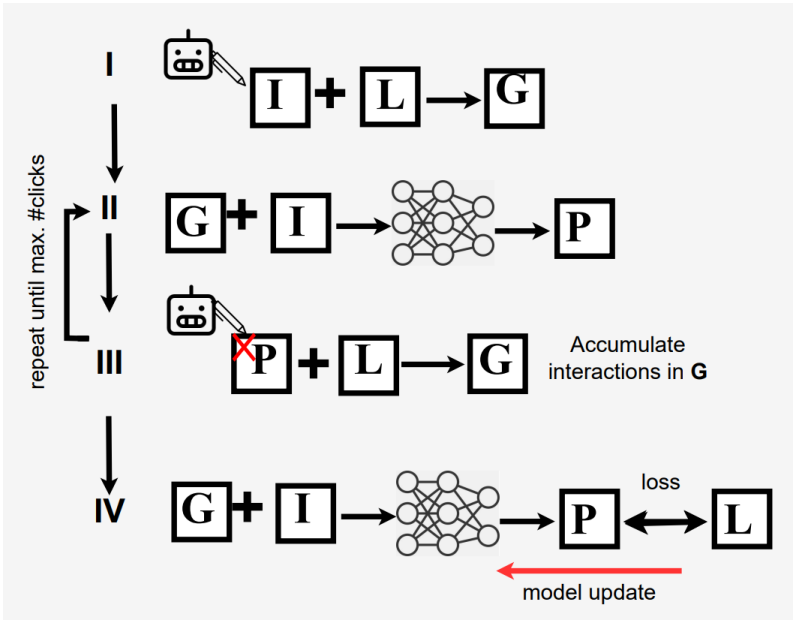
P Prediction

Robot User II

II



Interactive model predicts based on image + guidance signal



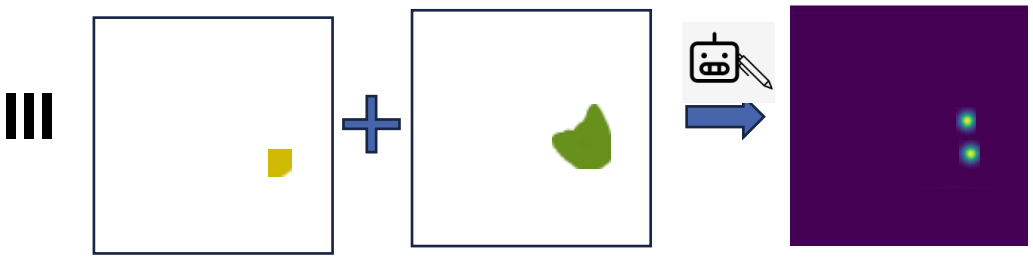
I Input image

L Ground-truth label

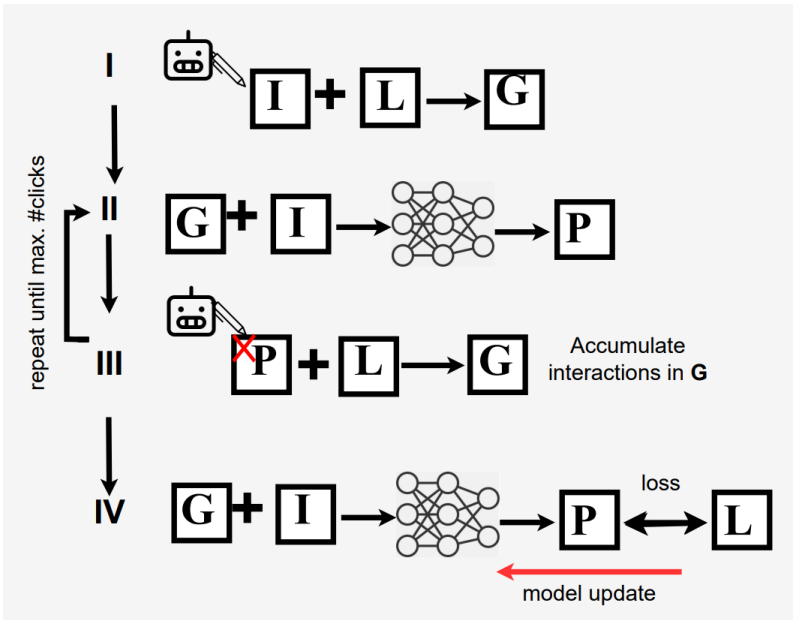
G Guidance Signal

P Prediction

Robot User III



Robot user generates a refinement click in the missegmented area



I Input image

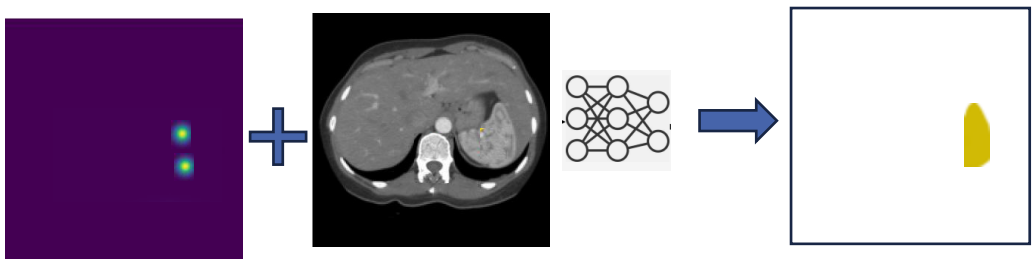
L Ground-truth label

G Guidance Signal

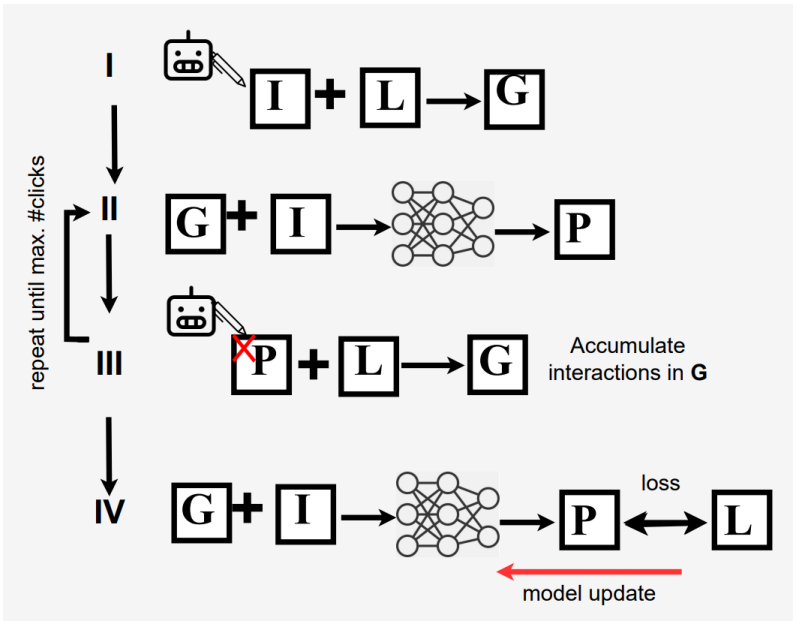
P Prediction

Robot User II

II



Interactive model predicts based on image + guidance signal



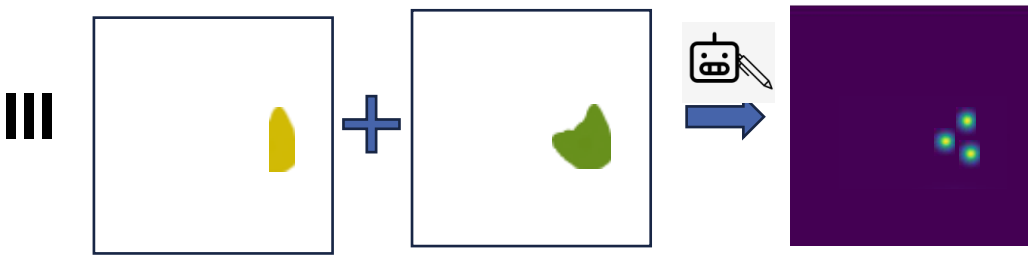
I Input image

L Ground-truth label

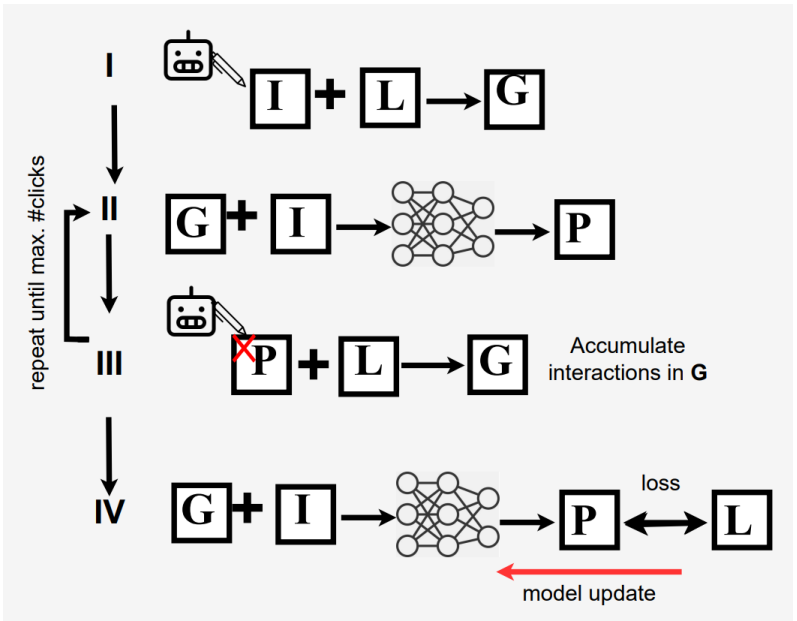
G Guidance Signal

P Prediction

Robot User III



Robot user generates a refinement click in the missegmented area



I Input image

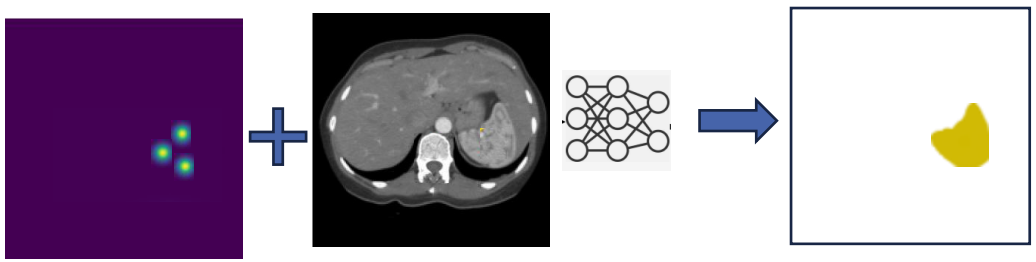
L Ground-truth label

G Guidance Signal

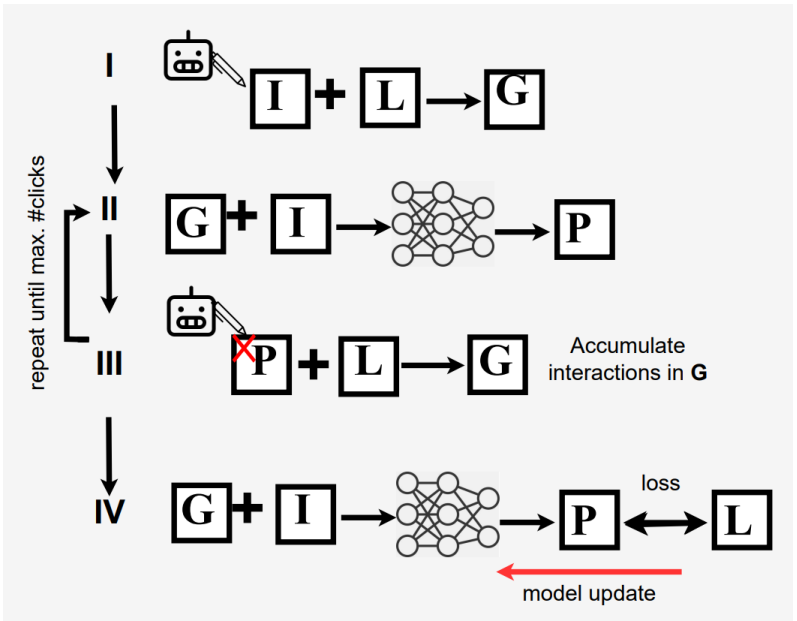
P Prediction

Robot User II

II



Interactive model predicts based on image + guidance signal



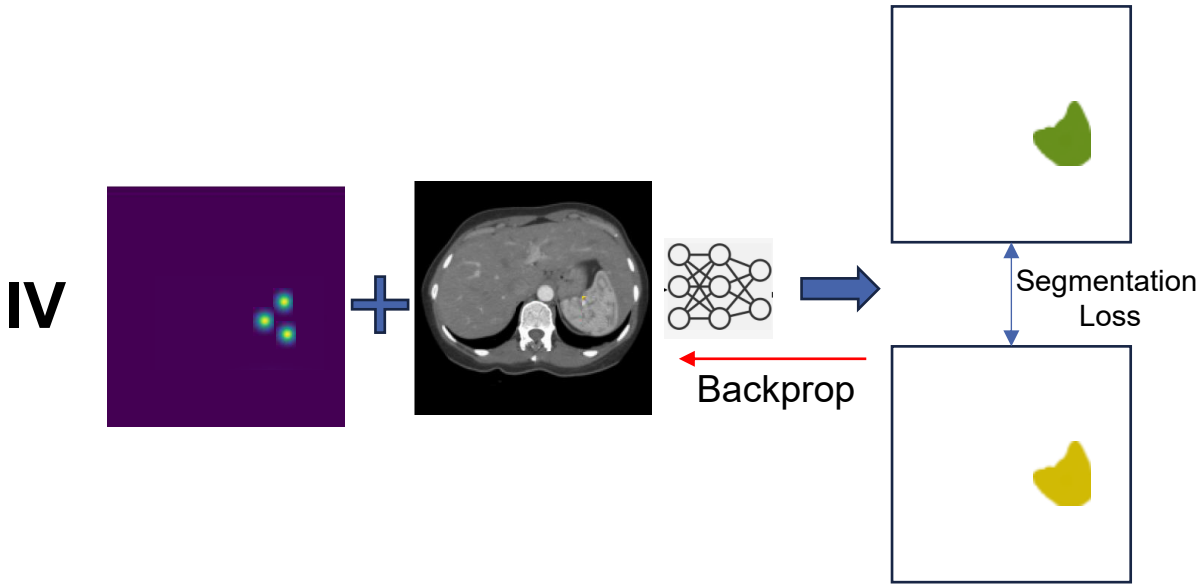
I Input image

L Ground-truth label

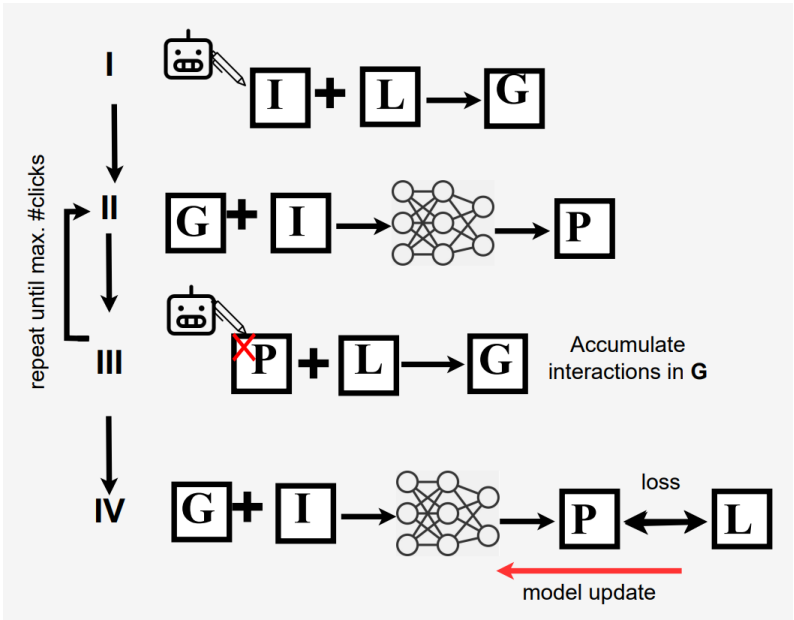
G Guidance Signal

P Prediction

Robot User IV



Compute loss after gathering all clicks and update model



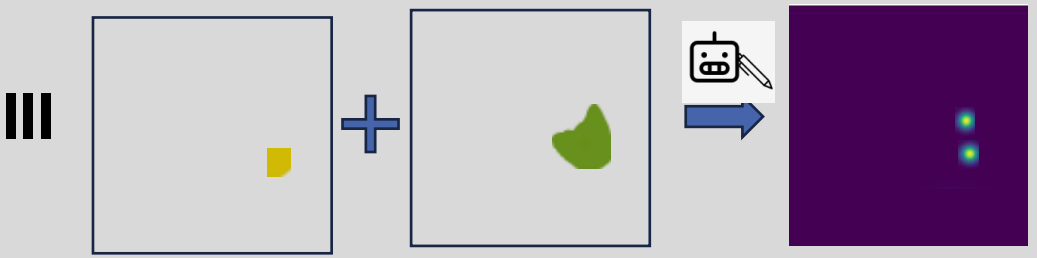
I Input image
L Ground-truth label
G Guidance Signal
P Prediction

Interactive models are usually updated after ALL prediction steps are performed

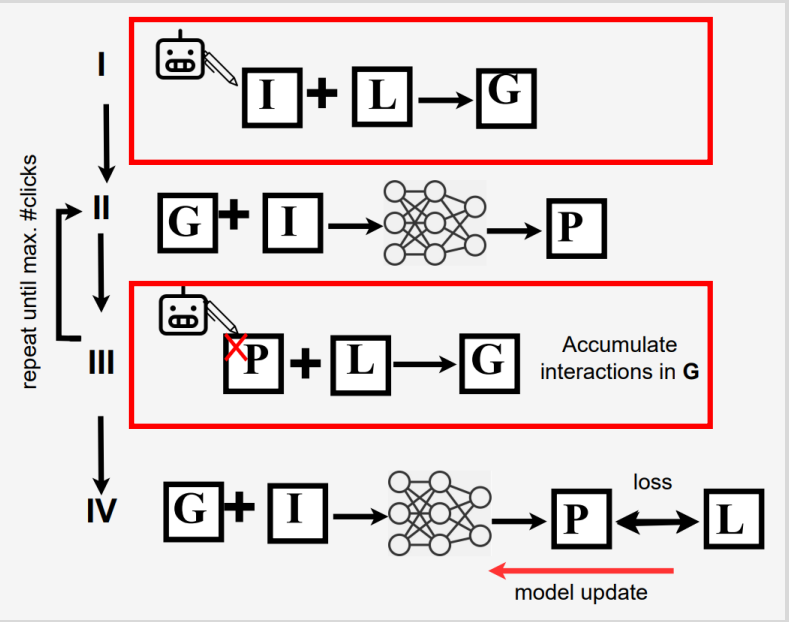
Discuss with your neighbour (3 min)

- **Question**: How could the robot user decide **where exactly** to put new interactions, for example **clicks** (steps I and III)?

Robot user generates an initial click

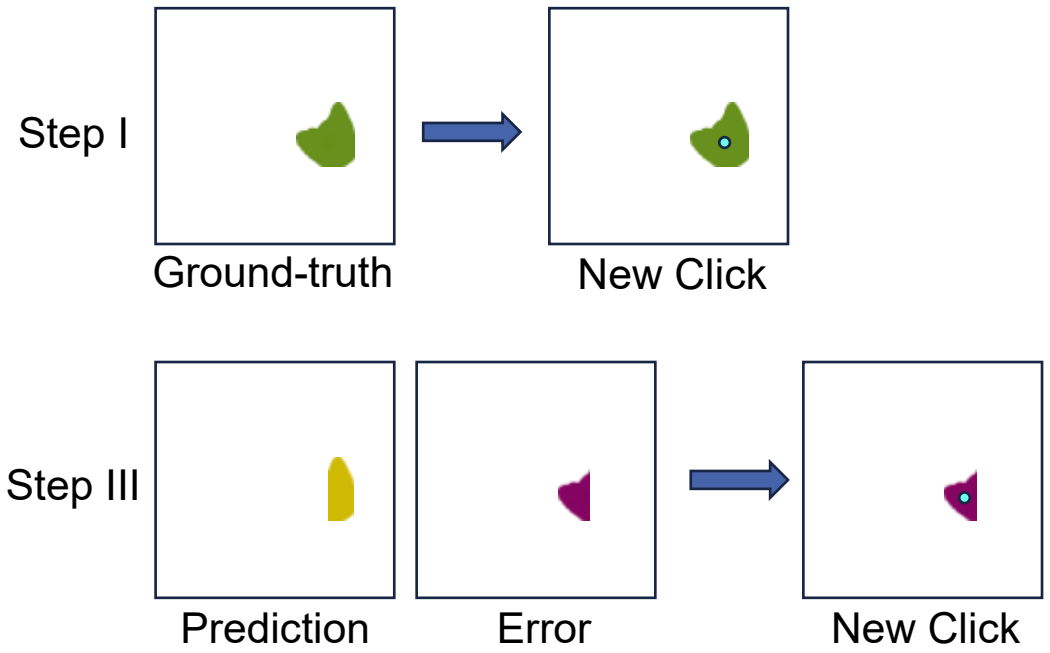


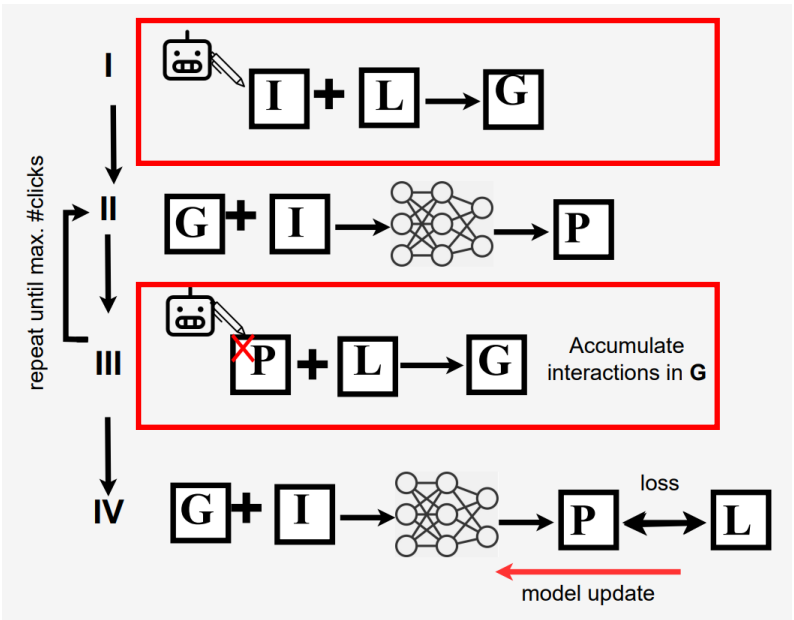
Robot user generates a refinement click in the missegmented area



Robot User

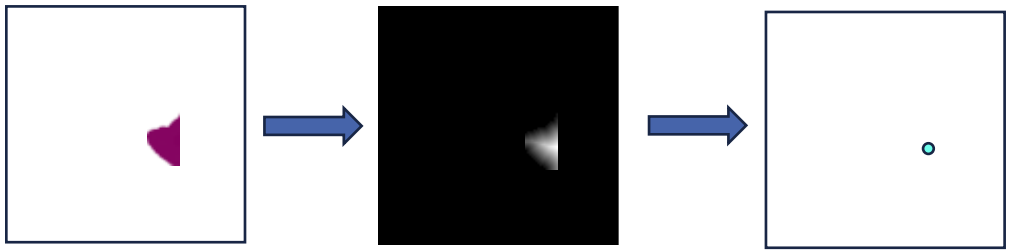
- How does the robot user decide where to put new interactions (steps I and III)?
- For clicks → Most often: center of ground-truth or center of error





Robot User - Clicks

- Many different strategies for generating clicks from ground-truth (GT) and errors
  - Center of GT/error
  - Random click in GT/error
    - Uniform sampling
    - Distance transform-based sampling
      - Distance transform used as prior sampling distribution
      - Click is near the center but not “exactly” in the center
      - Serves as regularization
  - Stratified sampling
    - Sample only clicks near the boundary
    - Sample clicks along longest axis
- The generated clicks can be perturbed
  - Random X,Y shift to simulate user error
    - No user knows where exactly the center is
    - Acts as a regularization



Error | Euclidean Distance Transform | New sampled click

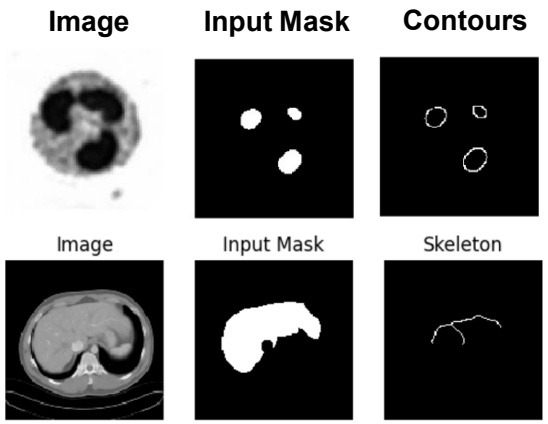
Stratified Long Axis Sampling [23]

Stratified Boundary Sampling [24]

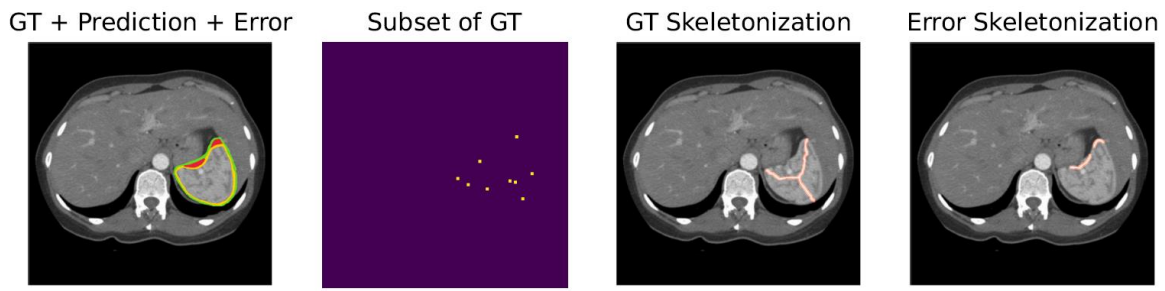
Ground-truth Mask

Robot User - Scribbles

- Most often simulated through **skeletonization** of GT mask
  - **Skeleton** := set of pixels within the mask which are maximally distant from the mask boundary
- Other option:
  - **Boundary contours** → 1-pixel wide boundary of the GT mask



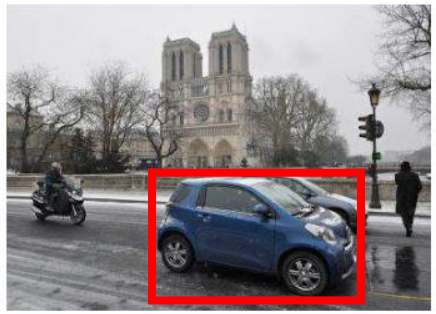
Source: Wong, Hallee E., et al. "ScribblePrompt: Fast and Flexible Interactive Segmentation for Any Medical Image." *arXiv preprint arXiv:2312.07381* (2023).



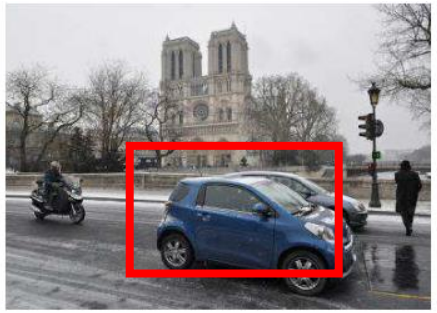
Source: Marinov, Zdravko et al. "Deep Medical Interactive Segmenation: A Systematic Review and Taxonomy" [1]

Robot User – Bounding Box

- Typically, non-iterative
  - Only one prediction step
- Simulated bounding box is either
  - **Perfect** := GT Box
  - **Perturbed** := Randomly shifted in X, Y directions to simulate user error
    - Regularizes the model
  - **Relaxed** := Extended by some margin to include more context



Perfect

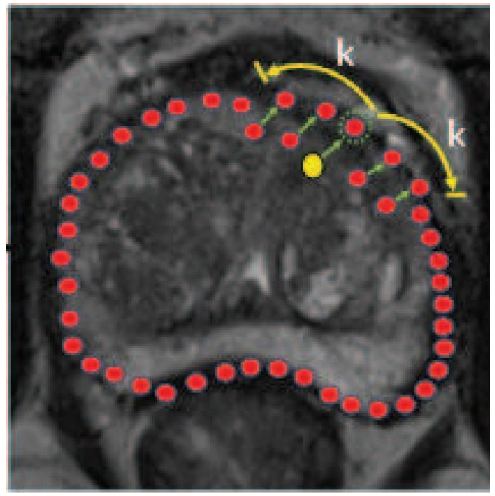


Perturbed

Relaxed

Robot User – Boundary Polygon

- **Worst vertex correction** [25]
  - The vertex with the largest error is “dragged” to the correct position
    - Often with its k-neighbours



Source: Tian, Zhiqiang, et al. "Graph-convolutional-network-based interactive prostate segmentation in MR images." *Medical physics* 47.9 (2020): 4164-4176.

When to use a Robot User?

- During training
  - Simulate interactions for a large amount of images
    - Typically infeasible to do with **real annotators → simulate it**
- During evaluation
  - Simulate an annotator using the interactive model
    - Instead of conducting time-intensive and expensive annotation studies

When to use a Robot User? (3 min)

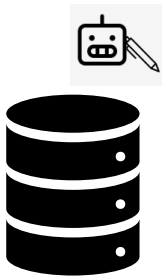
- During training
  - Simulate interactions for a large amount of images
    - Typically infeasible to do with **real annotators → simulate it**
- During evaluation
  - Simulate an annotator using the interactive model
    - Instead of conducting time-intensive and expensive annotation studies
    - What is a potential problem when doing this?

Evaluation with Robot User

Two routes to evaluating an interactive model

Conduct a user study with **real human experts**

- Usually on a small sample size and few annotators
- The evaluated performance is the REAL performance when used by experts



Simulate the interactions on a test dataset

- Usually the test split of a public dataset
- Since it is simulated, sample size can be quite large
- Most current approaches go down this route and use a Robot User during evaluation

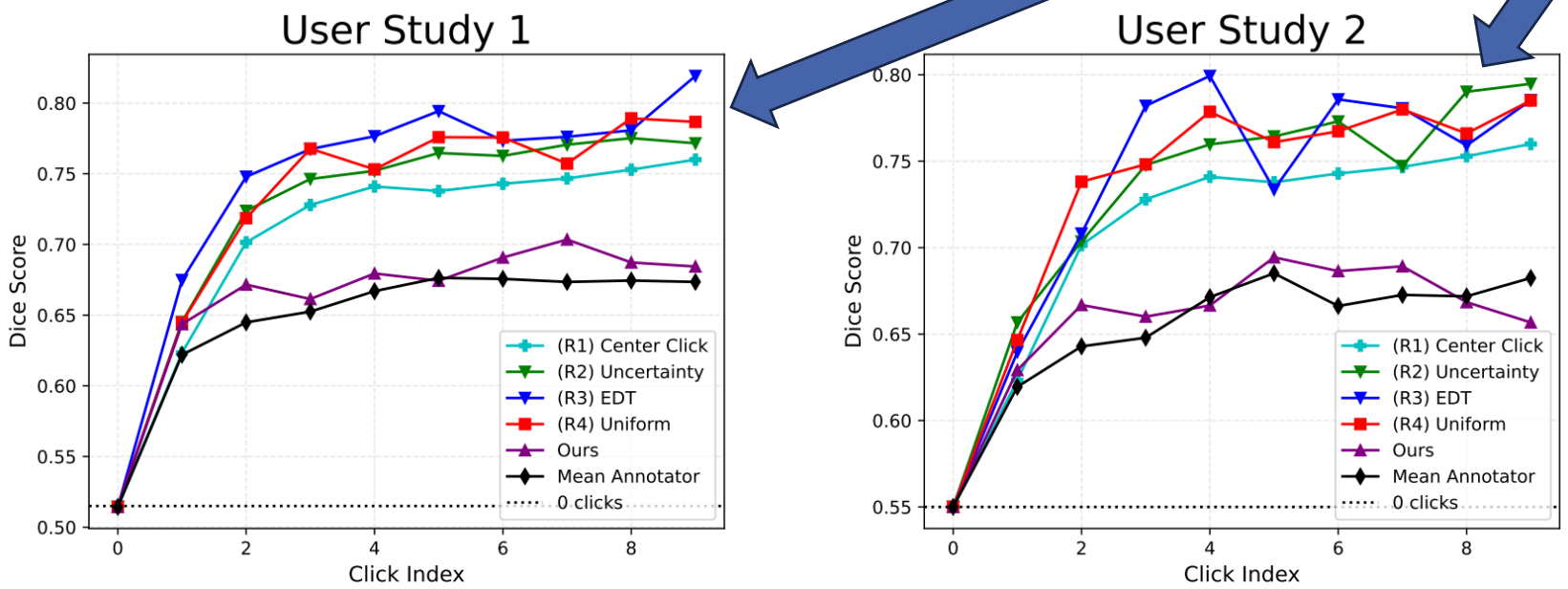
Are these simulations realistic?
Would the results be the same as a real user study?

Evaluation with Robot User

- Comparison of 4 popular robot users and a real user study
  - 8 medical annotators from various backgrounds
  - External collaboration with University Clinic Essen, AI for Medicine (IKIM)
- Dataset: AutoPET, n=20 volumes

THIS is what you would report if you use off-the-shelf simulation methods

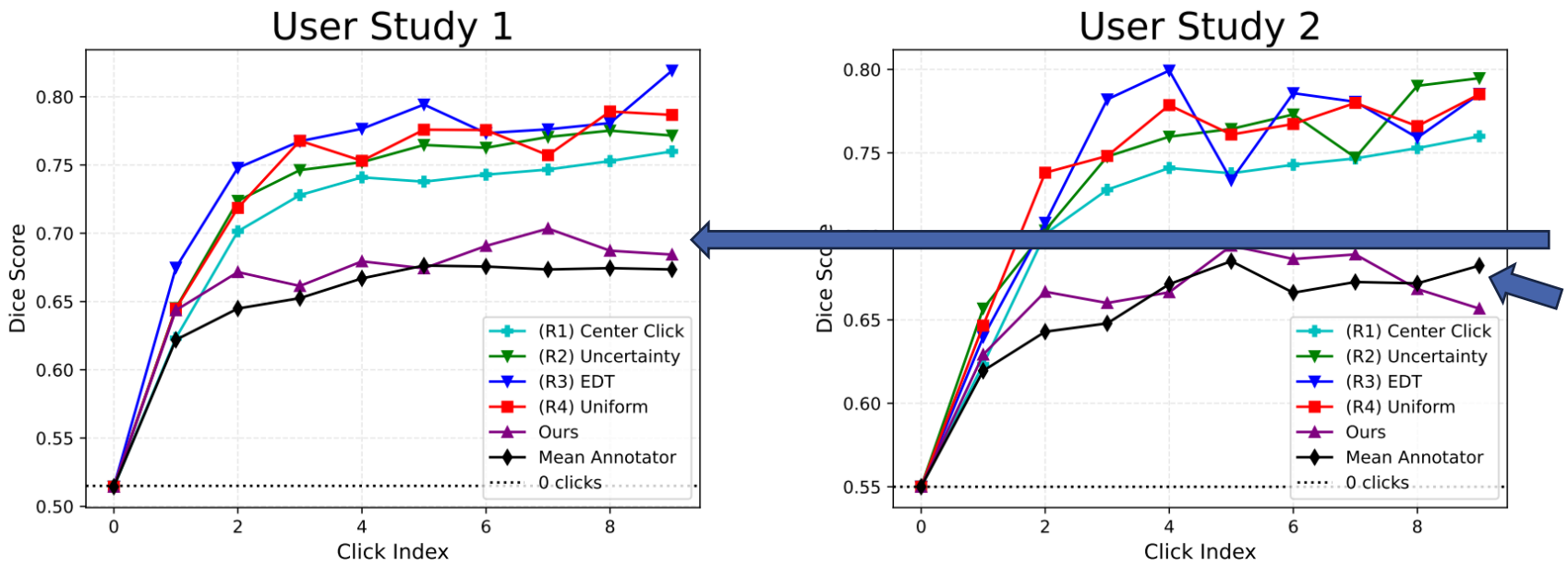
"Our method is SOTA and is awesome and great! Just look at the numbers!"



Marinov, Zdravko, et al. "Rethinking Annotator Simulation: Realistic Evaluation of Whole-Body PET Lesion Interactive Segmentation Methods." MICCAI 2024.

Evaluation with Robot User

- Comparison of 4 popular robot users and a real user study
  - 8 medical annotators from various backgrounds
  - External collaboration with University Clinic Essen, AI for Medicine (IKIM)
- Dataset: AutoPET, n=20 volumes



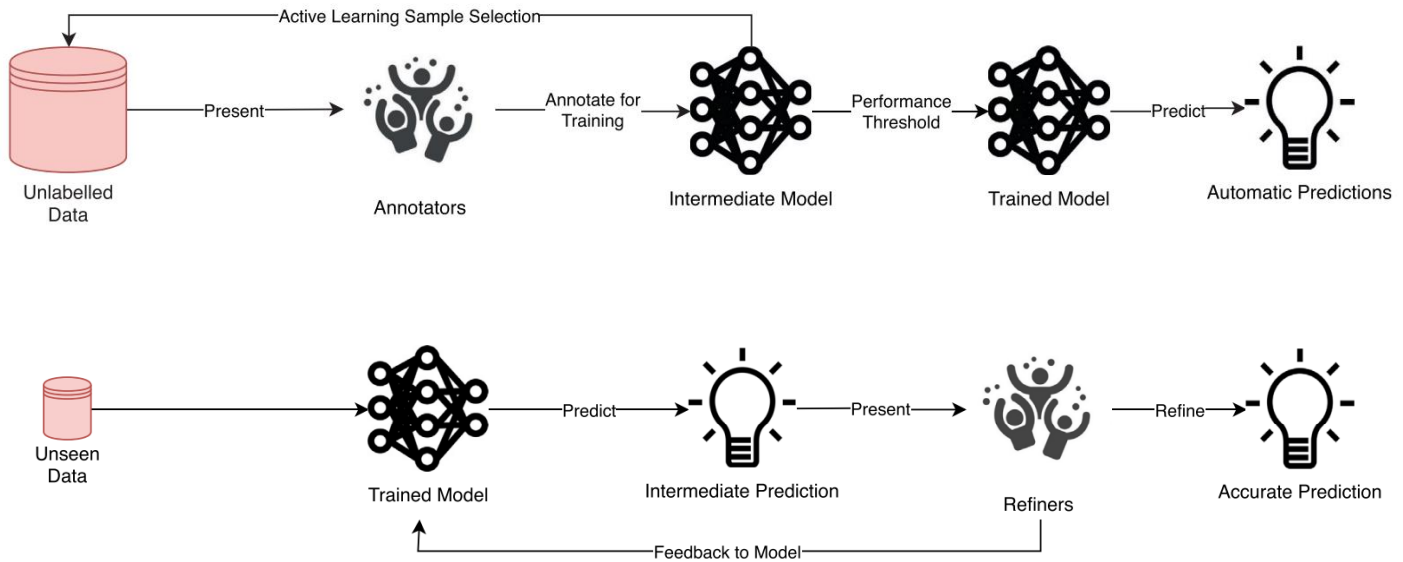
THIS (black lines) is what is actually the performance in a real user study (same model, same data)

Marinov, Zdravko, et al. "Rethinking Annotator Simulation: Realistic Evaluation of Whole-Body PET Lesion Interactive Segmentation Methods." MICCAI 2024.

ACTIVE LEARNING

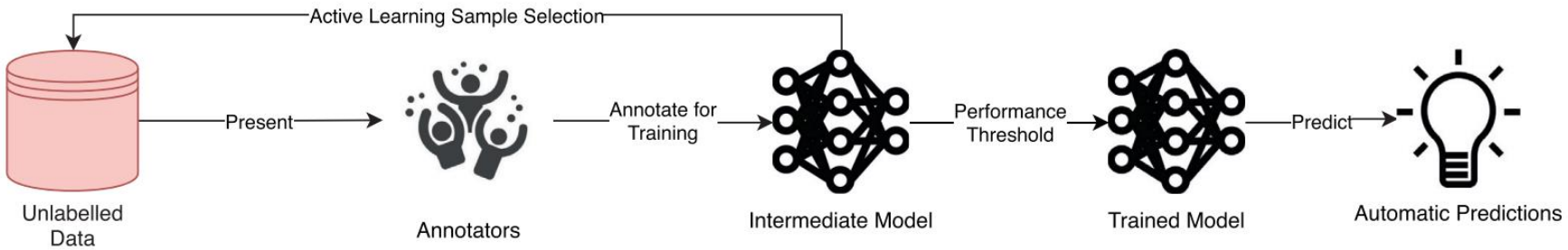
Active Learning for Interactive Segmentation [26]

- Two types of active learning
  - Focused on data annotation
  - Focused on model refinement



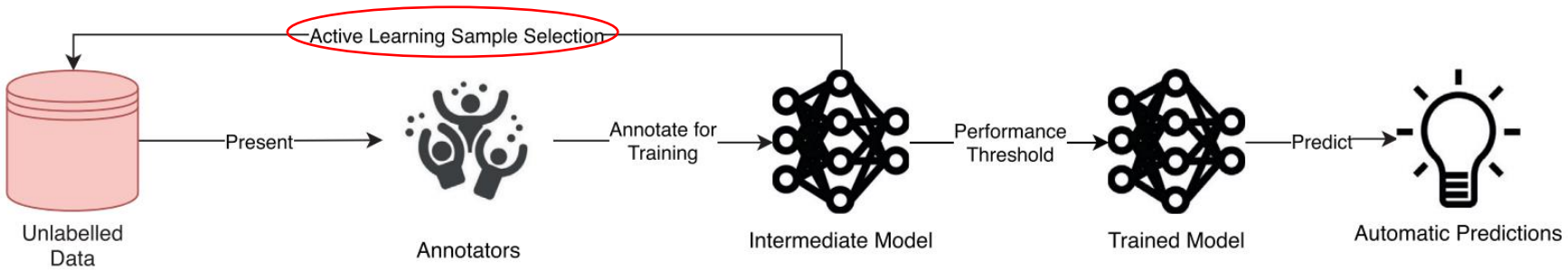
Active Learning for Data Annotation [26]

- Annotators label a few samples (annotation budget) with interactive segmentation
  - Model is trained on annotation budget
  - Predicts on the rest of unlabelled data
  - “Most informative samples” are selected for further annotation and added to annotation budget
  - All steps are repeated until the model reaches a certain performance on an independent test dataset
- In the end:
  - Most important samples are labelled and can be used for model training
  - A model is already trained well and can be deployed



Most Informative Sample

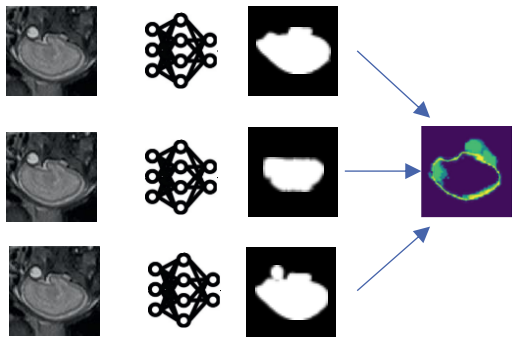
- What is an “informative” sample?
- **Informative** := Adding annotations to it would benefit the model training
  - Idea := Select only top N informative samples instead of all unannotated samples
- Metrics for informativeness:
  - Most often associated with prediction uncertainty [27]
    - Ensembles
    - MC-Dropout
    - Test-time augmentation
    - Auxiliary network



Most Informative Sample [28]

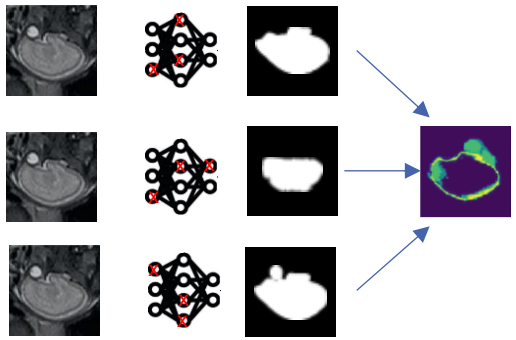
- Most often associated with prediction uncertainty [27]
  - Ensembles
  - MC-Dropout
  - Test-time augmentation
  - Auxiliary network

Ensembles



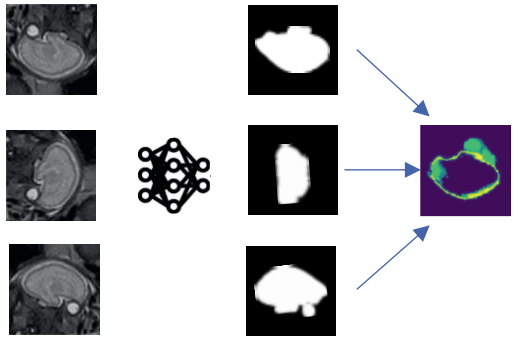
Variance of model predictions
Ensemble of **3 different** models

MC-Dropout



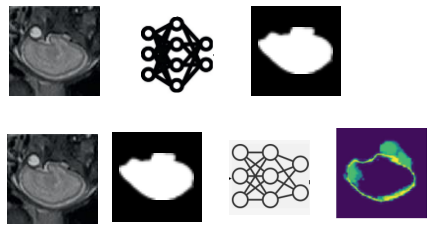
Variance of model predictions
Predictions from same model
with different “deactivated” weights

Test-time augmentation



Variance of model predictions
Predictions from same model
with same augmented input

Auxiliary network

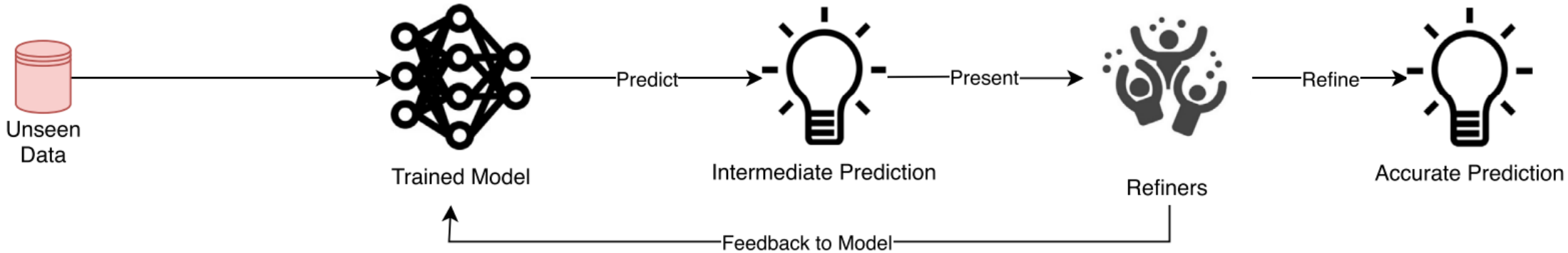


Train a special auxiliary network
to predict the uncertainty

Image Source: Wang, Guotai, et al. "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks." *Neurocomputing* 338 (2019): 34-45.

Active Learning for Model Refinement [26]

- Goal: Improve model, not annotate more data
- Same sampling selection strategies as for Data Annotation
  - Most informative samples or hard-sample mining (worst model performance)
- Final goal is to deploy a robust model

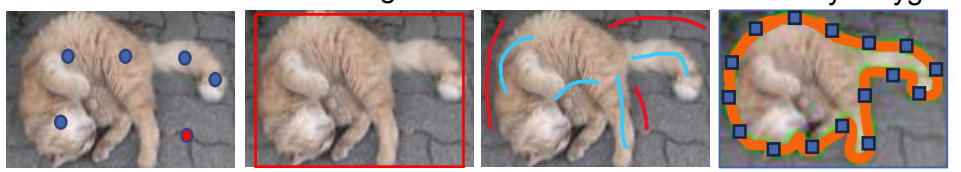


CONSTRUCTING THE INTERACTIVE PIPELINE

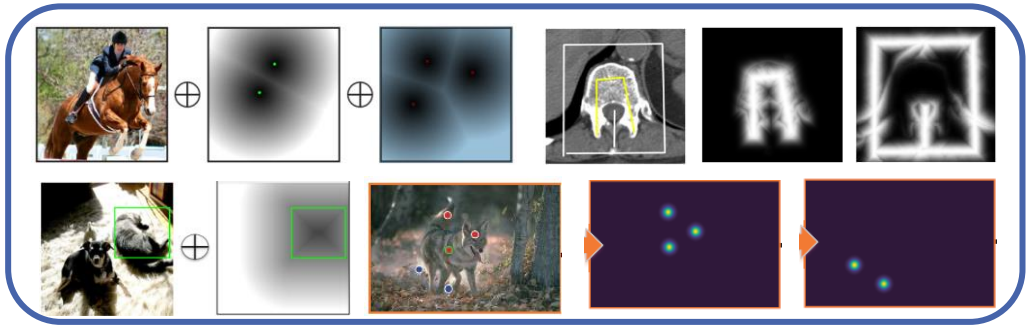
Deep Learning-based Interactive Segmentation

All components are here… Let's build an interactive model together!

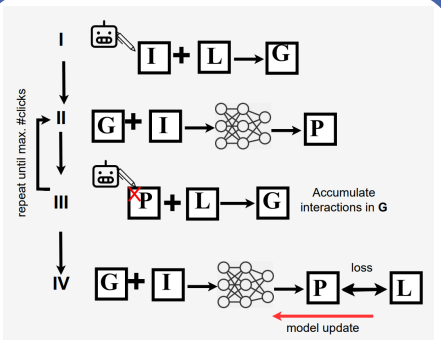
Types of Interactions



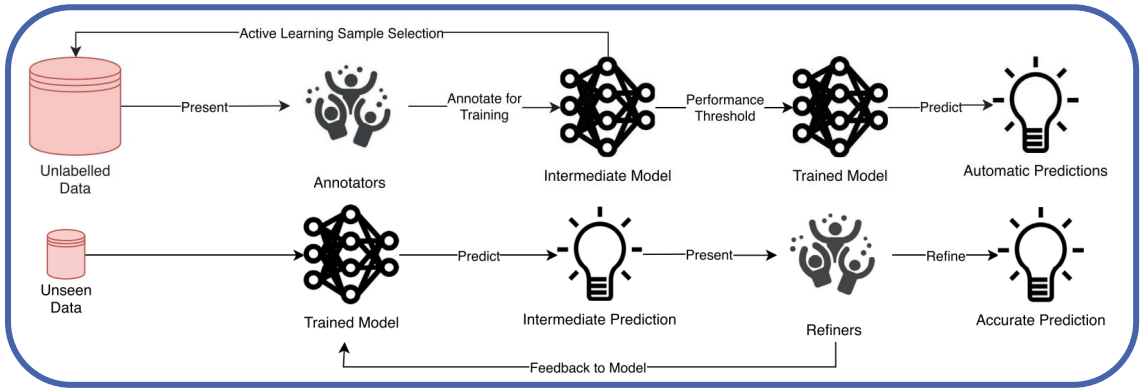
Guidance Signal



Robot User



Active Learning



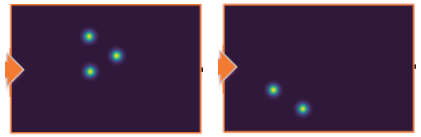
Working Example

All components are here… Let's build an interactive model together!

Type of Interaction

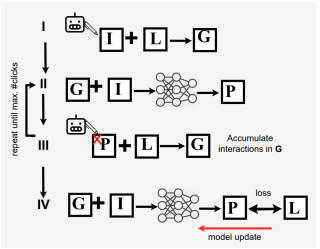
Clicks

Guidance Signal



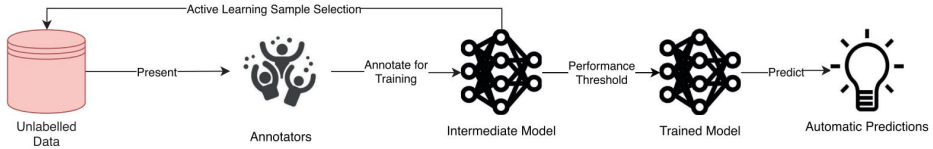
Gaussian Heatmaps

Robot User



Center of Object (I) / Error (III)

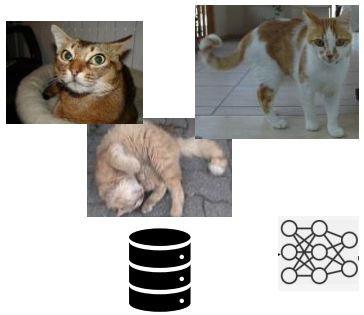
Active Learning



Focused on Data Annotation
Sample Selection: MC Dropout

Working Example

Training Phase

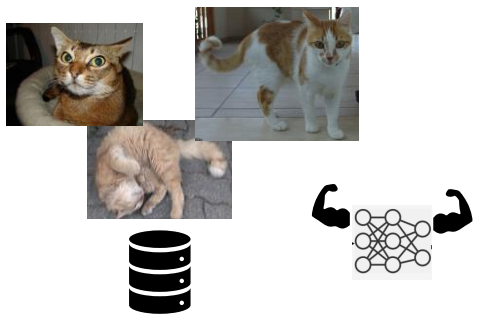


100 labelled cats
90 train / 10 test

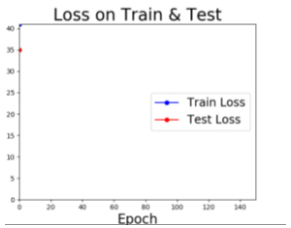
Cat Images: https://www.kaggle.com/datasets/crawford/cat-dataset

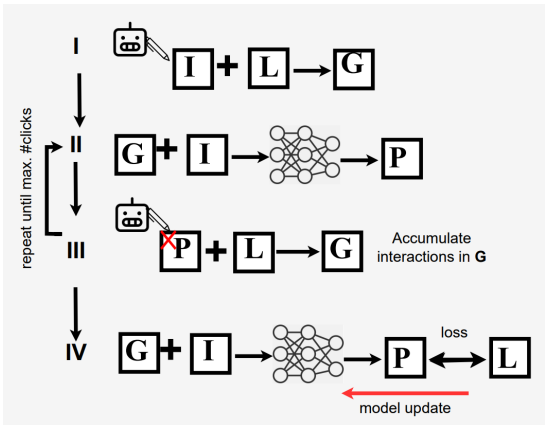
Working Example

Training Phase



100 labelled cats
90 train / 10 test





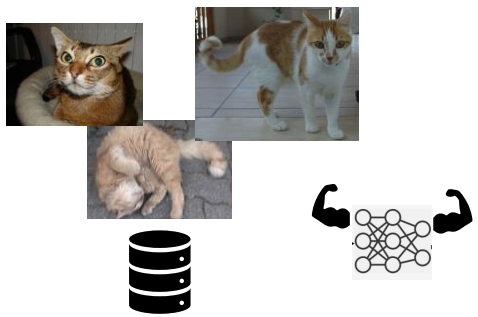
Simulate central clicks for steps I and III
For each cat image: Simulate 10 iterative clicks
Update model after the 10th click

10 x 100 = 1000 forward passes per epoch!
Interactive training is inherently slow!
(linearly slower than non-interactive)

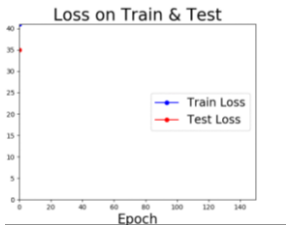
Cat Images: https://www.kaggle.com/datasets/crawford/cat-dataset

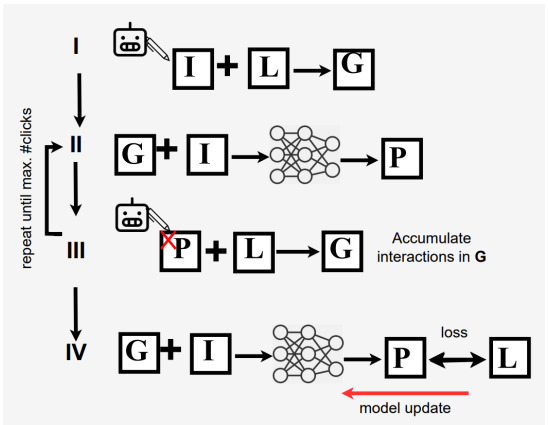
Working Example

Training Phase



100 labelled cats
90 train / 10 test

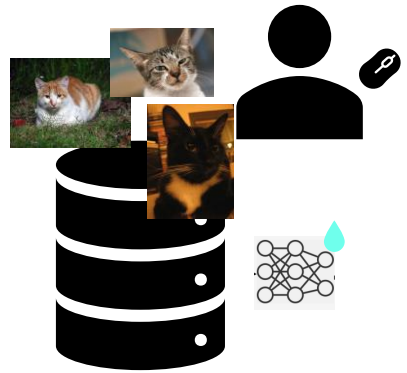




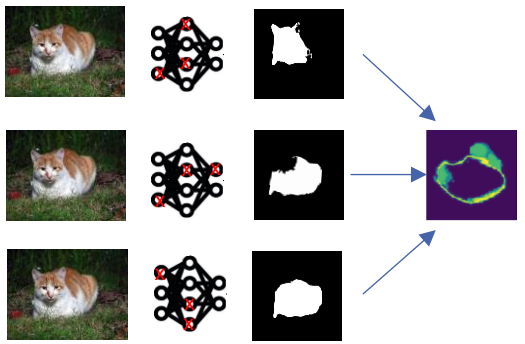
Simulate central clicks for steps I and III
For each cat image: Simulate 10 iterative clicks
Update model after the 10th click

10 x 100 = 1000 forward passes per epoch!
Interactive training is inherently slow!
(linearly slower than non-interactive)

Application Phase



10.000 unlabelled cats



Compute Uncertainty for all 10.000 images

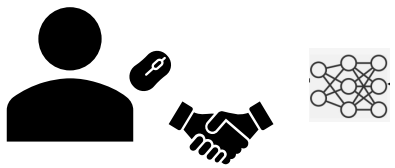
Provide top 50 for interactive annotation with clicks
(Annotator can make as many clicks as wanted, not exactly 10)

Fine-tune model with new 50 annotations

Cat Images: https://www.kaggle.com/datasets/crawford/cat-dataset

Working Example

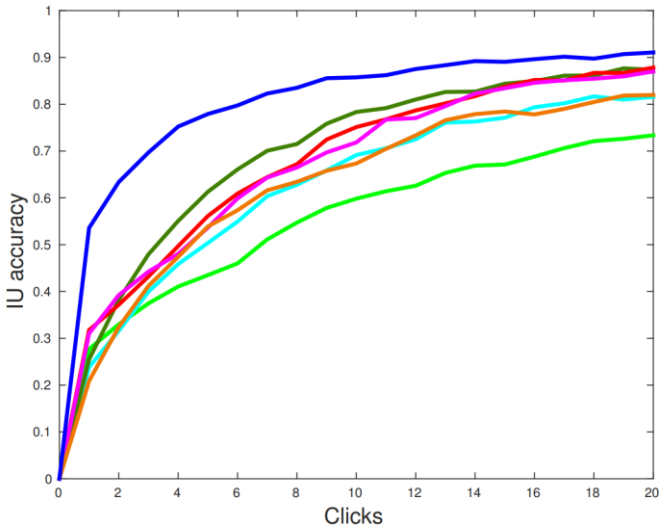
- After annotating 500 / 10.000 cats, the model starts to produce almost perfect segmentations for all cats!
  - We have both a good model for cats
  - and 10.000 annotated cats for more training



Cat Images: https://www.kaggle.com/datasets/crawford/cat-dataset
Demo model: https://segment-anything.com/demo

Evaluation Metrics

- Segmentation Performance
  - **NoC@90**
    - Number of Clicks (NoC) at 90% performance (typically IoU or Dice)
  - **IoU@10** or **Dice @10**
    - IoU or Dice at 10 clicks
  - **NoC / IoU** curves
  - **NoC / Dice** curves
  - **Consistent Improvement** (CI)
    - % of iterations where adding an interaction improves the segmentation



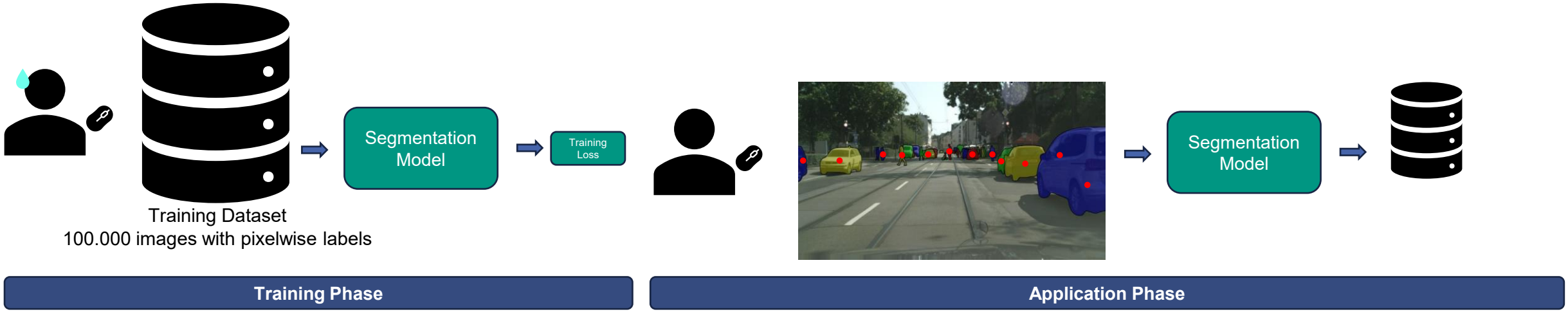
Evaluation Metrics

- Usability
  - **User Time**
    - Time it takes to annotate an image in seconds
  - **Machine Time**
    - Inference time for an image
  - **Scribble Length**
    - Mean number of pixels in scribbles
  - **NASA-TLX** Score
    - Perceived workload in terms of mental demand, frustration etc.
  - **System Usability Scale** (SUS)
    - Likert-scale questionnaire to quantify usability

APPLICATION IN MEDICAL IMAGE ANALYSIS

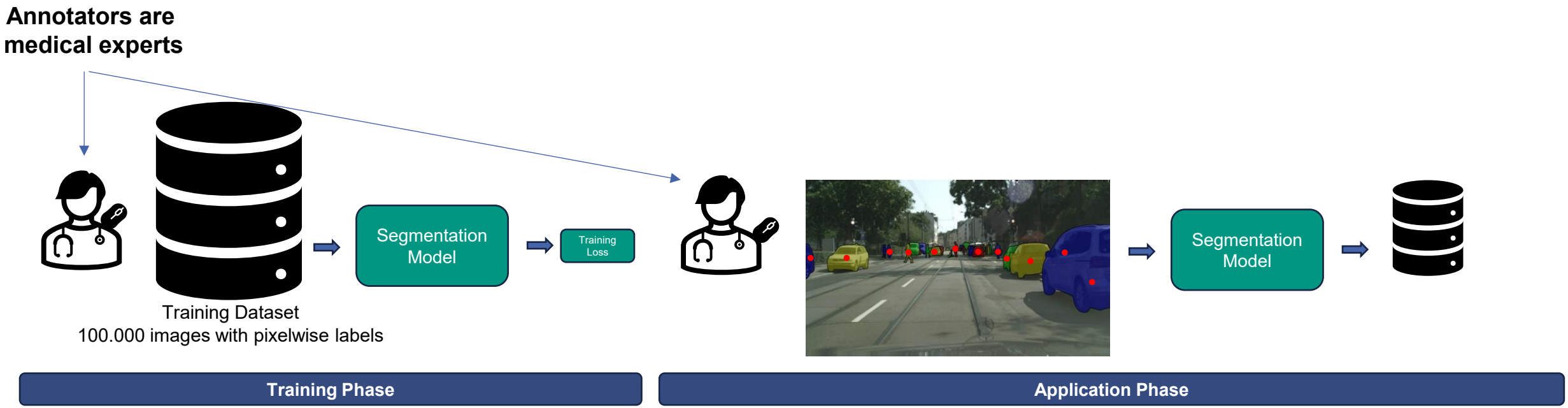
Application in Medical Image Analysis

- Similar to natural images
  - With a few differences



Application in Medical Image Analysis

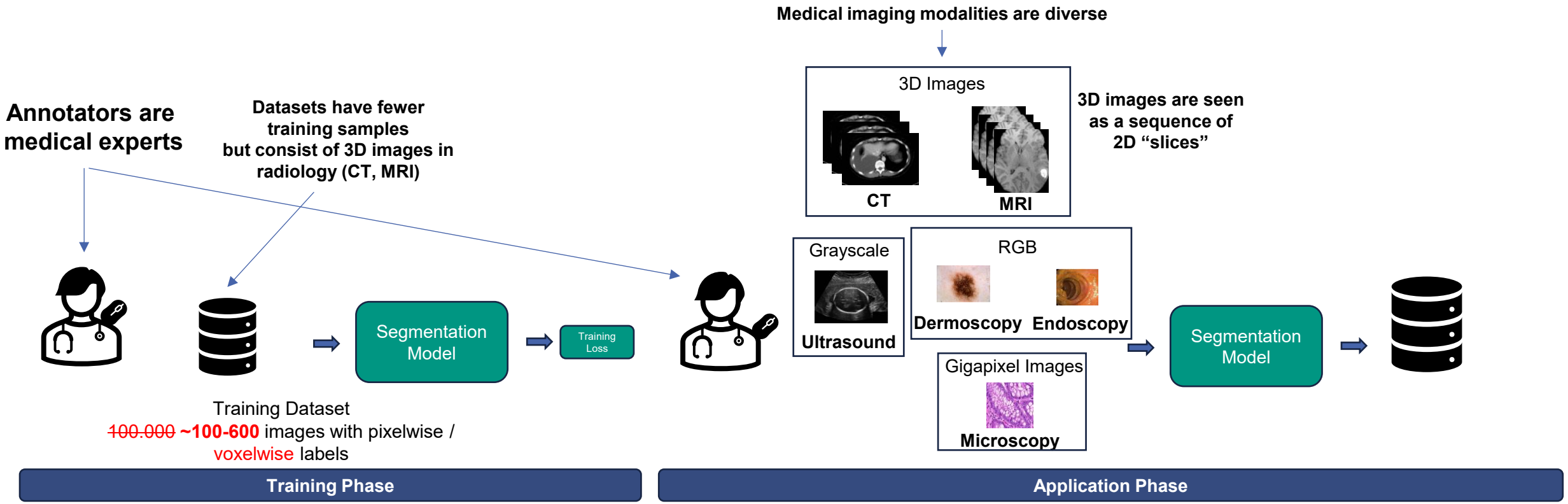
Differences



Application in Medical Image Analysis

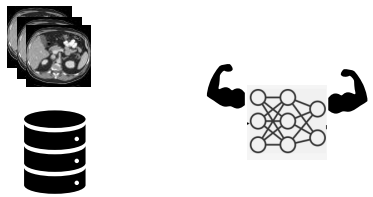
CT: Computed tomography
MRI: Magnetic Resonance Imaging

Differences

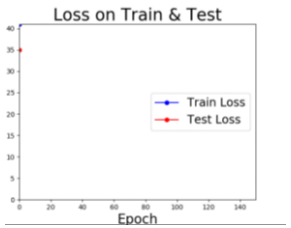


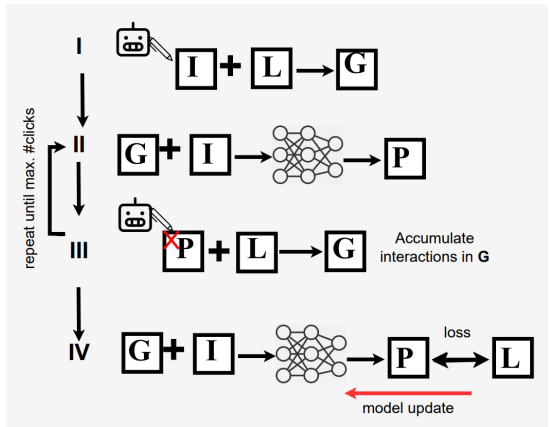
Medical Data Example: 3D CT Images

Training Phase



100 labelled **livers in CT**
90 train / 10 test

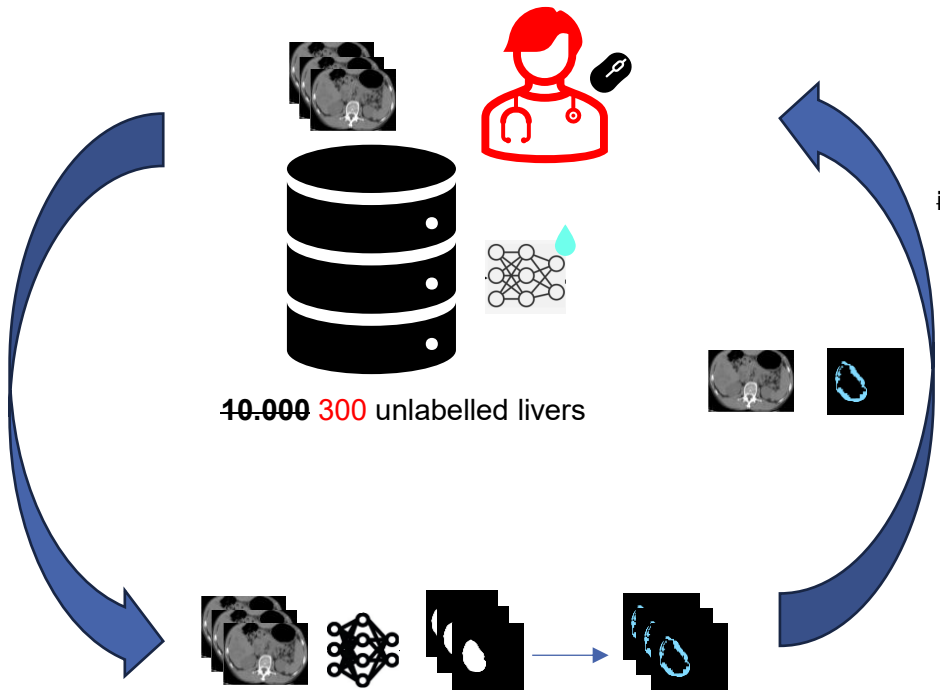




Simulate central clicks for steps I and III
For each cat image: Simulate 10 iterative clicks
Update model after the 10th click

10 x 100 = 1000 forward passes per epoch!
Interactive training is inherently slow!
(linearly slower than non-interactive)

Application Phase



~~10.000~~ 300 unlabelled livers

~~Provide top 50 for interactive annotation with clicks~~
~~Fine-tune model with new 50 annotations~~

Provide slice with highest uncertainty

Update prediction but keep model frozen
(updates usually corrupt the model)

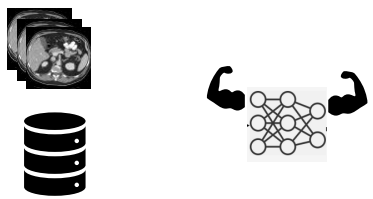
Repeat until clinician is satisfied and then move on to next CT volume

Compute Uncertainty for ~~all 10.000 images~~ all slices of the current CT image

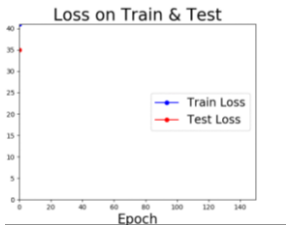
CT Images: https://github.com/bowang-lab/MedSAM/blob/main/assets/MedSAM_supp.pdf

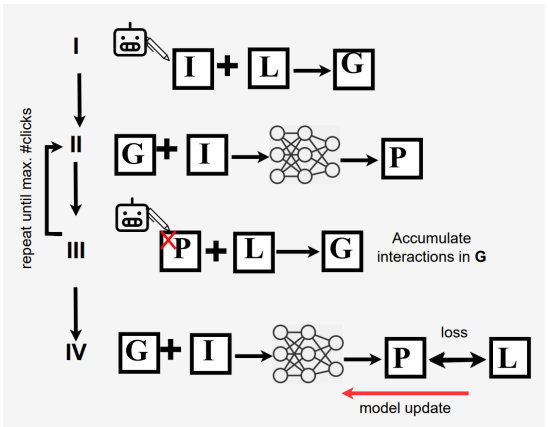
Medical Data Example: 3D CT Images

Training Phase



100 labelled **livers in CT**
90 train / 10 test

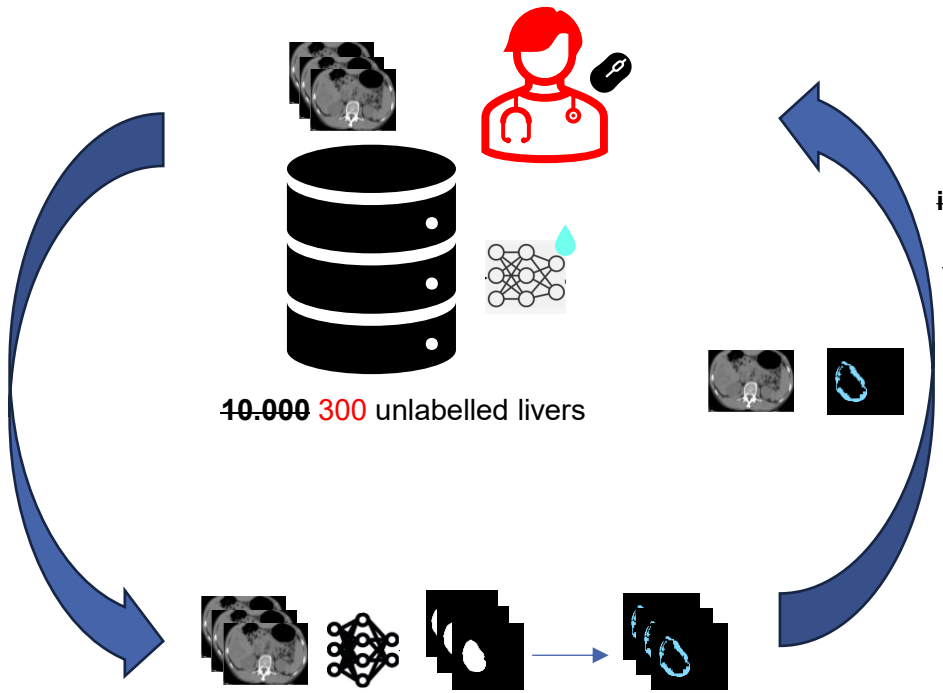




Simulate central clicks for steps I and III
For each cat image: Simulate 10 iterative clicks
Update model after the 10th click

10 x 100 = 1000 forward passes per epoch!
Interactive training is inherently slow!
(linearly slower than non-interactive)

Application Phase



~~10.000~~ 300 unlabelled livers

~~Provide top 50 for interactive annotation with clicks~~
~~Fine-tune model with new 50 annotations~~

Provide slice with highest uncertainty

Update prediction but keep model frozen
(updates usually corrupt the model)

Repeat until clinician is satisfied and then move on to next CT volume

Compute Uncertainty for ~~all 10.000 images~~ all slices of the current CT image

CT Images: https://github.com/bowang-lab/MedSAM/blob/main/assets/MedSAM_supp.pdf

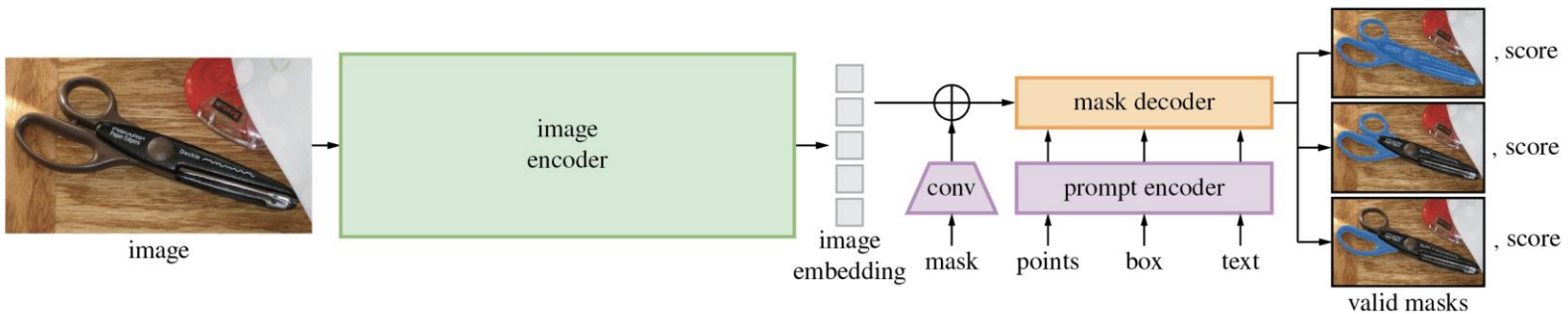
SEGMENT ANYTHING MODEL (SAM)

Segment Anything Model (SAM)

- Interactive Segmentation at a scale
  - Pre-trained on 1 billion masks from 11 million images
  - First model towards a foundation segmentation model
- Great zero-shot performance
  - Notion of “thing”
- Interaction Types
  - Click, Bounding Box, Text

SAM Architecture – Components

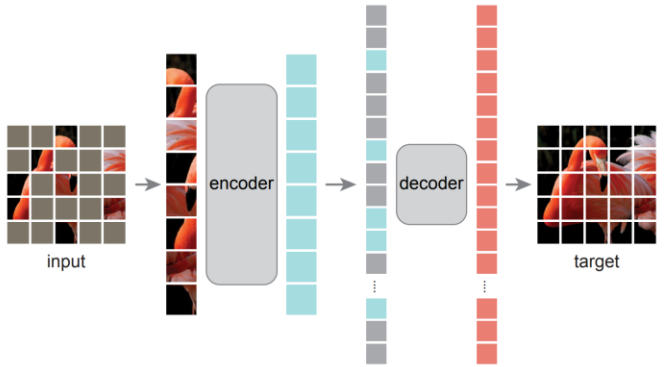
- Image encoder
- Mask encoder
- Prompt encoder
- Mask decoder



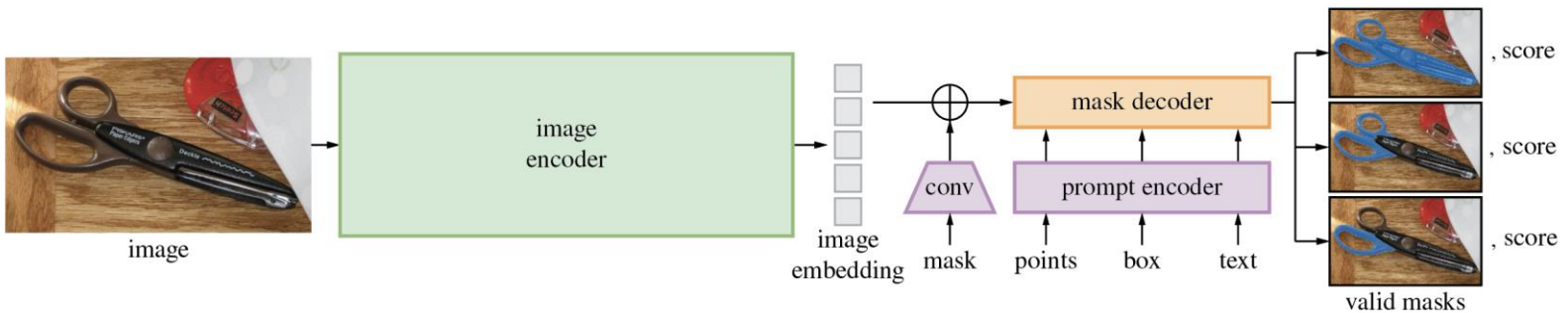
SAM Architecture [9]

SAM Architecture – Image Encoder

- Image encoder
  - Pre-trained Masked Autoencoder ViT-H [29]
  - 256-dimensional representation of the image
  - Large and heavy but powerful
    - Needs to be ran only once per image!



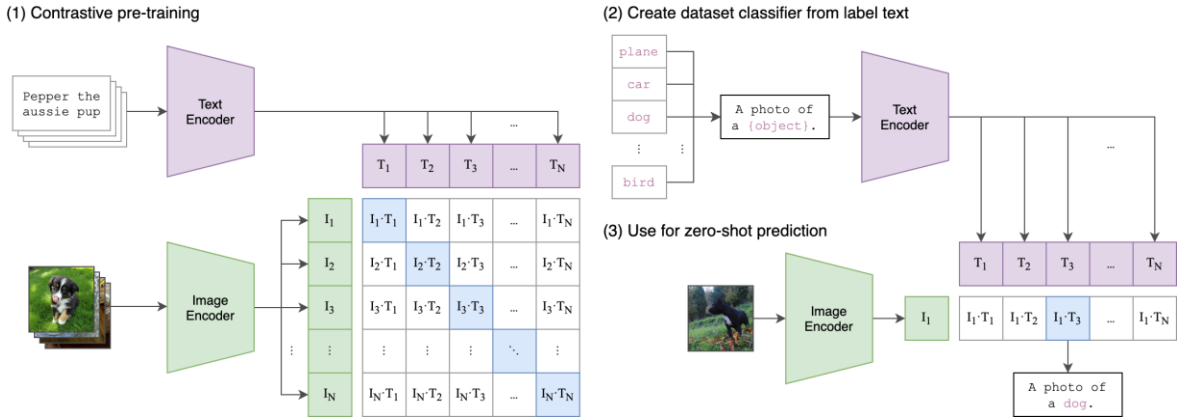
Masked Autoencoder Training [29]



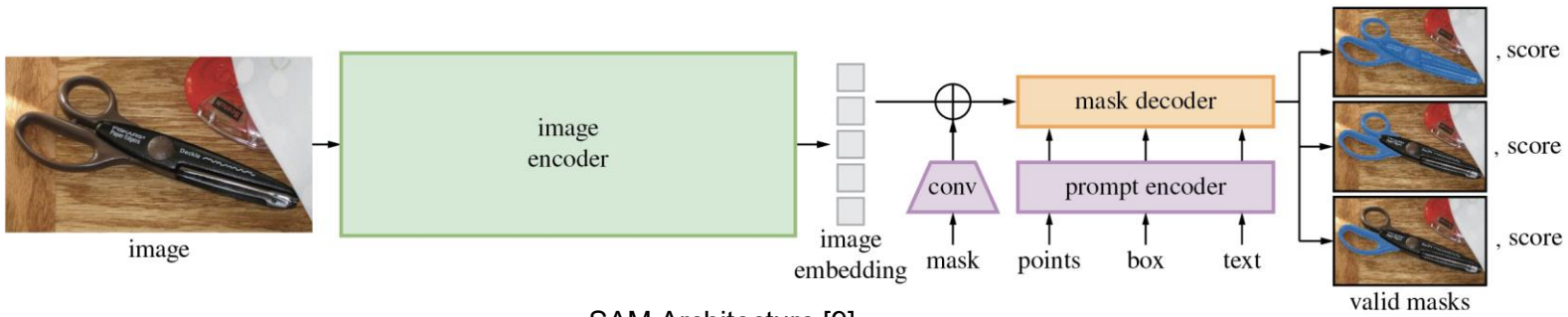
SAM Architecture [9]

SAM Architecture – Prompt Encoder

- Prompt encoder
  - Text
    - CLIP text encoder [30]



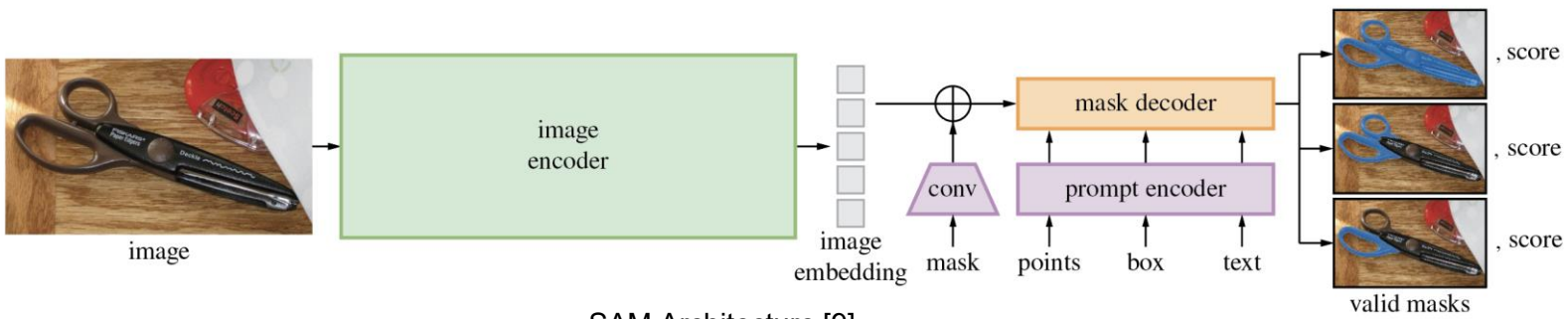
Reminder: CLIP [30]



SAM Architecture [9]

SAM Architecture – Prompt Encoder

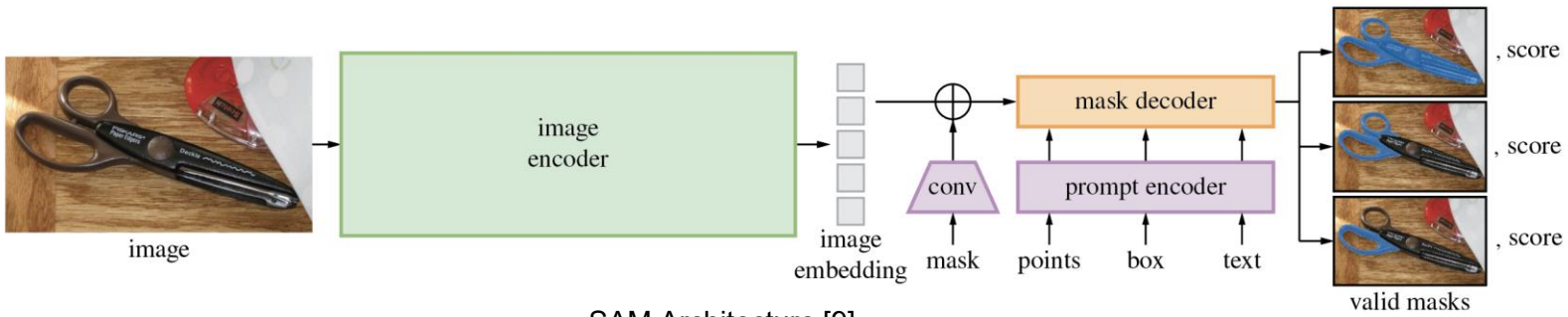
- Prompt encoder
  - Points (x, y) + Label (foreground / background)
    - (x, y) → Positional Encoding (256-dim) → Summed with “foreground” or “background” learnable weights
  - Bounding Box
    - $(x, y)_{\text{top left}}$ $(x, y)_{\text{bottom right}}$
      - Positional embeddings → Summed with “top-left” and “bottom-right” learnable weights
- SAM learns internally how to understand these prompts!
  - Positional encoding → Spatial location
  - “foreground”, “background”, “top-left”, “bottom-right” weights → *Prompt meaning*



SAM Architecture [9]

SAM Architecture – Mask Encoder

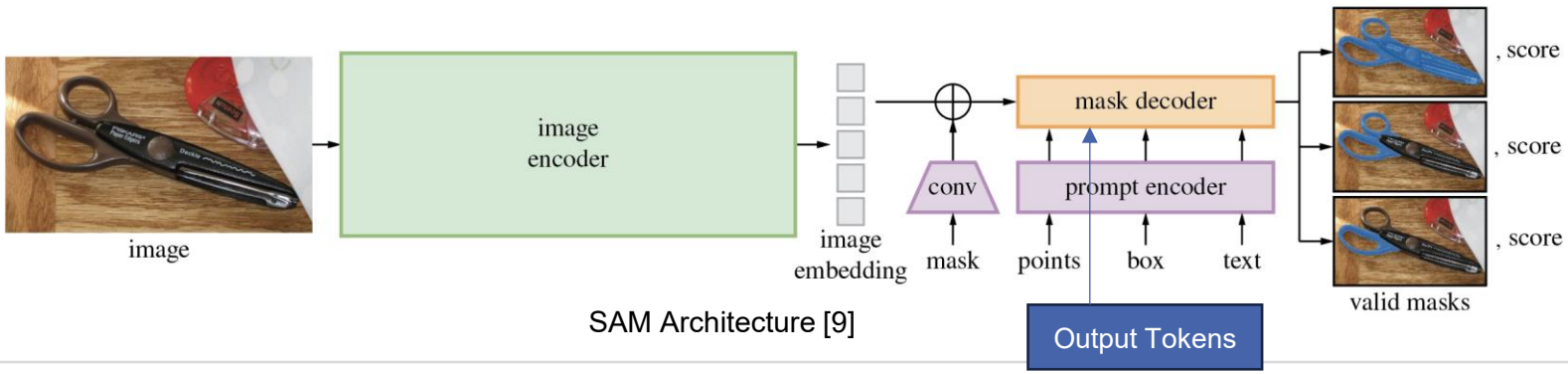
- Mask encoder
  - 2D CNN with downscaling
  - Special “no-mask” embedding
  - Point-wise sum with image embedding (+)



SAM Architecture [9]

SAM Architecture – Mask Decoder

- Really light-weight → Must be fast and interactive!
  - Encoder is slow (but powerful) but is ran only once over the image
- Inputs
  - Prompts
    - **Output tokens (+) Prompt tokens**
      - Output tokens: Force model to put the output in this token
        - 3 segmentation mask predictions
        - IoU per mask
  - Image (+) Mask

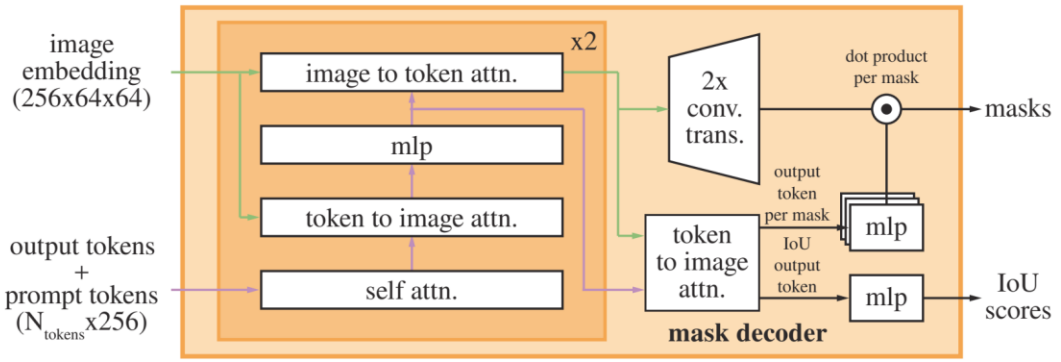


SAM Architecture [9]

Not shown in original image (omitted for clarity)

SAM Architecture – Mask Decoder

- Forward Pass
  - Self-attention
    - [Prompt tokens (+) Output tokens]
  - Cross-attention (token-to-image)
    - Query: Tokens
    - Keys, Values: Image Embedding
  - Linear MLP
  - Cross-attention (image-to-token)
    - Query: Image Embeddings
    - Keys, Values: Tokens
- Why 2 Cross-attention blocks?
  - 2 outputs from decoder
    - Attended sequence of prompt tokens → Used to produce the output tokens and predicted IoU scores
    - Attended Image embedding → Combined with output tokens to produce the 3 masks



SAM Decoder Architecture [9]

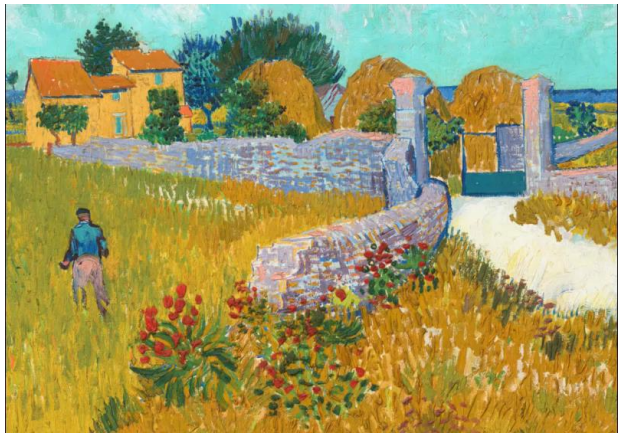
SA-1B Dataset

- 1 billion masks from 11 million images
- 4 stages
  - **Small interactive pre-training on public datasets**
    - Robot user
      - 50% chance → uniform random sampling of 8 iterative clicks
      - 50% chance → GT bounding box +- some perturbation
  - **Manual**
    - Warm-up interactive annotation with clicks/bounding boxes
      - Manual corrections to predictions
    - Fine-tune SAM on its own (corrected) predictions
  - **Semi-automatic**
    - SAM is applied to whole dataset
      - Most confident predictions are filled out and shown to annotators
      - Annotators fill out additional low-confidence objects
  - **Automatic**
    - 32x32 points grid of image are used as clicks
    - 1024 predictions are aggregated to form more stable masks

SAM Automatic Stage

SAM Examples

- General notion of a “thing”
- Associates spatial with semantic context

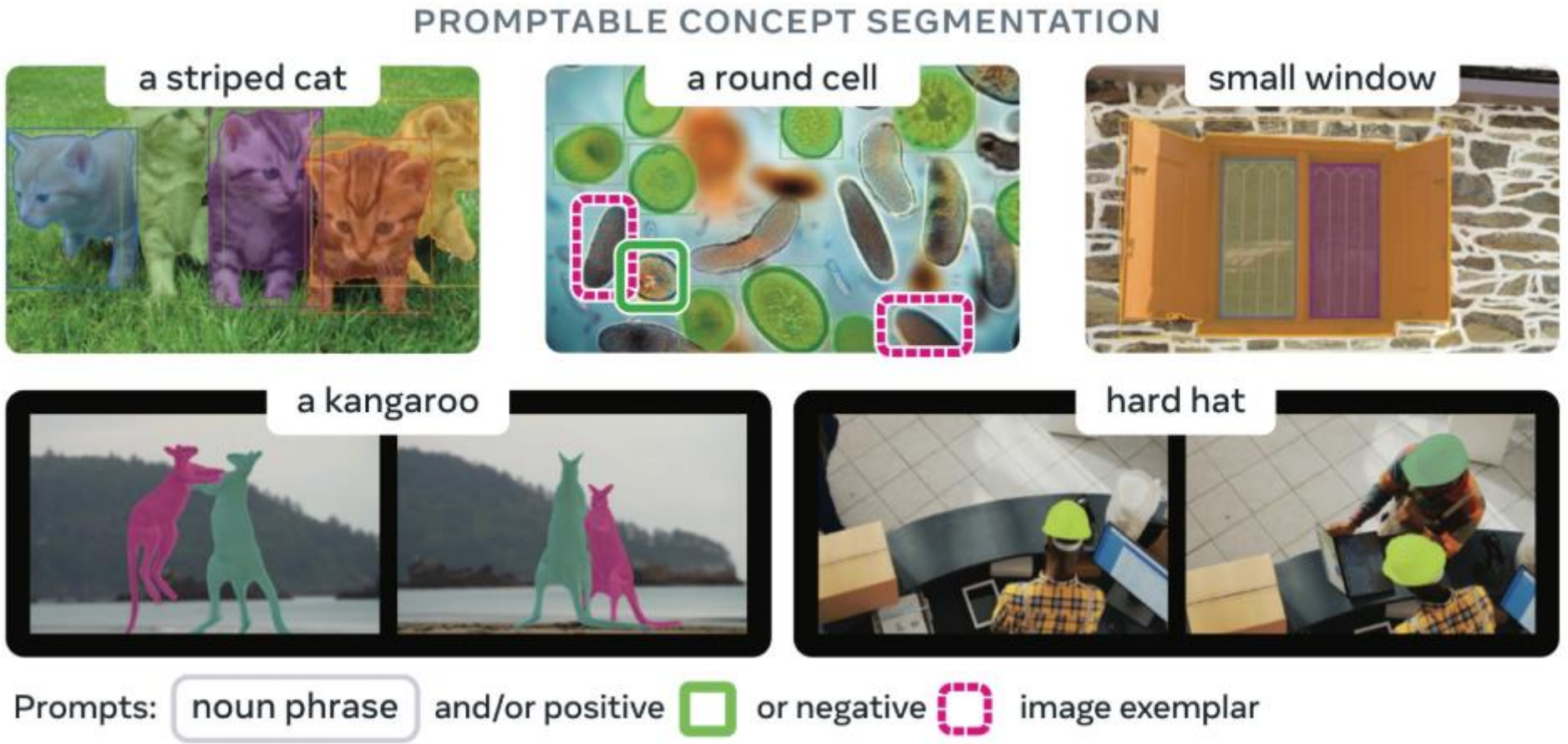


Examples from https://segment-anything.com/

SAM2 – Videos!

Source: https://ai.meta.com/sam2/

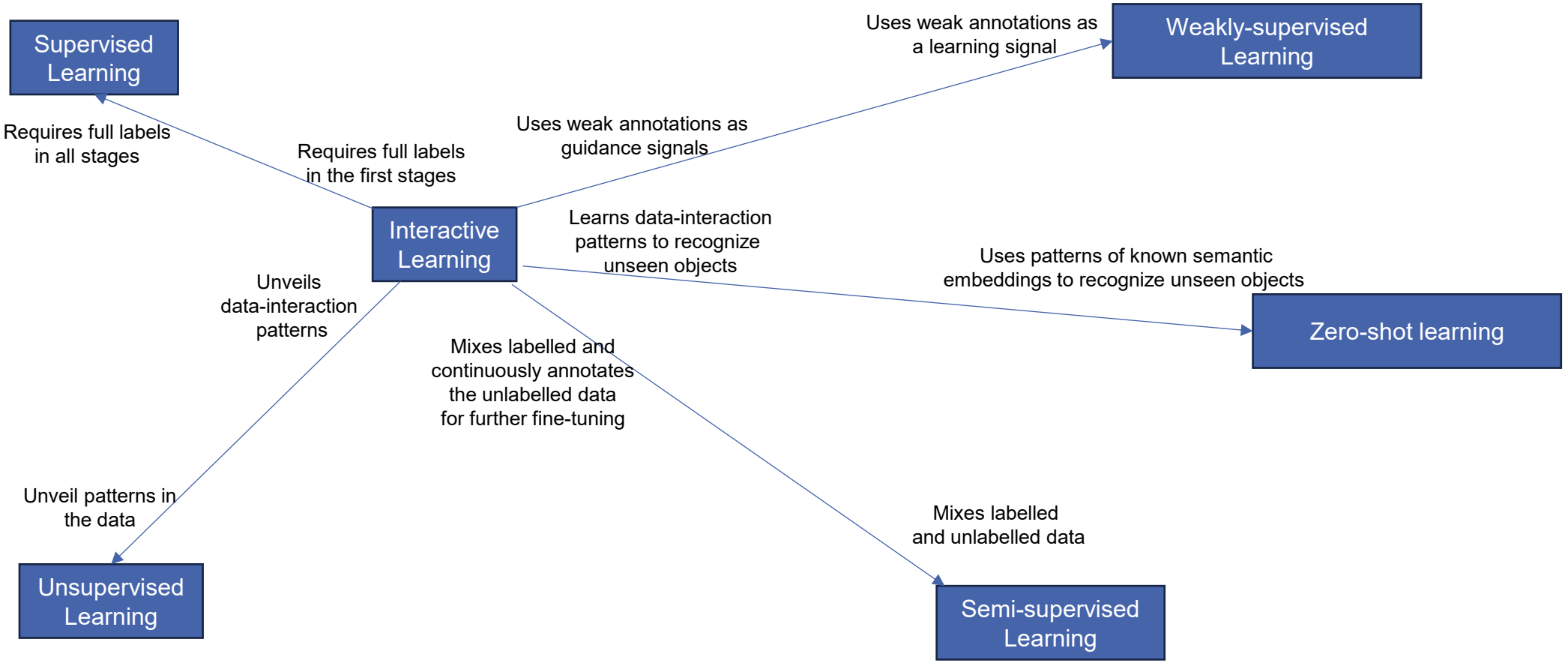
SAM3 - Concepts! (coming in the next weeks)



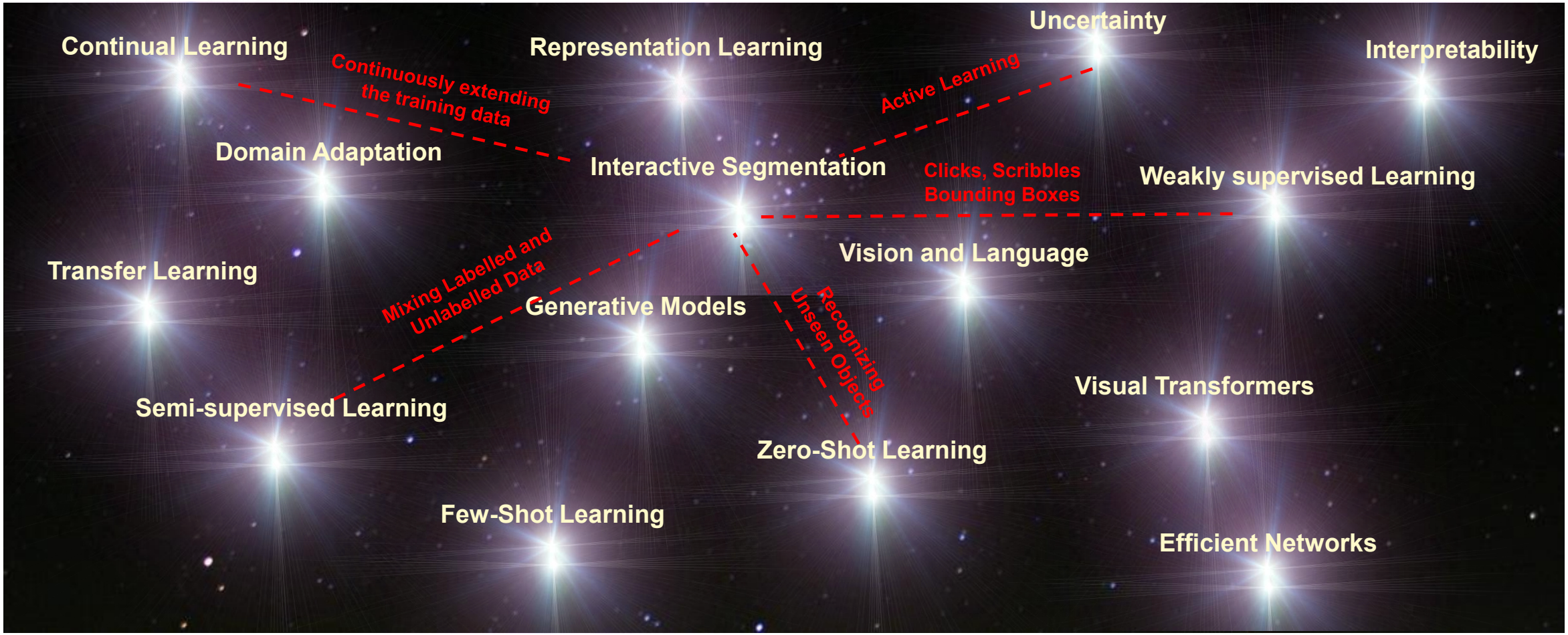
Source: https://docs.ultralytics.com/models/sam-3/

DIFFERENCES TO OTHER LEARNING PARADIGMS

Differences to Other Paradigms



Constellations in Interactive Segmentation



References

- [1] Marinov, Zdravko, et al. "Deep Interactive Segmentation of Medical Images: A Systematic Review and Taxonomy." *arXiv preprint arXiv:2311.13964* (2023).
- [2] Jain, Suyog Dutt, and Kristen Grauman. "Click carving: Interactive object segmentation in images and videos with point clicks." *International Journal of Computer Vision* 127 (2019): 1321-1344.
- [3] Kass, Michael, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active contour models." *International journal of computer vision* 1.4 (1988): 321-331.
- [4] Boykov, Yuri Y., and M-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images." *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*. Vol. 1. IEEE, 2001.
- [5] Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [6] Sofiiuk, Konstantin, Ilya A. Petrov, and Anton Konushin. "Reviving iterative training with mask guidance for interactive segmentation." *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022.
- [7] Hadlich, Matthias, et al. "Sliding Window FastEdit: A Framework for Lesion Annotation in Whole-body PET Images." *arXiv preprint arXiv:2311.14482* (2023).
- [8] Wong, Hallee E., et al. "ScribblePrompt: Fast and Flexible Interactive Segmentation for Any Medical Image." *arXiv preprint arXiv:2312.07381* (2023).
- [9] Kirillov, Alexander, et al. "Segment anything." *arXiv preprint arXiv:2304.02643* (2023).
- [10] Liu, Qin, et al. "Simpleclick: Interactive image segmentation with simple vision transformers." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.
- [11] Ding, Henghui, et al. "Phraseclick: toward achieving flexible interactive segmentation by phrase and click." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer International Publishing, 2020.
- [12] Li, Zhuwen, Qifeng Chen, and Vladlen Koltun. "Interactive image segmentation with latent diversity." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
- [13] Cheng, Ho Kei, Yu-Wing Tai, and Chi-Keung Tang. "Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- [14] Maninis, Kevis-Kokitsi, et al. "Deep extreme cut: From extreme points to object segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [15] Zhang, Shiyin, et al. "Interactive object segmentation with inside-outside guidance." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
- [16] Acuna, David, et al. "Efficient interactive annotation of segmentation datasets with polygon-rnn++." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018.
- [17] Rupprecht, Christian, et al. "Guide me: Interacting with deep networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
- [18] Wang, Bin, et al. "Gazesam: What you see is what you segment." *arXiv preprint arXiv:2304.13844* (2023).
- [19] Aresta, Guilherme, et al. "iW-Net: an automatic and minimalistic interactive lung nodule segmentation deep network." *Scientific reports* 9.1 (2019): 11591.
- [20] Sun, Jinquan, et al. "A point says a lot: An interactive segmentation method for MR prostate via one-point labeling." *Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8*. Springer International Publishing, 2017.
- [21] Wong, Hallee E., et al. "ScribblePrompt: Fast and Flexible Interactive Segmentation for Any Medical Image." *arXiv preprint arXiv:2312.07381* (2023).
- [22] Xu, Ning, et al. "Deep GrabCut for Object Selection." *Procedings of the British Machine Vision Conference 2017*. British Machine Vision Association, 2017.#
- [23] Dupont, Camille, Yanis Ouakrim, and Quoc Cuong Pham. "UCP-net: unstructured contour points for instance segmentation." *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021.
- [24] Xu, Ning, et al. "Deep interactive object selection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [25] Tian, Zhiqiang, et al. "Graph-convolutional-network-based interactive prostate segmentation in MR images." *Medical physics* 47.9 (2020): 4164-4176.
- [26] Budd, Samuel, Emma C. Robinson, and Bernhard Kainz. "A survey on active learning and human-in-the-loop deep learning for medical image analysis." *Medical Image Analysis* 71 (2021): 102062.
- [27] Jungo, Alain, and Mauricio Reyes. "Assessing reliability and challenges of uncertainty estimations for medical image segmentation." *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22*. Springer International Publishing, 2019.
- [28] Wang, Guotai, et al. "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks." *Neurocomputing* 338 (2019): 34-45.
- [29] He, Kaiming, et al. "Masked autoencoders are scalable vision learners." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.
- [30] Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *International conference on machine learning*. PMLR, 2021.