Probabilistic Salient Object Ranking

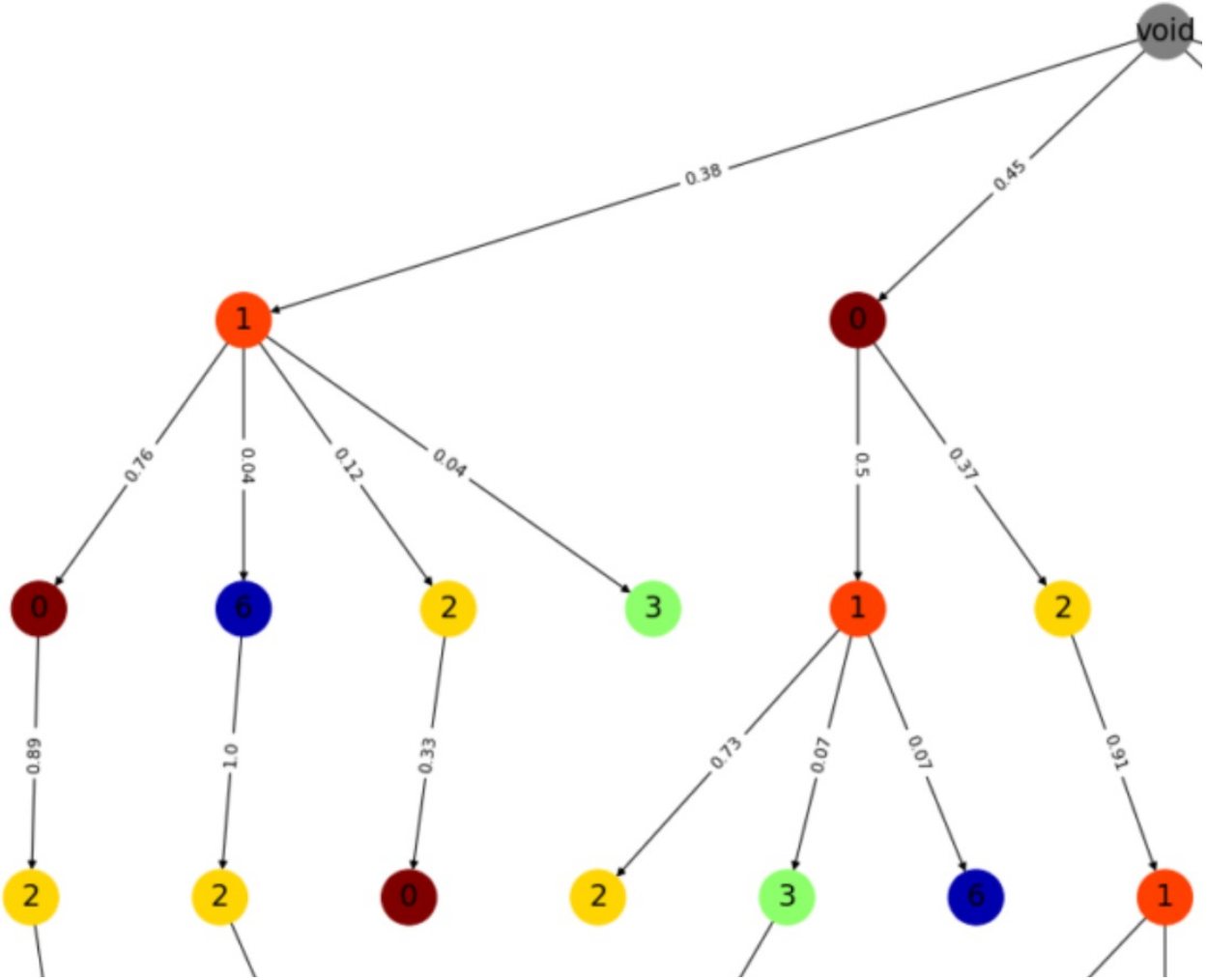
Guan Huankang

Updated abstract

Salient Object Ranking (SOR) aims to study how humans visually explore complex scenes by predicting an ordered sequence of objects that will attract our attention. Existing SOR approaches typically model this ranking deterministically, assuming a single, fixed ranking of attention. However, such deterministic SOR fails to capture the true nature of human attention. We observe that human attention shifts exhibit variability and stochasticity, showing that the next object of fixation is not a definitive choice but rather a probability distribution. Yet, existing SOR methods and evaluation metrics cannot account for this inherent randomness. To bridge this gap, we propose **ProbSOR**, a novel **Prob**abilistic **S**alient **O**bject **R**anking model that explicitly learns the uncertainty of attention shifts by incorporating Group Relative Policy Optimization (GRPO). We leverage a Vision-Language Model (VLM) as the foundation for ProbSOR and extend it with autoregressive mask generation. We also propose a new metric tailored to ProbSOR, as existing SOR metrics only support deterministic rankings. Additionally, we establish a ProbSOR benchmark comprising 15,000 probabilistic SOR samples for evaluation purposes. Extensive experiments demonstrate that ProbSOR achieves strong performance in salient object ranking and mask generation.

Problem


3
1
0
2
5
4


void
0.38
0.45
1
0
0.76
0.04
0.12
0.04
0.5
0.37
0
6
2
3
1
2
0.89
1.0
0.33
0.73
0.07
0.07
0.91
2
2
0
2
3
6
1

Problem

ior. We observe that human viewing behaviors are influenced by prior viewing trajectory and exhibit variability, meaning that the next object of fixation should be represented as a conditional probabilistic distribution rather than a definitive choice, while previous SOR methods and evaluation metrics fail to account for this inherent randomness. Motivated by this, we propose to reformulate SOR into a

- Our model: accept multiple answers/ranks
- Previous model: a single absolute GT/rank

- If we consider top 3 objects
  - GT: [0, 1, 2] SA-SOR: 1.0 probability: 0.16
  - Pred: [1, 0, 2] SA-SOR: 0.5 probability : 0.26
- If we consider top 2 objects
  - GT: [0, 1] SA-SOR: 1.0 probability : 0.23
  - Pred: [1, 0] SA-SOR: -1.0 probability : 0.29

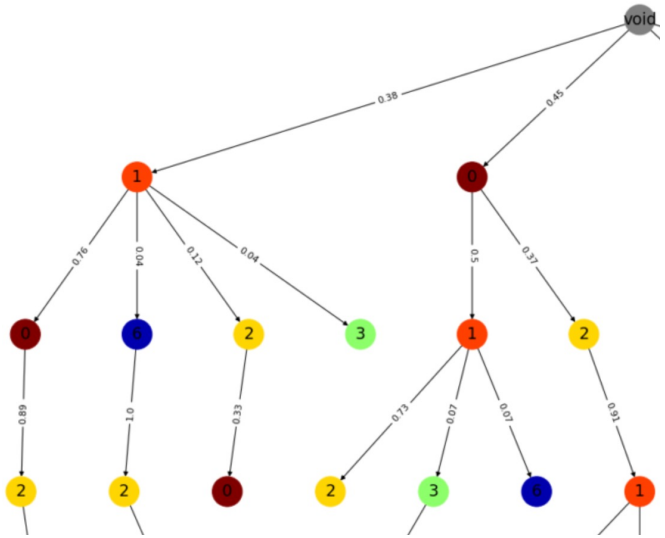
New Evaluation Metric

- Model Input: RGB image
- Model Output: Salient Object Ranking (# dynamic)
- Metrics: *Reward-based tree traversal strategy*
  - Starts at root node
  - Go down the tree following the predicted ranking
  - At each node: collects reward = human arrival probability
  - Final score: cumulative rewards along path
  - Core Concept: It evaluates SOR by how well they mirror real human attention shifts at each decision point – replacing deterministic 'right/wrong' judgments with plausibility scoring.

New Evaluation Metric

Key features of new metrics:

- Human-Centric Benchmark
  - Rewards = Empirical attention shift probabilities
  - Directly encodes how humans actually allocate attention to objects
- Stochastic Consistency
  - Scores SOR by cumulative attention shift probabilities (not binary or linear association between two sequences.)
  - A rank through high-probability nodes scores well → even if it differs from individual human ranks.
  - Naturally accommodates attention variability across viewers
- Tree Traversal = Simulated Visual Exploration
  - Path following: Mimic attentional shifts between objects
  - Cumulative rewards: Quantifies overall behavioral alignment
- Probabilistic Validation
  - Deterministic metric: "Is object B always viewed after object A?" ❌
  - Our metric: "How probable is this sequence given real human data?" ✅



Previous metrics: Pearson correlation coefficient (measure the strength of a linear association between two variables. It is defined as the covariance of the two variables divided by the product of their standard deviations.)

Comparison

Existing Metrics	Our Metrics
Assume one “correct” rank	Accept multiple valid ranks
Penalize deviations	Rewards *statistically likely* choices
Ignore viewer variability	Embeds population-level attention shift distribution

In essence: Instead of asking "Is this the exact path humans take?" (which doesn't exist), we ask "How behaviorally plausible is this path?" – quantified through observable human attention shift probabilities.

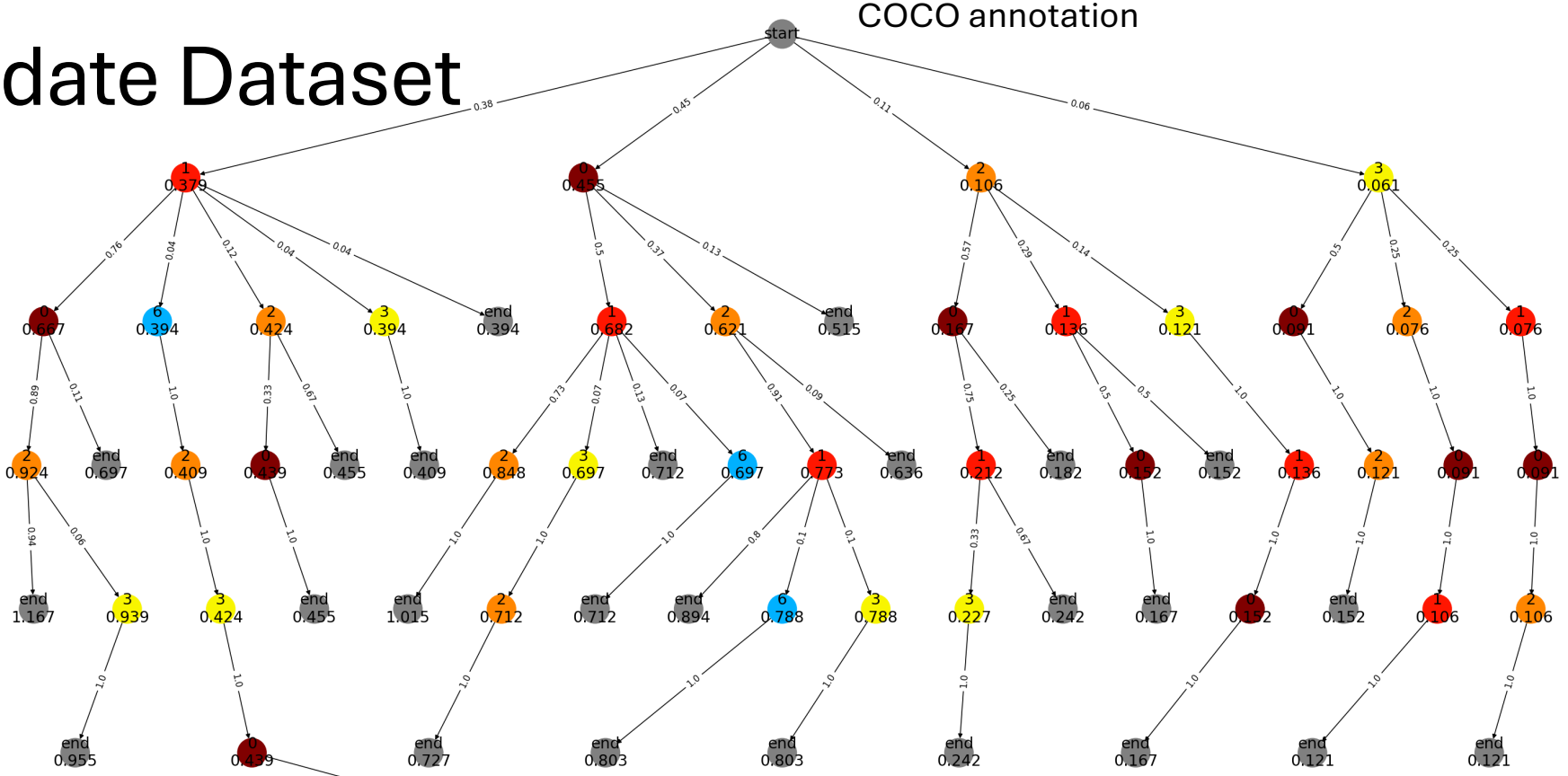
New Evaluation Metric

Examples:

Salicon (15k)
COCO annotation

ARSR, IRSR, HyperSOR

Update Dataset



- **The tree contains salient and less salient objects**
- **--> Exclude less salient objects to make the entire tree easier to interpret.**

Update Dataset

We can estimate the probability that an object will be noticed or visited as:

$$\mathrm{P}(A\ is\ visited) = \frac{The\ number\ of observers\ who\ visit\ A}{The\ number\ of\ observers}$$

Salient objects are the most visually prominent elements in a scene. Therefore, we can identify salient objects by selecting those with a higher visitation probability *P*—meaning they are more likely to attract attention.

We simply adopt thresholding method:

- Assume an **average** of 5 salient objects per image (based on empirical observations), OR
- Compute the visitation probability threshold using existing benchmark datasets, e.g., ARSR or IRSR.

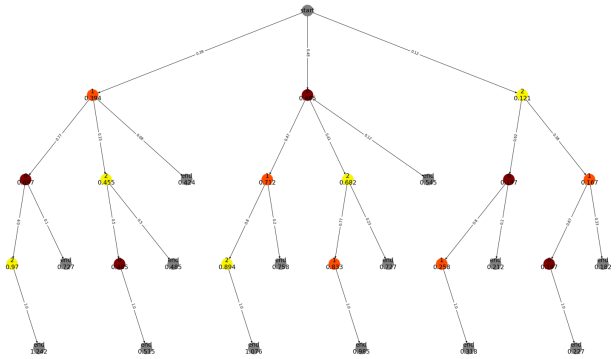
Top 5 = 0.12

ARSR: 0.18

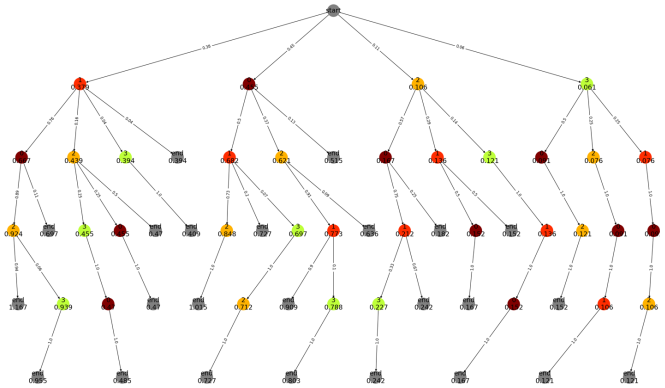
IRSR: 0.30

Update Dataset

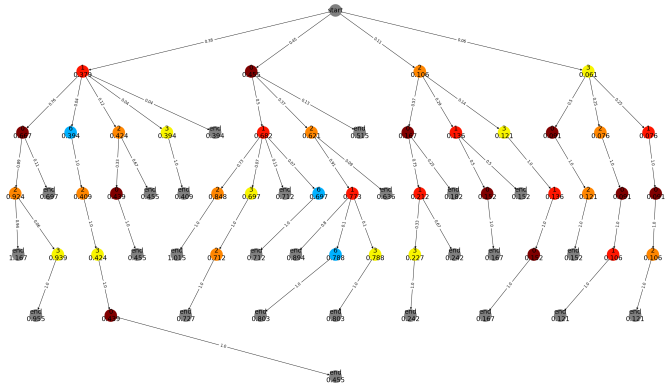
1. Filter out those less salient objects, i.e., P(OBJ is visited) < 0.18
2. Construct the dataset based on previous discussion



Threshold = 0.18

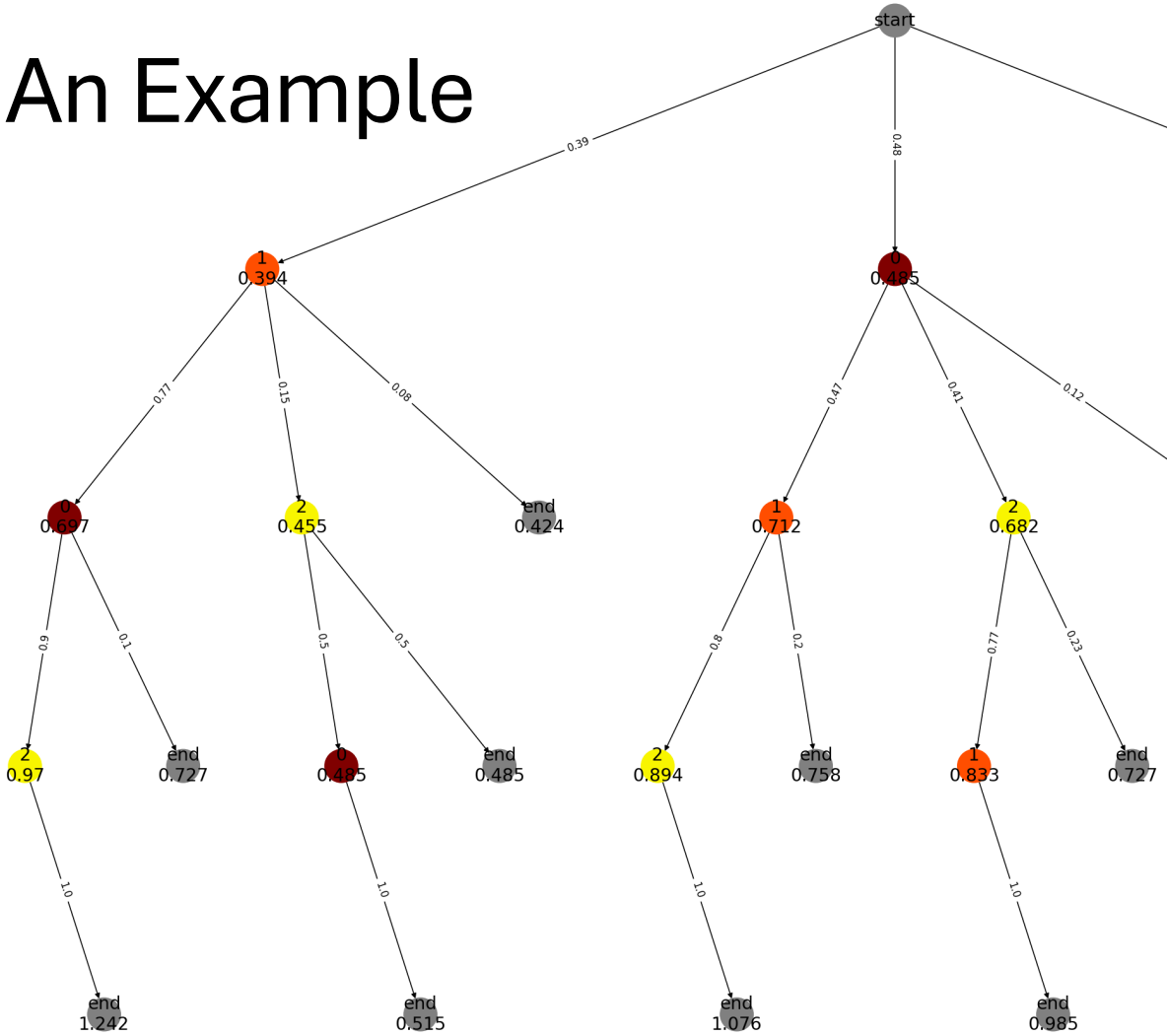


Threshold = 0.12

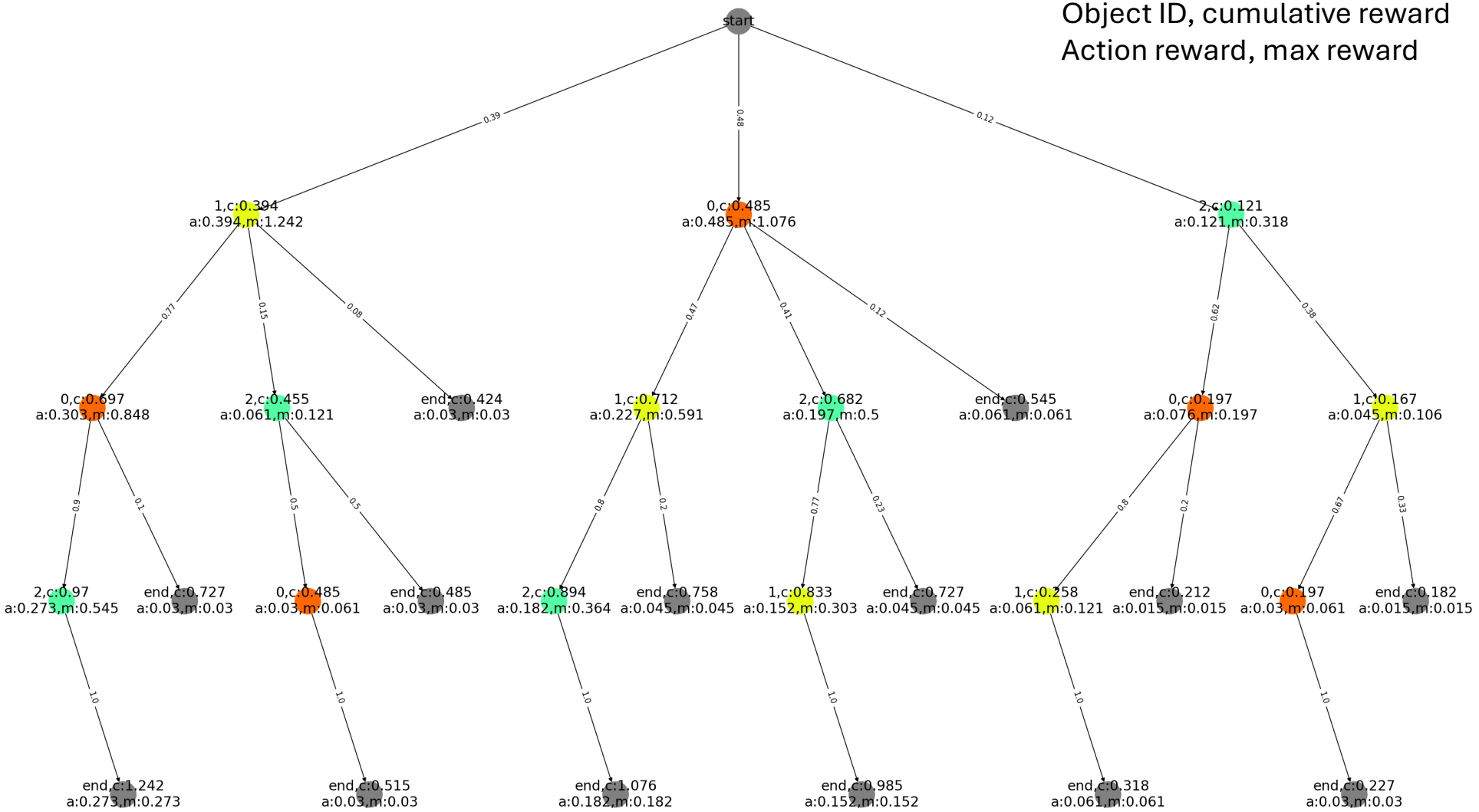


No Threshold

An Example


start
0.39
0.48
0.12
1
0.394
0
0.485
2
0.121
0.77
0.15
0.08
0.47
0.41
0.12
0.62
0.38
0
0.697
2
0.455
end
0.424
1
0.712
2
0.682
end
0.545
0
0.197
1
0.167
0.9
0.1
0.5
0.5
0.8
0.2
0.77
0.23
2
0.97
end
0.727
0
0.485
end
0.485
2
0.894
end
0.758
1
0.833
end
0.727
1.0
1.0
1.0
1.0
end
1.242
end
0.515
end
1.076
end
0.985

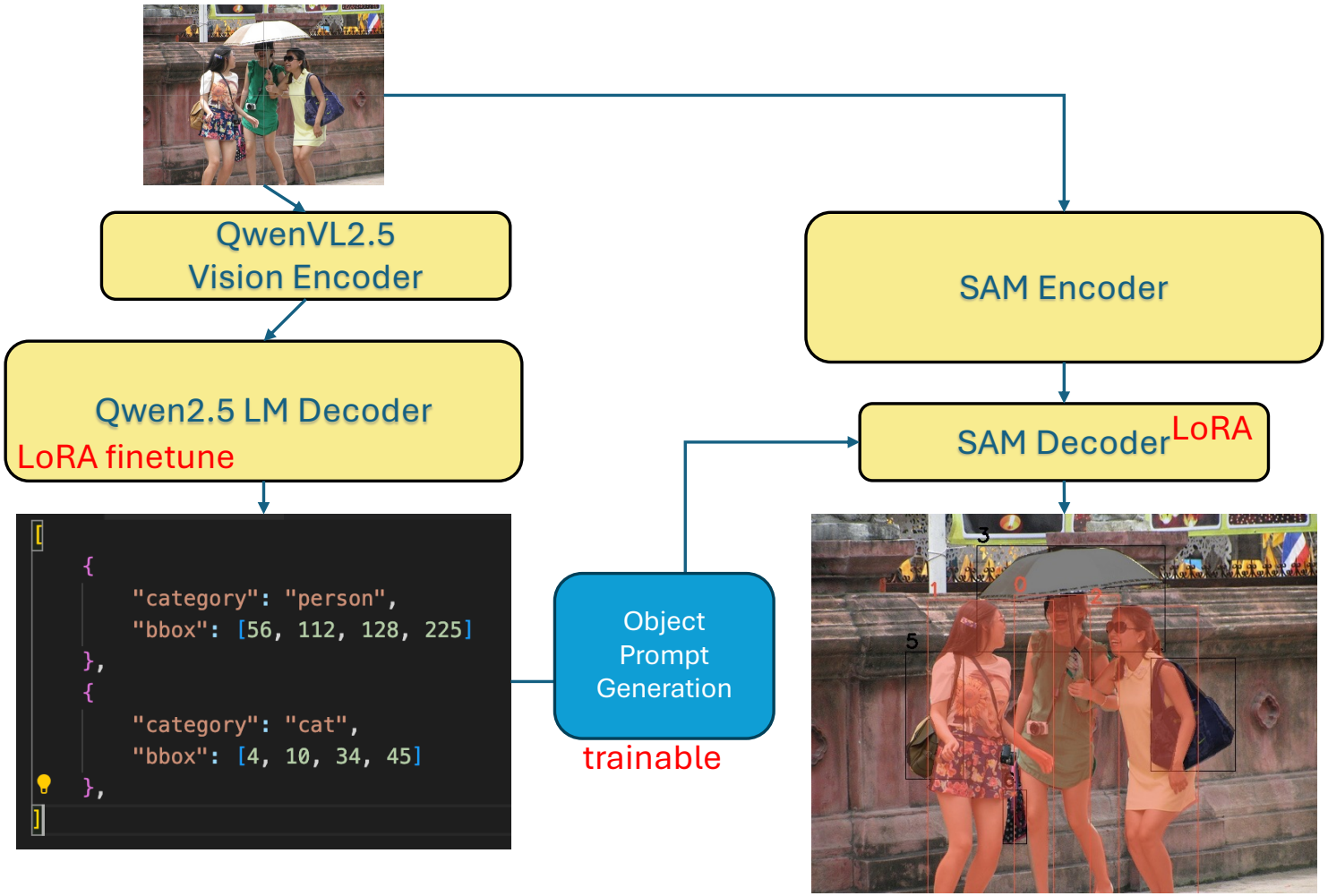

3
1
0
2
5


Object ID, cumulative reward
Action reward, max reward
start
0.39
0.48
0.12
1,c:0.394
a:0.394,m:1.242
0,c:0.485
a:0.485,m:1.076
2,c:0.121
a:0.121,m:0.318
0.77
0.15
0.08
0.47
0.41
0.12
0.62
0.38
0,c:0.697
a:0.303,m:0.848
2,c:0.455
a:0.061,m:0.121
end,c:0.424
a:0.03,m:0.03
1,c:0.712
a:0.227,m:0.591
2,c:0.682
a:0.197,m:0.5
end,c:0.545
a:0.061,m:0.061
0,c:0.197
a:0.076,m:0.197
1,c:0.167
a:0.045,m:0.106
0.9
0.1
0.5
0.5
0.8
0.2
0.77
0.23
0.8
0.2
0.67
0.33
2,c:0.97
a:0.273,m:0.545
end,c:0.727
a:0.03,m:0.03
0,c:0.485
a:0.03,m:0.061
end,c:0.485
a:0.03,m:0.03
2,c:0.894
a:0.182,m:0.364
end,c:0.758
a:0.045,m:0.045
1,c:0.833
a:0.152,m:0.303
end,c:0.727
a:0.045,m:0.045
1,c:0.258
a:0.061,m:0.121
end,c:0.212
a:0.015,m:0.015
0,c:0.197
a:0.03,m:0.061
end,c:0.182
a:0.015,m:0.015
1.0
1.0
1.0
1.0
1.0
1.0
end,c:1.242
a:0.273,m:0.273
end,c:0.515
a:0.03,m:0.03
end,c:1.076
a:0.182,m:0.182
end,c:0.985
a:0.152,m:0.152
end,c:0.318
a:0.061,m:0.061
end,c:0.227
a:0.03,m:0.03

Model Design

1. OPG module: Object Prompt Generation module bridges the QwenVL and SAM for mask prediction.
2. QwenVL is finetuned for SOR prediction.
3. SAM decoder is finetuned to generate object mask.

Three novel things:
1. OPG module
2. Training strategy
3. Metrics



Results: (mask prediction is good enough)

```
runs.summary["Table-000000301867-1"]
```

Filter

	image	data
1	 000000301867	{'name': '000000301867', 'width': 640, 'height': 480, 'records': [{'rank': 1, 'category': 'person', 'bbox': {'x1': 150.0, 'y1': 106.07142857142857, 'x2': 280.0, 'y2': 473.5714285714286}, 'mask': None, 'mask_rle': {'size': [480, 640], 'counts': b'`^W2c1T=?PE^NR8i1eG`NU8f1eG`NU8g1cGaNj1oNP3e2PKbNh1POT3c2mJdNg1POW3c2fJiNj1iN^3e2\\JlNP {'rank': 2, 'category': 'person', 'bbox': {'x1': 255.71428571428572, 'y1': 98.57142857142857, 'x2': 384.28571428571433, 'y2': 473.5714285714286}, 'mask': None, 'mask_rle': {'size': [480, 640], 'counts': b'bfh3f0P>c0@?C:E;mC[N\\:R2TEhNU:[3B`0A<_Oc0_Oi0[Ob0@=D7I9G6K6I6K5K5K4L5L3L3M2N3N0O2N101N1O1O101mNlEVMV:h2QFRMPf5X3ZI_Lm0;i5\\4VJdKl2VOaNU5bNfKf2@eNi4dNhK^2LjN\\4hNhKW25POQ4iNkKR2:UOj3hNnKW[NAlKoNb4^1CAkKTOd4W1CBjKWOg4S1@BmK[Of4n0AAlK@m4f0@WOf2`0Q6K5K4Kanh3'}}, {'rank': 3, 'category': 'person', 'bbox': {'x1': 324.28571428571433, 'y1': 114.64285714285715, 'x2': 485.71428571428567, 'y2': 473.5714285714286}, 'mask': None, 'mask_rle': {'size': [480, 640], 'counts': b'gmi4`0<3Q=S1K4L5L2N3L3N2O1eGQNb3Q2]LQN`3Q2^LRN`3n1aLSN]3n1dLRNZ3n1gLSNW3n1jL7DF;;GD8<KA4`0N^OgNCkKo0b5ZO^N0kKh0j5TOYN9iKd0`6aNdMS3\\2lLbMV3`2gL_M\\3b2aL]MfLP1YO]NQ4d0dLo0AVNn3k0`Ln0HPNj3S1\\Ln0Y6ROeIm0^6SOaIm0`6SO^Im0d6SO[Il0f6UOXIl 'end_of_detection': True, 'output_text': '<|vision_end|>This is the input image with height = 448 and width = 448.<|im_end|>\n<|im_start|>assistant\n|1|person|105,99,196,442|\n|2|person|179,92<|im_end|>'}

Export as CSV Columns... Reset table

Training Strategy

1. **Exp-1:** Select the best SOR as GT to train the model
   - Global optimal solution: predict the best reward sequence

2. **Exp-2:** Reward-based Learning
   - Image -> Model -> Output ~ Rewards
   - Allow the model to predict different sequences
   - GRPO

3. **Metrics:** reward, normalized reward, iou, mae

Issues

- Inference step is slow, requiring 3 – 9 seconds for a single image. Our evaluation set contains 5000 images, then it takes up to 8.5 hours to finish the evaluation.
- Training time is acceptable that it requires <8 hours on 1 A100 to finetune the model. Use small image as input (like 448), we can use 1 4090GPU to train.
- Solutions:
  - Use a smaller VLM or smaller image as input
  - there may be technical optimizations that could help improve inference speed

Mask Prediction (ongoing)

Resize=1036, bbox

QwenVL2.5
Vision Encoder

Qwen2.5 LM Decoder
LoRA finetune

```
[1][person](417,215,620,1036)
[2][person](240,238,498,1036)
[3][person](521,248,788,1036)
[4][umbrella](344,91,720,430)
[5][background](0,0,1036,1036)
```

Mask token x2

For each detected object, we forward multiple tokens to segmentation branch

Mask Decoder
Trainable
Resize=480x640

Resize=1024
SAM Encoder

1. Mask tokens (as placeholder)
2. [1][person](x1, y1, x2, y2) as seg inputs
3. LISA <seg> -> mask
4. 2 tokens -> 1 mask

Contributions

1. We reformat the SOR to probability-based.

2. The first VLM for SOR, (considering the uncertainty of human attention shift (GRPO), new mask generation strategy (under revision))

3. New Benchmark and new metric

Probabilistic Salient Object Ranking

Anonymous Author(s)
Affiliation
Address
email

Abstract

Salient Object Ranking (SOR) aims to study how humans visually explore complex scenes by predicting an ordered sequence of objects that will attract our attention. Existing SOR approaches typically model this ranking deterministically, assuming a single, fixed ranking of attention. However, such deterministic SOR fails to capture the true nature of human attention. We observe that human attention shifts exhibit variability and stochasticity, showing that the next object of fixation is not a definitive choice but rather a probability distribution. Yet, existing SOR methods and evaluation metrics cannot account for this inherent randomness. To bridge this gap, we propose **ProbSOR**, a novel **Prob**abilistic **S**alient **O**bject **R**anking model that explicitly learns the uncertainty of attention shifts by incorporating Group Relative Policy Optimization (GRPO). We leverage a Vision-Language Model (VLM) as the foundation for ProbSOR and extend it with autoregressive mask generation. We also propose a new metric tailored to ProbSOR, as existing SOR metrics only support deterministic rankings. Additionally, we establish a ProbSOR benchmark comprising 15,000 probabilistic SOR samples for evaluation purposes. Extensive experiments demonstrate that ProbSOR achieves strong performance in salient object ranking and mask generation.

1 Introduction

2 Related Works

3 Benchmark

4 Methodology

5 Experiments

Metrics: $reward = \sum_{i=1}^{n} r_i$, where $r_i = p_i$, p_i is the possibility of reaching $node_i$ from the top of the tree. $normalized\_reward = \frac{reward}{optimal\_reward}$ is used to report the relative goodness of the SOR prediction. We report iou, mae to assess the quality of mask prediction.