Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion (Supplementary)

Dinesh Jayaraman and Kristen Grauman

The University of Texas at Austin
{dineshj,grauman}@cs.utexas.edu

This document provides information supplementary to the main paper. Procedure block 1 lists the steps involved in the forward pass during training/inference. Fig 1 provides a detailed schematic diagram explaining the architectures of and connections amongst the modules of our active vision system. Dataset-specific details of training-related choices are provided in Sec 1. Finally, at the end of this document, we present some examples of view sequences selected by our approach.

Procedure 1 Forward propagation (training/inference time)

Input: 3D instance X, together with:
- projection function $P(X, \boldsymbol{p})$ denoting the view captured from camera pose $\boldsymbol{p}$
- proprioception function $f(\boldsymbol{p})$ of the current position, which is known to the active vision system *e.g.* wrist position, or gravity direction.

Output: $\hat{y}$, the predicted label for instance X.

procedure ForwardOneStep($t, \boldsymbol{a}_{t-1}, \boldsymbol{p}_{t-1}, \hat{\boldsymbol{a}}_t$) ▷ 1 forward propagation step
  $\boldsymbol{m}_t \leftarrow$ actor($\boldsymbol{a}_{t-1}, f(\boldsymbol{p}_{t-1})$) ▷ motor command sampled from $\mathcal{C}$ to adjust camera
  $\boldsymbol{p}_t \leftarrow \boldsymbol{p}_{t-1} + \boldsymbol{m}_t$ ▷ camera pose update
  $\boldsymbol{x}_t \leftarrow P(X, \boldsymbol{p}_t)$ ▷ capture new view
  $\boldsymbol{s}_t \leftarrow$ sensor($\boldsymbol{x}_t, \boldsymbol{m}_t$) ▷ per-view processing
  $\boldsymbol{a}_t \leftarrow$ aggregator($\boldsymbol{a}_{t-1}, \boldsymbol{s}_t$) ▷ evidence fusion
  if $t > 1$ **then** ▷ relevant only at training time
    $\hat{\boldsymbol{a}}_t \leftarrow$ lookahead($\boldsymbol{a}_{t-1}, \boldsymbol{m}_{t-1}, f(\boldsymbol{p}_t)$) ▷ look-ahead prediction of current time-step feature
    look-ahead error $\zeta_t \leftarrow d(\boldsymbol{a}_t, \hat{\boldsymbol{a}}_t)$
  return $\boldsymbol{a}_t, \boldsymbol{p}_t, \zeta_t$

$\boldsymbol{a}_0 \leftarrow \mathbf{0}$ ▷ initialization
$\boldsymbol{p}_0 \leftarrow$ random position
for t =1,2,...T **do** ▷ move, observe, aggregate in a loop
  $\boldsymbol{a}_t, \boldsymbol{p}_t, \zeta_t \leftarrow$ ForwardOneStep($t, \boldsymbol{a}_t, \boldsymbol{p}_{t-1}$)
$\hat{y} \leftarrow$ classifier($\boldsymbol{a}_t$) ▷ final class prediction
return $\hat{y}$

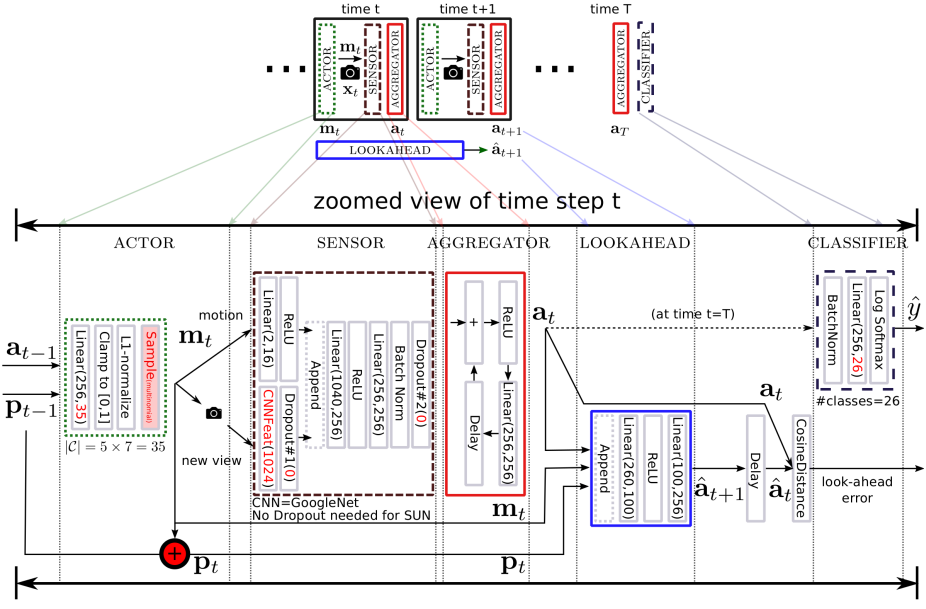


Fig. 1: A detailed schematic diagram showing the architectures of and connections amongst our active vision system modules. The small schematic at the top (repeated from Fig1 in the paper) presents a succinct bird's-eye view of information flow within as well as between time steps, and the large schematic below zooms into the operations at some given time step t in more detail. Processing proceeds from left to right, with arrows to disambiguate where necessary. In the bottom schematic, "Linear(a,b)" denotes fully connected layers which transform a-length vector inputs to b-length vector outputs. The "Clamp" operator in ACTOR is a squashing function that sets both upper and lower limits on its inputs. The red "Sample" layer in ACTOR takes the weights of a multinomial pdf as input and samples stochastically from the distribution to produce its output (gradients cannot be backpropagated through this layer; it is trained through REINFORCE [1] instead of SGD from the classification loss). "Delay" layers store inputs internally for one time-step and output them at the next time-step. Other layer names in the schematic are self-explanatory. Input and output sizes of some layers are marked in red to denote that these are parameters derived from dataset-related choices — these are set for our SUN360 experiments in this schematic, and explanations are shown below each module. Note that AGGREGATOR is a recurrent neural network, and LOOKAHEAD may be considered a "predictive" autoencoder, that reduces its input features (appended together with the current agent motion $\boldsymbol{m}_t$) to a more compact representation in its bottleneck layer before producing its prediction of its next time-step input.

1 Dataset-specific training details

Training for both datasets followed the general procedure described in Sec 3 of the main paper. In this section, we mention details of the training procedures specific to the two datasets. For SUN360, the full system shown in the schematic of Fig 1 was trained end-to-end in one shot. No dropout was employed in either of the Dropout layers in Fig 1.

For GERMS, the paucity of data and the relative simplicity of the task created some difficulties during training. First, for convergence, it was necessary to initialize end-to-end training at $T = 3$ with the weights of the same network trained end-to-end at $T = 1$. Given the shortcomings of the dataset though (small training set and ease of task), the $T = 1$ network had already achieved zero training error. This meant, the $T = 3$ network initialized with $T = 1$ weights did not have sufficient error gradients from the softmax classification loss for training to proceed. To handle this, we introduced dropout rates of 2e-5 and 0.5 at the CNN feature layer (as a kind of data augmentation) and preceding the AGGREGATOR input layer, respectively.

The other choices and hyperparameters involved in training are listed below: (marked with [*] when selected through cross-validation; others were fixed based on values known to work well for similar systems in the past)

- Gradient descent variant: Stochastic gradient descent (batch-size 32) with momentum (0.9)
- Starting learning rate [*]: 7e-3 (SUN360), 3e-3 (GERMS)
- Learning rate schedule: Linear decay towards a minimum learning rate of 1e-5 after 800 epochs (both datasets)
- Weight decay rate (L2 regularization on network weights): 5e-3 for both
- Lookahead λ [*]: 1.5 (SUN360), 0.05 (GERMS). Explanation for this hyperparameter: As specified in Sec 3 in the main paper, the net error from which derivatives are computed for modules other than ACTOR is:

$$\sum_{t=1}^{T}(\text{classification } L_{\text{softmax}} \text{ at } t) + \lambda \sum_{t=1}^{T} d_{\text{lookahead}}(\hat{\boldsymbol{a}}_t, \boldsymbol{a}_t) \tag{1}$$

- Weight initialization: uniform between 0 and 0.1 for all weights, except the recurrent network feedback weights (Linear(256,256) block inside AGGREGATOR in Fig 1), whose weights and biases were initialized to identity $\mathcal{I}_{256\times256}$ and zero $\mathbf{0}_{256\times1}$ respectively.
- Convergence criterion: Training is terminated when validation classification accuracy does not fall for 50 epochs. The rough number of epochs (full passes over training data) required for convergence are 250 for SUN360 and 150 for GERMS. Total training time is approximately 2 hours for SUN360 and 1.5 hours for GERMS.

If the paper is accepted, code and processed data will be made available for easy reproducibility.

2 Examples of selected views

At the end of this document, we provide some examples of views selected by our approach when presented with SUN360 panoramas. Each example consists of two rows of $T = 3$ panels each. The top row corresponds to the views selected, and the bottom row shows where on the 12×12 view grid they are selected from. The eye graphics at the corners of the top row views indicate the elevation angle of each view (legend provided in the bottom row, left in each example). Note that views appear inverted when elevation angle is >90 or < -90 degrees, because these correspond to the agent "bending over backwards".

In each example, the first view is chosen at random for presentation to the system, and the remaining views are chosen automatically by the system itself, within a 5×7 grid neighborhood of the preceding view. In the bottom row of panels (view grids), the transparent red square shows the view that is currently being observed, and the 5×7 grid of transparent yellow squares shows the views that are available for selection at the next time step.

True category names are indicated above each example (left top), and the likelihood of the true category, as returned by the CLASSIFIER module of our system at every time-step, is shown in parentheses above each corresponding view. Note that these are *aggregated* likelihood scores *i.e.*, the score at time t is based on observations at times 1,2...t. In addition, we list the top-3 most likely categories (among 26 total) returned by our model above each view.

As can be seen from these examples (also explained in accompanying captions for a few cases, as an illustration), in many cases, (1) the automatic motion/view selection performed by our ACTOR module, serves to increase the confidence of the right category, and helps disambiguate among confusing categories. (2) AGGREGATOR efficiently fuses information across time-steps, often showing evidence of accounting for the agent's motions relative to previous views.

References

1. Williams, R.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. In: JMLR (1992)

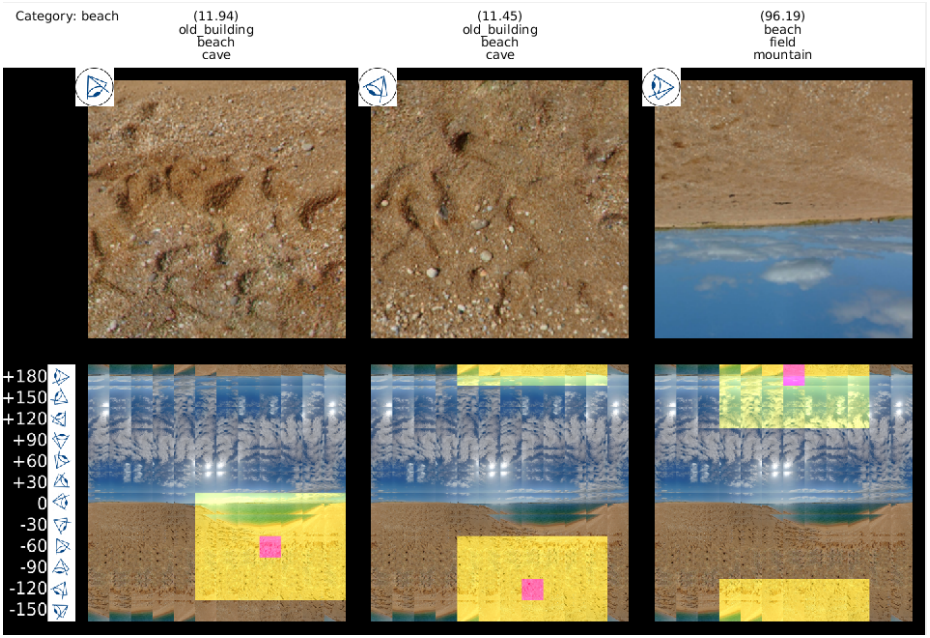


Fig. 2: The wet dry sand in this beach resembles old mud buildings, but once the agent looks up to see the sky and water at the horizon, it becomes 96% confident of "beach".

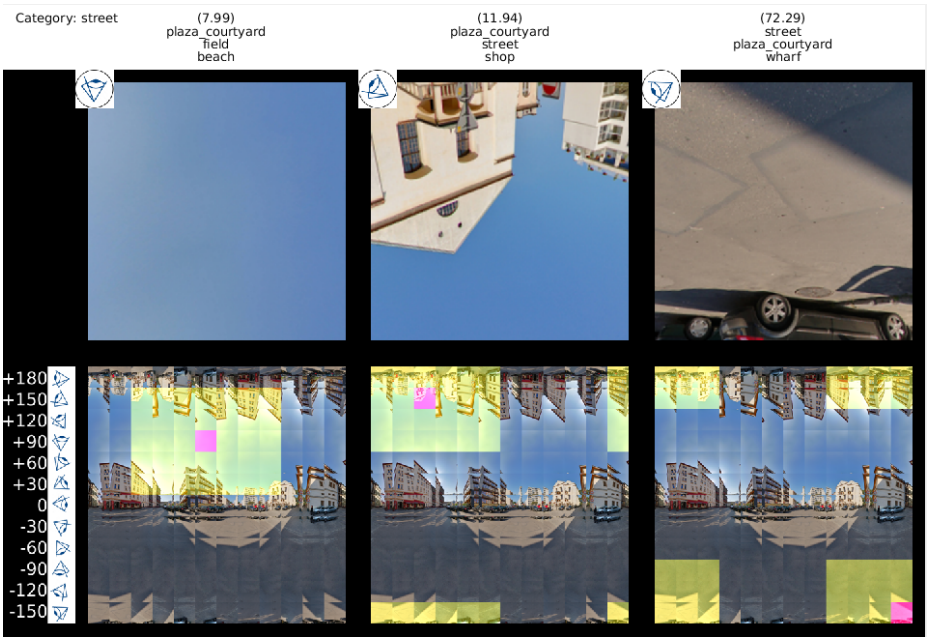


Fig. 3: The random starting view (at T=1) of the sky is not informative, but still enough to start guessing outdoor categories rather than indoor. The architecture of the building facade suggests "plaza", but with low confidence, at $T = 2$. When the system looks down to disambiguate its top guesses ("plaza"— typically paved with brick and "street"— typically tarmac), it sees car wheels and tarmac, immediately suggesting "street" with high confidence.

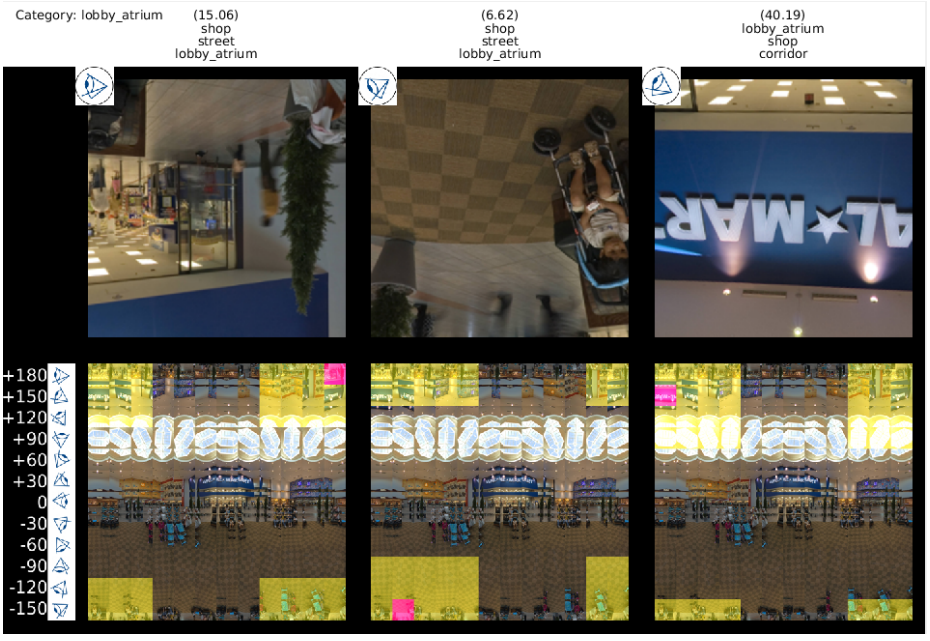
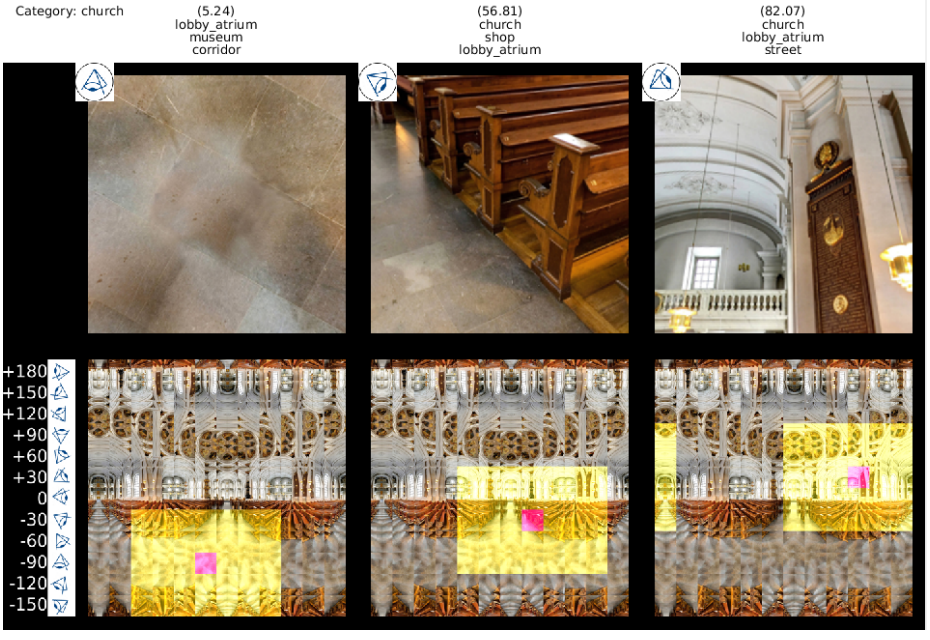
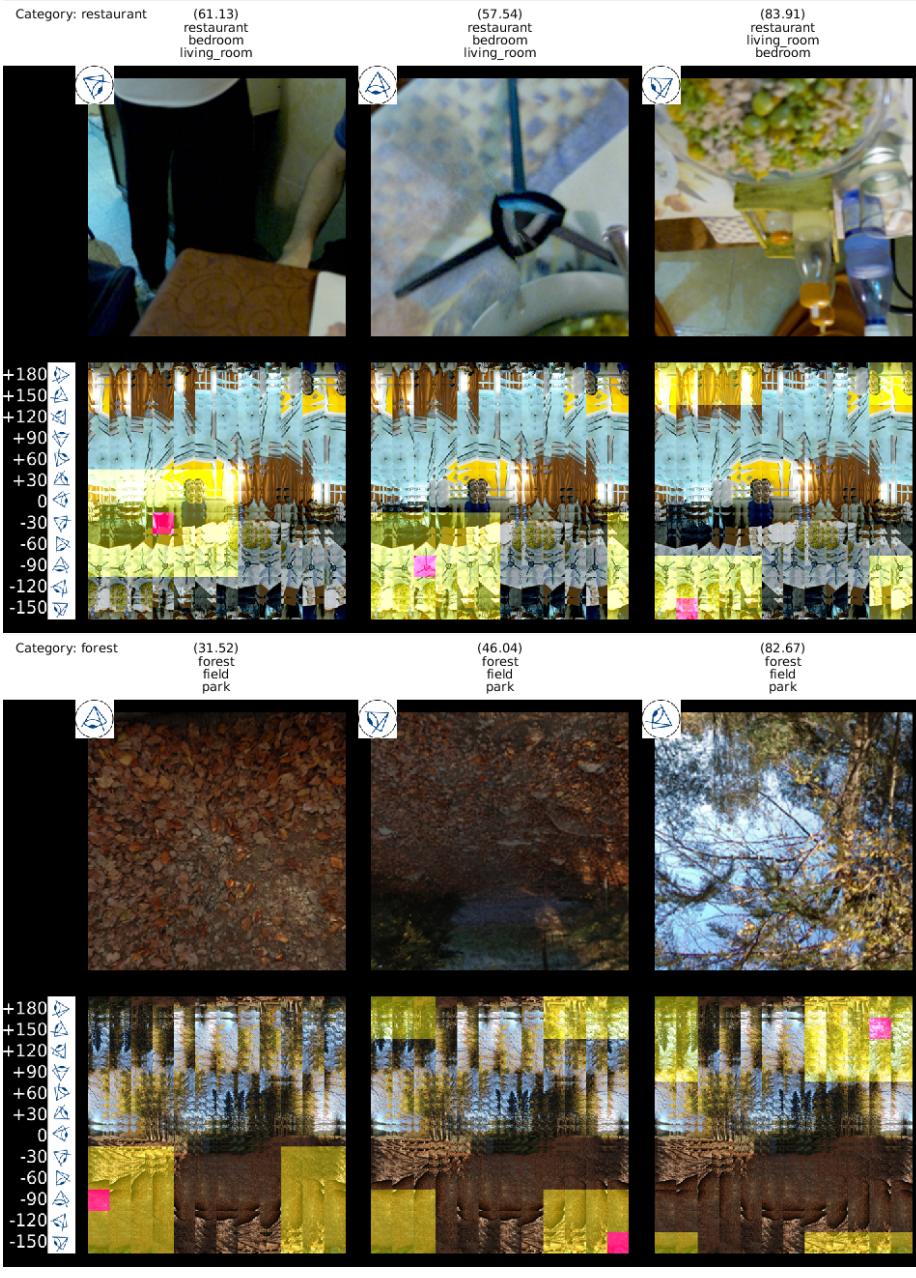
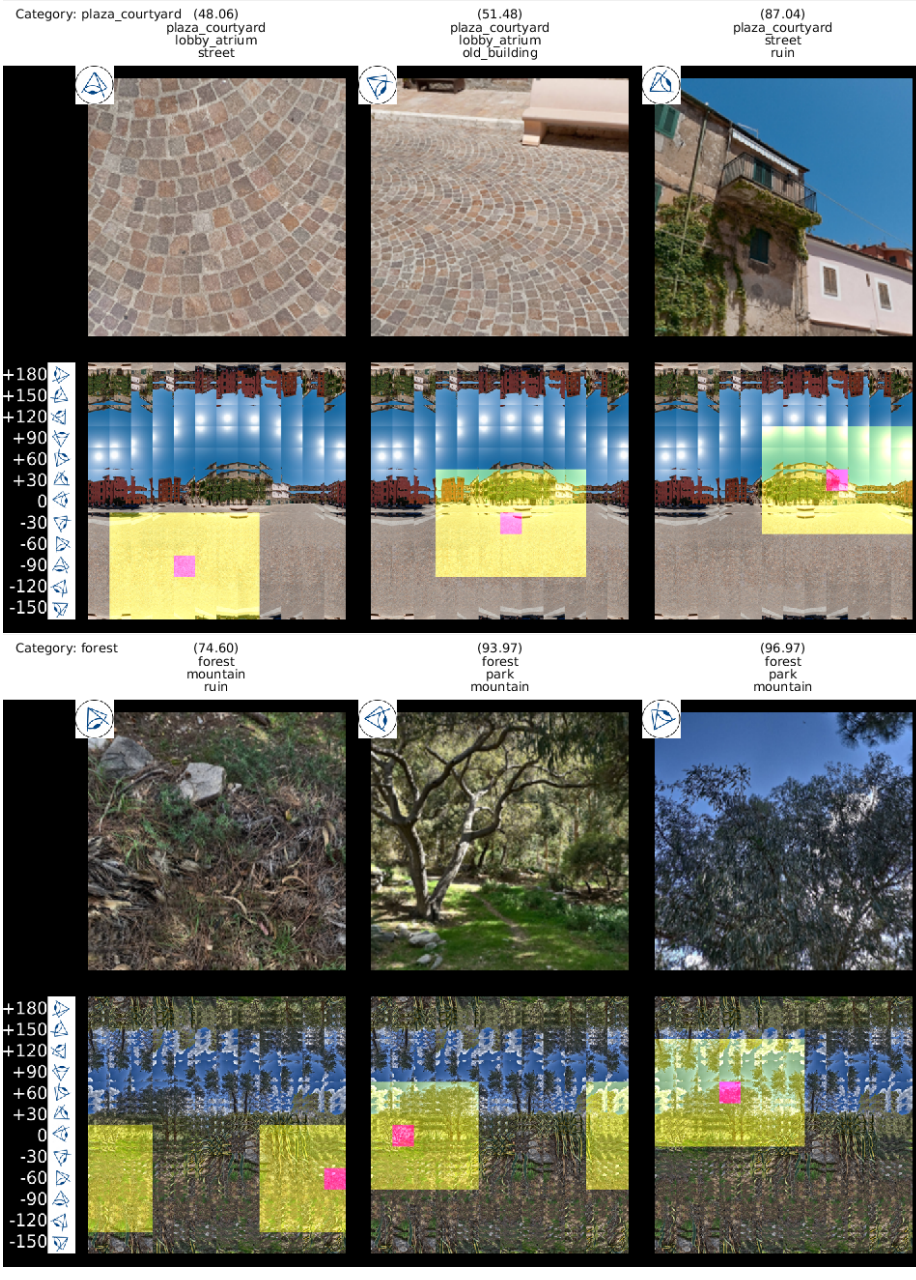
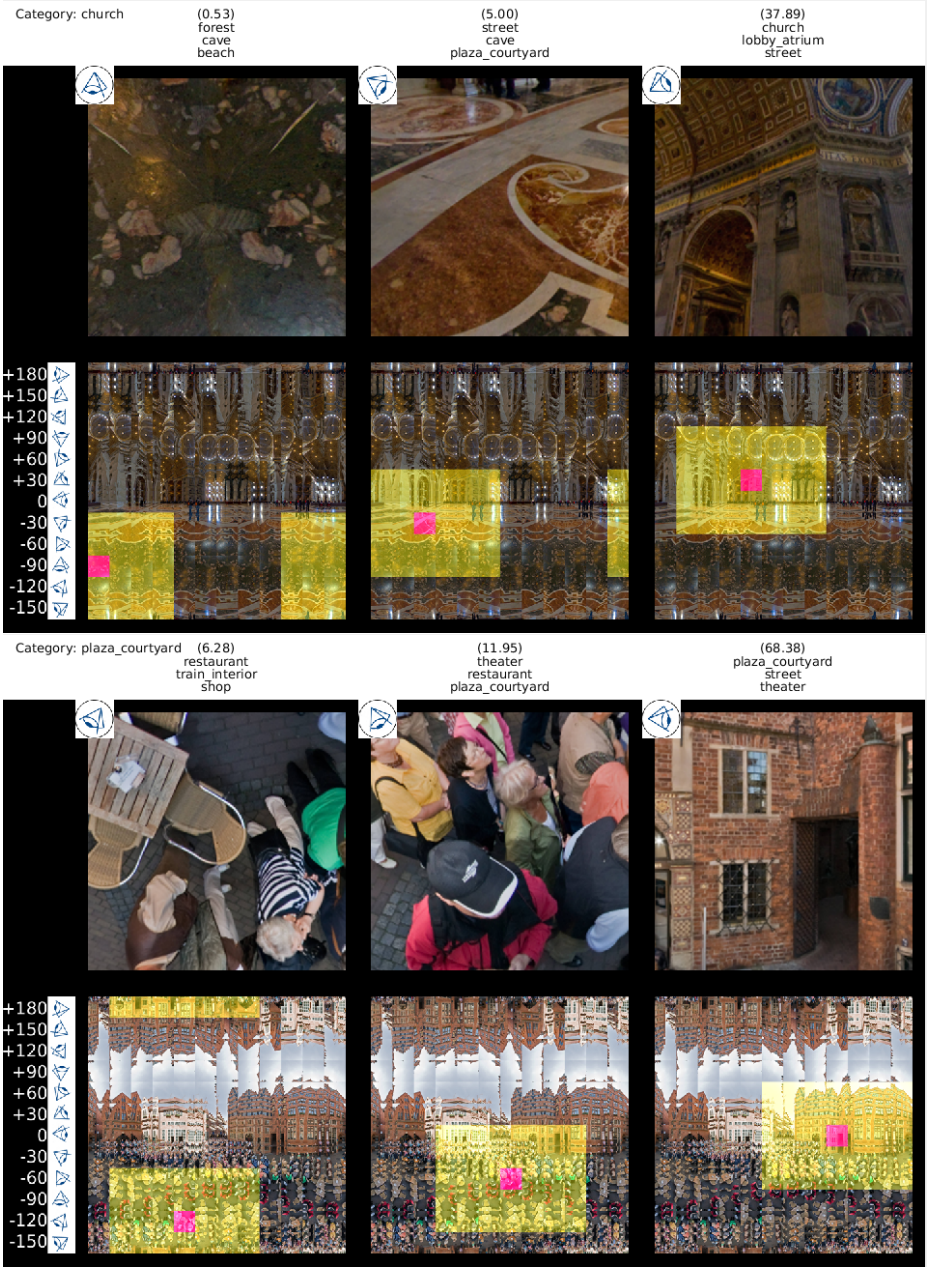


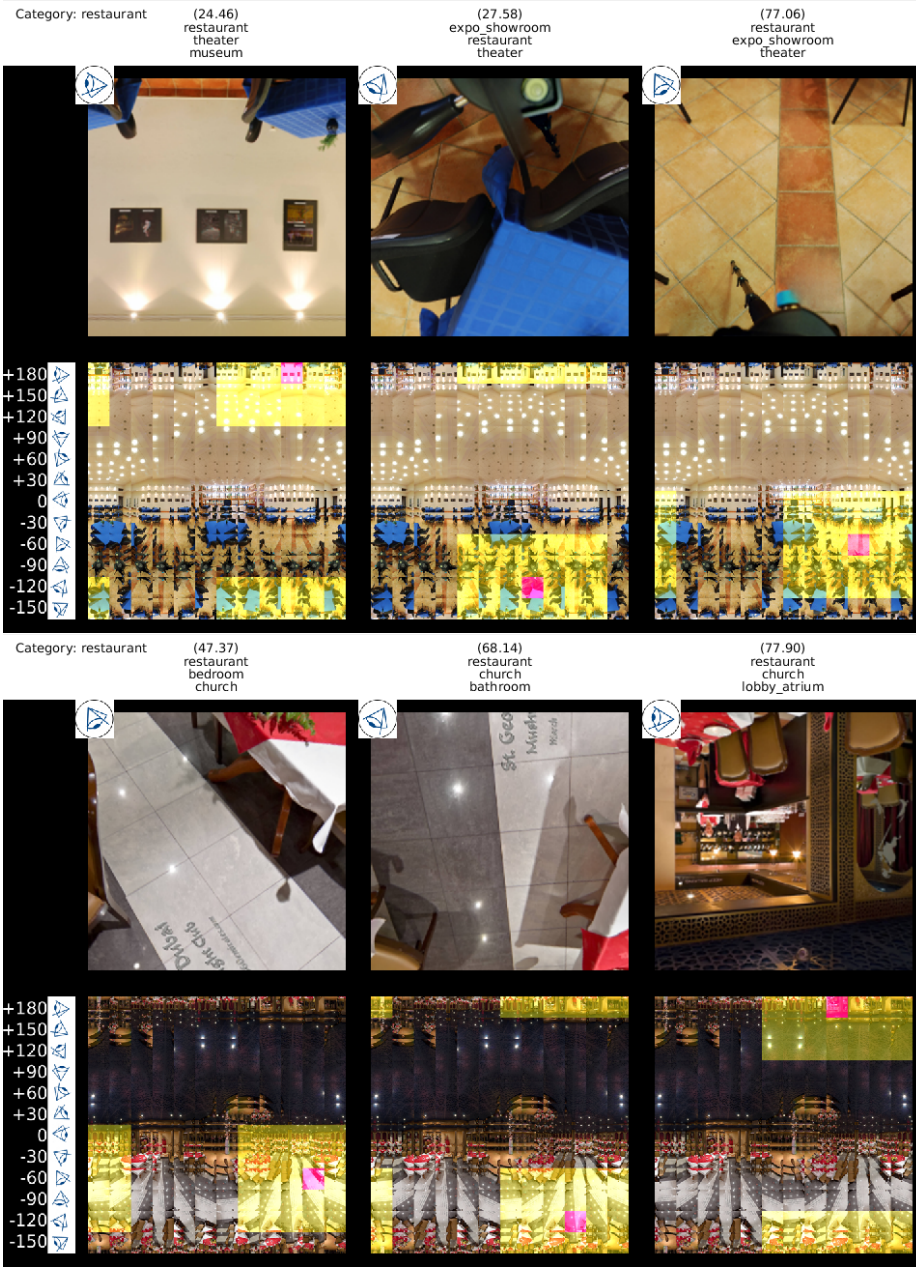
Fig. 4: The view shown to the agent at $T = 1$ suggests a store-front, and thus "shop", but after exploring the scene and finding a large indoor space near the shop, the agent revises its guess correctly to "lobby atrium".

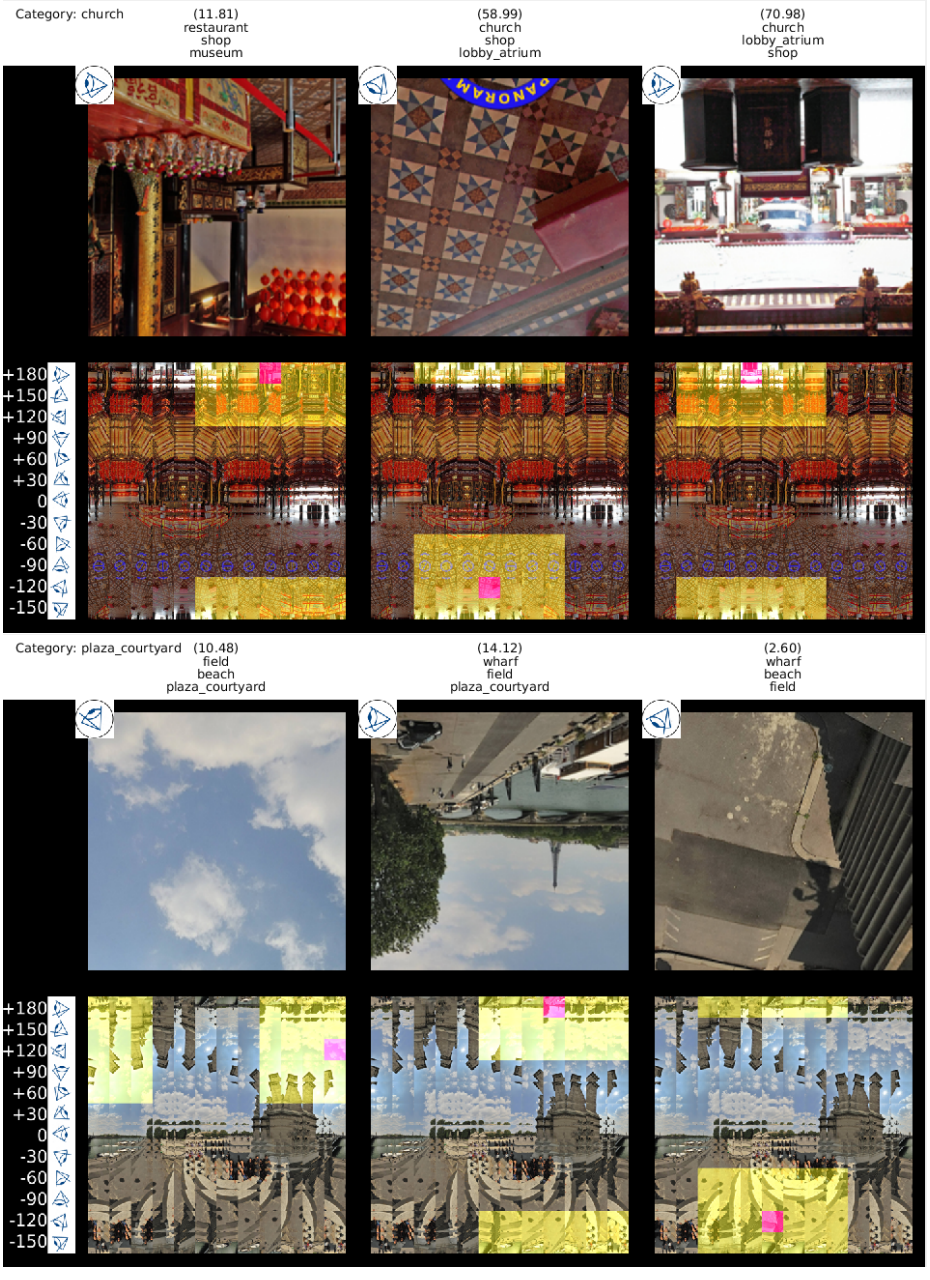


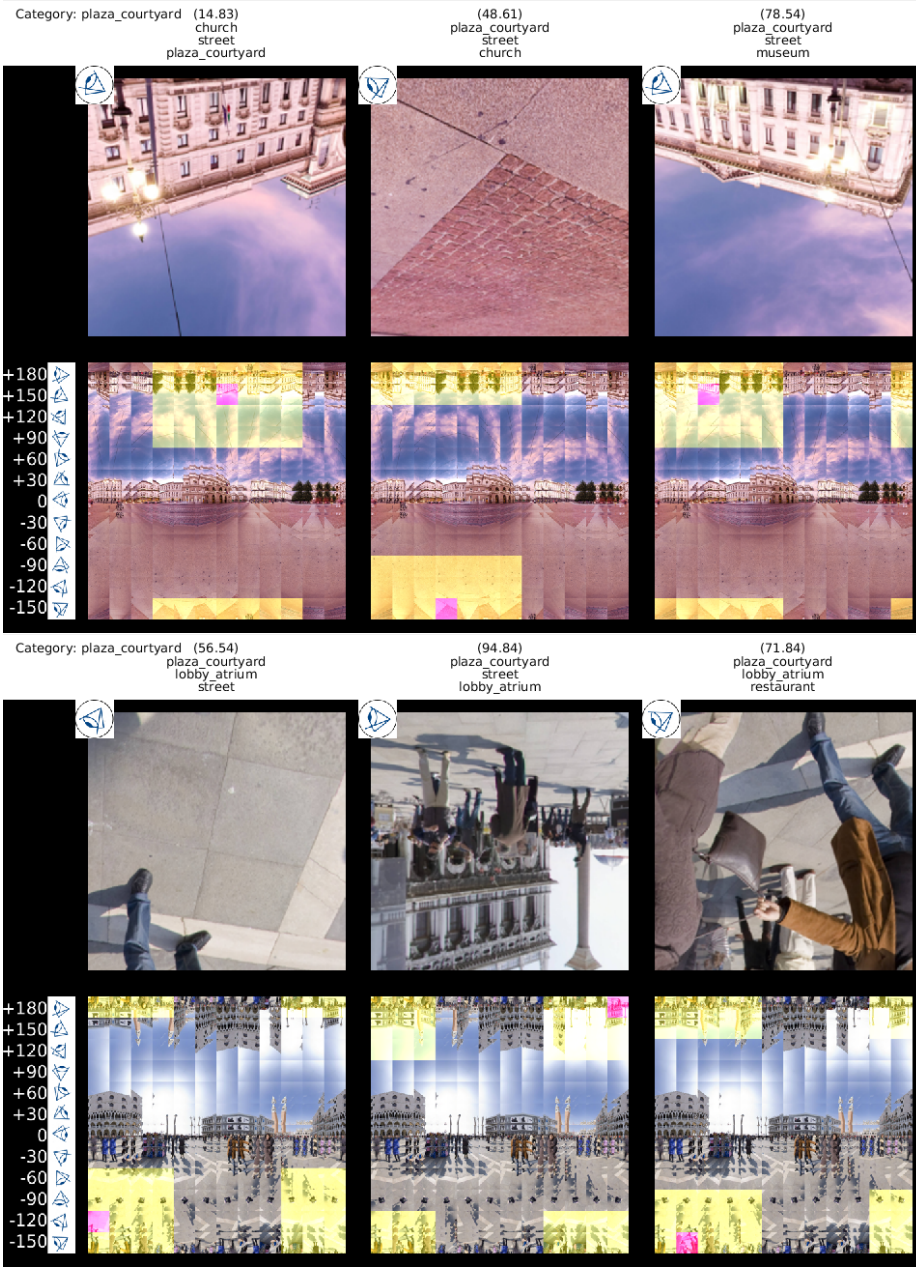

Category: restaurant
(61.13)
restaurant
bedroom
living_room
(57.54)
restaurant
bedroom
living_room
(83.91)
restaurant
living_room
bedroom
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: forest
(31.52)
forest
field
park
(46.04)
forest
field
park
(82.67)
forest
field
park
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150

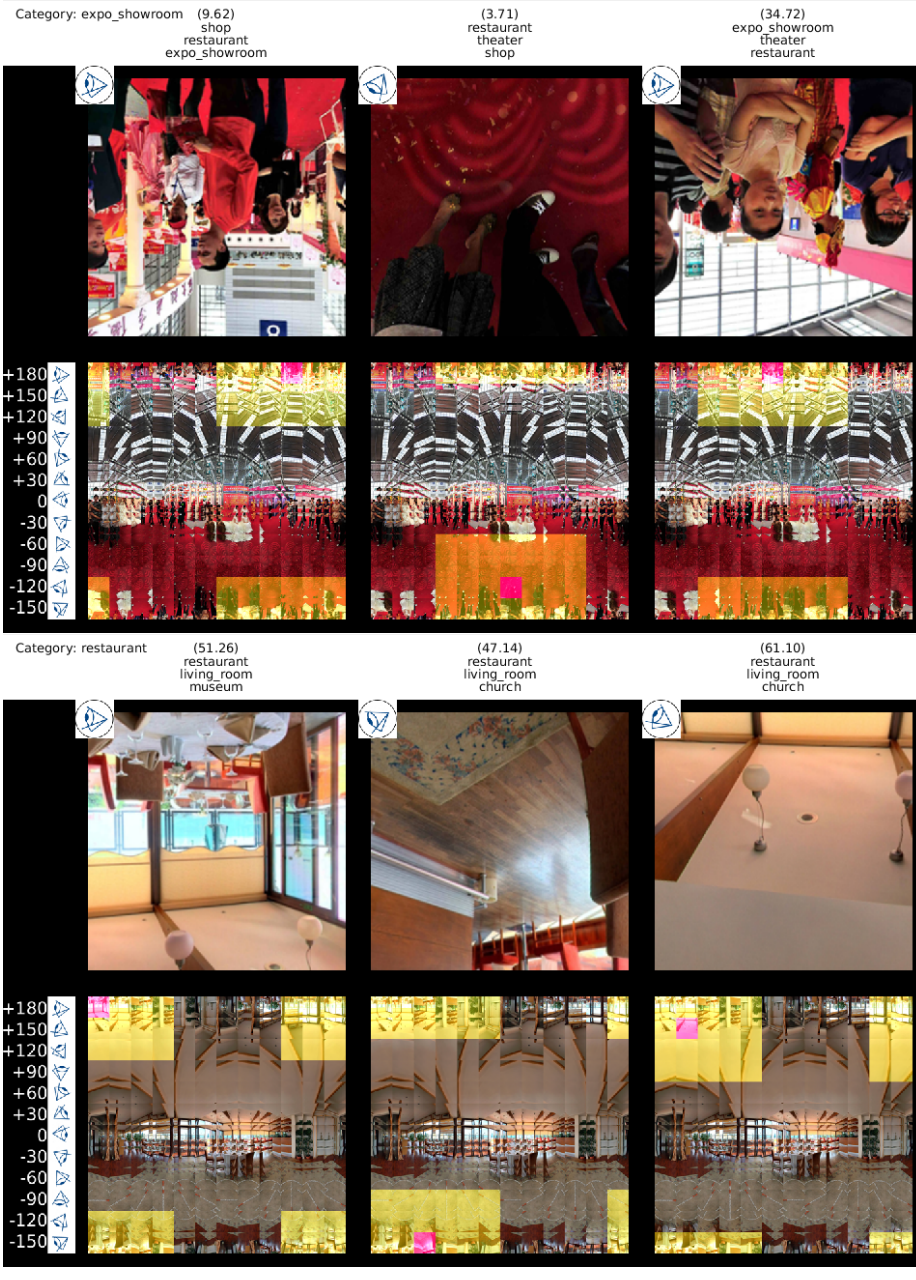

Category: plaza_courtyard (48.06)
plaza_courtyard
lobby_atrium
street
(51.48)
plaza_courtyard
lobby_atrium
old_building
(87.04)
plaza_courtyard
street
ruin
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: forest
(74.60)
forest
mountain
ruin
(93.97)
forest
park
mountain
(96.97)
forest
park
mountain
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150

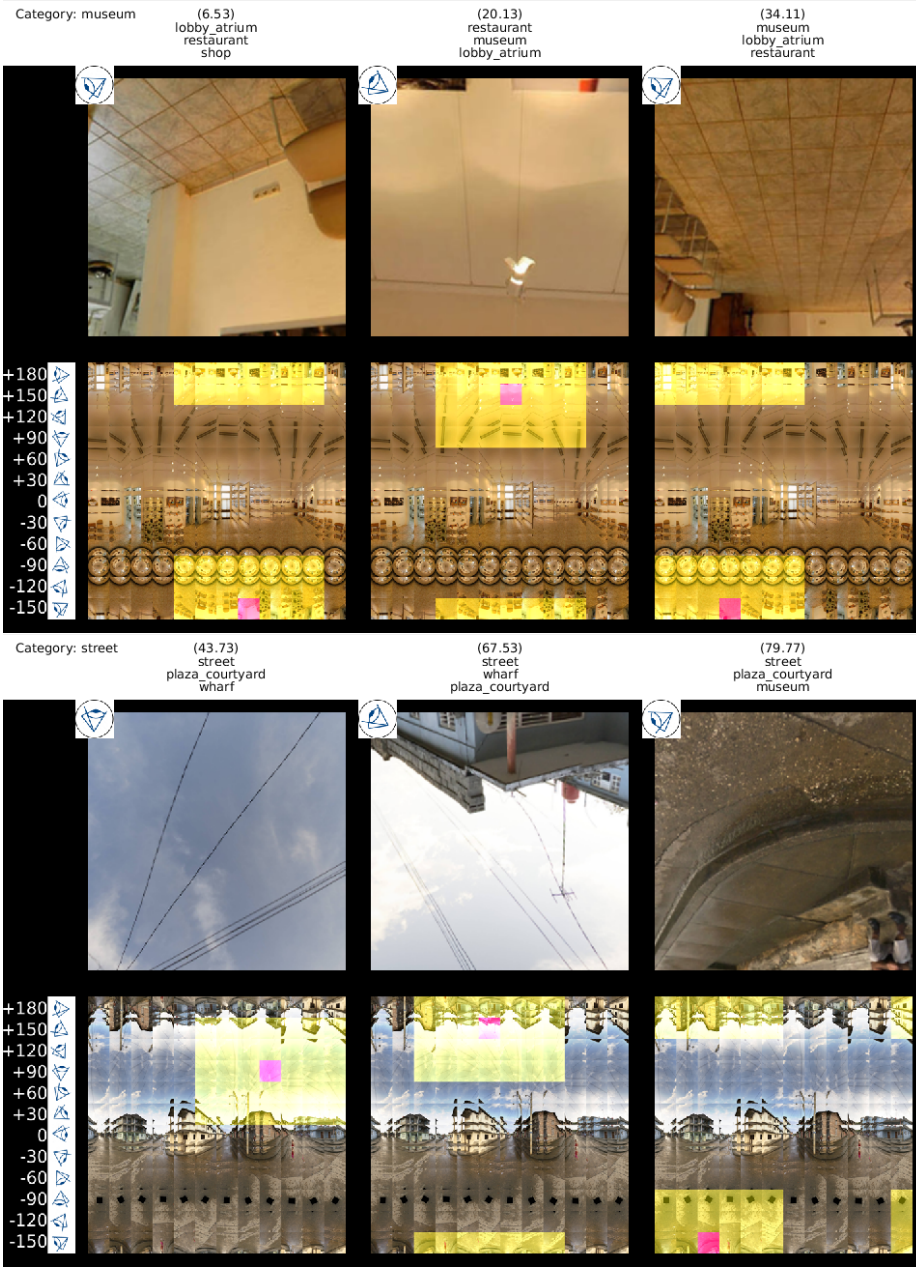

Category: church
(0.53)
forest
cave
beach
(5.00)
street
cave
plaza_courtyard
(37.89)
church
lobby_atrium
street
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: plaza_courtyard
(6.28)
restaurant
train_interior
shop
(11.95)
theater
restaurant
plaza_courtyard
(68.38)
plaza_courtyard
street
theater
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150

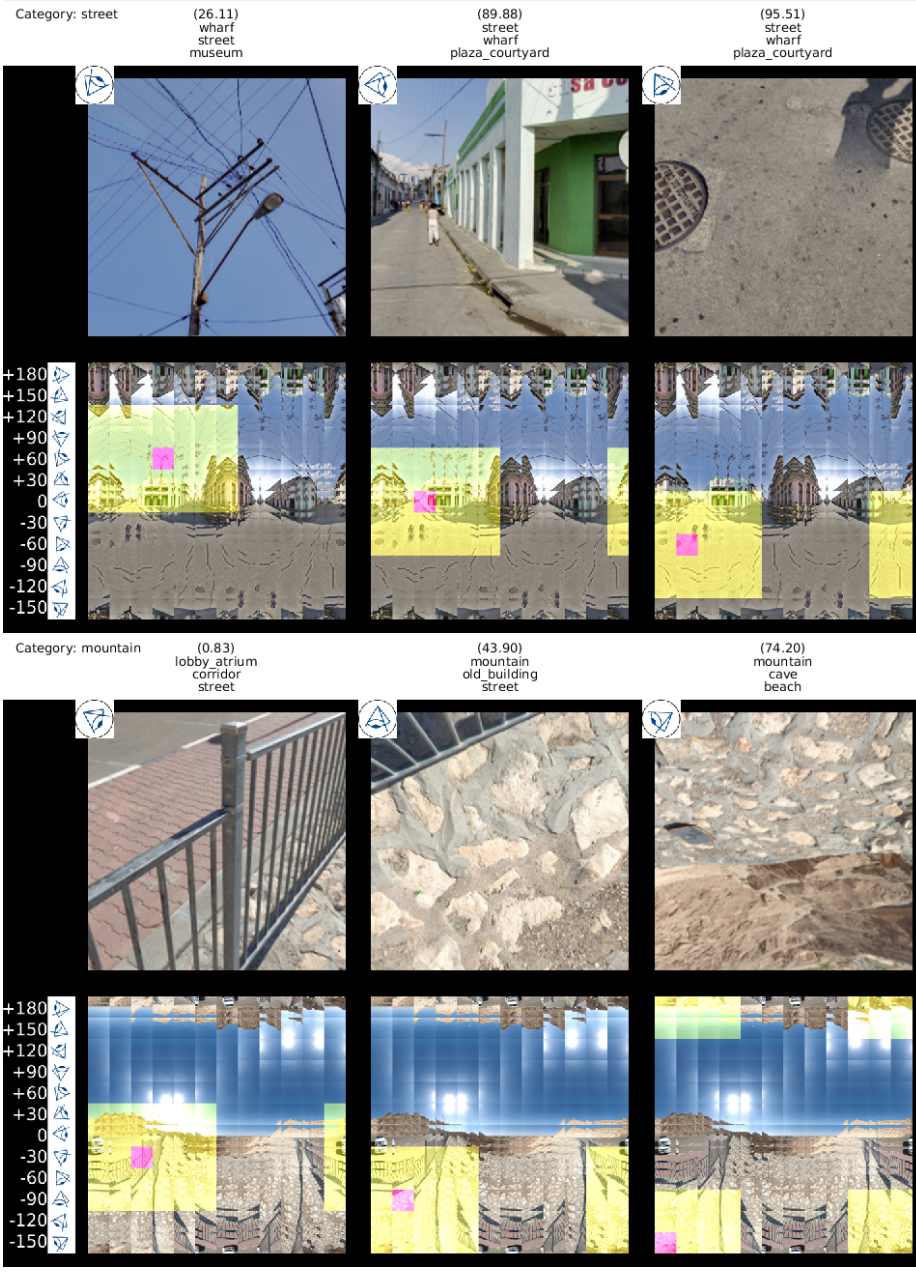

Category: restaurant
(24.46)
restaurant
theater
museum
(27.58)
expo_showroom
restaurant
theater
(77.06)
restaurant
expo_showroom
theater
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: restaurant
(47.37)
restaurant
bedroom
church
(68.14)
restaurant
church
bathroom
(77.90)
restaurant
church
lobby_atrium
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150

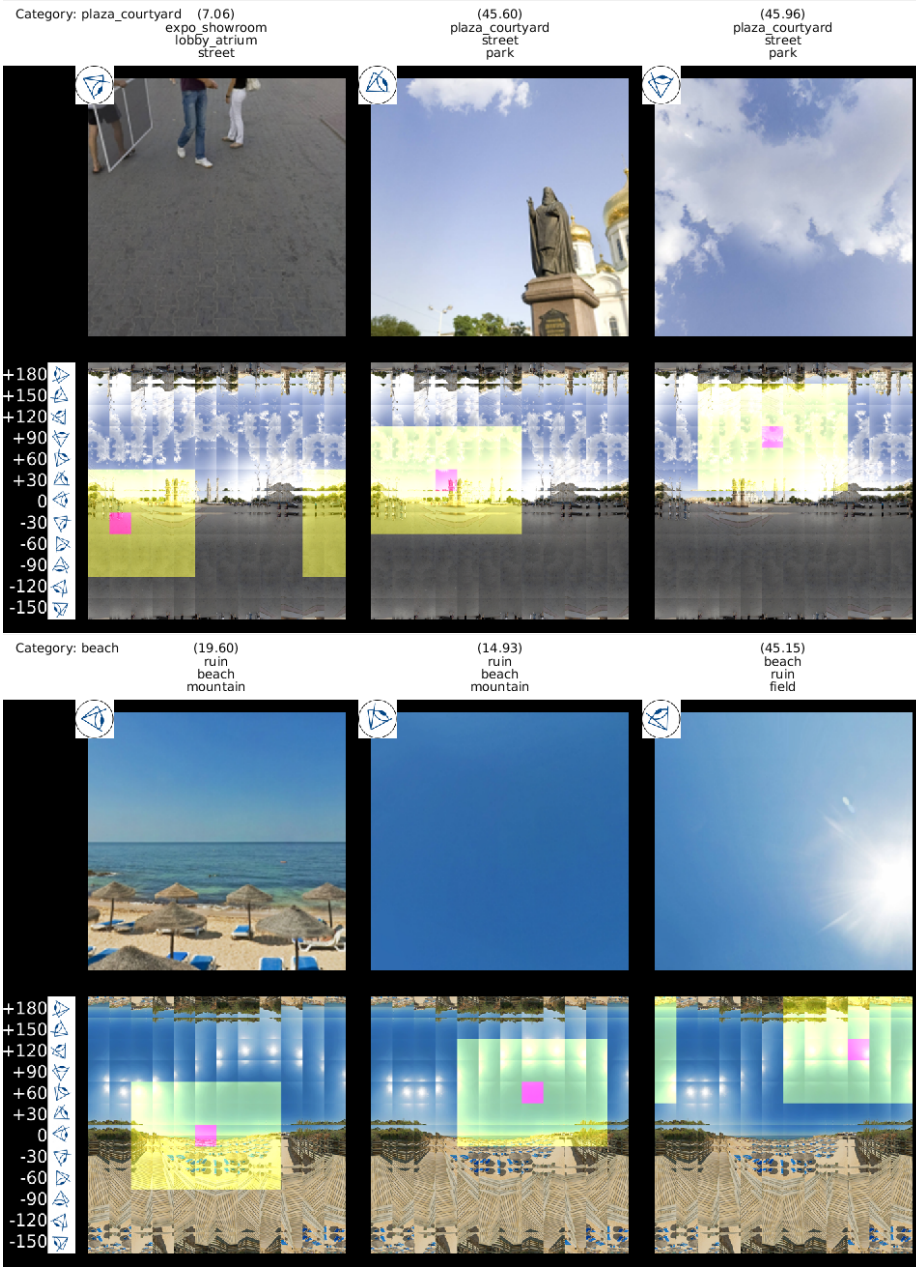

Category: church
(11.81)
restaurant
shop
museum
(58.99)
church
shop
lobby_atrium
(70.98)
church
lobby_atrium
shop
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: plaza_courtyard
(10.48)
field
beach
plaza_courtyard
(14.12)
wharf
field
plaza_courtyard
(2.60)
wharf
beach
field

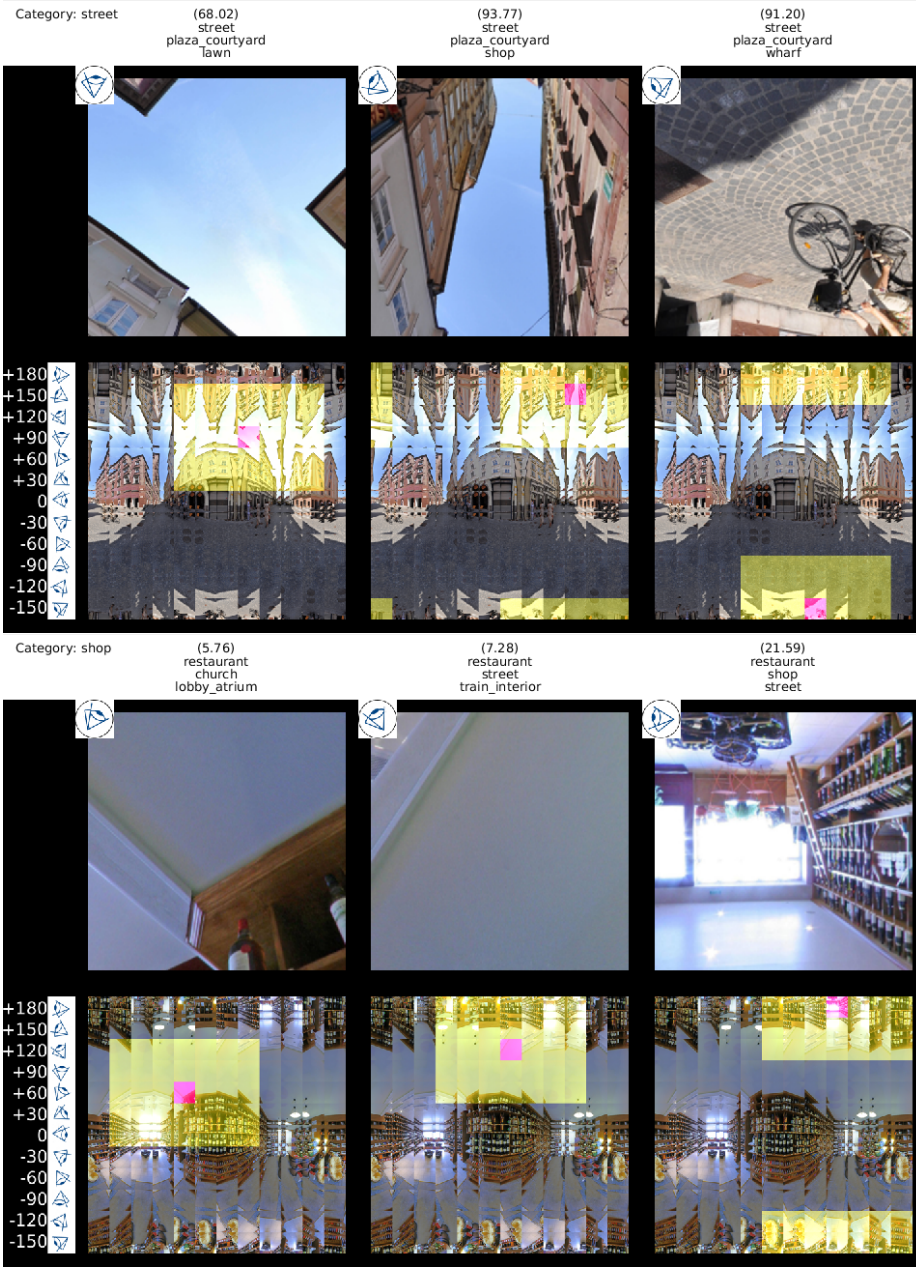

Category: plaza_courtyard (14.83)
church
street
plaza_courtyard
(48.61)
plaza_courtyard
street
church
(78.54)
plaza_courtyard
street
museum
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: plaza_courtyard (56.54)
plaza_courtyard
lobby_atrium
street
(94.84)
plaza_courtyard
street
lobby_atrium
(71.84)
plaza_courtyard
lobby_atrium
restaurant
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150

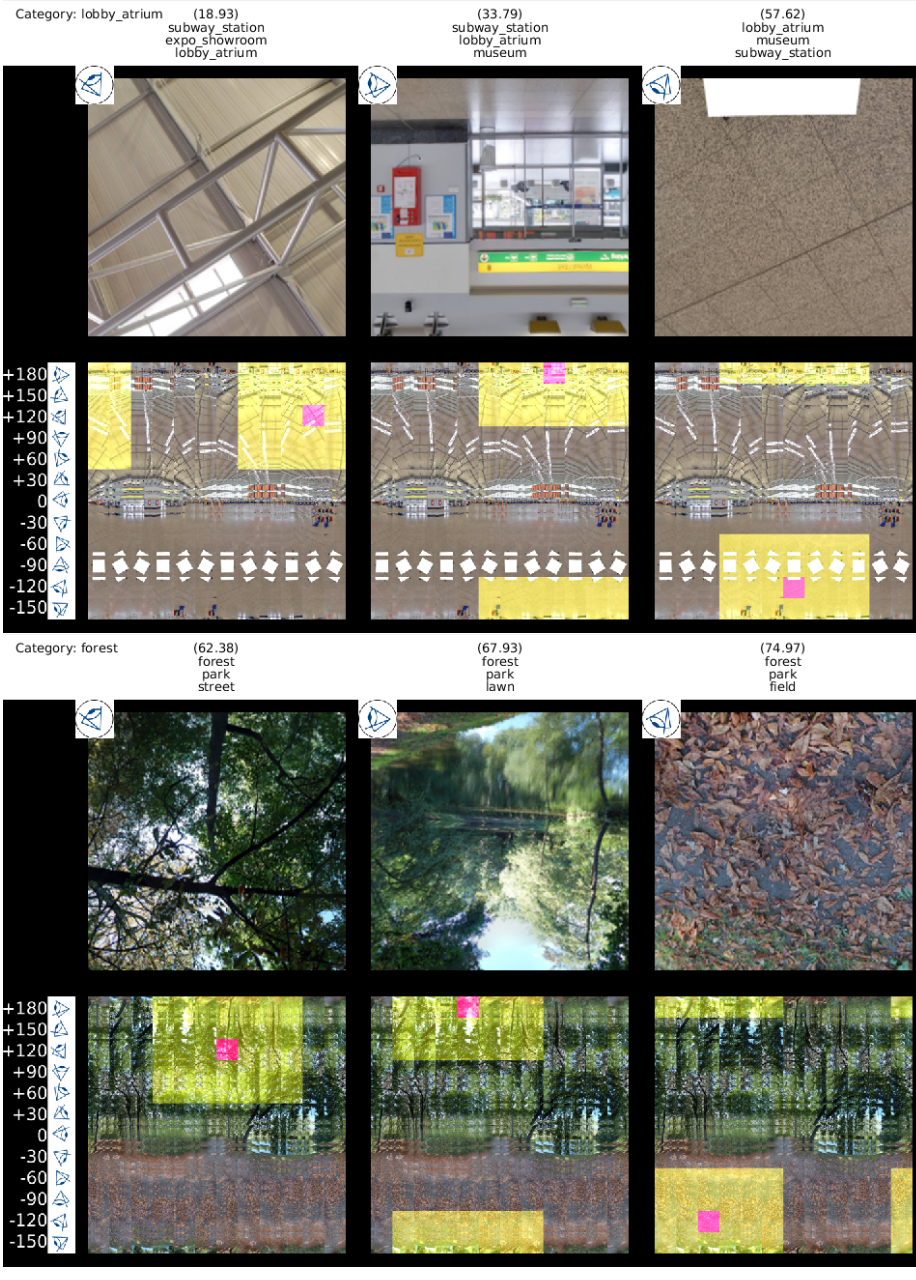

Category: expo_showroom (9.62)
shop
restaurant
expo_showroom
(3.71)
restaurant
theater
shop
(34.72)
expo_showroom
theater
restaurant
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: restaurant (51.26)
restaurant
living_room
museum
(47.14)
restaurant
living_room
church
(61.10)
restaurant
living_room
church
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150


Category: museum
(6.53)
lobby_atrium
restaurant
shop
(20.13)
restaurant
museum
lobby_atrium
(34.11)
museum
lobby_atrium
restaurant
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: street
(43.73)
street
plaza_courtyard
wharf
(67.53)
street
wharf
plaza_courtyard
(79.77)
street
plaza_courtyard
museum
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150


Category: street
(26.11)
wharf
street
museum
(89.88)
street
wharf
plaza_courtyard
(95.51)
street
wharf
plaza_courtyard
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: mountain
(0.83)
lobby_atrium
corridor
street
(43.90)
mountain
old_building
street
(74.20)
mountain
cave
beach
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150


Category: plaza_courtyard (7.06)
expo_showroom
lobby_atrium
street
(45.60)
plaza_courtyard
street
park
(45.96)
plaza_courtyard
street
park
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: beach
(19.60)
ruin
beach
mountain
(14.93)
ruin
beach
mountain
(45.15)
beach
ruin
field
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150


Category: street
(68.02)
street
plaza_courtyard
lawn
(93.77)
street
plaza_courtyard
shop
(91.20)
street
plaza_courtyard
wharf
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: shop
(5.76)
restaurant
church
lobby_atrium
(7.28)
restaurant
street
train_interior
(21.59)
restaurant
shop
street
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150


Category: lobby_atrium
(18.93)
subway_station
expo_showroom
lobby_atrium
(33.79)
subway_station
lobby_atrium
museum
(57.62)
lobby_atrium
museum
subway_station
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150
Category: forest
(62.38)
forest
park
street
(67.93)
forest
park
lawn
(74.97)
forest
park
field
+180
+150
+120
+90
+60
+30
0
-30
-60
-90
-120
-150

3 Failure cases

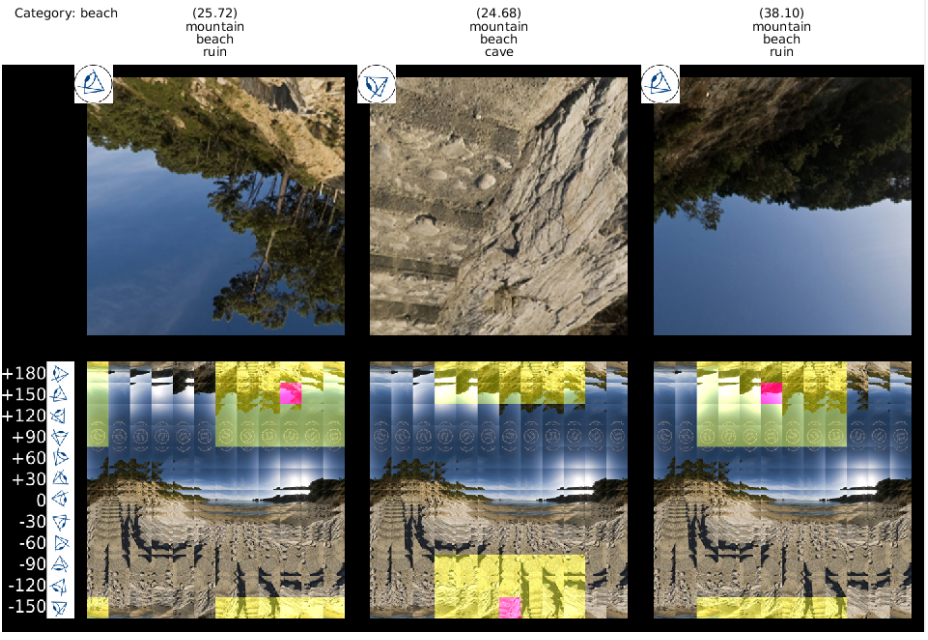


Fig. 5: Failure case: the agent grows increasingly confident about its starting "mountain" guess as the sea and sand are restricted to small portion of the original panorama (near the center of the view grid), and the agent observes neither of these.

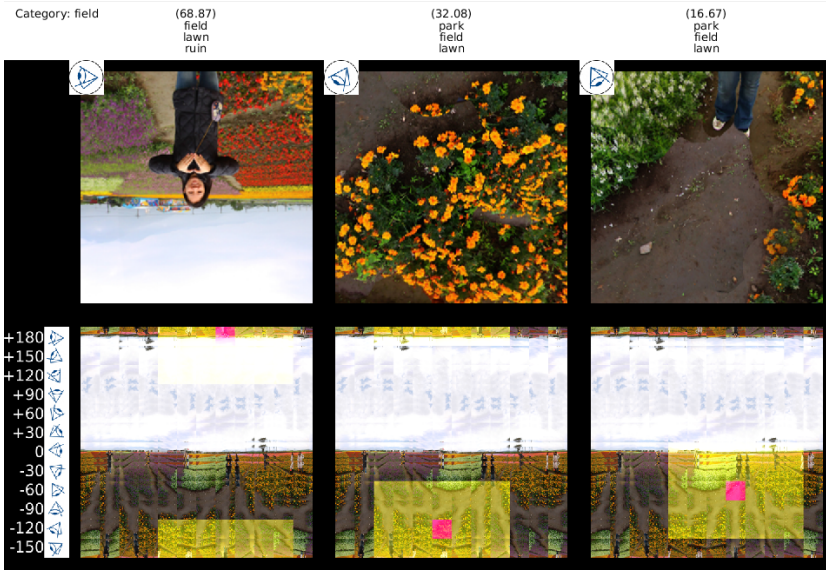


Fig. 6: Failure case: ambiguous categories. This field with multi-colored flowers, neat paths for walking (seen at T=2 and 3), is easily mistaken for a park.

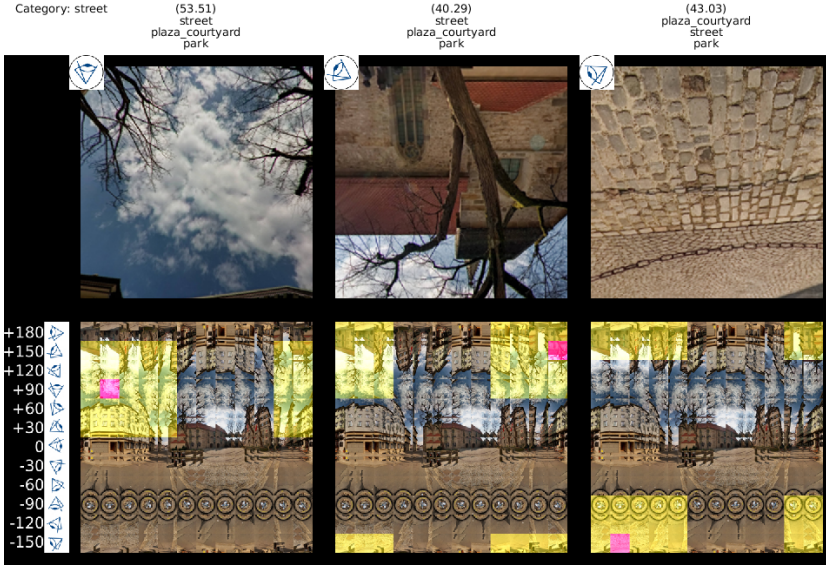


Fig. 7: Failure case: overcorrecting during category disambiguation. The top category after the first two views is "street", but confidence is relatively low (40.29%) and "plaza courtyard" is the top competing category. The system therefore opts to look at the ground where streets typically have tarmac patches, and plazas often have brick pavements. Unfortunately, this European street is brick-paved, and the system overcorrects to guess "plaza courtyard".

4 Random action selection

To illustrate the difficulty of the action selection task, we show views selected by a random action selection baseline, next to those selected by our method, starting from the same starting view.

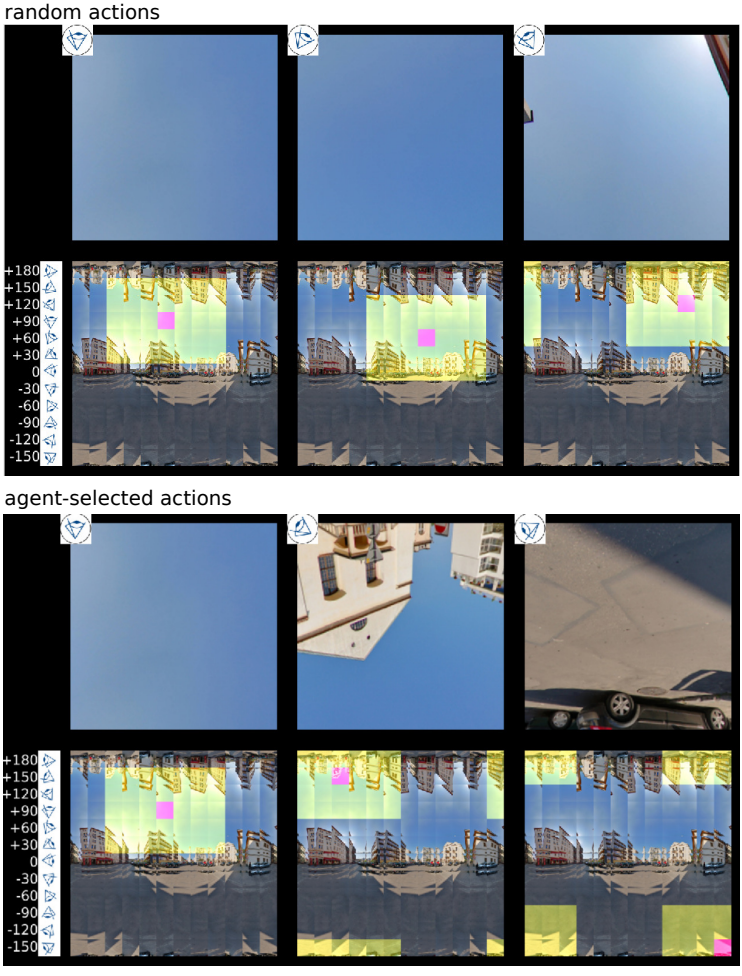


Fig. 8: The random agent selects all uninformative sky views, while the our agent selects disambiguating views

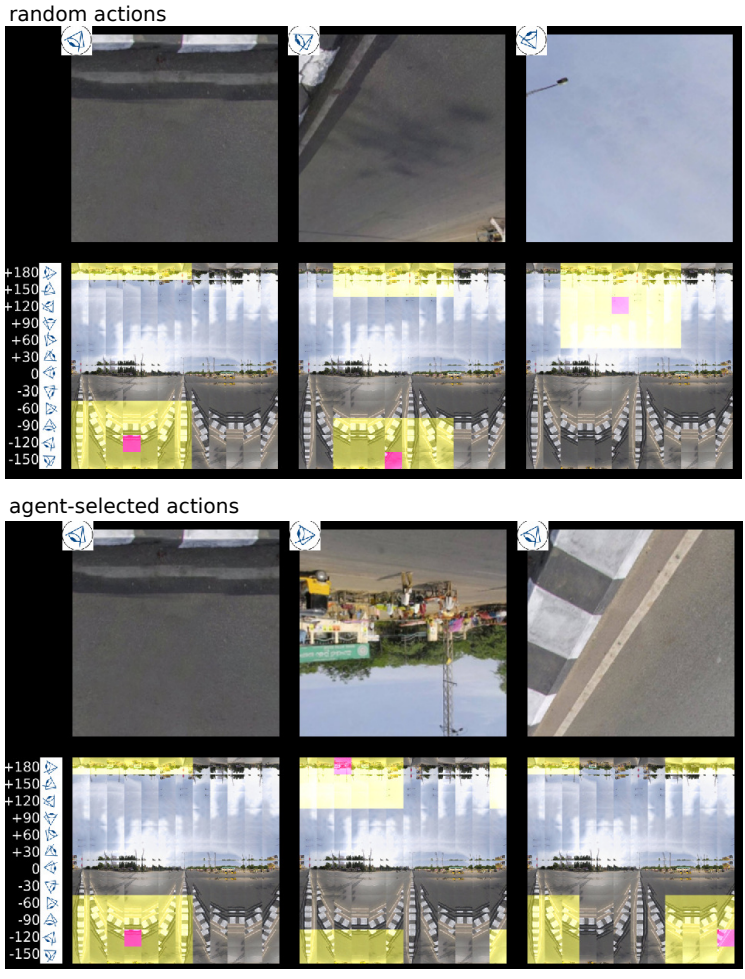


Fig. 9: The random agent views uninformative ground and sky views, while our agent starting at the same view, selects an informative view of the street, and then reviews the pavement curb from a more favorable viewpoint.