Supplementary Material for Gazing at Rewards: Eye Movements as a Lens into Human and AI Decision-Making in Hybrid Visual Foraging

S1. Implementation details of Hybrid Visual Foraging

S1.1. Human psychophysics experiments

The search grid contained either 90, 105, or 120 items, and the positions of these items shuffled every 3 seconds to prevent a fixed reading strategy from the top left to the bottom right of the screen.

Each experiment consists of 10 blocks, where the target objects and their values remain consistent across trials within the same block, but the prevalence of targets as well as the number of target objects may vary across the trials within the block. In each foraging trial, subjects searched for $N \in \{1, 2, 4\}$ target objects, each with a varying number of target instances. Targets and distractors in the hybrid foraging search arrays were randomly selected from a pool of 2,400 unique items used in [14]. The order of the blocks was counterbalanced across subjects.

Each experiment takes 1 hour to complete. A total of 15 subjects were recruited, yielding 750 trials, containing 50514 eye fixations and 12851 mouse clicks. All the experiments are conducted with the subjects' informed consent and according to the protocols approved by the Institutional Review Board of our institution. Each subject was compensated with monetary rewards.

S1.2. Foraging environments for AI models

We sub-sample various combinations of these experimental parameters for the procedural generation of foraging environments: First, the total number of items on the search array is fixed at 105, where 73 serve as distractors, and 32 are designated as target instances. Second, a fixed set of 4 items is randomly selected from the pool of 2,400 items and used as the set of target items throughout the AI model training. Third, there are always 4 target objects present on the search arrays. Fourth, the prevalence ratio among these 4 target items is randomly determined. Finally, the values of the four target items are consistently set at 2, 4, 8, and 12.

S1.3. In-domain and out-of-domain test conditions for AI models

To benchmark AI model performance in hybrid foraging tasks, we introduce two in-domain hybrid foraging conditions that align with the distribution of the training environments that the AI models were optimized to solve. To assess whether the AI models can generalize to out-of-distribution (OOD) hybrid visual search tasks, where experimental parameters differ from those encountered during training, we introduced five out-of-distribution conditions. Below is the summary of all seven conditions:

(1) **In-domain Uneven Value, Equal Prevalence (UnValEqPre)** The prevalence of all four targets was set at 25%, while their values varied, with one target worth 2, another 4, a third 8, and the fourth 16.

(2) **In-domain Uneven Value, Unequal Prevalence (UnValUnPre)** The first target had a value of 2 with 53% frequency, the second a value of 4 with 27%, the third a value of 8 with 13%, and the fourth a value of 16 with 7%.

(3) **OOD - Even Value, UnEqual Prevalence (EqValUnPre)** Each of the four target objects had a value of 8, but their prevalence varied, with 53% 27% 13%, and 7% respectively.

(4) **OOD - Unseen target objects (UTargets)** We replace the target and distractor objects from the pool of 2400 items used for training with unseen items, while maintaining the other experimental parameters.

(5) **OOD - Unseen value combinations (UValues)** The prevalence of all four targets was randomized, and their absolute values exceeded the range used for training, with their relative values changing in either arithmetic or geometric series. Specifically, the value combinations included (1, 2, 3, 4), (1, 2, 4, 8), (8, 9, 10, 11), (8, 16, 32, 64), (16, 18, 20, 22), and (16, 32, 64, 128).

(6) **OOD - Unseen total item numbers (UItemNum)** Unlike during training, when the total number of items on the screen was consistently 120, the search arrays were populated with either 90 or 105 items.

(7) **OOD - Unseen target object sizes (USetSize)** The set size of target objects was manipulated to include either one or two; specifically, the single target object was valued at 4, while the two target objects were valued at 4 and 16.

S1.4. Eccentricity dependent pooling

We replicated the eccentricity dependent pooling from [51], aligning it with neurophysiological recordings in macaque monkeys. A conversion of 30 pixels to 1 degree of visual angle (dva) was applied to match human behavioral experiments.

To examine how the layer-specific scaling factor γ_l affects search efficiency and average saccade amplitudes, we varied γ_l by a coefficient β. Tab. S1 shows that increasing β reduces saccade size and cumulative rewards in UnValEqPre. VF with $\beta = 1$ best matches human saccades and achieves comparable rewards.

	Humans	$\beta = 1$	$\beta = 2$	$\beta = 4$
Avg.Sac.Size (dva)	4.05	4.06	2.26	2.12
NormScore (%)	87.4	72.6	6.4	2.95

Table S1. Ablation of layer-specific scaling factor in eccentricity dependent pooling

S2. Reinforcement learning

We recall the Markov decision process (MDP) framework with finite state space $\mathcal{S}$ and action space $\mathcal{A}$. An MDP is defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, Pr, r, \gamma)$, where $Pr : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. Given an initial state s_0, the goal of reinforcement learning (RL) is to learn a policy π that maps a state $s \in \mathcal{S}$ to a distribution $\pi(\cdot \mid s)$ over the action space, aiming to maximize the expected cumulative discounted reward.

For any policy π, the action-value function $Q^\pi(s, a)$ represents the expected return starting from state s, taking action a, and thereafter following policy π. It is defined as $Q^\pi(s, a) = \mathbb{E}_{\pi,Pr}\left[\sum_{t=0}^{\infty} \gamma^h r(s_t, a_t) \mid s_0 = s, a_0 = a\right]$, where $\mathbb{E}_{\pi,Pr}(\cdot)$ denotes the expectation over trajectories generated by following π under the transition dynamics Pr. The state-value function $V_s^\pi(s)$ is the expected return starting from state s and following π, while the advantage function $A^\pi(s, a)$ is given by $A^\pi(s, a) = Q^\pi(s, a) - V_s^\pi(s)$, quantifying the relative advantage of taking action a in state s under policy π.

S2.1. Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a policy gradient method designed to improve the stability and efficiency of policy updates. PPO ([109]) uses a surrogate objective function with a clipping mechanism to prevent large, destabilizing updates. The surrogate objective function is defined as:

$$L^{\text{CLIP}} = \mathbb{E}_{\pi_{\theta_{\text{old}}}}\left[\min\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)}\tilde{A}^{\pi_{\theta_{\text{old}}}}(s, a), \text{clip}\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)}, 1 - \delta_0, 1 + \delta_0\right)\tilde{A}^{\pi_{\theta_{\text{old}}}}(s, a)\right)\right].$$

where $\tilde{A}^{\pi_\theta}$ is an estimate of the advantage function, and δ_0 is a hyperparameter controlling the extent of clipping.

In this formulation, the first term inside the $\min$ operator is the standard policy gradient objective, while the second term applies the clipping mechanism to ensure that the policy update does not result in excessively large changes. This clipping mechanism is crucial for maintaining the stability of the learning process. Our VF leverages Generalized Advantage Estimation (GAE) [108] for advantage calculation and TD(λ) for value estimation [11]. This choice is motivated by the computational efficiency of TD(λ) compared to Monte Carlo sampling [56], as noted in the work of GAE.

S2.2. Additional training and implementation details

In practice, rather than learning two separate policies for actions at different times i.e., the mouse click at t and the fixation at $t + 1$, we modify the click policy $\pi_c(\cdot|s)$ to output the binary click decision at $t + 1$, aligning it with the fixation policy. Empirically, this leads to more efficient training and faster convergence. Importantly, this modification does not alter the hybrid foraging setup, as VF can fixate on the same grid cell consecutively. In other words, VF may initially decide not to click the item fixated at $t + 1$ but can later decide to click it by fixating on the same item again at the next time step.

The search image I_S has a resolution of 1024×1024 pixels, while the target images I_T are 64×64 pixels, corresponding to the size of one cell within a 16×16-sized grid in I_S. The search feature map ϕ_S has dimensions $32 \times 32 \times 512$, while the target feature maps $\phi_T^{1:N}$ are $2 \times 2 \times 512$. We implemented the target modulation function $\mathcal{M}$ with a stride of 2, resulting in M_F with dimensions $16 \times 16 \times N$, where the spatial size matches the grid size of the search image.

Our VF was trained over 3 million timesteps in the first stage, taking approximately 3 days, and over 0.6 million timesteps in the second stage, taking approximately 1 day. All training was conducted on a single NVIDIA RTX A6000 GPU.

S2.3. Deep Q-leaning

Value-based reinforcement learning method solves MDP problem by getting an optimal value function. The optimal value function is defined by $V_s^*(s) = \sup_\pi V_s^\pi(s)$ and similarly $Q^*(s, a) = \sup_\pi Q^\pi(s, a)$. We use deep Q-learning (DQN) as a baseline method, which obtains Q^* based on the update $Q_{i+1}(s_t, a_t) = (1 - \alpha_t) Q_i(s_t, a_t) + \alpha_t (r_t + \gamma \max_a Q_i(s_{t+1}, a))$,

where $\alpha_t \in (0, 1)$ is the learning rate. We employ the ε-greedy approach for action selection based on a value function, which means that we pick $\arg\max_a Q_i(s, a)$ with $1 - \varepsilon$ probability and a random action with probability ε.

As our baseline, we do not incorporate target feature modulation or target value modulation. Instead, we designed a deep neural network (DNN) to predict the value function in an end-to-end fashion. This DNN takes as input a search image, target images, and target values. It uses two 2D-CNNs to extract features from the search and target images, respectively, then concatenates the search features, target features, and target values. An MLP with three fully connected blocks outputs an approximate state-action value for each action. Following the standard DQN used in [92], our approach incorporates the key techniques of target networks and experience replays.

S2.4. PPO Hyperparameters

This hyperparameters used at two training stages are listed as follow:

Training stage	Stage 1	Stage 2
Discount (γ)	0.99	0.99
GAE parameter (λ) [108]	0.95	0.95
Batch size	512	512
Epochs	5	1
PPO clip range	0.05	0.05
Entropy coefficient [90]	0	0.001
Learning rate	2e-4	2e-4

Table S2. PPO Hyperparameters.

S3. Additional experiment results

S3.1. Ablations reveal critical component designs

We systematically ablated several essential components in our VF model and reported their results in Tab. S3. (1) Rather than using reinforcement learning, we train VF on human eye movements and mouse clicks through supervised learning (**Behavior Cloning**). The lower Norm. Score of Behavior Cloning indicates that human eye movement data is limited, leading to model overfitting. Hence, the model struggles to generalize to unseen target value and prevalence combinations. (2) We replace the transformer-based decision-making module with a 2D-CNN, referred to as **VF(2D-CNN)**. The lower Norm. Score of this ablated model indicates that the transformer architecture, with its ability to capture long-range dependencies and global context through self-attention, leads to better decision-making. (3) We ablate VF by replacing the learnable value encoder with explicit value embeddings and directly feeding them into the transformer (**Explicit Val. Emb.**). The small drop in Norm. Score suggests that a learnable value embedding is more effective for making better decisions. (4) We remove the permutations of target and value pairs (**W/o Augmentation**), resulting in a significant drop in Norm. Score, especially under the EqValUnPre condition. This indicates that the data augmentation in VF is crucial for enhancing generalization to OOD hybrid foraging tasks.

Ablations	UnVal EqPre	UnVal UnPre	EqVal UnPre
Behavior Clone	61.7	48.5	60.1
VF (2D-CNN)	**75.3**	63.7	70.0
Explicit Val. Emb.	69.2	56.7	61.8
W/o Augmentation	51.3	52.0	52.2
Full VF (ours)	72.6	**67.1**	**81.6**

Table S3. Ablation studies reveal critical design choices of our VF model. Norm.Score for various ablated models are reported over UnValEqPre, UnValUnPre, and EqValUnPre conditions. See Sec. S3.1 for ablated models. Best is in bold.

S3.2. Human motor response

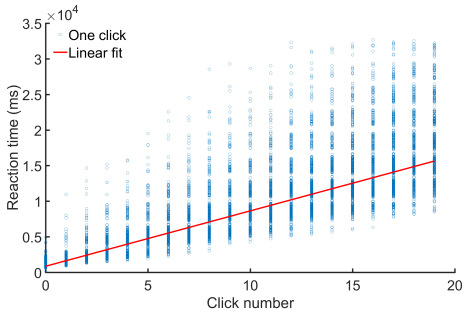

Figure S1. Human reaction time in a trial as a function of click numbers. We recorded clicks in all subjects' trials and showed the result of the linear fit.

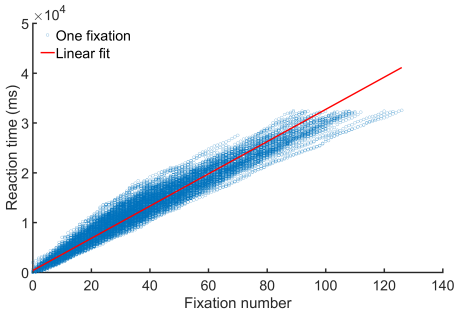

Figure S2. Human response time in a trial as a function of fixation numbers. We recorded fixations in all subjects' trials and showed the result of the linear fit.

S3.3. Human fixation duration

Fixation durations are longer on targets with higher values. We also investigate human fixation durations on targets with varying values under the UnValEqPre and UnValUnPre conditions. From Appendix Fig. S3, surprisingly, we found that humans tend to spend more time fixating on higher-value targets compared to those with lower values. For example, under the UnValEqPre condition, the mean eye fixation duration is 344 milliseconds on targets valued at 16, while the duration is 309 milliseconds on targets valued at 2. This may be attributed to the enhancement of learning and memory, where longer fixation durations facilitate cognitive processing and reinforce associations between previous decision-making strategies and positive outcomes.

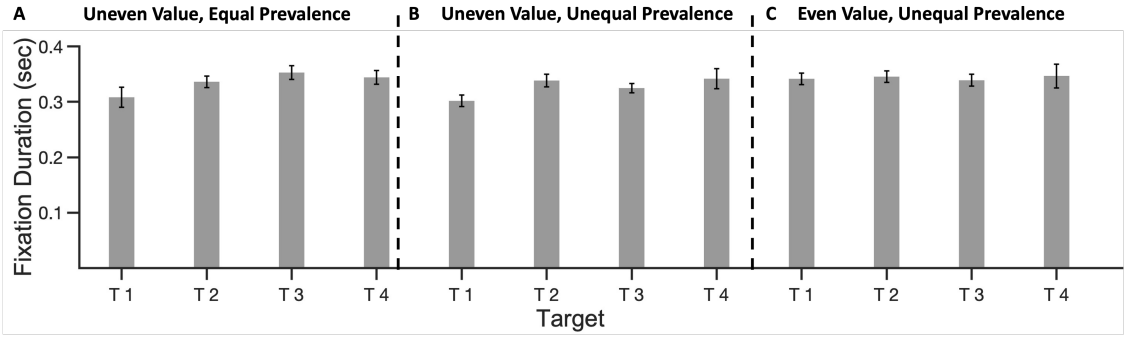

Figure S3. Eye fixation duration for different types of targets in UnValEqPre (T1: $mean = 309ms$, T2: $mean = 336ms$, T3: $mean = 353ms$, T4: $mean = 344ms$), UnValUnPre (T1: $mean = 302ms$, T2: $mean = 339ms$, T3: $mean = 325ms$, T4: $mean = 342ms$) and EqValEqPre (T1: $mean = 342ms$, T2: $mean = 346ms$, T3: $mean = 339ms$, T4: $mean = 347ms$). Fixation durations are significantly different for targets with different values in UnValEqPre condition ($p = 0.13$) and UnValUnPre ($p = 0.12$). Fixation durations are not significantly different for targets with same value in EqValEqPre ($p = 0.98$).

S3.4. Average reward within fixation area

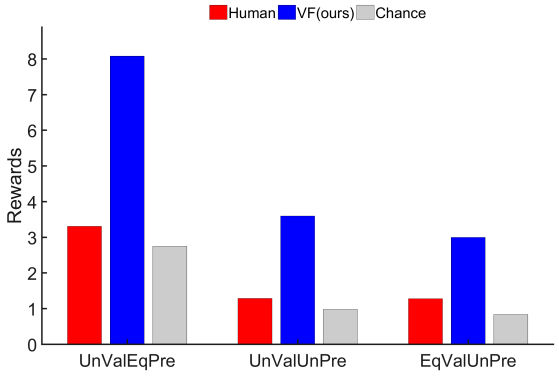

Figure S4. Mean rewards of all target objects within a radius of 1.5 degrees of visual angle around each fixation predicted by our VF model (UnValEqPre: $mean = 8.08$, UnValUnPre: $mean = 3.60$, and EqValEqPre: $mean = 3.00$), made by human subjects (UnValEqPre: mean=3.30, UnValUnPre: mean=1.29, and EqValEqPre: mean=1.28) and predicted by the chance model (UnValEqPre: $mean = 2.75$, UnValUnPre: $mean = 0.98$, and EqValEqPre: $mean = 0.84$). For all three conditions, both human subjects and our VF model tend to fixate on regions associated with average rewards significantly higher than that derived from random fixations. We conducted two-tailed t-tests. All p-values are below 0.01.

S3.5. Click behavior

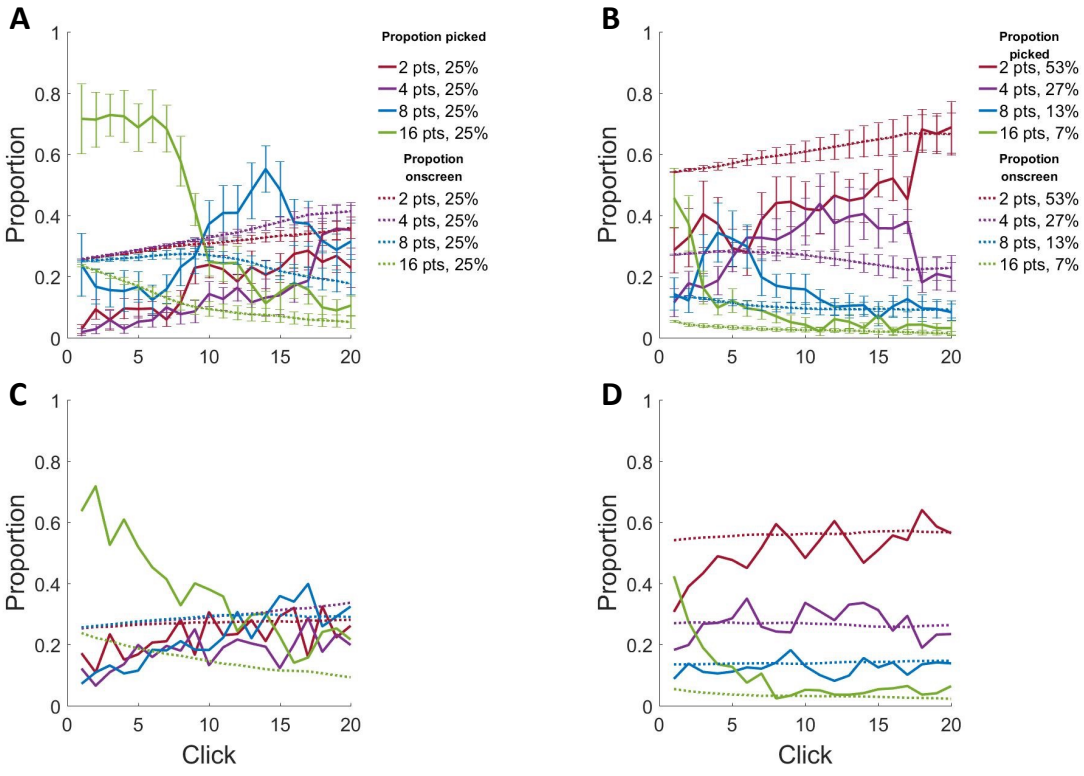

Figure S5. Proportion as a function of number of clicks for (A) humans in UnValEqPre, (B) humans in UnValUnPre, (C) VF model in UnValEqPre, and (D) VF model in UnValUnPre. Solid lines are click proportions of different types of targets. Dash lines are proportions of different targets that remain on screen. Colors indicate the target types.

S3.6. Scanpath similarity

We calculated the results of ScanMatch and Fixation Edit Distance (FED) to assess the scanpath similarity between humans and AI models (Tab. S4). Best are in bold. Results show that our VF generates more human-like scanpaths than chance but with lower similarity than within-subject scanpaths across repeated trials and between subjects performing the same trials.

	Within-subject	Between-subject	VF	Chance
ScanMatch ↑	**0.3556**	0.3173	0.2474	0.1596
FED ↓	**39.60**	47.50	55.90	60.10

Table S4. Scanpath similarity.

S3.7. Penalty ablation

The -1 penalty for distractors ensures consistent rewards for both the agent and humans. Clicking on blank areas wastes time and reduces available clicks, so we apply a -0.01 intrinsic penalty to discourage this and prevent suboptimal behavior in VF. We conduct ablation experiments to assess the impact of this instinctive reward and find that it has no significant effect on the final result (see Tab. S5).

	-1	-2	-0.01	0
NormScore (%)	75.49	76.5	72.6	79.9

Table S5. NormScore of UnValEqPre condition trained with different instinctive reward.

S3.8. External baseline

We introduced four external baselines that do not utilize foveated vision: (1) IVSN: This baseline iteratively selects the maximum from four-channel similarity maps and applies infinite inhibition of return. (2) IVSN-NN: A variant of IVSN, where the attention map is modulated by value, incorporating an additional neural network module trained via behavior cloning. (3) pre-GF: The pre-trained GazeFormer model [93]. (4) GF: The GazeFormer model fine-tuned on our in-domain data. We tested these models in our OOD tasks. Our VF outperforms all of them (see Tab. S6). The inferior performance of GF and pre-GF compared to our VF is due to their failure to account for descriptive texts of multiple targets with different values during the OOD foraging.

	EqVal UnPre	UValues	UItemNum	USetSize
GF	1.6	0.4	0.61	0.39
pre-GF	0.93	0.83	0.18	0.79
IVSN	76.03	47.6	47.05	**77.76**
IVSN-NN	72.72	46.85	47.18	64.03
Ours	**81.63**	**70.87**	**65.16**	72.34

Table S6. NormScore (%) of external baselines tested in OOD tasks. Best is in bold.

S3.9. Qualitative results of humans and our VF model

We visualized the scanpaths and click locations of our VF model and a human subject in Fig. S6B and C, respectively. As item positions were shuffled every three seconds in our experiment, we only depict clicks and fixations that occurred before the first shuffle. Both the human and the model primarily clicked on the highest-value targets (red balls), selecting them in 3 out of 6 clicks, indicating that target values strongly influenced their click decisions.

S4. Future works

First, we observed that humans occasionally clicked on items they were not directly fixating on, while VF assumes eye movements always align with the locations at which foraging decisions are made. Second, a strong priming effect was evident in humans, especially when target values were equal, showing the long-lasting influence of prior experiences on human decisions. Our VF currently lacks the ability to model such long-term dependencies, as it does not have a working

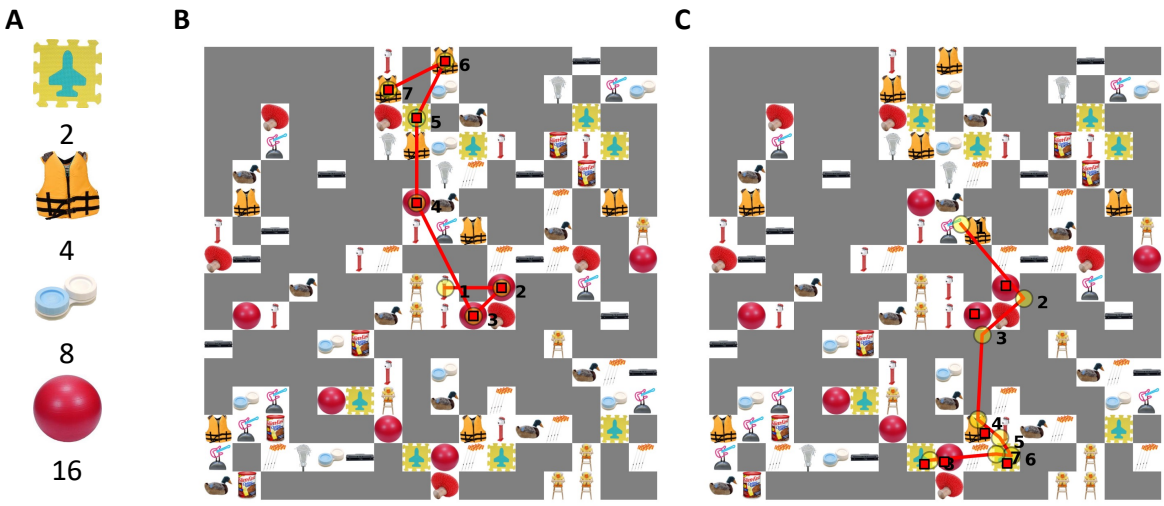

Figure S6. Qualitative results of our VF model evaluated under the UnValEqPre condition. (A) displays the targets along with their corresponding values for this example trial. (B) illustrates the model's scanpaths and click locations, while (C) presents those of a human subject. Yellow dots indicate fixations, with connecting red lines representing visual scanpaths, and black numbers denoting fixation order. Red squares mark clicked items.

memory integrating reinforcements from past actions into current decisions. Third, in hybrid foraging, humans actively compare fixated items with those in memory, a process known as memory search. Our VF assumes perfect memory search, where all targets are compared to the fixated item simultaneously. Fourth, fixation duration is another important aspect of human eye movement decisions. However, our VF model currently lacks the ability to capture fixation duration. Lastly, real-world environments may present additional challenges, such as target occlusions and physical constraints imposed by scene contexts. Extending the study of hybrid visual foraging beyond simplistic stimuli in controlled experimental settings remains an intriguing research direction.