Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Viewport selection for automatic StarCraft observation with learned scene interest and switching costs

Ho-Taek Joo

hotaek87@gm.gist.ac.kr

Gwangju Institute of Science and Technology

JongWon Kim

Gwangju Institute of Science and Technology

Article

Keywords:

Posted Date: November 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11345263/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Viewport selection for automatic StarCraft observation with learned scene interest and switching costs

Ho-Taek Joo[1*] and Jongwon Kim[1]

[1]Gwangju Institute of Science and Technology, Department of AI Convergence, Gwangju, Republic of Korea.

*Corresponding author(s). E-mail(s): hotaek87@gm.gist.ac.kr;

Abstract

Automatic esports observers must decide both which scene to show and when to leave the current view. We study this problem in StarCraft: Brood War by learning scene interest from multiple human viewing trajectories and combining predicted interest with a cost for changing the camera position. This separates the prediction of relevant scenes from the decision to interrupt the current view. Under a common evaluation protocol, selection based on current interest achieved mean human-view overlap of 0.6257, compared with 0.5753 for a retrained Mask R-CNN observer. Paired match comparisons favored this controller over the detector and two rule-based observers. Uniform sampling of training candidates improved selection in both current-state and history-based models. At the primary switching cost, camera changes fell by about 40% relative to zero cost, with a small loss in overlap and longer remaining moves. An additional distance penalty reduced movement with a small observed overlap loss. Adding learned future viewing value provided no consistent benefit over current-interest selection. These results identify training-candidate sampling and explicit switching cost as useful, separately testable components of automatic viewport selection.

Keywords: automatic esports observation, sequential viewport selection, switching cost, multi-observer supervision, StarCraft: Brood War

1 Introduction

An esports observer operates the broadcast camera rather than playing the game. In StarCraft: Brood War, combat, troop movement, production and expansion can occur in different places at the same time, while viewers see only one part of the map. Early automatic observers addressed this selection problem with human-defined rules. Events such as an attack or scouting activity received a priority, and dwell rules determined how long the camera should remain at the selected scene. Public examples include the Student StarCraft AI Tournament (SSCAIT) Observer Module and the Starcraft AI Tournament Manager's AutoObserver, referred to here as the AIIDE rule observer [1, 2]. These systems make camera decisions easy to trace, but the relevant events and their priorities must be specified in advance.

Learning-based observers replaced hand-written definitions of scene interest with human viewing data. Joo *et al.* represented a human viewport as a two-dimensional spatial object and trained Mask R-CNN to detect watched regions at sampled frames [3, 4]. Their region-of-common-interest target used overlapping human views to learn locations that attracted shared attention. This approach learns where people look without requiring a hand-written priority for each game event.

Detecting an interesting region, however, does not by itself determine whether the camera should move there now. Suppose the broadcast shows production at one base while combat begins elsewhere. The observer must weigh the interest of the new event against continuing the current view. The screen selected now also becomes the current camera position at the next step. These dependencies motivate treating automatic observation as a sequence of connected camera choices rather than a series of independent location predictions.

Reinforcement learning provides a framework for describing connected decisions through states, actions and rewards [5]. In camera control, this makes it possible to express both the value of showing an interesting scene and the cost of interrupting the current view. Previous studies have used reinforcement and inverse reinforcement learning to represent cinematographic objectives and learn camera preferences from demonstrations [6, 7]. In a recorded game, camera choices do not change combat or unit motion, but they do determine where the next camera decision begins. This structure lets us study the benefit of future viewing value separately from the ability to predict current scene interest.

Human observation adds another important aspect: there may be several reasonable screens at the same moment. One observer may show the main engagement while another follows an expansion or production activity. Learning a single correct coordinate would discard these alternative choices. We instead use overlap with multiple human viewports as a graded measure of scene interest, defined here by spatial agreement with the recorded views. A candidate can therefore receive credit for showing a plausible scene without exactly copying one person's camera. Figure 1a illustrates this ambiguity, and Fig. 1b–c shows how human-derived interest and the previous camera contribute to selection.

We develop a multi-observer viewport selector that learns current scene interest and applies an explicit cost when the camera changes. A second controller adds a learned estimate of future viewing value within the same decision framework. The study makes

three contributions. First, it combines candidate-level interest learning with explicit transition control while separating game progression from camera memory. Second, controlled comparisons hold the number of training candidates fixed while examining sampling, human-view aggregation, recent-state features and future-value correction. Third, a conditional bound on continuation-value differences and fixed-model analyses of cost magnitude and distance dependence clarify when future value can change a choice and how transition cost changes camera behavior. Comparisons with a retrained detector and two rule observers assess spatial agreement with human viewing under a common protocol.

2 Related work

2.1 Automatic esports observers

Rule-based observers connect game events directly to camera actions. SSCAIT uses event priorities and focus durations, while the Tournament Manager AutoObserver follows units involved in attacks and selected construction activity [1, 2]. Event-driven directing also appears beyond StarCraft: Valve's Source SDK HLTV director schedules player and fixed-camera shots from game events [8]. The shared design principle is to make viewing choices through explicit event and timing rules.

Human-trained observers move the source of scene importance from those rules to recorded viewing behavior. The Mask R-CNN observer of Joo *et al.* learns spatial targets from individual viewport masks and a region of common interest [3]. This provides a learned estimate of spatial interest. Our formulation examines how such interest can guide repeated screen choices by separating candidate-interest learning from the cost of changing views.

2.2 Sequential camera control and view selection

Learning from human camera operation predates the present setting. Chen and Carr learned sports-camera behavior from calibrated broadcast footage and player positions [9]. Chen *et al.* subsequently used recurrent decision trees to plan camera trajectories from noisy player detections, incorporating previous predictions and balancing framing accuracy against smoothness [10]. These studies establish learning from demonstration and temporally dependent camera decisions as prior approaches. Our setting concerns discrete selection among map viewports with a fixed charge for changing coordinates, rather than continuous pan-angle planning.

Automatic cinematography also combines scene relevance with continuity. AutoCam scores spatiotemporal glimpses in panoramic video and connects them by dynamic programming in the Pano2Vid task [11]. Deep 360 Pilot uses recurrent features and policy-gradient training to select viewing directions in sports videos, considering relevant objects and trajectory smoothness [12]. Reinforcement-learning camera control and learning from camera demonstrations provide further ways to encode the objectives of a viewing sequence [6, 7].

In some settings, camera choice also changes what the system can observe. Active-camera navigation coordinates viewing with robot exploration, while MVSelect chooses

which images to process for multiview recognition and detection [13, 14]. Here, the complete structured game map is available regardless of the displayed viewport. The camera changes what the audience sees and where the next move starts, rather than which game-state information the controller receives.

2.3 Learning scene interest from recorded games

Offline reinforcement learning uses fixed experience without further interaction with the environment [15]. In its usual setting, rewards are observed for the actions taken in the recorded data. Human viewport annotations provide a different form of feedback: their overlap with any candidate screen can be computed at a recorded training state. A model can therefore learn candidate interest by supervised regression. The distinction between recorded-action feedback and broader action feedback is also relevant to offline contextual decision learning [16].

Learning interest and deciding whether to change views are separate parts of the problem. Work on online learning with switching costs and smoothed online optimization likewise distinguishes the quality of a choice from the cost of changing it [17, 18]. In the present setting, a supervised interest model supplies the current score, and camera memory supplies the switching cost. This also permits a direct comparison with a controller that adds a learned estimate of future viewing value, while keeping the game data and candidate views fixed.

2.4 Multiple observers and multiple valid views

Human attention data offer an alternative to manually specifying which events matter. Atari-HEAD records player gaze and actions, providing one example of attention supervision in games [19]. Spectator camera trajectories describe a different behavior: which part of the game an observer chooses to show to an audience. For StarCraft, prior work evaluates a predicted viewport by its overlap with regions watched by multiple people [3].

Variation between observers can represent plausible alternatives, not only annotation error. This is consistent with the broader treatment of human label variation discussed by Plank [20]. We retain those alternatives by assigning interest to candidate screens from the human views within each match. The resulting score measures spatial agreement with recorded observation. The exact aggregation and its use in training and evaluation are defined in Methods.

3 Methods

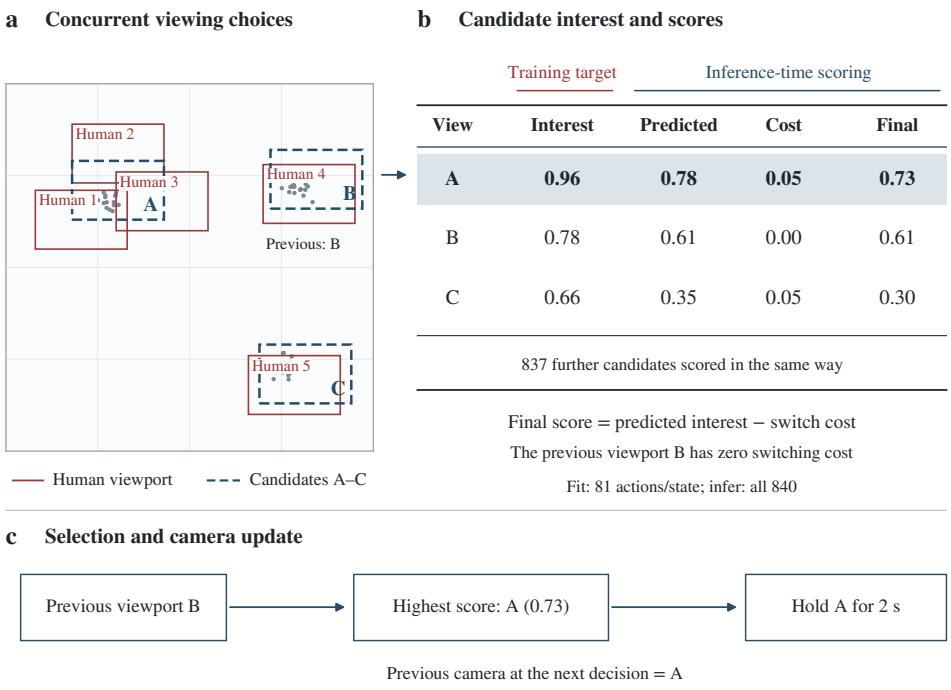


Fig. 1 Overview of one sequential viewport decision. (a) Five human viewports (red solid outlines) and three example candidates A–C (blue dashed outlines) share one game map. (b) The candidate table separates human-derived training interest targets from predicted interest scores, switching costs and final inference scores. Interest targets can be defined for all 840 candidates; each training snapshot fits 81 sampled actions, whereas inference scores all 840. Only three rows are illustrated. (c) With B as the previous camera, the highest-scoring viewport A is selected and retained until the next decision. Its coordinate becomes the next previous-camera state. Coordinates and values are schematic, not measurements from an evaluated match.

3.1 Overview

The proposed method decides whether to keep showing the current screen or move to another scene. It first compares the scenes that could be shown, then asks whether a new scene offers enough advantage to justify a camera change. We define a *candidate viewport* by placing a screen of fixed size at a location on the map; Fig. 1a shows three of the 840 possible candidates. We call the method the multi-observer viewport selector (MO-VS).

The basis for comparing scenes is learned from the screens people actually watched. For each candidate, overlap with the regions viewed by five observers provides a *scene-interest score* (Fig. 1b). When observers watch different scenes, each candidate can receive credit for its overlap instead of treating just one screen as correct. A regression

model learns to predict this score from the game state. Once trained, it can score candidate scenes without access to human viewports.

Screen selection then combines predicted interest with the current camera position. To avoid moving whenever another view has only a small score advantage, a fixed *switching cost* is subtracted from candidates that change the screen. The candidate with the highest resulting score is selected, with the current view included as an option to stay. In Fig. 1c, for example, the camera moves from B to A because A's advantage is large enough to cover the cost. The selected screen is held for 2 s, after which the process repeats from that position. We use *control reward* to mean scene interest minus switching cost.

This current-interest selection rule is the primary method, MO-VS-immediate. A separate comparison adds a prediction of the value of subsequent scenes to correct the current choice. We also evaluate short-term game history as an additional input. Camera choices leave game progression unchanged but set the starting position for the next camera decision, allowing these methods to be compared within the same sequential formulation [21].

3.2 Data and preprocessing

We reanalyzed game states and human viewport trajectories from the existing StarCraft observer dataset [3]. The analysis uses 17 matches with linked game states and five human-view annotations, totaling 10,507 decision points and 500,997 evaluation frames. Four of the 21 candidate matches lacked the required viewport annotations and were excluded. Inclusion criteria are detailed in Supplementary Methods, "Analysis-set eligibility"; training and evaluation were separated by match (Section 3.6).

Ethical approval and informed consent.

The human observer viewport trajectories analyzed in this study were collected in March 2022 under a research protocol approved by the Institutional Review Board of Gwangju Institute of Science and Technology (GIST), Republic of Korea (Approval No. 20210806-HR-62-01-02; approved on 17 August 2021). The approved protocol covered the collection of participant behavioral data and game-related logs for human-centered game artificial intelligence research. All methods involving human participants were carried out in accordance with the relevant guidelines and regulations and the approved research protocol. Informed consent was obtained from all participants prior to data collection.

The state $G_t \in \mathbb{R}^{9\times128\times128}$ contains four channels for each player—workers, ground units, air units and buildings—plus the source's neutral-vision channel. The first eight channels count objects at their tile positions, allowing multiple objects per tile; the last stores the supplied tile-wise vision state. These are channel sums rather than binary occupancy indicators. Arrays were stored every 8 frames, and every sixth array defines a decision, giving a 48-frame interval. Time and rate statistics use a nominal conversion of 24 frames per second. Training interest targets describe the human views at that decision snapshot, not the mean over the subsequent holding interval.

Training uses the source human-view masks, while evaluation transforms camera coordinates according to the original evaluation procedure. Corresponding locations

differ by at most one tile per axis. Mapping a human location to the nearest candidate grid point is used only to sample training candidates; it does not move the human view itself. Coordinate transformations are detailed in Supplementary Methods, "Time and coordinate conventions".

The 11 available replay headers identify Fighting Spirit. Evaluation holds out matches rather than maps.

3.3 MDP and state-action representation

The controller is represented as a finite-horizon MDP approximation $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ [5]. The horizon ends with the match, while discounting emphasizes near-term scenes within that horizon: $\gamma = 0.5$ halves the weight of each successive 2-s step. Its state distinguishes what is happening on the map from where the camera was last placed:

$$s_t = (G_t, H_t, c_{t-1}). \tag{1}$$

In this state, G_t is the current game-state map and c_{t-1} is the policy's previous camera. MO-VS-immediate uses (G_t, c_{t-1}). The history extension adds $H_t = (G_{t-2\mathrm{s}}, G_{t-4\mathrm{s}})$, the maps observed 2 and 4 s earlier. Every controller starts at viewport top-left coordinate $(56, 60)$ tiles and subsequently updates camera memory from its own selected coordinates. History padding is specified in Supplementary Methods, "State representation and interest regression".

Each action places a 20×12-tile viewport on the map. The horizontal and vertical top-left limits are $128 - 20 = 108$ and $128 - 12 = 116$ tiles; spacing candidate positions by 4 tiles gives

$$\mathcal{A} = \{0, 4, \ldots, 108\} \times \{0, 4, \ldots, 116\}, \qquad |\mathcal{A}| = 28 \times 30 = 840. \tag{2}$$

Choosing $a_t = c_{t-1}$ retains the current view; any other coordinate changes it. With $s = (G, H, c)$ and $s' = (G', H', c')$, the transition separates replay progression from camera feedback:

$$P(s' \mid s, a) = P_G(G', H' \mid G, H)\, \mathbb{I}[c' = a]. \tag{3}$$

The replay supplies the next game state, while $c' = c_t = a_t$ stores the selected action as the previous camera for the next decision. The game transition is independent of camera action. The compact state remains an approximation, however, because different game configurations can have the same channel summaries and subsequently evolve differently.

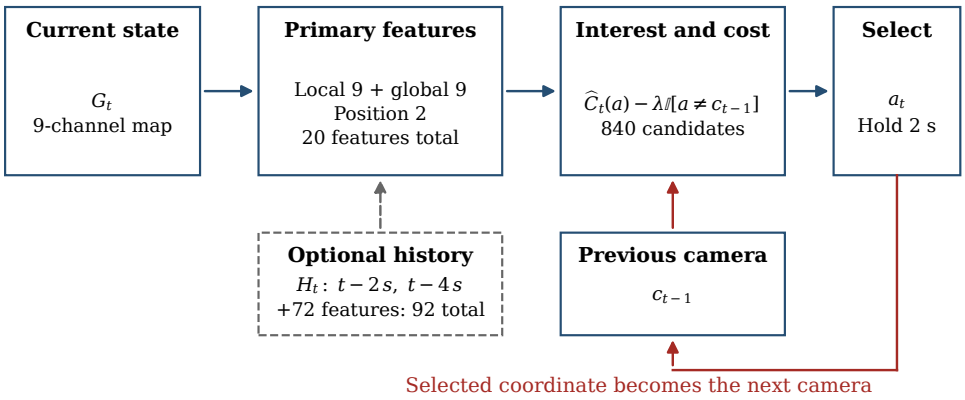


Fig. 2 State features and camera feedback in the immediate controller. Game maps supply local and map-wide candidate features to the interest regressor. The optional history branch adds earlier maps and their changes. Predicted interest is combined with a switching cost determined by the previous camera; the selected coordinate becomes the previous-camera state at the next decision. Camera choice does not change replay progression. The future-value extension is a separate comparison, not a required stage of this pipeline.

The interest model evaluates local objects inside each candidate together with map-wide activity (Fig. 2). Previous-camera memory enters the switching cost directly, while the optional history branch supplies earlier maps and their changes.

Each candidate is described by what it contains, activity across the map, and its location. At time τ, let $\boldsymbol{\ell}_\tau(a) \in \mathbb{R}^9$ and $\boldsymbol{g}_\tau \in \mathbb{R}^9$ summarize channel sums inside the candidate and across the full map. We reduce differences in magnitude by logarithmic normalization with fixed scales, then concatenate the summaries into 18 features, $\boldsymbol{m}_\tau(a) = [\boldsymbol{\ell}_\tau(a), \boldsymbol{g}_\tau]$. Adding the two normalized candidate coordinates $\boldsymbol{p}(a)$ gives the 20-dimensional current-state representation. Exact normalization formulas are in Supplementary Methods, "State representation and interest regression".

$$\boldsymbol{\phi}_t^{\text{static}}(a) = [\boldsymbol{m}_t(a), \boldsymbol{p}(a)] \in \mathbb{R}^{20}. \tag{4}$$

The 4-s-history representation adds the two earlier 18-dimensional summaries and their changes over successive 2-s intervals:

$$\begin{aligned} \boldsymbol{\phi}_t^{\text{temporal}}(a) = \big[&\boldsymbol{\phi}_t^{\text{static}}(a), \boldsymbol{m}_{t-2\text{s}}(a), \boldsymbol{m}_{t-4\text{s}}(a), \\ &\boldsymbol{m}_t(a) - \boldsymbol{m}_{t-2\text{s}}(a), \\ &\boldsymbol{m}_{t-2\text{s}}(a) - \boldsymbol{m}_{t-4\text{s}}(a)\big] \in \mathbb{R}^{92}. \end{aligned} \tag{5}$$

All historical blocks refer to the same absolute candidate location, and differences are computed after logarithmic normalization. The additional 72 features describe whether activity at that location is increasing or decreasing, while the two current coordinate features remain unchanged.

3.4 Multi-observer interest scores and control reward

The scene-interest score measures how well a candidate view lies within the regions watched by people. Let $\mathrm{VP}(a)$ be candidate a and $\mathrm{VH}_{t,i}$ the viewport of observer i at time t. We compare two ways to combine the five human views. The first uses their complete union:

$$C_t(a) = \frac{\left|\mathrm{VP}(a) \cap \left(\bigcup_{i=1}^{5} \mathrm{VH}_{t,i}\right)\right|}{|\mathrm{VP}(a)|}. \tag{6}$$

The primary, evaluation-aligned interest forms five unions by excluding one observer at a time and averages the resulting overlaps:

$$\widetilde{C}_t(a) = \frac{1}{5}\sum_{j=1}^{5} \frac{\left|\mathrm{VP}(a) \cap \left(\bigcup_{i\neq j} \mathrm{VH}_{t,i}\right)\right|}{|\mathrm{VP}(a)|}. \tag{7}$$

Training uses the archived human masks, whereas frame evaluation applies the same aggregation to the evaluator-coordinate views defined above. Both scores divide by the fixed candidate area. They measure how well one predicted screen is placed within human attention, rather than coverage of the entire human-view union or conventional intersection-over-union. The exclusions occur within each frame and are an aggregation rule, not observer-held-out validation.

The difference between the interest targets can be understood pixel by pixel. In $\widetilde{C}_t$, a candidate pixel receives 0 if nobody viewed it, 4/5 if exactly one person viewed it, and 1 if at least two people viewed it. Thus a uniquely observed region retains a positive weight, while shared regions receive full weight. A score of one means that every candidate pixel remains covered after any single observer is excluded.

The control reward subtracts a fixed cost when the camera changes:

$$r_t(s_t, a) = \widetilde{C}_t(a) - \lambda \mathbb{I}[a \neq c_{t-1}], \qquad \lambda = 0.05. \tag{8}$$

The main comparisons use $\lambda = 0.05$ as a fixed reference setting, rather than a validation-selected optimum. This is a margin of five percentage points in predicted overlap: another view must exceed the current view by this amount to justify moving. A separate sensitivity analysis evaluates the effect of changing the cost. The union condition retains the same cost and substitutes C_t for $\widetilde{C}_t$. Because $0.8C_t(a) \leq \widetilde{C}_t(a) \leq C_t(a) \leq 1$, this comparison changes both the weighting of human-viewed regions and the size of the switching cost relative to interest.

3.5 Learning scene interest and future-value correction

The candidate interest scores provide a continuous learning target: several viewports can have positive overlap with human attention at the same time. We therefore estimate each candidate's interest rather than assign a single coordinate as the correct class. This retains differences in overlap among plausible alternatives and allows one shared predictor to score all candidate locations from their state-action features.

At each training decision, the primary controller draws 81 of the 840 candidates uniformly without replacement. Under the same candidate budget, the target-aware comparison first includes high-interest candidates and candidates nearest the recorded human views, then fills the remaining positions randomly.

LightGBM is a gradient-boosted decision-tree method that combines successive trees to reduce prediction error [22]. It can represent nonlinear interactions between numerical features, such as local unit counts, map-wide activity and candidate coordinates. We selected this model because the inputs are compact feature vectors rather than raw images, and the same regression architecture can be retained across the interest-target, history and value comparisons. The loss function and regression settings were held fixed across conditions; details are provided in Supplementary Methods, "State representation and interest regression".

The fitted regressor thus predicts candidate current interest scores. Throughout this section, R_t denotes the chosen human-derived target, either C_t or $\widetilde{C}_t$, and $\widehat{C}_t$ is its learned prediction. At selection, subtracting the switching cost makes a new viewport preferable only when its predicted interest exceeds that of staying by enough to cover the cost. The primary policy is

$$a_t = \arg\max_{a \in \mathcal{A}} \left[\widehat{C}_t(a) - \lambda \mathbb{I}(a \neq c_{t-1}) \right]. \tag{9}$$

The current-state and 4-s-history models differ in their inputs but use the same screen-selection rule. Implementation details, including tie resolution, are provided in Supplementary Methods, "State representation and interest regression".

The optional value extension asks a different question: could a viewport with slightly lower current interest make subsequent viewing choices more valuable after switching costs? Because camera actions do not change the recorded game sequence and interest targets are available for every candidate, this question can be evaluated by working backward through a training match. We form D_t^{target} by adding the current interest to the best discounted continuation:

$$D_t^{\text{target}}(a) = R_t(a) + \gamma \max_{b \in \mathcal{A}} \left[D_{t+1}^{\text{target}}(b) - \lambda \mathbb{I}(b \neq a) \right]. \tag{10}$$

The target is computed backward to the end of the training match with zero terminal continuation. It is a post-decision value: moving to a is charged at selection, while equation (10) accounts for subsequent switches. Subtracting R_t avoids counting current interest twice; removing the across-action mean then focuses the second regressor on continuation differences that can change the candidate ordering:

$$F_t^{\text{target}}(a) = D_t^{\text{target}}(a) - R_t(a), \qquad F_t^{\text{rel}}(a) = F_t^{\text{target}}(a) - \frac{1}{|\mathcal{A}|} \sum_{u \in \mathcal{A}} F_t^{\text{target}}(u). \tag{11}$$

The backward targets describe optimal continuation along each recorded training sequence and provide hindsight supervision for a second regressor. At test time, this regressor uses only current-state features or their history extension. Centering uses all 840 actions during training-target construction and again for its predictions at

inference. We preserve the immediate estimate as the base score and add the predicted relative continuation as a bounded correction. Clipping at δ limits its magnitude, while β controls its weight. Both terms remain on the current-interest scale:

$$S_t^F(a) = \widehat{C}_t(a) + \beta \operatorname{clip}\left(\widehat{F}_t^{\mathrm{rel}}(a), -\delta, \delta\right), \quad a_t = \arg\max_a \left[S_t^F(a) - \lambda \mathbb{I}(a \neq c_{t-1})\right]. \tag{12}$$

At $\beta = 0$, this rule reduces to immediate selection. The future-value comparison fixes $\gamma = 0.5$, halving the weight every 2-s step. The correction's weight and magnitude limit were selected on validation matches using interest minus switching cost. Comparison conditions are described in the Results, and the parameter search is detailed in Supplementary Methods, "Fixed-budget sampling and value controls".

The constant cost of switching also limits how strongly the exact continuation target can distinguish two current actions. If $M_{t+1} = \max_b D_{t+1}^{\mathrm{target}}(b)$, then equation (10) implies

$$\begin{aligned} F_t^{\mathrm{target}}(a) &= \gamma \max\{D_{t+1}^{\mathrm{target}}(a), M_{t+1} - \lambda\}, \\ \max_a F_t^{\mathrm{target}}(a) - \min_a F_t^{\mathrm{target}}(a) &\leq \gamma\lambda. \end{aligned} \tag{13}$$

The first maximum lies between $M_{t+1} - \lambda$ and M_{t+1}, giving the bound on future-only target variation. Current interest scores can vary more widely, and learned predictions also contain regression error. Equation (12) independently limits the magnitude of the applied correction to $\beta\delta$.

At $\lambda = 0.05$ and $\gamma = 0.5$, exact continuation values can differ across candidates by at most 0.025. With the same candidate set available and a constant switching charge, a best-versus-runner-up gap above this bound in exact immediate control scores prevents exact continuation from changing the maximizing action at the same state and previous camera. This property concerns exact targets; it does not guarantee learned-policy performance or rule out benefits of lookahead under other conditions. With an incomplete observed state, the path-specific hindsight targets also need not be the Bellman-optimal values of a general stochastic MDP. Score-gap diagnostics are provided in Supplementary Methods, "Exact continuation and action-score gaps".

3.6 Evaluation

To assess screen selection on matches not used for training, we divided the 17 matches into five groups for match-level cross-validation. Each group served as the test set in turn, while one of the remaining matches was reserved for selecting model settings. Models were then refitted on all non-test matches and evaluated on the test group. Each match was tested exactly once, and MO-VS and Mask R-CNN shared the same split.

All methods' screen choices were evaluated on the same 2-s schedule, holding each selected view until the next decision. We calculated overlap with human-viewed regions at every frame using equation (7). Scores were averaged first within each match and then equally across the 17 matches, so longer matches did not dominate the result. MO-VS and Mask R-CNN were each trained with 3 random seeds, and their scores were averaged within matches.

Performance differences were calculated by comparing methods on the same matches. We obtained 95% confidence intervals by repeatedly resampling matches using the bootstrap procedure. Camera behavior was measured by screen changes per minute, time spent at the same view, and distance traveled when switching. Detailed procedures are provided in Supplementary Methods, "Match split and fitting order" and "Paired contrasts, multiplicity and metric interpretation".

We also recomputed the fixed-score detector contrast after omitting each match, outer test group or detector training run in turn, and examined all selector–detector run pairings (Supplementary Methods, "Sensitivity to matches and training runs").

4 Results and analysis

We first compare automatic observers, then examine the contributions of the learning components and the effect of switching cost on camera behavior.

4.1 Observer comparison

Table 1 Human-view overlap of automatic observers on the same matches and 2-s evaluation schedule. Both learned methods average 3 seeds within each match, followed by equal weighting of matches. Difference is MO-VS-immediate minus the listed method; intervals are paired 95% match-bootstrap confidence intervals. Differences are computed from unrounded match means.

Method	Overlap	Difference	95% interval
MO-VS-immediate	0.6257	Reference	–
Mask R-CNN	0.5753	+0.0504	[0.0267, 0.0791]
SSCAIT rule observer	0.4943	+0.1314	[0.1162, 0.1468]
AIIDE rule observer	0.4923	+0.1334	[0.1175, 0.1500]

We first asked whether MO-VS-immediate achieves greater agreement with human-viewed regions than detection-based or event-driven observers (Table 1). Mask R-CNN learns from the five individual viewport masks and selects its highest-confidence detection. SSCAIT uses event priorities and dwell rules, whereas AIIDE focuses on eligible combat and construction events. We re-evaluated archived SSCAIT and AIIDE camera trajectories from the source data at the common decision cadence. The learned systems share match splits and validation assignments; detector training settings are given in Supplementary Methods, "Mask R-CNN comparator".

MO-VS-immediate had higher overlap than the retrained detector and both rule observers, with positive paired intervals for all three comparisons. Supplementary Methods, "Descriptive score references", places the overlap scale in context using uniform-random and fixed-center viewports, alongside human agreement with the other observers.

We then asked whether the mean advantage depended on particular matches, training runs or the primary operating specification. It persisted across match,

outer-test-group and detector-training-run omission checks (Supplementary Methods, "Sensitivity to matches and training runs") and in two separate one-factor variants, zero switching cost and union-interest training (Supplementary Methods, "Switching-cost sensitivity and control diagnostics").

In a post-hoc diagnostic using one original detector training run, spatial retention approximately matched the switching rate of MO-VS-immediate while producing lower overlap. This comparison evaluates a simple way to reduce detector motion; its results are conditional on the motion-selected retention setting (Supplementary Methods, "Retention matched to detector switching rate").

4.2 Effects of the learning components

Table 2 Mean human-view overlap for interest aggregation, training-candidate sampling and state-history conditions. Aligned interest with uniform sampling is the reference. Entries average 3 seeds within each match and then weight matches equally. All conditions fit 81 candidates per training state and use switching cost 0.05.

Condition	Current state (20D)	4-s history (92D)
Union interest / uniform	0.6238	0.6234
Aligned interest / target-aware	0.6057	0.6070
Aligned interest / uniform (reference)	0.6257	0.6225

Table 2 asks whether selection benefits more from concentrating training on rare, high-interest scenes or from representing the full candidate set more evenly. Rows compare candidate sampling and interest aggregation; columns compare current-state and short-history inputs under the same camera-selection rule.

Aligned interest aggregates human views in the same way as the evaluation metric, whereas union interest measures overlap with all five viewers' regions combined. Uniform sampling draws candidates without regard to their interest. Target-aware sampling favors high-interest candidates and locations associated with human views. The latter retains rare interesting scenes, while the former represents the full set of alternatives encountered at inference more evenly.

To assess future-value learning, we added a predictive correction to the immediate controller. A capacity control instead adds a second current-interest regressor, separating the benefit of a future target from that of fitting another model. The two extensions use the same setting-selection budget. Detailed settings and absolute scores are in Supplementary Methods, "Fixed-budget sampling and value controls".

Table 3 Paired differences in human-view overlap for the component comparisons. Each difference is the first named condition minus the second, with a 95% match-bootstrap interval. Aligned interest and uniform candidate sampling are retained except where the contrast changes them. History minus current compares these reference models; 20D and 92D denote current-state and 4-s-history representations. Multiplicity sensitivity is reported in Supplementary Methods, "Paired contrasts, multiplicity and metric interpretation". Differences are computed from unrounded match means.

Comparison	Difference	95% interval
History minus current	−0.0032	[-0.0077, 0.0014]
20D: uniform minus target-aware	+0.0200	[0.0088, 0.0330]
20D: aligned minus union	+0.0019	[-0.0008, 0.0044]
20D: future minus immediate	+0.0004	[-0.0001, 0.0009]
20D: future minus capacity	+0.0015	[-0.0002, 0.0032]
92D: uniform minus target-aware	+0.0156	[0.0078, 0.0233]
92D: aligned minus union	−0.0008	[-0.0029, 0.0013]
92D: future minus immediate	0.0000	[-0.0003, 0.0003]
92D: future minus capacity	−0.0009	[-0.0030, 0.0010]

The most consistent improvement in Table 3 came from uniform candidate sampling. It increased overlap for both representations, and this direction remained after accounting for the multiple comparisons. Short history and the change in interest aggregation did not provide clear additional improvements. For this feature-based model, choosing which candidates enter training was therefore an important design decision.

Future correction changed overlap much less than sampling and did not establish an additional benefit over either immediate selection or the capacity control (Supplementary Fig. S1). A diagnostic using exact interest scores and the primary controller's previous camera found that 62.29% of evaluated decisions had an immediate-score gap above the bound in equation (13). Adding exact continuation cannot change the immediate maximizing action in those states. This diagnostic concerns exact scores at a common reference camera, not prediction errors in the learned models (Supplementary Methods, "Exact continuation and action-score gaps").

Candidate-level diagnostics clarify the sampling result. Uniform sampling reduced overprediction of background interest and improved the score of the selected viewport. However, prediction error increased when only high-interest candidates were considered. The benefit was thus in selecting among alternatives, rather than predicting every type of scene more accurately. All-action training and a diagnostic using a shared previous camera are reported in Supplementary Methods, "Fixed-discount repetitions and sampling diagnostics".

4.3 Switching cost and camera behavior

Table 4 Switching-cost sweep for the primary MO-VS-immediate controller, with its fitted interest models held fixed. The controller uses current-state features, aligned interest and uniform training-candidate sampling. Dwell is the mean of trajectory-level median times at one viewport; jump is the mean distance conditional on switching. The primary comparison uses cost 0.05.

Cost	Overlap	Changes/min	Dwell (s)	Jump (tiles)
0.00	0.6304	16.15	2.00	19.16
0.01	0.6304	13.58	2.16	20.64
0.02	0.6294	12.16	2.41	21.23
0.05	0.6257	9.63	3.04	22.43
0.10	0.6150	7.38	4.16	23.86
0.20	0.5887	4.81	5.73	25.96

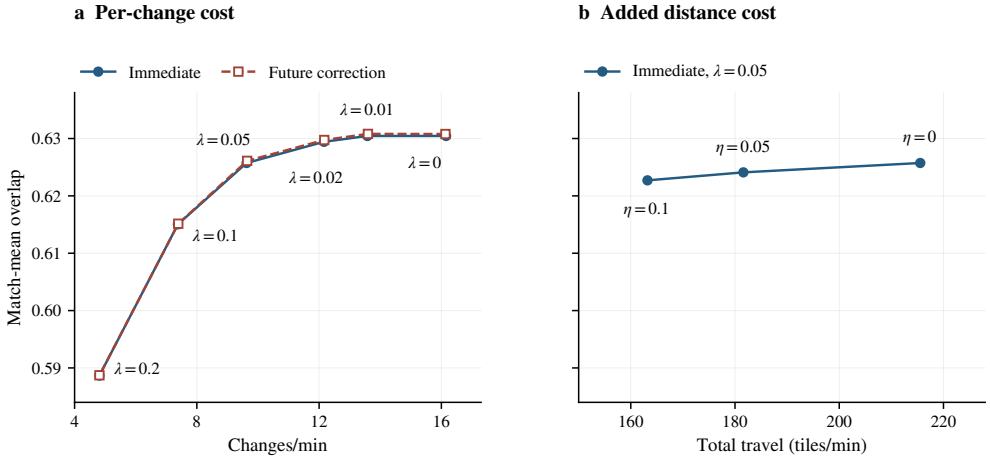


Fig. 3 Overlap and camera motion under fixed fitted models. (a) MO-VS-immediate and its future-correction extension as the per-change cost varies; Changes/min counts viewport coordinate changes per minute. **(b)** The immediate controller with an additional distance penalty, retaining the per-change cost at 0.05; total travel is the summed movement distance per minute. Panels share the overlap scale but use different horizontal motion measures. Labels give the per-change cost in (a) and distance-penalty weight in (b); lines connect the evaluated settings. Scores average seeds within matches and weight matches equally. No models are refitted during either sweep.

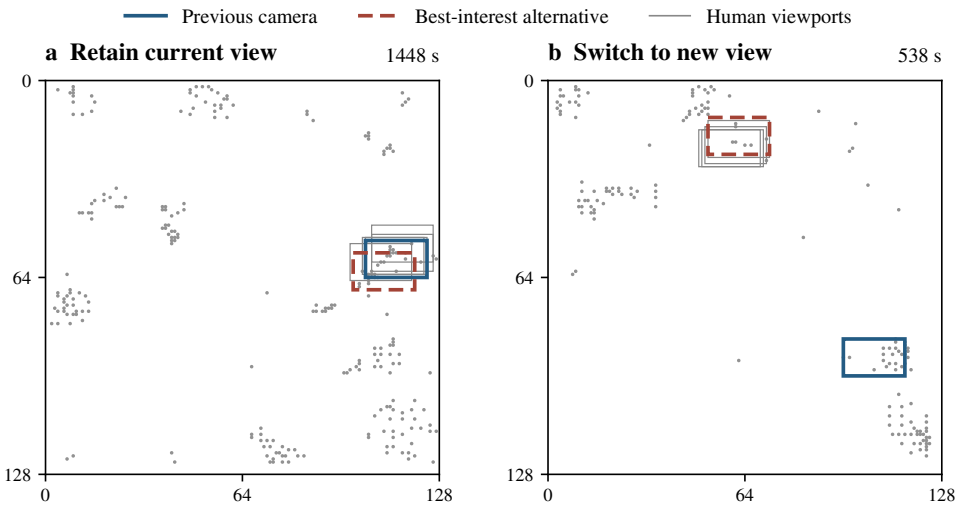
The fixed-model sweep in Table 4 and Fig. 3a separates the effect of the per-change cost from interest prediction. Larger costs retained each view longer and reduced camera changes. At the primary setting of 0.05, switching fell by about 40% relative to zero cost, with a small reduction in overlap. A smaller cost of 0.01 reduced changes

while nearly preserving the zero-cost overlap, illustrating a different operating trade-off. Higher costs reduced switching further but also reduced agreement with human-viewed regions.

The immediate and future-informed controllers followed similar trends in Fig. 3a. As changes became less frequent, the remaining moves became longer. A constant cost suppresses switching irrespective of distance. Switching frequency alone therefore gives an incomplete account of camera behavior; dwell time and movement distance provide complementary information. Supplementary Fig. S2 places these curves alongside baseline observers and human-view references; the human points provide agreement context rather than a direct performance ranking.

To distinguish how often the camera moves from how far it travels, we added a distance penalty while holding the interest models and primary per-change cost fixed (Fig. 3b). Increasing the penalty reduced both mean jump length and total travel per minute, with a small observed loss of overlap (Supplementary Methods, "Distance-sensitive switching cost").

4.4 Illustrative stay and switch decisions



Candidate	Interest	Cost	Score
Previous	0.6369	0.00	0.6369
Alternative	0.6538	0.05	0.6038

$\Delta\widehat{C} = 0.0168 < \lambda = 0.05$

Selected: (104, 52)

Candidate	Interest	Cost	Score
Previous	0.3047	0.00	0.3047
Alternative	0.4621	0.05	0.4121

$\Delta\widehat{C} = 0.1575 > \lambda = 0.05$

Selected: (52, 12)

Fig. 4 Retaining or changing the screen under the same rule. Blue solid rectangles show previous cameras and red dashed rectangles the highest-interest alternatives. Gray outlines are human viewports; points are occupied object tiles. The tables give predicted interest, switching cost and the resulting selection score. Interest differences $\Delta\widehat{C}$ are computed before rounding the displayed scores. Examples were chosen after evaluation to illustrate staying and switching; the selection procedure is documented in Supplementary Methods, "Illustrative camera decisions".

In Fig. 4a, another view has slightly higher predicted interest, but its gain is too small to offset the switching cost, so the current screen is retained. In panel (b), the alternative's interest gain exceeds that cost and the camera moves. The examples show how judging a scene's interest and deciding whether to move the camera are connected.

5 Discussion

The results identify candidate sampling and the switching cost as consequential design choices in this viewport-selection setting. Uniform sampling produced the most consistent component gain, while cost magnitude and distance dependence changed camera motion without retraining the interest model. The primary immediate controller

achieved higher overlap than Mask R-CNN and the two rule observers under the common evaluation protocol.

The sampling result suggests that exposing a model to more interesting scenes is not sufficient on its own. At inference, those scenes compete with many background alternatives. Uniform sampling represented these alternatives more evenly and reduced their overprediction. Selection improved even though prediction error increased among high-interest candidates, demonstrating why candidate-level accuracy and the quality of the chosen viewport should be assessed together.

The continuation bound helps interpret the small observed effect of future correction. Every candidate remains accessible at the next decision, and the current camera affects continuation through a constant switching cost. Under these conditions, equation (13) limits the differences in exact future scores across current choices. Learned prediction error is an additional, unresolved influence on the correction's effect. The analysis therefore connects the value of lookahead to the specific action and transition-cost structure.

Larger constant switching costs held screens longer but increased the distance of the remaining moves. The distance-dependent extension reduced jump length and total travel, with a small observed loss of overlap, showing that the structure of the transition penalty also matters. Retaining one fixed viewport is also different from tracking a moving event. Switching frequency, dwell and distance should therefore be interpreted together.

Detector performance varied across training runs, and the match-bootstrap intervals condition on the fitted models. Training used a fixed budget whose upper limit was selected in one fold of the original detector run.

The same matches supported repeated method development, making this a retrospective analysis. Independent matches, other maps and direct viewer evaluation are the next tests of generality and broadcast quality.

Together, learned scene interest and an explicit switching cost provide complementary controls over where to look and when to move. The experiments show the importance of examining training-candidate composition and the camera-selection rule separately, alongside any increase in value-model complexity.

Data availability. The accompanying Supplementary Data and Code package contains the match splits, prediction trajectories, match-level scores, paired-analysis inputs and diagnostic summaries. The Supplementary Information describes the corresponding original and follow-up experiments separately. Source game-state arrays and human trajectories from the original observer study [3] are not redistributed in this package. The source game-state arrays are not deposited in the linked public repository and are available on reasonable request from the corresponding author, Ho-Taek Joo (hotaek87@gm.gist.ac.kr). The supplied derived data permit verification of the summary comparisons; reconstructing the source overlap labels or retraining requires access to the original arrays and trajectories.

Human trajectories and replay files for part of the source dataset are available through the Starcraft_data repository. The repository provides human trajectories for 14 analyzed matches and replays for 11; preprocessed state arrays and the remaining

human trajectories are not included. Data coverage is detailed in Supplementary Methods, "Public source-data inventory".

Code availability. Code for training, evaluation, and generation of the tables and figures is provided with its settings in the supplementary code package. Reproduction procedures are described in Supplementary Methods, "Reproducibility scope", distinguishing reanalysis of supplied results from retraining that requires source data.

Funding. This work was supported by the InnoCORE program of the Ministry of Science and ICT (26-InnoCORE-01).

Author contributions. Ho-Taek Joo contributed to conceptualization, methodology, investigation, formal analysis, visualization, and writing of the original draft. Jongwon Kim contributed to conceptualization and supervision.

Competing interests. The authors declare no competing interests.

References

[1] Plankton555. *SSCAIT-ObserverModule*. GitHub repository, commit `b9b43c4` (accessed 19 August 2026). https://github.com/Plankton555/SSCAIT-ObserverModule

[2] Churchill, D. *Starcraft AI Tournament Manager: AutoObserver*. GitHub repository, commit `73aa5d0` (accessed 19 August 2026). https://github.com/davechurchill/StarcraftAITournamentManager

[3] Joo, H.-T., Lee, S.-H., Bae, C.-M. & Kim, K.-J. Learning to automatically spectate games for Esports using object detection mechanism. *Expert Syst. Appl.* **213**, 118979 (2023). https://doi.org/10.1016/j.eswa.2022.118979

[4] He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. In *Proc. ICCV*, 2961–2969 (2017). https://doi.org/10.1109/ICCV.2017.322

[5] Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction*, 2nd ed. (MIT Press, 2018).

[6] Passalis, N. & Tefas, A. Deep reinforcement learning for controlling frontal person close-up shooting. *Neurocomputing* **335**, 37–47 (2019). https://doi.org/10.1016/j.neucom.2019.01.046

[7] Hanocka, R., Assa, J., Cohen-Or, D. & Giryes, R. Learning camera control in dynamic scenes from limited demonstrations. *Comput. Graph. Forum* **41**, 427–437 (2022). https://doi.org/10.1111/cgf.14444

[8] Valve. Source SDK 2013: HLTV director. Source code (accessed 23 September 2026). https://github.com/ValveSoftware/source-sdk-2013/blob/master/src/game/server/hltvdirector.cpp

[9] Chen, J. & Carr, P. Mimicking human camera operators. In *2015 IEEE Winter Conference on Applications of Computer Vision*, 215–222 (2015). https://doi.org/10.1109/WACV.2015.36

[10] Chen, J., Le, H. M., Carr, P., Yue, Y. & Little, J. J. Learning online smooth predictors for realtime camera planning using recurrent decision trees. In *Proc. CVPR*, 4688–4696 (2016). https://doi.org/10.1109/CVPR.2016.507

[11] Su, Y.-C., Jayaraman, D. & Grauman, K. Pano2Vid: Automatic cinematography for watching 360° videos. In *Computer Vision–ACCV 2016*, Lecture Notes in Computer Science **10114**, 154–171 (2017). https://doi.org/10.1007/978-3-319-54190-7_10

[12] Hu, H.-N. *et al.* Deep 360 Pilot: Learning a deep agent for piloting through 360° sports videos. In *Proc. CVPR*, 3451–3460 (2017). https://doi.org/10.1109/CVPR.2017.153

[13] Chen, P. *et al.* Learning active camera for multi-object navigation. In *Adv. Neural Inf. Process. Syst.* **35** (2022). https://proceedings.neurips.cc/paper_files/paper/2022/hash/b84b9d1fe05c5e74d8f9466f063327a5-Abstract-Conference.html

[14] Hou, Y., Gould, S. & Zheng, L. Learning to select views for efficient multi-view understanding. In *Proc. CVPR*, 20135–20144 (2024). https://doi.org/10.1109/CVPR52733.2024.01903

[15] Levine, S., Kumar, A., Tucker, G. & Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint* arXiv:2005.01643 (2020).

[16] Brandfonbrener, D., Whitney, W. F., Ranganath, R. & Bruna, J. Offline contextual bandits with overparameterized models. In *Proc. 38th Int. Conf. Mach. Learn.*, 1049–1058 (2021).

[17] Dekel, O., Ding, J., Koren, T. & Peres, Y. Bandits with switching costs: $T^{2/3}$ regret. In *Proc. STOC*, 459–467 (2014). https://doi.org/10.1145/2591796.2591868

[18] Andrew, L., Barman, S., Ligett, K., Lin, M., Meyerson, A., Roytman, A. & Wierman, A. A tale of two metrics: Simultaneous bounds on competitiveness and regret. *Proc. Mach. Learn. Res.* **30**, 741–763 (2013). https://proceedings.mlr.press/v30/Andrew13.html

[19] Zhang, R. *et al.* Atari-HEAD: Atari human eye-tracking and demonstration dataset. *Proc. AAAI Conf. Artif. Intell.* **34**, 6811–6820 (2020). https://doi.org/10.1609/aaai.v34i04.6161

[20] Plank, B. The "Problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proc. EMNLP*, 10671–10682 (2022). https://doi.org/10.18653/v1/2022.emnlp-main.731

[21] Wan, J., Sinclair, S. R., Shah, D. & Wainwright, M. J. Exploiting exogenous structure for sample-efficient reinforcement learning. *arXiv preprint* arXiv:2409.14557 (2024).

[22] Ke, G. *et al.* LightGBM: A highly efficient gradient boosting decision tree. In *Adv. Neural Inf. Process. Syst.* **30**, 3146–3154 (2017).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- supplementarymethodsen20261005v39.pdf