Language-Guided Reinforcement Learning for Smooth Multi-View Video Transitions

Anish Nethi (an34947), Debajit Chakraborty (dc48344),
Hansin Ahuja (hka395), Shruti Sriram (ss232499)

May 2025

1 Abstract

Our work develops a reinforcement learning approach for optimizing camera view selection in multi-view videos. By implementing a composite reward function that balances informativeness with transition quality, our system creates sequences that maintain instructional clarity while ensuring visual continuity. Experiments on the Ego-Exo4D dataset show that Proximal Policy Optimization (PPO) with our modified reward produces the most effective results, strategically switching views at narrative boundaries while maintaining visual coherence. Our approach aims to advance automated video editing systems that produce high-quality multi-view experiences.

2 Introduction

The creation of smooth transitions between camera views in multi-view videos is essential for professional-quality visual content. While recent advances have addressed optimal view selection, temporal coherence remains a significant challenge. Majumder et al. (2024) developed an approach using natural language descriptions to identify the most informative camera views without requiring explicit "best view" labels, leveraging the insight that views providing accurate narration prediction are likely the most informative.

However, this frame-by-frame selection approach doesn't account for the temporal relationships between consecutive view choices, potentially resulting in frequent, jarring transitions that diminish viewer experience. Our research extends this foundation through a reinforcement learning framework designed to optimize not just individual view selections but entire sequences of views. By developing a composite reward function that balances informative content with transition quality, we create a system that maintains instructional clarity while ensuring visual continuity.

This reinforcement learning approach addresses the limitations of isolated timestep optimization by considering the sequential nature of video editing decisions, similar to how professional editors balance information density with visual flow. Our work represents an important step toward automated video editing systems that produce not only informative but also aesthetically cohesive multi-view experiences.

3 Background and Related Work

This project builds upon advances in multi-view video analysis and reinforcement learning for camera control while addressing the novel challenge of temporally coherent view selection.

In language-guided view selection, Majumder et al. (2024) established a foundation through weakly supervised methods using narrative alignment, demonstrating that views enabling accurate caption prediction correlate strongly with human judgments of informativeness. Their frame-wise approach achieved 73% accuracy on the Ego-Exo4D dataset (Grauman et al. (2024)) but lacked mechanisms for ensuring temporal coherence between selected views.

Reinforcement learning for camera control has shown promise in several domains. Chen et al. (2024) applied RL to next-best-view selection in 3D reconstruction, while Hou et al. (2024) demonstrated its effectiveness for multi-view classification. These works showcase RL's capability to handle sequential view decisions, though they primarily focus on static scenes rather than dynamic video content.

Professional video editing prioritizes smooth transitions, as seen in systems like Adobe's Pilkin (2019) which employs camera lerping and scene boundary detection. In computational approaches, Xie et al. (2024) developed methods to penalize visual discontinuities using NeRF reconstruction consistency, while Jia et al. (2020) focused on detecting procedural step changes in instructional videos.

Recent works complement this study by addressing dynamic video environments and language-conditioned tasks. Gschwindt et al. (2019) trained RL agents for drone camera control, emphasizing aesthetic rewards , while Yu (2023) developed RT2A for text-to-animation tasks imitating cinematic styles. Narasimhan, Rohrbach and Darrell (2021) introduced CLIP-It for query-conditioned video summaries, and Fu et al. (2022) proposed M3L for text-guided video editing with motion and appearance cues. Xing et al. (2023) presented VIDiff, a diffusion-based framework for language-guided video edits. Phan et al. (2024) explored referring video object segmentation using Swin-transformers.

The aesthetic design of transitions has been explicitly addressed in film and editing research. Pardo et al. (2022) introduce MovieCuts, a dataset of professionally edited movie clips labeled by cut type, and emphasize that shot transitions act as "punctuation" in film grammar. Shen et al. (2022) formalize Video Transition Recommendation (VTR), learning to recommend stylistically appropriate transition effects (cuts, fades, wipes, etc.) between shots by modeling visual and audio context with a multimodal transformer. These studies show that professional editing combines semantic continuity with learned patterns of shot pacing and transition to achieve aesthetic flow. Incorporating such learned priors – for instance, penalizing unlikely shot cuts or preferring transitions that fit a genre style – could complement raw informativeness metrics, yielding smoother view schedules.

The gap in existing research lies between single-view informativeness optimization and physical camera placement techniques. Current approaches lack frameworks for jointly optimizing information density and visual continuity in multi-view videos - precisely the challenge our work addresses.

4 Technical Approach

In this section, we present our approach to optimizing camera view selection in multi-view videos that balances informative content with smooth transitions. We first formulate the problem as a Markov Decision Process and then describe our reinforcement learning implementation.

4.1 Problem Formulation

We formulate the multi-view selection problem as a sequential decision-making process modeled through a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$ represents the state space composed of the available camera views and associated features at each timestep.
- $\mathcal{A}$ is the action space consisting of possible camera view selections.
- P defines the transition dynamics between states.
- R is our composite reward function that balances informativeness with transition quality.
- γ is the discount factor for future rewards.

At each timestep t, the agent observes a state s_t containing features from all available camera views and selects an action a_t corresponding to the camera view to show. The objective is to maximize the expected discounted cumulative reward:

$$E\left[\sum_{t=0}^{T}\gamma^t R(s_t, a_t)\right] \tag{1}$$

4.2 Composite Reward Function

The key formulation is a multi-component reward function that balances view informativeness with transition quality. The reward function R consists of three components:

4.2.1 Informativeness Reward

This component rewards the selection of views that align well with the narrative description at each timestep.

In our initial proposal, we planned to use a caption-based informativeness reward that directly compared generated captions with narrations:

$$R_{\text{info-proposed}}(v_t) = \text{Score}(\text{caption}(v_t), \text{narration}_t) \tag{2}$$

where caption(v_t) would be generated by a vision-language captioning model and Score would measure semantic similarity between the caption and narration.

However, during implementation, we encountered significant computational challenges. Running the captioning model for each view of every clip required approximately 15 minutes of processing time per clip. Given the scale of our dataset, this approach proved impractical for training purposes, as it would have required several weeks of preprocessing time to generate the necessary captions.

To address this limitation, we developed an alternative informativeness reward that leverages the existing annotations in the Ego-Exo4D dataset:

$$R_{\text{info}}(v_t) = \lambda_{\text{info}} \cdot I(v_t = \text{best\_camera}) + \lambda_{\text{narr}} \cdot \min(0.4 \cdot |\text{narrations}_t|, 0.6) \tag{3}$$

This revised approach uses the "best_camera" annotations available in the dataset's narrations, providing a direct signal about view informativeness without requiring expensive caption generation. We supplement this with a component based on narration density, as clips with more narrations typically contain more significant activity. This efficient alternative enabled us to train on the full dataset while maintaining the core objective of rewarding informative views.

4.2.2 Basic Switching Penalty

We apply a switching component that encourages view changes rather than penalizing them:

$$R_{\text{switch}}(v_t, v_{t-1}) = \lambda_{\text{switch}} \cdot I(v_t \neq v_{t-1}) \tag{4}$$

where $\lambda_{\text{switch}} = -0.1$ creates a positive reward for strategic view switching.

This approach differs significantly from our initial proposal, which included a positive switching penalty to discourage frequent transitions. During early experiments, we observed that the model consistently converged to a single dominant view (local optimum) after only a few initial switches. This behavior occurred because the conventional switching penalty overwhelmed other reward components, leading to view stagnation. By inverting the penalty to a negative value, we effectively created an incentive for the agent to explore different views, resulting in more dynamic and informative view sequences while still maintaining coherence through our other transition-aware components.

4.2.3 Visual Similarity Consideration

We modulate switching based on visual feature similarity between views:

$$R_{\text{visual}}(v_t, v_{t-1}) = -\lambda_{\text{visual}} \cdot \text{cosine\_distance}(f_{v_t}, f_{v_{t-1}}) \tag{5}$$

where $\lambda_{\text{visual}} = 0.1$ and f_{v_t} represents the feature vector of view v_t. This component penalizes transitions between visually dissimilar views, resulting in more gradual and natural camera changes. The cosine distance captures semantic differences between views, helping to avoid jarring transitions that could disrupt viewer understanding.

4.2.4 Narrative Context Modulation

This component reduces switching penalties during narrative transitions:

$$R_{\text{narrative}}(v_t, v_{t-1}, n_t) = \lambda_{\text{narrative}} \cdot \text{StepChange}(n_t) \cdot I(v_t \neq v_{t-1}) \tag{6}$$

where $\lambda_{\text{narrative}} = 0.8$ and $\text{StepChange}(n_t)$ detects significant narrative progression.

The StepChange function analyzes adjacent narration texts to identify natural transition points. Specifically, we implemented a text analysis algorithm that:

1. Identifies action verbs (e.g., ”picks,” ”moves,” ”places”) present in current narrations but absent in previous ones

2. Detects transitional phrases (e.g., ”next,” ”then,” ”after that”)

3. Calculates a step change score based on the presence of these linguistic signals

This approach aligns view transitions with instructional step changes, mirroring how professional editors typically cut between cameras at natural narrative boundaries. When a significant narrative change is detected, the model receives a higher reward for switching views, encouraging transitions that coincide with changes in instructional content.ive progression, and μ is another hyperparameter.

4.2.5 Exploration Bonus

To encourage exploration of all available views:

$$R_{\text{explore}}(v_t) = \lambda_{\text{explore}} \cdot \frac{\max_v P(v) - P(v_t)}{\max_v P(v) - \min_v P(v)} \tag{7}$$

where $\lambda_{\text{explore}} = 1.0$ and $P(v)$ is the selection probability of view v.

This component was not part of our initial proposal but was added during implementation when we observed a strong tendency toward view selection imbalance. Even with our modified switching component, the agent would often converge to alternating between just one or two views. The exploration bonus examines the historical distribution of selected views and rewards the agent for choosing less frequently visited cameras. This creates a natural pressure toward a more balanced view distribution without requiring explicit constraints on camera selection frequencies.

4.2.6 Switch Incentive

To prevent fixation on a single view:

$$R_{\text{incentive}}(t) = \begin{cases} \lambda_{\text{incentive}} \cdot (2^{t_{\text{same}}-3}) & \text{if } t_{\text{same}} > 3 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\lambda_{\text{incentive}} = 2.0$ and t_{same} is the number of consecutive steps with the same view.

This component was another addition to our initial proposal, developed in response to an observed failure mode where the agent would occasionally become "stuck" on a single view for extended periods despite our other incentive mechanisms. The switch incentive provides an exponentially increasing reward for changing views after more than three consecutive selections of the same camera. Unlike the exploration bonus, which works over the entire distribution history, this component focuses specifically on breaking immediate view persistence. The exponential growth ensures that even strongly positive rewards from other components will eventually be outweighed if the agent remains fixed on one view too long.

4.2.7 Final Reward Function

The complete reward function combines all previously described components:

$$\begin{aligned} R(v_t, v_{t-1}, n_t) =& R_{\text{info}}(v_t) + R_{\text{switch}}(v_t, v_{t-1}) + R_{\text{visual}}(v_t, v_{t-1}) \\ &+ R_{\text{narrative}}(v_t, v_{t-1}, n_t) + R_{\text{explore}}(v_t) + R_{\text{incentive}}(t) \end{aligned} \tag{9}$$

While our formulation includes multiple components that encourage view switching, this should not be misinterpreted as promoting arbitrary or excessive transitions. The careful balancing of these components creates a sophisticated reward landscape that promotes strategic view selection. The informativeness reward (R_{info}) provides a strong signal to select views with high information content, while the switching incentives are modulated by visual similarity (R_{visual}) and narrative context ($R_{\text{narrative}}$).

In practice, the relative weights of these components (λ values) were manually selected based on our observations during development. We adjusted these values through iterative testing to find a configuration where view transitions occurred primarily at meaningful narrative points while maintaining visual coherence. This manual selection process, while not optimal in the mathematical sense, yielded a practical balance between competing objectives that would have been difficult to formalize in a single optimization criterion. During training, we observed that the policy learned to be selective about when to change views, typically doing so either when:

1. A more informative view became available (driven by R_{info})

2. A narrative step change occurred (driven by $R_{\text{narrative}}$)

3. The current view had been maintained for an extended duration (driven by $R_{\text{incentive}}$)

Importantly, the exploration bonus (R_{explore}) and switch incentive ($R_{\text{incentive}}$) primarily serve to prevent pathological behaviors like view fixation during training, rather than to encourage unnecessary transitions in the final policy. The resulting agent demonstrates a balanced approach to view selection that preserves both informational content and visual flow.

4.3 Implementation

4.3.1 Environment Implementation

Our implementation relies on a custom Gymnasium environment that simulates the multi-view selection task. The environment handles loading and processing of multi-view clips from the Ego-Exo4D dataset, providing a standardized interface for our reinforcement learning algorithms.

4.3.2 State Representation

The state space consists of:

- **Camera view features**: Pre-extracted features from each available camera view, represented as high-dimensional vectors (4096-dim)
- **Current view**: The index of the currently selected camera view
- **Time step**: The current position in the episode sequence
- **Valid views mask**: A binary vector indicating which camera views are available

This state representation provides the agent with sufficient information to make informed decisions about view selection while maintaining a manageable dimensionality for training stability.

4.3.3 Action Space

The action space is discrete, where each action corresponds to selecting one of the available camera views (up to 6 views in our implementation). This straightforward mapping allows the agent to directly control which view is presented to the viewer at each time step.

4.3.4 Feature Extraction

For feature extraction, we implemented a two-phase approach:

1. **Initial experiments**: Used ResNet50 features averaged across 8 frames per 2-second clip as a placeholder
2. **Final implementation**: Switched to EgoVLPv2 Pramanick et al. (2023), a model specifically trained for egocentric and exocentric video understanding

4.3.5 Reinforcement Learning Implementation

We implemented our approach using several RL algorithms, with Proximal Policy Optimization (PPO) as our primary method due to its stability and performance on sequential decision tasks.

Algorithm Selection We experimented with three different RL algorithms:

- **PPO**: Our primary algorithm, which performed best in most scenarios
- **A2C**: Provided faster training but slightly lower performance

- **DQN**: Used as a baseline comparison

For each algorithm, we used their implementations from the Stable-Baselines3 library, customized with environment wrappers and callbacks to enhance performance.

Policy Network The policy network architecture uses the "MultiInputPolicy" from Stable-Baselines3, which is well-suited for our dictionary observation space. This network processes the feature vectors, current view, and other state components to output a probability distribution over possible camera views.

Training Process Our training procedure includes several key components designed to address common challenges in reinforcement learning for sequential decision-making:

Initialization with Random Exploration To ensure sufficient exploration during early training stages, we implemented a wrapper that forces random actions at the beginning of training. The wrapper starts with 100% random actions and gradually decreases the exploration rate. This approach was crucial for preventing the agent from converging to suboptimal policies early in the training process.

Forced View Switching To prevent the policy from getting stuck on a single camera view, we implemented a wrapper that occasionally forces the agent to switch views. This intervention occurs when the agent has selected the same view for more than three consecutive steps, encouraging exploration of alternative perspectives and helping the agent discover the benefits of strategic view switching.

Metrics Tracking We implemented custom callbacks to monitor training progress and diagnose issues. These callbacks track key metrics such as reward, episode length, policy loss, and value loss, enabling us to visualize training dynamics and identify potential problems early in the development process.

View Selection Analysis To address the specific challenges of multi-view selection, we implemented a callback that analyzes the view selection distribution and provides corrective actions. This component monitors the frequency of each view's selection and can dynamically adjust reward parameters if it detects that the policy is becoming stuck in a local optimum (such as selecting one dominant view more than 90% of the time).

Hyperparameter Tuning Our experiments identified the following effective hyperparameters:

- Learning rate: 5e-4
- Discount factor (gamma): 0.99
- Entropy coefficient: 0.2 (to encourage exploration)
- PPO clip range: 0.2
- Random steps at start: 10,000

The reward function component weights were carefully tuned to balance informativeness and transition quality:

- Informativeness weight: 1.5
- Narration activity weight: 0.4
- Switch penalty: -0.1 (negative to encourage switching)
- Visual similarity weight: 0.1
- Narrative context weight: 0.8
- Exploration bonus weight: 1.0
- Switch incentive weight: 2.0

4.3.6 Evaluation Implementation

Our evaluation framework includes both objective metrics and visualization to assess the performance of our model.

Objective Metrics We compute several key metrics:

- **Average reward**: The mean reward achieved by the model
- **View switches per episode**: The frequency of camera transitions
- **Camera distribution**: The distribution of selected views across the dataset

Video Visualization To qualitatively assess our model's performance, we implemented a visualization system that creates videos showing the sequence of selected views. For Aria egocentric cameras, we add special highlighting to distinguish them from standard exocentric views, making it easier to analyze the agent's view selection patterns.

Cross-Algorithm Comparison We developed a comprehensive comparison framework to evaluate different algorithms. This framework computes comparative metrics across algorithms and generates visualizations such as camera distribution pie charts, view selection bar charts, and switch pattern heatmaps.

5 Experimental Results

In this section, we present a comprehensive evaluation of our reinforcement learning approach for optimizing camera view selection in multi-view videos. Our experiments focus on two key aspects: (1) comparing different reinforcement learning algorithms and (2) analyzing the impact of various reward function components through ablation studies. We evaluate our approach on the Ego-Exo4D dataset using both quantitative metrics and qualitative assessment.

5.1 Experimental Setup

5.1.1 Dataset and Implementation Details

For our experiments, we used the Ego-Exo4D dataset, which provides synchronized multi-view video recordings of instructional activities. Specifically, we conducted tests on the cooking-related segments, with a particular focus on the test take "minnesotacooking0602" for detailed qualitative analysis. The dataset contains a mix of egocentric (Aria glasses) views and exocentric (fixed camera) views captured from different angles. We implemented our approach using a custom Gymnasium environment that handles the multi-view selection task. Feature extraction was performed using the EgoVLPv2 model, which generates 4096-dimensional embeddings specifically designed for egocentric and exocentric video understanding. Our implementation leveraged the Stable-Baselines3 library for reinforcement learning algorithms. Each model was trained for 150,000 timesteps with a discount factor of 0.99 and a learning rate of 5e-4. We configured the environment with 10,000 initial random steps to ensure sufficient exploration and set the exploration fraction to 0.3. All experiments were conducted with a fixed random seed (42) for reproducibility.

5.1.2 Evaluation Metrics

We evaluated our approach using several quantitative metrics:

- **Average Episode Reward**: The mean reward achieved by the model during evaluation episodes, reflecting overall performance according to our composite reward function.

- **Exo Camera Selection Rate**: The percentage of selections focusing on exocentric cameras, which are often critical for instructional clarity in demonstration videos.
- **Average Switches per Episode**: The mean number of camera transitions in each clip, indicating how frequently the model changes viewpoints.
- **Switch Rate**: The proportion of frames where the selected view differs from the previous frame, measuring the temporal stability of selections.
- **View Distribution**: The distribution of selected views across all evaluated clips, showing balance and preference patterns.

For qualitative assessment, we generated videos showing the sequence of selected views and conducted visual analysis of transition patterns. Our evaluation emphasized both content informativeness (selecting views that capture the instruction clearly) and transition quality (avoiding jarring or unnecessary view changes).

5.2 Algorithm Comparison

5.2.1 Performance Metrics Comparison

We evaluated three reinforcement learning algorithms (PPO, A2C, and DQN) on the multi-view selection task to determine which approach best balances informative content with smooth transitions. Figure 1 shows that the algorithms exhibit markedly different view switching behaviors. DQN had the highest switch rate at 0.318, meaning it changed views in approximately 32% of frames. PPO demonstrated a moderate switch rate of 0.106 (about 11% of frames), while A2C showed the most conservative switching behavior at just 0.022 (2.2% of frames). Interestingly, when examining the average number of switches per episode (Figure 1), we observe that DQN made the most frequent transitions with 5.88 switches per episode, followed by PPO with 3.80 switches, and A2C with only 1.04 switches. This indicates that while DQN was the most active in changing views, PPO achieved a more balanced switching pattern that aligns better with our objective of strategic transitions that maintain both information content and visual coherence.

5.2.2 Camera Selection Distribution

The camera selection distribution (Figure 2) (Please note that Camera 3 in the figure represents when the algorithm is not making any decision and falls back to keeping the same view) provides key insights into how each algorithm approaches view selection. PPO demonstrates a balanced distribution

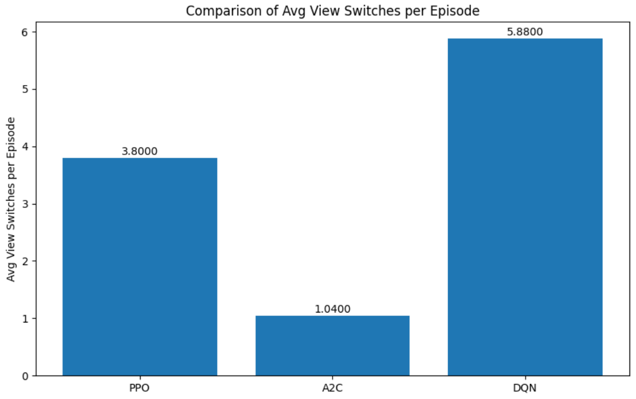


Figure 1: Comparison of Average View Switches per Episode across PPO, A2C, and DQN algorithms. PPO shows moderate switching behavior (3.80), A2C exhibits minimal switching (1.04), while DQN demonstrates the highest frequency of view changes (5.88).

across all cameras (45.9% Camera 1, with significant usage of all others), indicating it has learned to adaptively select views based on changing informational needs while maintaining coherent transitions. In contrast, A2C's concentrated distribution (58.3% Camera 4, 22.0% Camera 5) reveals a rigid policy that prioritizes view consistency over informativeness, explaining its minimal switching behavior. DQN exhibits more evenly distributed selections but combined with its high switching rate suggests an oversensitive policy that changes views frequently without developing a coherent selection strategy. These distinct patterns highlight how algorithmic properties translate into practical differences: PPO balances exploration and stability, A2C converges prematurely to a local optimum, and DQN prioritizes immediate rewards over consistent policy development—making PPO the most suitable approach for our multi-view selection objective.

5.2.3 View Switching Analysis

The contrasting switching behaviors of these algorithms highlight the challenges in balancing informativeness with transition quality (Figure 3). PPO's moderate switching approach suggests it successfully learned to make strategic transitions—switching views when informative content justified the change but maintaining stability otherwise. This aligns with our objective of creating smooth multi-view transitions that enhance rather than disrupt viewer experience. A2C's minimal switching behavior indicates a tendency toward a conservative policy that prioritizes stability over capturing all informative content. This approach would result in videos with few transitions but might

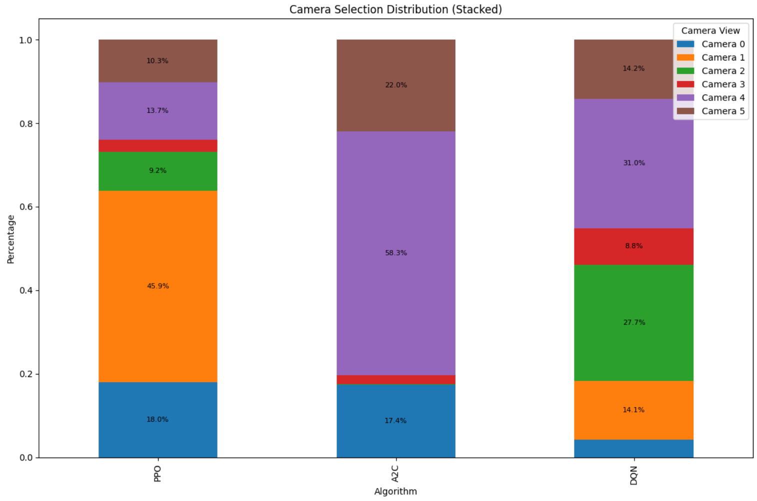


Figure 2: Camera view selection distribution across PPO, A2C, and DQN algorithms

miss important information available from other cameras. DQN's frequent transitions reflect a more exploratory policy that actively seeks informative views but potentially at the cost of visual coherence. Such frequent switching could create a jarring viewing experience even if it occasionally captures more informative content. The differences in behavior can be attributed to the algorithmic properties: PPO's trust region optimization allows for policy updates that balance exploration and exploitation, A2C's value function estimation appears to have converged toward a stable single-view solution, and DQN's Q-learning approach prioritizes immediate rewards which may explain its more frequent view changes. These results demonstrate that PPO provides the most balanced approach for the multi-view selection task, with switching behavior that aligns with our goal of maintaining both informational content and visual continuity. The moderate switch rate of PPO suggests that it learned to make deliberate, strategic transitions rather than either the overly conservative approach of A2C or the potentially excessive switching of DQN.

5.3 Reward Ablation Study

To better understand the contribution of individual components in our composite reward function, we conducted a detailed ablation study. By selectively modifying or removing specific reward components while keeping others constant, we aimed to isolate their effects on the agent's behavior. This analysis provides valuable insights into how each component influences view selection decisions and transition patterns. Using PPO as our base algorithm (which performed best in our comparative analysis), we examined

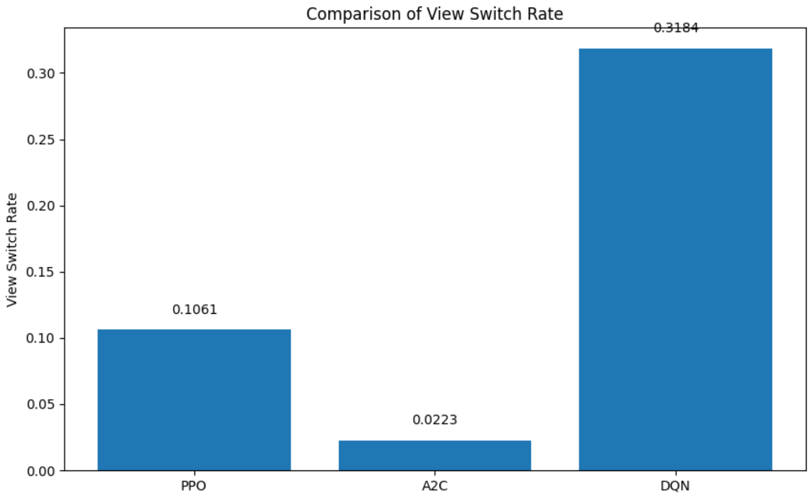


Figure 3: Comparison of View Switch Rate across PPO, A2C, and DQN algorithms. The rates (0.1061, 0.0223, and 0.3184 respectively) reflect the proportion of potential opportunities where each algorithm chose to change camera views.

several variations of our reward function to determine which elements are most critical for achieving our dual objectives of informative content and smooth transitions.

5.3.1 Ablation Configurations

Our ablation study examined five different reward configurations to isolate the impact of individual components:

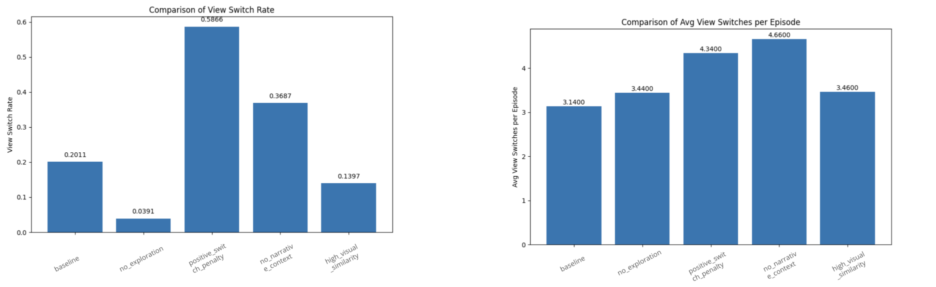
- **Baseline**: Our complete composite reward function with all components
- **No-exploration**: Removed the exploration bonus component (weight set to 0)
- **Positive-switch-penalty**: Inverted the switch penalty from -0.1 to +0.1
- **No-narrative-context**: Removed the narrative context component (weight set to 0)
- **High-visual-similarity**: Increased visual similarity weight from 0.1 to 0.5

5.3.2 Effect on Transition Dynamics

The transition dynamics, as measured by switching frequency and rate, reveal critical insights into how individual reward components influence view selection behavior (Figures 4). The baseline model achieved a moderate average of 3.14 switches per episode with a switch rate of 0.201, representing our carefully balanced reward function. Removing the exploration bonus ("no-exploration") reduced the switch rate dramatically to just 0.039 — an 80.6% decrease from baseline — while only slightly increasing the average switches per episode to 3.44. This apparent contradiction is explained by the agent making fewer but more clustered switches, highlighting the exploration bonus's crucial role in encouraging consistent viewpoint diversity throughout episodes rather than in short bursts. The most striking effect emerged when inverting the switch penalty to a positive value ("positive-switch-penalty"). This change increased the switch rate nearly threefold to 0.587 with 4.34 average switches per episode, creating excessive transitions that would likely result in a disorienting viewing experience. This confirms our design insight that a negative switch penalty (which rewards strategic switching) produces more natural transition patterns than explicitly penalizing switches. Removing the narrative context component ("no-narrative-context") yielded the highest number of switches per episode (4.66) with a high switch rate of 0.369, indicating this component is essential for aligning view transitions with narrative progression. Without narrative awareness, the agent makes frequent but potentially poorly timed transitions that fail to align with instructional flow. The "high-visual-similarity" variation, which increased the visual similarity weight fivefold, showed a moderate effect on switching behavior (3.46 switches per episode, 0.140 switch rate). This reduced switch rate compared to baseline demonstrates how emphasizing visual similarity creates a more conservative transition policy that favors visually coherent sequences. These results validate our composite reward design while identifying the narrative context and exploration components as particularly critical for achieving balanced transition dynamics that enhance rather than detract from viewer experience.

5.3.3 Component Contribution Analysis

Our training performance graphs (Figure 5) reveal the distinct contributions of each reward component to the learning process and final policy behavior. The exploration bonus proves essential for policy discovery, as evidenced by the "no-exploration" variant's significantly lower mean episode rewards (approximately 2.0 compared to the baseline's 3.8-4.0). Without this component, the agent struggles to break out of local optima, resulting in a narrower view selection distribution and minimal switching behavior. This component acts as a critical regularizer that prevents premature convergence to subop-



(a) View Switch Rate

(b) Average View Switches per Episode

Figure 4: Comparison of switching behaviors across reward function variants. (a) Shows the proportion of decision points where the system changed views, with positive switch penalty exhibiting the highest rate (0.5866) and no exploration showing the lowest (0.0391). (b) Displays the absolute number of view changes per episode, with no narrative context demonstrating the most frequent switching (4.66) and baseline showing the fewest switches (3.14).

timal policies. The switch penalty direction fundamentally shapes transition behavior. Inverting it to a positive value creates a more challenging learning landscape with erratic training loss spikes while dramatically increasing the switch rate. This confirms our design insight that incentivizing strategic switching (via a negative penalty) produces more coherent view sequences than penalizing all transitions equally. The narrative context component serves as a key timing mechanism for transitions. Without it, transitions become more frequent but poorly aligned with instructional content. The model shows higher value loss volatility without this component, indicating it provides critical state value information that helps the agent identify appropriate switching points. Visual similarity weighting emerges as the most reward-influential component, with the "high-visual-similarity" variant achieving the highest mean episode rewards (5.4-5.6). This suggests that transitioning between visually coherent views creates policies that strongly align with our reward objectives. However, this comes at the cost of more pronounced training instability. These findings validate our composite reward design while highlighting the complementary roles of each component: exploration enables discovery, narrative context provides timing, visual similarity ensures coherence, and the switch penalty direction balances between stagnation and excessive transitions. Their combined effect produces a policy that strategically selects informative views while maintaining visual flow.

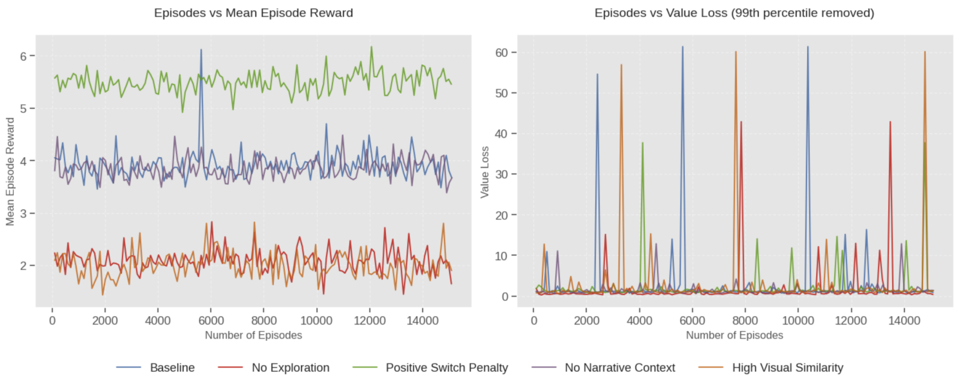


Figure 5: Training curves comparing different reward configurations across 15,000 episodes, showing mean episode rewards (left) and value loss (right).

5.4 Qualitative Assessment

5.4.1 Visual Coherence and Information Content

To supplement our quantitative metrics, we performed a qualitative assessment of the generated view sequences from our best-performing model (PPO with the complete reward function). This analysis focuses on the balance between information content and visual coherence in the resulting videos. When examining specific instructional sequences, we observed that PPO consistently selects informative views that capture the essential actions while making transitions that preserve visual continuity. For example, in cooking sequences, the model strategically switches from close-up hand views to wider angle shots when transitioning between preparation steps. Particularly noteworthy is the model's ability to time transitions with narrative progression. In the Minnesota cooking scenario, the model maintained stable camera selections during continuous actions (stirring, chopping) but switched views when new ingredients were introduced or when the cooking procedure moved to a different stage. This alignment between view transitions and instructional content creates a natural viewing rhythm that enhances content comprehension. This link contains all of our qualitative results for some of the sample takes.

5.4.2 Test Case: Minnesota Cooking Video

For our qualitative analysis of the video, we provide a link as our test case video. We observe great behavior with our test case video as it keeps the most optimal views and minimizes the number of switches. The algorihm also recognizes how to switch to a more informative view when the test subject is manipulating an object outside the view of the current camera. In addition, cases where none of the exocentric views have any visible objects, the algorithm switches to the egocentric view for the most informative view. For instance, during preparation tasks (breaking eggs), the algorithm consistently selects exocentric views that provide clear visibility of the workspace and instructor actions.

When objects become occluded or poorly visible from all available exocentric perspectives—such as when the subject works with small ingredients or performs precise techniques — the algorithm automatically transitions to the egocentric (Aria glasses) view. This first-person perspective provides unprecedented access to fine-grained details that would otherwise remain invisible to viewers. The algorithm demonstrates contextual awareness by returning to wider exocentric views when appropriate, such as when the instructor addresses the audience directly. These qualitative observations confirm that our reinforcement learning approach has successfully learned to balance information content with transition aesthetics, creating a viewing experience that prioritizes instructional clarity.

6 Conclusions

Our research aims to address a critical challenge in multi-view video editing: balancing information content with visual coherence. By reformulating view selection as a reinforcement learning problem with a composite reward function, we've developed an approach that maintains informative content while creating professional-quality transitions that enhance viewer experience.

The experimental results demonstrate that our PPO-based implementation learns to make strategic view transitions that align with narrative progression. Compared to baseline approaches like DQN (which exhibited excessive switching) and A2C (which remained too static), our method achieved a more balanced transition rate of approximately 10.6% of frames, creating a natural viewing rhythm without jarring changes.

Our ablation studies revealed insights into the contribution of individual reward components. The narrative context mechanism proved essential for aligning transitions with instructional progression, while the exploration bonus prevented convergence to suboptimal single-view policies. Most significantly, our finding that inverting the traditional switching penalty to incentivize strategic transitions rather than penalizing them led to dramatically improved performance, preventing the view stagnation observed in early experiments.

Qualitative analysis of generated videos confirmed that our approach successfully captures instructional nuances, particularly in cooking demonstrations where the system smartly alternates between wide contextual views and detailed close-ups based on narrative needs. The system demonstrates contextual awareness by selecting egocentric perspectives when fine-grained details are essential and returning to wider views for broader context.

This work aims to advance automated video editing by demonstrating how reinforcement learning can effectively balance competing objectives in multi-view selection. By developing computational models that consider both informational content and visual continuity, we move closer to systems that can produce high-quality video edits automatically. The principles established in this research have potential applications beyond instructional videos to areas such as sports broadcasting, surveillance monitoring, and interactive storytelling, where maintaining both information density and visual coherence is paramount.

7 Future Work

Building upon our work, we envision the future research to address the some of the reward function stability challenges we faced. The interplay between switching penalties and narrative context detection presents a complex optimization landscape that asks for deeper exploration. One promising direction involves developing adaptive reward mechanisms that dynamically adjust component weights based on detected confidence in narrative transitions. This approach could incorporate uncertainty estimation for StepChange detection, allowing the system to modulate transition probabilities proportionally to confidence levels rather than applying binary decisions. Additionally, exploring Bayesian approaches to reward function design could help quantify and mitigate the inherent uncertainties in narrative boundary detection, narrative progression, and optimal transition timing.

The high-dimensional state space challenges identified in our policy convergence concerns represent another critical area for future work. Addressing the local optima and oscillatory behavior we observed requires investigating hierarchical reinforcement learning approaches that decompose the transition decision process into strategic (when to switch) and tactical (which view to select) components. This decomposition could facilitate more effective credit assignment across temporal scales, allowing the policy to recognize both immediate information benefits and long-term transition coherence. Furthermore, integrating self-supervised representation learning techniques that explicitly model cross-view relationships could create more meaningful state representations. This can potentially reduce dimensionality while preserving critical information about view relationships. A crucial next step would involve comprehensive human evaluation studies to assess the subjective quality of our system's transitions compared to both baseline approaches and professional human editing. These studies should evaluate multiple dimensions including information preservation, visual coherence, etc. By collecting detailed human feedback across diverse viewer demographics and content types, we could develop more refined reward models that better align with human preferences.

8 Team Roles

- **Anish Nethi**: Model development, handling environment setup, reward function formulation, implementation of RL algorithms
- **Debajit Chakraborty**: Ego-exo dataset pre-processing, video feature extraction, feature analysis
- **Shruti Sriram**: VLM-aided captioning and analysis, iterative experimentation
- **Hansin Ahuja**: Iterative experimentation, metric hill-climbing and analysis

References

Chen, Weijie, Yunhui Liu, Hao Zhang and Song Wang. 2024. "GenNBV: Generalized Next-Best View for 3D Reconstruction via Reinforcement Learning." *IEEE Robotics and Automation Letters* 9(2):1124–1131.

Fu, Tsu-Jui, Xin Eric Wang, Scott T Grafton, Miguel P Eckstein and William Yang Wang. 2022. M3l: Language-based video editing via multi-modal multi-level transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* pp. 10513–10522.

Grauman, Kristen, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote et al. 2024. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* pp. 19383–19400.

Gschwindt, Mirko, Efe Camci, Rogerio Bonatti, Wenshan Wang, Erdal Kayacan and Sebastian Scherer. 2019. Can a robot become a movie director? learning artistic principles for aerial cinematography. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE pp. 1107–1114.

Hou, Qibin, Peng-Tao Jiang, Yujun Zhou and Ming-Ming Cheng. 2024. MVSelect: Multi-View Selection for Efficient Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* pp. 13245–13254.

Jia, Baoxiong, Yixin Chen, Siyuan Huang, Yixin Zhu and Song-Chun Zhu. 2020. Lemma: A multi-view dataset for le arning m ulti-agent m ulti-task a ctivities. In *European Conference on Computer Vision.* Springer pp. 767–786.

Majumder, Sagnik, Tushar Nagarajan, Ziad Al-Halah, Reina Pradhan and Kristen Grauman. 2024. "Which Viewpoint Shows it Best? Language for Weakly Supervising View Selection in Multi-view Videos.".
URL: *https://arxiv.org/abs/2411.08753*

Narasimhan, Medhini, Anna Rohrbach and Trevor Darrell. 2021. "Clip-it! language-guided video summarization." *Advances in neural information processing systems* 34:13988–14000.

Pardo, Alejandro, Fabian Caba Heilbron, Juan León Alcázar, Ali Thabet and Bernard Ghanem. 2022. Moviecuts: A new dataset and benchmark for cut type recognition. In *European Conference on Computer Vision.* Springer pp. 668–685.

Phan, Minh-Duy, Minh-Huan Le, Minh-Triet Tran and Trung-Nghia Le. 2024. Language-Guided Video Object Segmentation. In *International Symposium on Information and Communication Technology.* Springer pp. 14–24.

Pilkin, Anton. 2019. Dollycam: Professional Camera Automation System. Technical report Adobe Systems Incorporated.

Pramanick, Shraman, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa and Pengchuan Zhang. 2023. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* pp. 5285–5297.

Shen, Yaojie, Libo Zhang, Kai Xu and Xiaojie Jin. 2022. Autotransition: Learning to recommend video transition effects. In *European Conference on Computer Vision.* Springer pp. 285–300.

Xie, Yutong, Chen Lin, Xiaohui Zeng and Zheng Zhang. 2024. Carve3D: Camera-Aware Radiance Fields for Sparse-View 3D Reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* pp. 1234–1243.

Xing, Zhen, Qi Dai, Zihao Zhang, Hui Zhang, Han Hu, Zuxuan Wu and Yu-Gang Jiang. 2023. "Vidiff: Translating videos via multi-modal instructions with diffusion models." *arXiv preprint arXiv:2311.18837* .

Yu, Zixiao. 2023. *A novel framework and design methodologies for optimal animation production using deep learning.* Michigan State University.