Can ModelScopeT2V Understand Arrow of Time?

Paanugoban Kugathasan

Abstract—This paper examines the concept of Arrow of Time(AoT) and it's inclusion in video generation. Specifically, this study will focus on ModelScopeT2V, a multi-modal text-to-video model that was released in 2023 by the Alibaba Group. By analyzing this state of the art model, we aim to understand the current capabilities in video generation and creating temporally coherent content. Through training which entailed developing a classifier, we evaluated the model's capacity in generating realistic videos in both forward and reverse directions. While the model achieved success in producing realistic forward sequences, it faced shortcomings in generating reverse content. This was verified by our experiments involving machine and human evaluators. Notably, this may be an issue with recognition of reverse prompts and vocabulary. Further exploration, potentially incorporating elements like disciminators for AoT could be beneficial. This research aims to contribute to the ongoing discourse on reliability and authenticity of AI generated content in various sectors.

Index Terms—Arrow of Time(AoT), Video Classification, Text to Video Generation

1 Introduction

The concept of the Arrow of Time (AoT), first proposed by British astronomer Sir Arthur Stanley Eddington in 1928 in his work 'The Nature of the Physical World' [1], is a fundamental principle in the understanding of temporal dynamics. Eddington's AoT posits that the flow of time is intrinsically unidirectional and irreversible [2]. A vivid illustration of AoT can be seen in the asymmetry of events such as a glass shattering (Figure 1), where the reconstitution of the fragmented pieces into their original form is virtually impossible.

Our research focuses on the implications of AoT in the context of synthetic video generation, particularly through the lens of multi-modal text-to-video conversion. Since 2022, we have witnessed an explosive growth in text-to-image technologies which have revolutionized the field of AI-generated imagery [3]. This evolution has necessitated a deeper understanding of the realism of AI-generated content, as discussed in studies like 'Analysis of Appeal for Realistic AI-Generated Photos' [3]. Our study extends this inquiry to text-to-video models, exploring their potential impacts across various industries.

Videos are ubiquitous across numerous sectors. The advent of advanced text-to-video tools welcomes their integration into these domains, raising critical questions about the authenticity and reliability of such content in relation to AoT. In communication contexts, the correct sequencing of events, guided by AoT, is crucial for accurate message interpretation. Misrepresentation of AoT in instructional videos, for instance, could lead to significant misunderstandings. In the realm of social media, where concerns about misinformation are rampant, inaccuracies in temporal sequencing might exacerbate issues of trust and credibility. On the other hand, if these models struggle to achieve AoT prior to mass adoption, could it be used as a technique to classify real versus generated content. In the entertainment industry, the use of these tools under constraints of time and budget might result in content that savvy audiences could perceive as unauthentic. Another intriguing application is in gaming, where technologies like NVIDIA's Deep Learning Super Sampling (DLSS) rely on frame prediction for enhanced performance. The effectiveness of such technologies hinges on the realistic portrayal of temporal progression in AI-generated videos. While these technologies as far as we know aren't applied to videos yet, continued efforts to upscale video frames is ongoing.

This paper examines the ModelScopeT2V model by Alibaba Group, released in June 2023 with 1.7 billion parameters and available for testing on the Hugging Face platform [4]. While other leading AI research entities like Google and Meta have announced their own text-to-video models, their accessibility remains limited. Despite known limitations of ModelScopeT2V, including its inability to produce film-quality outputs and realistic representations of people and events [4], our analysis aims to assess its current capabilities. Understanding these limitations is vital for future enhancements and potential applications in the areas previously mentioned.

2 Related Work

In understanding and applying the Arrow of Time (AoT) to video analysis, we look at several pivotal studies. In 2014, "Seeing the Arrow of Time" explored this study by utilizing a dataset of YouTube videos, predominantly featuring physics-related content such as gravity, friction, and entropy, amongst others [6]. This study carefully selected 125 forward-played videos and 25 reverse-played videos, ensuring optimal conditions like good lighting and minimal camera movement or shake [6]. The researchers employed optical flow techniques to generate motion-patch descriptors, termed 'flow words' Through the application of

• *E-mail: paanugoban@cs.toronto.edu*

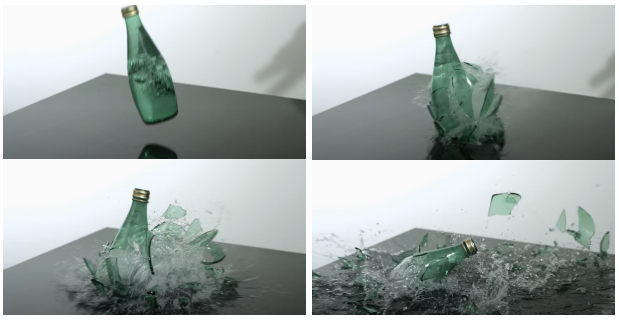

Fig. 1: Sequence of a glass shattering demonstrating the Arrow of Time [5]

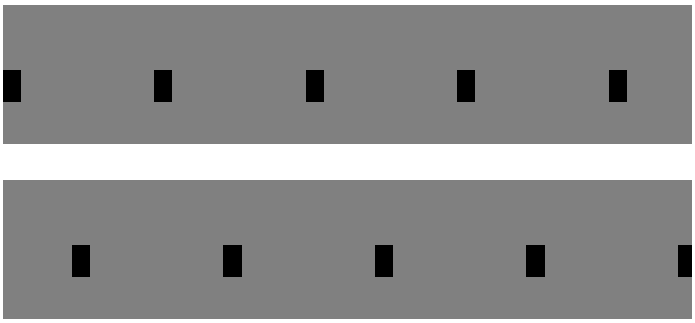

Fig. 2: Sequence of 10 basic frames

Support Vector Machines (SVM), they achieved remarkable classification accuracies of 90%, 77%, and 75% on different dataset partitions [6].

The 2018 paper "Learning and Using the Arrow of Time" delved deeper into the classification of videos, particularly aimed to extract insights about the visual world while circumventing artificial cues [7]. It expanded upon the 2014 study's methodology, which had a strong AoT emphasis but was limited to a smaller, specialized dataset [7]. The researchers implemented a convolutional neural network (convnet) architecture, utilizing optical flows and training the model end-to-end on three diverse datasets [7]. To enhance the authenticity of their experiment, they meticulously removed artificial elements like black bars and stabilized camera motion. Their model demonstrated an accuracy of 76% on the Flickr dataset, 72% on Kinetrics, and interestingly, human participants achieved an accuracy of 80% [7].

More recent advancements in this field came with the 2021 study "ArrowGAN: Learning to Generate Videos by Learning Arrow of Time" [8]. This research introduced the concept of an AoT discriminator, named Arrow-D, within the framework of Generative Adversarial Networks (GANs) for video generation. The objective was to enable the video GANs to better comprehend the AoT through the integration of the Arrow-D discriminator [8]. The discriminator utilized 3D convolutional networks to refine its performance. The results showed marked improvements in video generation quality across all datasets with the addition of the Arrow-D discriminator [8].

3 Proposed Method

To assess ModelScopeT2V's capacity for grasping the Arrow of Time (AoT), it is imperative that the model not only generate realistic reverse videos in response to reverse prompts but also that its outputs exhibit a discernible temporal direction [4]. Our classifier serves as a critical tool in our methodology, enabling us to evaluate the videos generated by ModelScopeT2V. By applying this classifier to the model's outputs, we can draw informed conclusions regarding ModelScopeT2V's proficiency in replicating both forward and reverse temporal sequences.

3.1 Training the classifier

Initially, we explored datasets such as 'Moment in Time' and 'Hollywood2' [9] [10], which encompassed a wide array of actions. The architectures considered included both 2D and 3D convolutional networks, as indicated in the related literature [7] [8]. However, these datasets presented challenges for our models, primarily due to the substantial variation in camera motion and perspectives, which proved difficult for the architectures to interpret accurately given the covariate shift. Since the text-to-video model would be generating a substantial volume of videos, extensive pre-processing would be computationally expensive for this analysis. Consequently, there was a need for a dataset that inherently offered stable camera motion and consistent perspectives.

To gauge the capability of our chosen architecture we conducted a basic validation test. This test involved creating a sequence, as depicted in Figure 2, featuring squares moving from left to right. Ultimately, we theorized that the model should at least be able to comprehend this basic movement. This approach allowed us to identify any foundational deficiencies in the model's learning process, which could then be addressed through the addition of layers or the optimization of hyperparameters, thereby enhancing the model's capacity to analyze more intricate motion patterns in real-world scenarios.

We found a new model capable of achieving high accuracy on the basic motion sequences, following training on a dataset comprising 1000 sets. The next step involved identifying an dataset that featured minimal camera motion. The decision to avoid eliminating artificial cues, stabilizing camera motion, and utilizing optical flows [7] was driven by our goal to ensure a high degree of consistency between the training data and the application environment, while also aiming to minimize computational expense.

After extensive research, we identified the UCF-101 dataset as an ideal match for our requirements [11]. This dataset is well-categorized, featuring 95 distinct classes that lend themselves to unique prompts, thereby facilitating a deep dive into various types of motions. Additionally, UCF-101 presented less variation in content and exhibited greater camera stabilization compared to other datasets. This strategic selection of UCF-101 played a pivotal role in enabling us to rigorously test and refine our model, ensuring its effectiveness and accuracy in the context of text-to-video model analysis.

Our initial step in training the classifier commenced with a specific category of videos, as illustrated in Figure 3. For each video, we extracted 10 evenly spaced frames throughout the video's duration. This method diverged from other literature, which often determined the frames per second

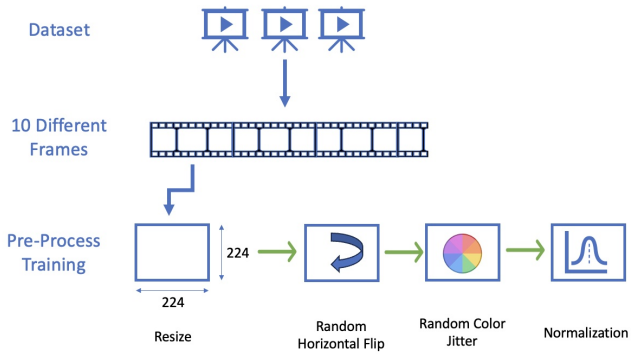

Fig. 3: Preprocessing of a Video

(FPS) during preprocessing to ensure a consistent interval between frames [7]. Subsequently, each frame was resized to a standard dimension of 224x224 pixels to align with the input requirements of our model. Following this, we applied various data augmentations, including random horizontal flips, color jitter adjustments, and normalization.

The architecture of our model was composed of a pre-trained ResNet 18, average pooling followed by an LSTM (Long Short-Term Memory) network, and a linear layer. We chose ResNet 18 for its proven efficacy in image classification, utilizing it to extract features from each frame [12]. By maintaining the original RGB format, size, and normalized values of each frame, we were able to leverage the benefits of the network's pre-trained state. The extracted features from each frame were then fed into the LSTM, which was trained so that the final hidden layer output from the 10th frame was used by the linear layer to classify the entire video.

Crucially, each video underwent dual classification: once in its original, forward direction and then again in reverse, after inverting the sequence of frames. This approach was designed to compel the model to focus on the motion characteristics distinguishing the two classifications, despite the individual frames possessing identical features. This methodology in Figure 4 aimed to deepen the model's understanding of motion as a key discriminator in temporal sequences.

3.2 Analysis

To effectively evaluate the performance of ModelScopeT2V videos, it was essential to establish specific criteria for generating these videos. Considering that the model was trained on UCF-101, we decided to align video prompts with it's categories. Due to the time-intensive nature of video generation, we opted for the model's simplest configuration, setting it to 25 inference steps and limiting it to 16 frames per video [13].

Initial experiments with ModelScopeT2V revealed a high sensitivity to the choice of prompt words. Given these constraints, producing a large volume of videos across all 95 categories was impractical. Therefore, our strategy pivoted to four videos for each category: two depicting forward motion and two illustrating reverse motion. To differentiate between the two, prompts for reverse videos included the

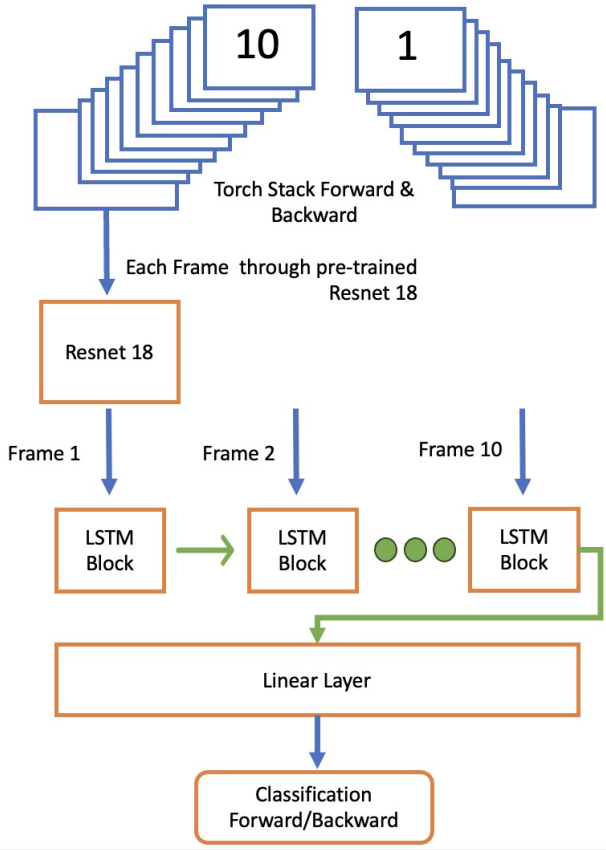

Fig. 4: Model Architecture

phrase "in reverse" at the end or "Reverse" at the beginning, while forward videos were prompted either neutrally, without any time cue, or with the word "forward" at the end.

We examined top five and bottom five categories in terms of classification accuracy. We then planned to test five different token lengths for prompts in forward and reverse directions, using the same prompt for each length within a category, resulting in a total of 10 unique videos per category. These prompts were to be generated using a Large Language Model (LLM), specifically GPT-4, allowing us to investigate the impact of prompt length on video quality and relevance.

In a more comprehensive approach, we aimed to utilize ModelScopeT2V's maximum token length of 77 for prompts [4]. From our initial experiment, we selected 15 categories: the top five, five chance, and the bottom five based on classification performance. For each category, we planned to create prompts at five different token lengths: 4, 10, 20, 40, and 70. We would generate 10 unique prompts for each length and category, ensuring varied vocabulary to counteract the model's sensitivity. Again, these prompts would be crafted using an LLM. This strategy was designed to provide insights into the effects of prompting and vocabulary variety on video generation.

3.2.1 Fine-Tuning the Classifier

Implementing the classifier trained on the UCF-101 dataset directly on videos produced by ModelScopeT2V presented

a potential issue. The distinct distribution characteristics of the videos from each source could hurt the classifier's accuracy. To address this discrepancy and potentially enhance classification accuracy, we proposed another experiment where we would fine-tune our classifier using videos generated by ModelScopeT2V.

The process of fine-tuning would serve a dual purpose. Firstly, it would allow us to adapt our classifier to the unique characteristics of the ModelScopeT2V-generated videos. Secondly, it would enable us to verify whether our classifier, once fine-tuned, could effectively discern temporal properties in these videos. Specifically, we aimed to confirm if the forward videos generated by ModelScopeT2V, when manually reversed, still aligned with the temporal patterns observed in the UCF-101 training dataset. We did this approach as it aligned with what we did for UCF-101 training. This would suggest that ModelScopeT2V was indeed producing videos with a discernible forward temporal direction.

To execute this fine-tuning process, we planned to utilize the forward videos generated from 15 selected categories. Post-fine-tuning, we would replicate the experiments conducted previously, now employing our enhanced, fine-tuned classifier. This iterative approach would provide a more nuanced understanding of the classifier's performance.

3.2.2 Human Survey

To corroborate the findings from our classification model, we planned to compare its performance with human judgment. We selected two videos from each of the 15 categories, one forward and one backward, to feature in a survey. This survey would present videos with five different token lengths, equally divided between those correctly and incorrectly classified by our model. The aim was to assess whether human participants could more accurately predict the temporal direction of these videos compared to random chance. This comparison would serve as a crucial test of our model's validity.

4 Experimental Results

We will now be going over the results of the experiments mentioned in the Proposed Method Section.

4.0.1 Classifier Training Results

For the classifier training, we used a split of 19,909 video sequences for training and 4,983 for testing, encompassing both forward and reverse versions. We employed One Cycle LR scheduler and an Adam optimizer with weight decay, setting the maximum learning rate at 1e-4. Our results, as detailed in Table 1, showed an accuracy of 86% on the training set and 78% on the test set. The classifier equally and accurately categorized forward and backward videos, with a slight tendency to misclassify backward videos as forward. Although further enhancements through more sophisticated models and larger datasets might improve accuracy, the current 78% accuracy is satisfactory for our analysis of the videos generated. This performance level serves as a baseline for evaluating the quality of ModelScopeT2V's output.

TABLE 1: Classifier Performance Metrics

	Predicted		Metric
Actual	**Backward**	**Forward**	
Backward	1920	571	Precision for Backward: 0.7786
Forward	546	1946	Recall for Backward: 0.7708
F1 for Backward:			0.7747
Precision for Forward:			0.7731
Recall for Forward:			0.7809
F1 for Forward:			0.7770
Total Videos:			4983
Accuracy:			0.780

Analyzing the curated videos revealed performance patterns across categories. Categories like Cliff Diving, Cricket Bowling, Playing Sitar, Knitting, and Billiards showed high performance. In contrast, categories like Lunges, Boxing Speed Bag, Applying Eye Makeup, and Pizza Tossing fared poorly. This outcome is encouraging, as the model struggles mainly in areas with less obvious temporal progression. Interestingly, it excels in Playing Sitar and Knitting, which might be assumed to lack clear temporal markers, suggesting a nuanced understanding of temporal sequences by the model.

4.0.2 Generated Video Results

The first experiment we performed was on the manually curated 388 videos with 4 prompts per category, 2 in forward, and 2 in reverse. The results from this experiment are shown in Table 2.

TABLE 2: Experiment 1 Results

	Predicted		Metric
Actual	**Backward**	**Forward**	
Backward	81	112	Precision for Backward: 0.554784
Forward	65	130	Recall for Backward: 0.419689
F1 for Backward:			0.477876
Precision for Forward:			0.537190
Recall for Forward:			0.66666
F1 for Forward:			0.59496
Total Videos:			388
Accuracy:			0.54

In our initial experiment, each category was represented by an average of 4 videos. Aware of the potential limitations of this small sample size for statistical significance, we conducted a focused analysis on the 5 best and 5 worst-performing categories, as detailed in Table 3. This targeted approach aimed to determine if initial findings genuinely reflected model performance or were merely artifacts of the limited dataset. By analyzing these specific categories with additional videos, we sought to draw more statistically robust conclusions about the model's capabilities.

One might contend that certain actions, by their very nature, present challenges for maintaining a consistent arrow of time, which could compromise the classifier's accuracy. However, our detailed analysis of the top 5 and bottom 5 categories reveals a nuanced picture. Out of these 10, 7 categories achieved an accuracy exceeding 60%, with only 3—namely TaiChi, and Military Parade—showing subpar

TABLE 3: Experiment 1 Category Accuracies

Categories Chosen	Forward Accuracy	Backward Accuracy
Fencing	0	0
Diving	50	0
Horse Riding	50	0
Cutting in Kitchen	0	0
Salsa Spin	100	100
Punch	100	100
Throw Discuss	100	100
Military Parade	100	100
Walking With Dog	100	100
Tai Chi	0	0

performance. Importantly, none fell below the 50% accuracy threshold or were among the most problematic categories for the classifier during training.

By selecting categories that span a broad spectrum of arrow of time complexity, we can rigorously assess the generator's proficiency. If we chose categories with clear defined arrow of time, the classifier might easily categorize the videos, thus failing to adequately test the generator's capabilities. Our approach ensures that the generator is challenged to produce videos that are not only classifiable but also retain the temporal coherence.

Moving to the second experiment, we created 5 different prompts for both forward and backward in each category, and for each prompt we generated 10 videos. The results are shown in Table 4. We see that the performance dipped considerably, and far more videos are being classified as forward, rather than backward.

TABLE 4: Experiment 2 Results

	Predicted		**Metric**
Actual	**Backward**	**Forward**	
Backward	155	345	Precision for Backward: 0.4889
Forward	162	338	Recall for Backward: 0.31
F1 for Backward:			0.3794
Precision for Forward:			0.4948
Recall for Forward:			0.676
F1 for Forward:			0.5714
Total Videos:			1000
Accuracy:			0.49

TABLE 5: Experiment 2 Category Accuracies

Categories	Forward Accuracy%	Backward Accuracy%
Fencing	66	34
Diving	64	34
Horse Riding	70	20
Cutting in Kitchen	80	10
Salsa Spin	62	42
Punch	70	28
Throw Discuss	62	30
Military Parade	66	38
Walking With Dog	72	30
Tai Chi	64	44

From Table 5, we can see that there is a significant disconnect between forward and backward classification. When we take Cutting Kitchen which has the highest forward accuracy and the worst backward accuracy and visualize it with GradCAM Fig 5 and Fig 6, we see that the classifier

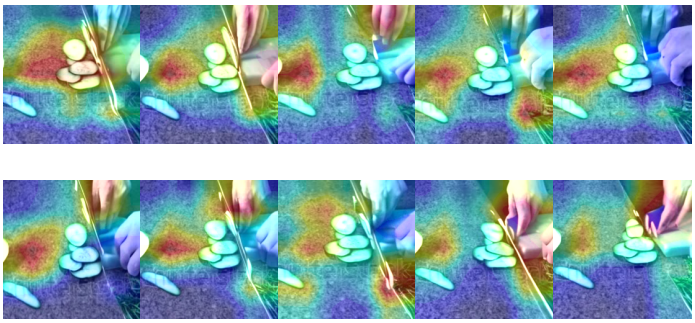

Fig. 5: Prompt: Chef Cutting Vegetables Backward

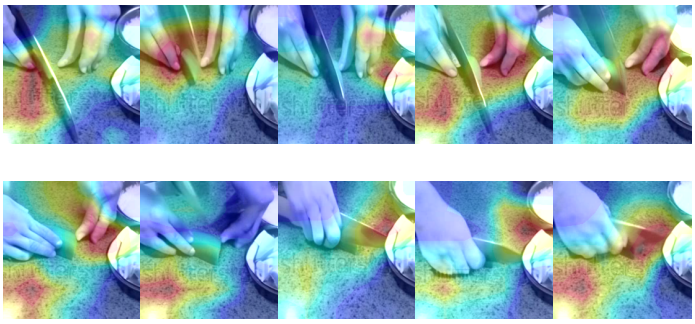

Fig. 6: Prompt: Preparing Ingredients by Cutting

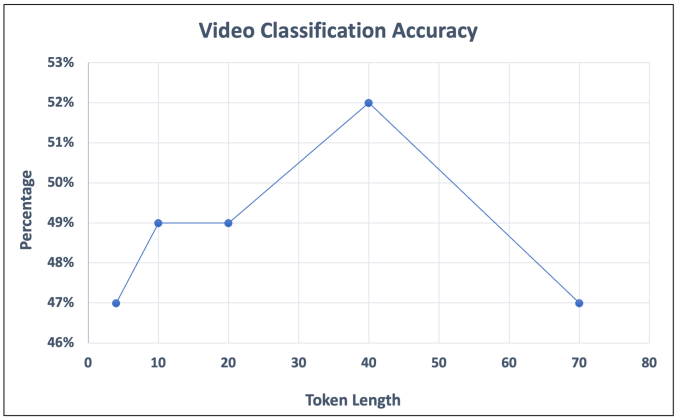

Fig. 7: Experiment 3 Token length Accuracy

focuses in the same area of motion as forward [14]. This suggests that the video's motion itself may not be conclusive enough to predict backward motion. We will look for this pattern in our Experiment 3.

In our last experiment, we wanted to specifically see if prompt length and prompt vocabulary made a significant impact. As this experiment was more robust, we went back to our experiment 1, and took 5 more categories that had an even split in forward and backward accuracy. We would now be using 10 different prompts for each of the 5 prompt lengths.

We see from Fig 7, that a prompt length of 40 had the highest accuracy. However, looking at the accuracy of the forward and backward videos for token lengths of 40, we continue to see poor reverse classification.

TABLE 6: Experiment 3 Results

	Predicted		Metric
Actual	**Backward**	**Forward**	
Backward	233	517	Precision for Backward: 0.4844
Forward	248	502	Recall for Backward: 0.3106
F1 for Backward:			0.3785
Precision for Forward:			0.4926
Recall for Forward:			0.6693
F1 for Forward:			0.5675
Total Videos:			1500
Accuracy:			0.49

TABLE 7: Experiment 3 Category Accuracies

Categories	Forward Accuracy%	Backward Accuracy%
Fencing	68	26
Diving	66	36
Horse Riding	64	34
Cutting in Kitchen	74	28
Salsa Spin	66	30
Punch	70	28
Throw Discuss	68	36
Military Parade	70	24
Walking With Dog	64	28
Tai Chi	62	42
Kayaking	60	20
Basketball Dunking	72	32
Pole Vault	54	42
Surfing	74	40
BabyCrawling	72	28

4.0.3 *Fine-tuned Results*

After our initial experiments, we noticed the classifier struggled with reverse generated videos. To investigate, we fine-tuned the classifier using the forward videos from experiment 3, with a 70/30 split for training and testing. If the test accuracy approaches our original results, it suggests the forward videos possess a discernible temporal direction. This is because, mirroring our approach with UCF-101, we manually flipped these forward videos and labeled them as reverse. Table 8 confirms this hypothesis, showing a 74% accuracy, indicating that ModelScopeT2V's forward videos indeed have a temporal direction, as seen by the higher accuracy in forward videos in earlier experiments.

TABLE 8: Fine Tuning Results

	Predicted		Metric
Actual	**Backward**	**Forward**	
Backward	168	57	Precision for Backward: 0.7433
Forward	58	167	Recall for Backward: 0.7466
F1 for Backward:			0.7450
Precision for Forward:			0.7455
Recall for Forward:			0.7422
F1 for Forward:			0.7438
Total Videos:			450
Accuracy:			0.74

We see that in Table 9,10 and 11 we repeat the results of our earlier tests, now with our fine-tuned model.

Now you might notice that our total videos we used for Experiment 3 dropped. That's because we were limited by the test videos we had available in the forward direction.

TABLE 9: Fine Tuned Classifier on Experiment 1

	Predicted		Metric
Actual	**Backward**	**Forward**	
Backward	98	95	Precision for Backward: 0.5355
Forward	85	110	Recall for Backward: 0.5077
F1 for Backward:			0.5212
Precision for Forward:			0.5365
Recall for Forward:			0.5641
F1 for Forward:			0.55
Total Videos:			388
Accuracy:			0.54

TABLE 10: Fine Tuned Classifier on Experiment 2

	Predicted		Metric
Actual	**Backward**	**Forward**	
Backward	225	275	Precision for Backward: 0.5939
Forward	192	308	Recall for Backward: 0.45
F1 for Backward:			0.4907
Precision for Forward:			0.5283
Recall for Forward:			0.616
F1 for Forward:			0.5687
Total Videos:			1000
Accuracy:			0.53

TABLE 11: Fine Tuned Classifier on Experiment 3

	Predicted		Metric
Actual	**Backward**	**Forward**	
Backward	97	128	Precision for Backward: 0.6258
Forward	58	167	Recall for Backward: 0.4311
F1 for Backward:			0.5105
Precision for Forward:			0.5661
Recall for Forward:			0.7422
F1 for Forward:			0.6423
Total Videos:			450
Accuracy:			0.59

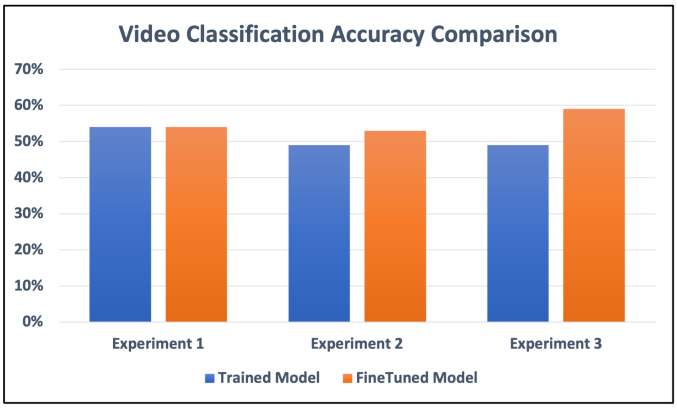

Fig. 8: Comparison of Model Acccuracies on various Experiments

This is because we already fine-tuned the model on train videos, which was a large portion of the experiment 3 videos. As a result, to keep relative proportion we used the same number of test videos we had for forward as backward. Unfortunately, despite the increase in accuracy

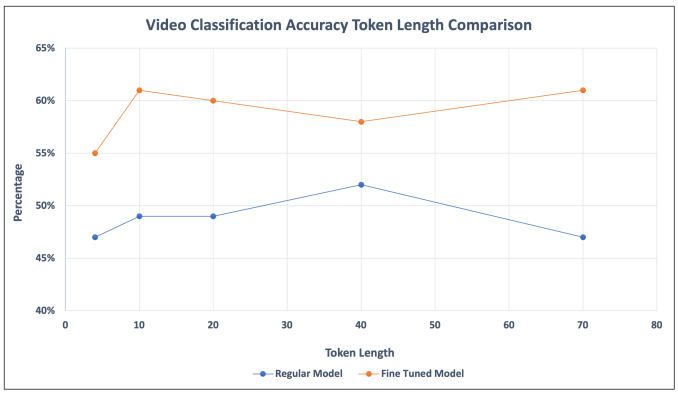

Fig. 9: Comparison of Model Acccuracies relating to Token Length

we continue to see the same pattern of low classification accuracy on reverse prompted videos. This is a strong indication, that due to the varied amount of prompts, categories and prompt lengths used, the ModelScopeT2V struggles to make reverse videos when asked to do so with reverse prompts. However, it does make reasonable forward videos that follow AoT.

TABLE 12: Fine Tuned Model Experiment 3 Results

Categories	Forward Accuracy%	Backward Accuracy%
Fencing	60	40
Diving	87	67
Horse Riding	80	50
Cutting in Kitchen	73	25
Salsa Spin	73	38
Punch	60	38
Throw Discuss	67	39
Military Parade	80	43
Walking With Dog	93	50
Tai Chi	80	42
Kayaking	53	44
Basketball Dunking	80	50
Pole Vault	60	39
Surfing	100	47
BabyCrawling	67	42

Definitive conclusions about the impact of token length remained elusive. Our findings suggest that larger token descriptions generally perform better than shorter ones, though the optimal length varies by application. In the horse riding category, Figures 10 and 11 show backward and forward prompts, respectively. Interestingly, while both videos focus on the correct motion, Figure 10 is incorrectly predicted as forward, whereas Figure 11 is accurate. For the baby crawling category, Figure 12's reverse prompt is correctly predicted, but Figure 13's forward motion prediction is wrong, possibly due to limited motion. This issue might affect the reverse prompts' accuracy, but since it should equally impact forward videos, we conclude that reverse prompts are not generated as accurately as forward ones.

4.0.4 *Human Results*

To validate the conclusions drawn by our classifier, we conducted a survey with 30 videos, encompassing all 15 categories with an equal mix of forward and reverse videos. The survey included 3 videos from each token length group. To

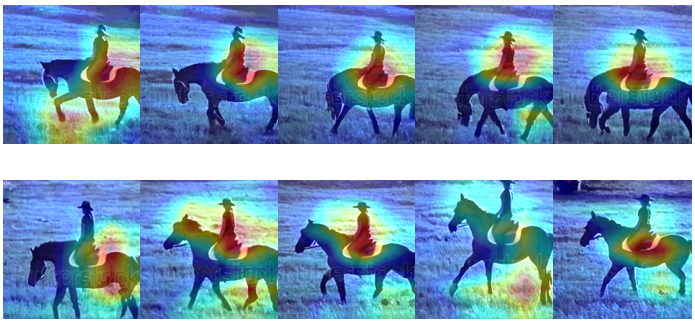

Fig. 10: Prompt: Traversing the landscape under a setting sun, the horse and rider's serene journey rewinds, the peaceful evening light receding gently

Fig. 11: Prompt: Horse riding adventure, exploring trails with enthusiasm and grace

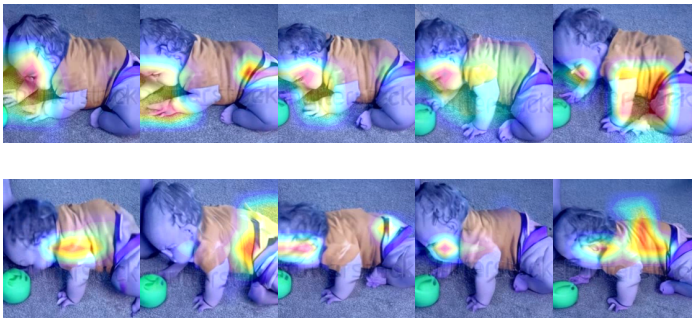

Fig. 12: Prompt: In their home's safe haven, the baby's crawling practice unfolds backward, each effort a display of emerging strength, watched over by parents whose pride remains evident in reverse

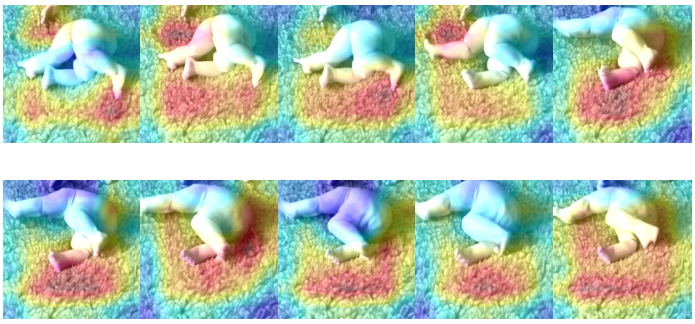

Fig. 13: Prompt: Baby's first crawl, tiny fingers gripping the soft carpet below

ensure a balanced evaluation, we included videos that were both accurately and inaccurately labeled by our model. This approach avoided biasing the survey towards the model's strengths or weaknesses. The results, as shown in Tables 13 and 14, largely aligned with the model's performance. While human accuracy was slightly higher, the difference was not substantial. Notably, humans faced similar challenges in correctly identifying the direction of reverse videos.

TABLE 13: Human Survey

	Predicted		Metric
Actual	**Backward**	**Forward**	
Backward	5	10	Precision for Backward: 0.8333
Forward	1	14	Recall for Backward: 0.3333
F1 for Backward:			0.4761
Precision for Forward:			0.5833
Recall for Forward:			0.9333
F1 for Forward:			0.7179
Total Videos:			30
Accuracy:			0.63

4.0.5 Key Statistics

We first calculate the entropy for the backward and forward sets for both humans and our fine-tuned model.

The entropy expression is:

$$H(X) = -\sum_{i} p(x_i) \log_2 p(x_i)$$

Results of the entropy score are in Table 14.

TABLE 14: Comparison of Entropy Scores, Cross-Entropy Scores, and Chi-Square Values

Model	E. Score	C-E. Score	Chi-Sq. Val.
Reg. Model	0.9056	1.09945	192.96
F. Tuned Model	0.9303	1.0741	43.55
Human	0.72192	1.32195	-

As we fine-tuned our model, it became more cautious when seeing the generated videos and classifying them, whereas our original model without fine-tuning was slightly more confident. Humans were more confident in their decisions as expected, and this matches with the high accuracy we observed in our results.

When we look at cross-entropy, we can use it to discover how each method's predictions align with the true distribution.

The cross-entropy expression is:

$$H(p, q) = -\sum_{i=1}^{n} p(x_i) \log_2 q(x_i)$$

Here, $p(x)$ is our true probability distribution, and $q(x)$ is our predicted distribution.

What we see in these results is that both the regular and fine-tuned models are different from the true distribution, with the fine-tuned model performing slightly better; however, humans are by far the worst. This is somewhat counter-intuitive as cross-entropy is typically a good measure of model performance. However, in this case, humans had the best accuracy on all the videos. This suggests that, given the high accuracy for forward videos, humans overwhelmingly could not get close to the reverse prompt true distribution. This aligns with our model's performance, which also had difficulty interpreting many reverse videos as forward.

The final metric we look at is the Chi-Square test, which will tell us how statistically different each model is compared to the true distribution.

The formula for this goodness of fit test is:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Since we have two classes, the degrees of freedom is 1. The critical value at the 0.05 significance level is 3.841.

We can see that indeed, the fine-tuned model, with its higher accuracy, is also statistically closer to the true distribution than the original model. For this particular table, both model Chi Values can't be directly compared due to their different dataset sizes, however, both are higher than the critical value.

5 Conclusion

In conclusion, we ran multiple tests to answer the question does ModelScopeT2V understand the arrow of time. While many aspects of the experiments could be optimized including using a more accurate model, more test data, and greater supervision of the video generation, the experiments suggest that while ModelScopeT2V does understand AoT when generating videos in forward time, however the reverse isn't true. Specifically, the reverse prompts used for ModelScopeT2V do not reliably generate videos that can are reversing an action. This was verified by our two models and human trials. We recommend more through investigation be performed to build on these results and to look into adjusting the spatial temporal portion of the generation model and increase training of model on reverse vocabulary. Related work such as Arrow-D mentioned in the relevant work could also be incorporated in some form to improve this understanding of Arrow of Time at generation.

Acknowledgments

The author would like to thank Dr. David Lindell for his mentorship, fellow colleagues including Kareem Alsawah, Tarit Kandpal and Edward Chen for their advice and encouragement, and a big thanks to everyone who completed the survey.

References

[1] S. A. Eddington, *Nature of the Physical World*. Cambridge Scholars Publishing, 1927.

[2] N. Rahaman, S. Wolf, A. Goyal, R. Remme, and Y. Bengio, "LEARNING THE ARROW OF TIME FOR PROBLEMS IN REINFORCEMENT LEARNING," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Montreal, QC, Canada, Apr. 2020.

[3] S. Göring, R. R. Ramachandra Rao, R. Merten, and A. Raake, "Analysis of appeal for realistic ai-generated photos," *IEEE Access*, vol. 11, pp. 38 999–39 012, 2023.

[4] J. Wang, H. Yuan, D. Chen, Y. Zhang, X. Wang, and S. Zhang, "Modelscope text-to-video technical report," 2023.

[5] T. L. Directory, "Breaking slow motion hd a green glass mineral water bottle dropping shattering in slow mo," Online video, Feb. 2013, available from: https://www.youtube.com/watch?v=t2NSxiFo1go. [Online]. Available: https://www.youtube.com/watch?v=t2NSxiFo1go

[6] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Scholkopf, and W. T. Freeman, "Seeing the arrow of time," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2043–2050.

[7] D. Wei, J. Lim, A. Zisserman, and W. T. Freeman, "Learning and using the arrow of time," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8052–8060.

[8] K. Hong, Y. Uh, and H. Byun, "Arrowgan : Learning to generate videos by learning arrow of time," 2021.

[9] M. Monfort, B. Zhou, S. A. Bargal, A. Andonian, T. Yan, K. Ramakrishnan, L. M. Brown, Q. Fan, D. Gutfreund, C. Vondrick, and A. Oliva, "Moments in time dataset: one million videos for event understanding," *CoRR*, vol. abs/1801.03150, 2018. [Online]. Available: http://arxiv.org/abs/1801.03150

[10] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[11] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012. [Online]. Available: http://arxiv.org/abs/1212.0402

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[13] ModelScope, "Damo-vilab/text-to-video-ms-1.7b · hugging face." [Online]. Available: https://huggingface.co/damo-vilab/text-to-video-ms-1.7b

[14] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016. [Online]. Available: http://arxiv.org/abs/1610.02391