Proceedings of Machine Learning Research vol 120:1–11, 2020 2nd Annual Conference on Learning for Dynamics and Control

Keyframing the Future: Keyframe Discovery for Visual Prediction and Planning

Karl Pertsch[*,†] PERTSCH@USC.EDU
Oleh Rybkin[*,‡] OLEH@SEAS.UPENN.EDU
Jingyun Yang[†] JINGYUNY@USC.EDU
Shenghao Zhou[‡] SHZHOU2@SEAS.UPENN.EDU
Konstantinos G. Derpanis[§] KOSTA@RYERSON.CA
Kostas Daniilidis[‡] KOSTAS@SEAS.UPENN.EDU
Joseph J. Lim[†] LIMJJ@USC.EDU
Andrew Jaegle[¶] AJAEGLE@UPENN.EDU

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M.Zeilinger

Abstract

Temporal observations such as videos contain essential information about the dynamics of the underlying scene, but they are often interleaved with inessential, predictable details. One way of dealing with this problem is by focusing on the most informative moments in a sequence. In this paper, we propose a model that learns to discover these important events and the times when they occur and uses them to represent the full sequence. We do so using a hierarchical Keyframe-Inpainter (KEYIN) model that first generates a video's keyframes and then inpaints the rest by generating the frames at the intervening times. We propose a fully differentiable formulation to efficiently learn this procedure. We show that KEYIN finds informative keyframes in several datasets with different dynamics and visual properties. KEYIN outperforms other recent hierarchical predictive models for planning. For more details, please see the accompanying arXiv report and the project website.[1]

Keywords: Subgoal-based Planning, Visual Planning, Learning Dynamics, Model-based Control

1. Introduction

When thinking about the future, humans focus on the important things that may happen (When will the plane depart?) without fretting about the minor details that fill each intervening moment (What is the last word I will say to the taxi driver?). Because the vast majority of elements in a temporal sequence contain redundant information, a temporal abstraction can make reasoning and planning both easier and more efficient. How can we build such an abstraction? Consider the example of a lead animator who wants to draw the next scene of a cartoon. Before worrying about every low-level detail, the animator first sketches out the story by *keyframing*, drawing the moments in time when the important events occur. The scene can then be easily finished by other animators who fill in the

* equal contribution
† University of Southern California
‡ University of Pennsylvania
§ Ryerson University and Samsung AI Centre Toronto
¶ DeepMind, work conducted while still at UPenn

1. Arxiv report: arxiv.org/abs/1904.05869, project website: `https://sites.google.com/view/keyin`.

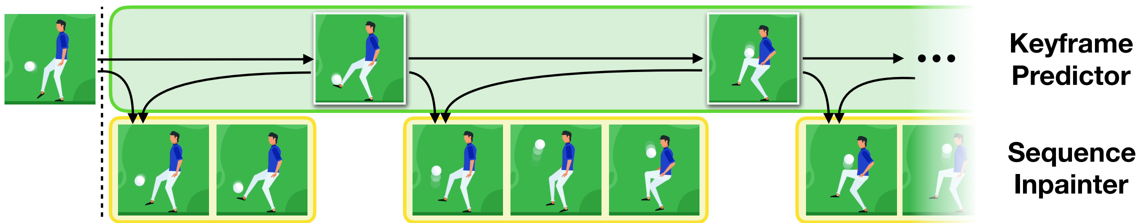

Figure 1: Keyframing the future. Instead of predicting one frame after the other, we propose to represent the sequence with the *keyframes* that depict the interesting moments of the sequence. The remaining frames can be inpainted given the keyframes.

rest from the story laid out by the keyframes. In this paper, we argue that learning to discover such informative keyframesis an efficient and powerful way to learn to reason about the future.

Our goal is to learn such an abstraction for learning dynamics of images. In contrast, much of the work on future image prediction and planning has focused on frame-by-frame synthesis (Oh et al. (2015); Finn et al. (2016); Ebert et al. (2018)). This strategy puts an equal emphasis on each frame, irrespective of the redundant content it contains or how useful it is for reasoning relative to other predicted frames. Other recent work has considered predictions that "jump" more than one step into the future, using either fixed-offset jumps (Buesing et al., 2018) or heuristics such as predictability of the frame (Neitz et al., 2018; Jayaraman et al., 2019; Gregor et al., 2019) to choose which frames to predict. In this work, we instead propose a method that predicts the keyframes that are most informative about the full sequence. We do so by ensuring that the full sequence can be recovered from the keyframes with an *inpainting* strategy, similar to how a supporting animator fleshes out the story keyframed by the lead (see Fig. 1). The keyframe structure allows us to reason about the sequence holistically when planning future actions while only using a small subset of the frames. Visual model-predictive control (MPC) methods that reason about every single future time step scale poorly if the task requires long-horizon planning. In contrast, our method enables visual planning over much greater horizons by using keyframes as subgoals in a hierarchical planning framework.

Our contributions are as follows. We formulate a hierarchical approach for the discovery of informative keyframes using joint keyframing and inpainting (KEYIN), and propose a soft objective that allows us to train the model in a fully differentiable way. We also propose a hierarchical planning algorithm for this model. We first analyze our model in a controlled setting to show that it can reliably recover the underlying keyframe structure on visual data. We then show that our model discovers hierarchical temporal structure on more complex datasets of demonstrations: an egocentric gridworld environment and a simulated robotic pushing dataset, which is challenging for current approaches to visual planning. Our approach outperforms existing hierarchical and non-hierarchical planning schemes on the pushing task, enabling long-horizon, hierarchical control.

2. Related work

Video modeling. Early deep probabilistic video models include autoregressive models that predict the pixels sequentially (Kalchbrenner et al., 2017; Reed et al., 2017). To reason about the images holistically, latent variable approaches were developed based on variational inference (Chung et al., 2015; Rezende et al., 2014; Kingma and Welling, 2014), including (Babaeizadeh et al., 2018; Denton and Fergus, 2018; Lee et al., 2018; Li and Mandt, 2018; He et al., 2018) and large-scale models such as (Castrejon et al., 2019; Villegas et al., 2019). We show how latent variable models can be used to learn temporal abstractions with a novel keyframe-based generative model.

Visual planning and model predictive control. Several groups (Oh et al., 2015; Finn et al., 2016; Chiappa et al., 2017) have proposed models that predict the future image observations given the agent's actions. Byravan et al. (2017); Hafner et al. (2018); Ebert et al. (2018) have shown that visual model predictive control based on such models can be applied to a variety of different settings. Fang et al. (2019) shows that a simple jumpy hierarchical prediction method improves planning performance in the real world. Concurrently, Nair and Finn (2020) design a hierarchical planning method that finds subgoals via extensive planning. In this work, we show that the hierarchical representation of a sequence in terms of keyframes allows more efficient hierarchical planning.

Hierarchical temporal structure. Recently, several neural methods were proposed to leverage temporal structure in video data for prediction. Neitz et al. (2018) and Jayaraman et al. (2019) proposed models that find and predict the least uncertain "bottleneck" frames. Kipf et al. (2019) propose a related method for video segmentation via generative modeling, and use it for hierarchical reinforcement learning. Kim et al. (2019) propose a method for learning temporal abstractions through hierarchical state-space models. Concurrently to our work, Shang et al. (2019) propose a keyframing method that learns to select frames that are informative about the action trajectory. In contrast to these works, KEYIN discovers informative keyframes via joint keyframing and inpainting.

3. Keyframing the future

Our goal is to develop a model that generates sequences by first predicting important observations (keyframes) and the time steps when they occur and then filling in the observations in between. To achieve this goal, in the following we (i) define a probabilistic model for joint keyframing and inpainting, and (ii) show how a maximum likelihood objective for this model leads to the discovery of keyframe structure.

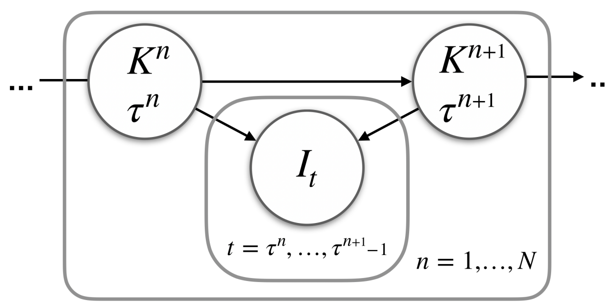

Figure 2: A probabilistic model for jointly keyframing and inpainting a future sequence. First, the model generates a sequence of keyframes $K^{1:N}$ and the corresponding temporal indices $\tau^{1:N}$ defining the structure of the underlying sequence. Second, the model inpaints the frames $I_{\tau^n:\tau^{n+1}-1}$ for each pair K^n and K^{n+1}.

3.1. A probabilistic model for joint keyframing and inpainting

To represent a sequence $I_{1:T}$ via a small set of keyframes, we propose a probabilistic model of the sequence that consists of two parts: the *keyframe predictor* and the *sequence inpainter* (see Fig. 2).

The *keyframe predictor* takes in C conditioning frames I_{co} and produces N keyframes $K^{1:N}$ as well as the corresponding time indices $\tau^{1:N}$. It factorizes in time as:

$$p(K^{1:N}, \tau^{1:N} | I_{co}) = \prod_n p(K^n, \tau^n | K^{1:n-1}, \tau^{1:n-1}, I_{co}). \tag{1}$$

From each pair of keyframes, the *sequence inpainter* generates the sequence of frames in between:

$$p(I_{\tau^n:\tau^{n+1}-1}|K^n, K^{n+1}, \tau^{n+1}-\tau^n) = \prod_t p(I_t|K^n, K^{n+1}, I_{\tau^n:t-1}, \tau^{n+1}-\tau^n), \tag{2}$$

which completes the generation of the full sequence. The inpainter additionally observes the number of frames it needs to generate $\tau^{n+1}-\tau^n$. The temporal spacing of the most informative keyframes is data-dependent: shorter keyframe intervals might be required in cases of rapidly fluctuating motion, while longer intervals can be sufficient for steadier motion. Our model handles this by predicting the keyframe indices τ and inpainting $\tau^{n+1}-\tau^n$ frames between each pair of keyframes.

3.2. Keyframe discovery

If a simple model is used for inpainting, most of the representational power of the model has to come from the keyframe predictor. We use a powerful stochastic latent variable model for the keyframe predictor and a simpler predictor network without stochastic latent variables for inpainting. Because of this structure, the keyframe predictor has to predict keyframes that describe the underlying sequence well enough to allow a simpler inpainting process to maximize the likelihood.

Specifically, to produce a complex multimodal distribution over K we use a per-keyframe latent variable z with prior distribution $p(z)$ and approximate posterior $q(z|I, I_{co})$.[2] We construct a variational lower bound on the likelihood of both I and K as follows:

$$\begin{aligned}\ln p(I, K|I_{co}) \geq \mathbb{E}_{q(z|I,I_{co})}\Bigg[\sum_{n=1}^{N} \underbrace{\ln \mathbb{E}_{p(\tau^n,\tau^{n+1}|z^{1:n},I_{co})}\left[p(I_{\tau^n:\tau^{n+1}}|K^{n,n+1}, \tau^{n+1}-\tau^n)\right]}_{\text{inpainting}} \\ + \underbrace{\ln p(K|z, I_{co})}_{\text{keyframing}}\Bigg] - \underbrace{D_{\mathrm{KL}}\left(q(z|I, I_{co})||p(z)\right)}_{\text{regularization}}.\end{aligned} \tag{3}$$

In practice, we use a weight β on the KL-divergence term, as is common in amortized variational inference (Higgins et al., 2017; Alemi et al., 2018; Denton and Fergus, 2018).

4. Continuous relaxation by linear interpolation in time

In principle, this model can dynamically predict the keyframe placement τ^n. However, learning a distribution over the discrete variable τ^n is challenging due to the expensive evaluation of the expectation over $p(\tau^n|z^{1:n}, I_{co})$ in Eq. 3. To learn the keyframe placement efficiently and in a differentiable manner, we propose a continuous relaxation of the objective.

Keyframe targets. Instead of sampling from τ^n to pick a target frame, we compute the *expected* target frame $\tilde{K}^n$ - we linearly interpolate between the ground truth images according to the predicted distribution over the keyframe's temporal placement τ^n: $\tilde{K}^n = \sum_t \tau_t^n I_t$, where τ_t^n is the probability that the n^{th} keyframe occurs at timestep t (see supplement, Fig. 7). When the entropy of τ^n converges to zero, the resulting continuous relaxation objective is equivalent to the original, discrete objective. [3]

We parametrize temporal placement prediction in terms of offsets δ with a maximum offset of J. The maximum possible length of the predicted sequence is then NJ. Large values of J may allow

2. For simplicity, the variable representing the full sequence is written without indices (I is the same as $I^{1:T}$).
3. We find this occurs most of the time in practice.

the model more flexibility, but this may also lead to the generation of sequences longer than the target $NJ > T$. To force the model to predict valid sequences at training time , we discard predicted frames at times $> T$ and normalize the placement probability over the first T steps. Specifically, for each keyframe we compute this probability as c^n: $c^n = \sum_{t \leq T} \tau^n_t$. The loss corresponding to the last two terms of Eq. (3) then becomes:

$$\mathcal{L}_{key} = \frac{\sum_n c^n \left(||\hat{K}^n - \tilde{K}^n||^2 + \beta \, D_{\text{KL}} \left(q(z^n | I, I_{co}, z^{1:n-1}) || p(z^n) \right) \right)}{\sum_n c^n}. \tag{4}$$

Inpainting targets. We produce a target image composed from the inpainted frames for each *ground truth* frame.[4] We note that as offsets δ have a maximum range of J, and in general have non-zero probability on each timestep, the inpainting network needs to produce J frames $\hat{I}^n_{1:J}$ between each pair of keyframes $\left(K^n, K^{n+1}\right)$. The expected targets are computed as: I_t: $\tilde{I}_t = \left(\sum_{n,j} m^n_{j,t} \hat{I}^n_j\right) / \sum_{n,j} m^n_{j,t}$. Here, $m^n_{j,t}$ is the probability that the j-th predicted image in segment n has an offset of t from the beginning of the predicted sequence, which can be computed from τ^n. To obtain a probability distribution over produced frames, we normalize the result with $\sum_{n,j} m^n_{j,t}$.

A detailed description of the loss computation can be found in the supplement, Sec. D. The full loss for our model is:

$$\mathcal{L}_{total} = \mathcal{L}_{key} + \beta_I \sum_t ||I_t - \tilde{I}_t||^2. \tag{5}$$

5. Keyframe-based planning

We next describe how we use the keyframe-based prediction model for long-horizon, keyframe-based planning. The hierarchical planning procedure is outlined in Fig. 3. We can generate keyframe trajectories $\hat{K}^{1:N}$ from our model by rolling out trajectories of latent variables z sampled from a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. The planning problem can be formalized as finding the set of latent variables z^* for which the resulting keyframe trajectory minimizes a given cost function c, e.g. the final distance to the goal image: $\min_z c(\hat{K}^N(z), I_{\text{goal}})$. To optimize this objective, we use the Cross-Entropy Method (CEM, Rubinstein and Kroese (2004)), which is conceptually simple and has given good results in similar settings in prior work (Hafner et al. (2018); Ebert et al. (2018)). CEM is a sampling-based optimizer that iteratively refits the sampling distributions to those parts of the latent space that resulted in trajectories of low cost. We describe the CEM procedure in more detail in the supplement, Sec. E and provide details on the used cost function in the experimental section.

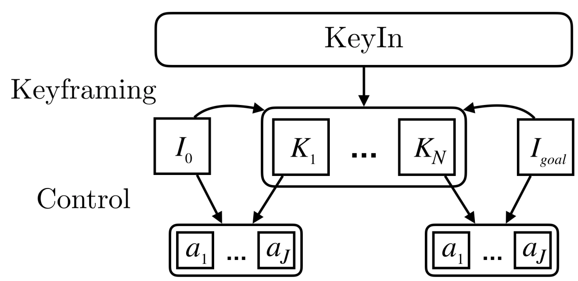

Figure 3: Keyframe-based planning. We use the keyframe model to plan a sequence of keyframes between the current observation image and the goal. A low-level controller, e.g. based on model predictive control, produces the actions, a_t, executed to reach each keyframe, until the final goal is reached.

4. This ensures that each ground truth frame contributes equally to the final loss.

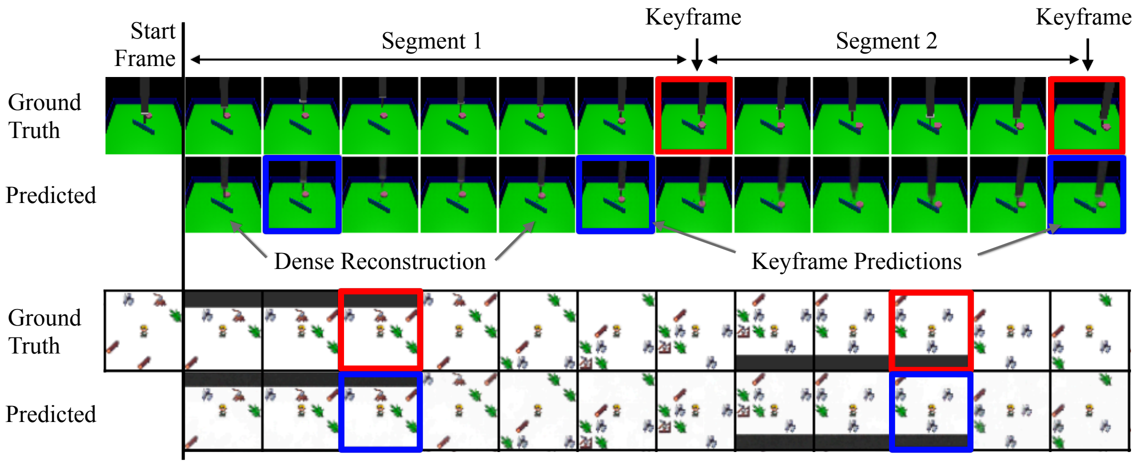

Figure 4: Example generations by KEYIN. The generation is conditioned on a single ground truth frame. Twelve of the 30 predicted frames are shown. We observe that for each transition between pushes and each action of the Gridworld agent our network predicts a keyframe either exactly at the timestep of the event or one timestep apart.

After planning a sequence of subgoals towards the goal, we execute the plan by sequentially reaching the subgoals using a low-level controller. KEYIN is agnostic to the choice of low-level controller used to reach the intermediate goals.

6. Experiments

We evaluate the quality of KEYIN's representation for future sequences by addressing the following questions: (i) Can it discover and predict informative keyframes? (ii) Can it model complex data distributions? (iii) Is the discovered hierarchy useful for long-horizon hierarchical planning?

We instantiate KeyIn using neural networks and train our model using a two-stage training procedure in which we first train the sequence inpainter to inpaint between ground truth frames sampled with random offsets and then train the keyframe predictor with the loss from Eq. 5 while freezing the weights of the inpainter. We found this lead to improved results. For further details on model architecture, training procedure and hyperparameters we refer to the supplement, Sec. A & B.

Datasets. We evaluate our model on three datasets containing structured long-term behavior. The *Structured Brownian motion* (SBM) dataset consists of binary image sequences of size 32×32 pixels in which a ball randomly changes directions after periods of straight movement of six to eight frames.

The *Gridworld Dataset* consists of 20k sequences of an agent traversing a maze with different objects. The agent sequentially navigates to objects and interacts with them. We use the same maze for all episodes and randomize the initial position of the agent and the task sketch. We use 64×64 pixel image observations and further increase the problem complexity by constraining the field of view to a 5×5 cells egocentric window.

The *Pushing Dataset* consists of 50k sequences of a robot arm pushing a puck towards a goal on the opposite side of a barrier. Each sequence consists of six consecutive pushes. We vary start and target position of the puck, as well as the placement of the barrier. The demonstrations were generated with the MuJoCo simulator (Todorov et al., 2012) at a resolution of 64×64 pixels. For more details on the data generation process, see supplement, Sec. C.

6.1. Keyframe discovery

To evaluate KEYIN's ability to discover keyframes, we train KEYIN on all three datasets with $N = 6$ keyframes, which can be interpreted as selecting the N most informative frames from a sequence. We show qualitative examples of keyframe discovery for the Gridworld and Pushing datasets in Fig. 4 and for the SBM dataset in the supplement, Fig. 10.

Table 1: F1 accuracy score for keyframe discovery on all three datasets. Higher is better.

METHOD	BROWNIAN	PUSH	GRIDWORLD
RANDOM	0.15	0.18	0.12
STATIC	0.21	0.18	0.25
SURPRISE	0.73	0.17	0.32
KEYIN (OURS)	**0.94**	**0.43**	**0.42**

For quantitative analysis, we define approximate ground truth keyframes to be the points of direction change for the SBM dataset, the moments when the robot lifts its arm to transition between pushes, or when the agent interacts with objects in the gridworld. We report F1 scores that capture both the precision and recall of keyframe discovery. We compare to random keyframe placement, a learned but static baseline that chooses identical keyframe placement for all sequences, and a method based on surprise that is similar to prior approaches (see supplement, Sec. F). The evaluation in Table 1 shows that KEYIN discovers better keyframes than alternative methods. We analyze the robustness of keyframe discovery to misspecified number of keyframes and image noise in supplement, Sec. G.

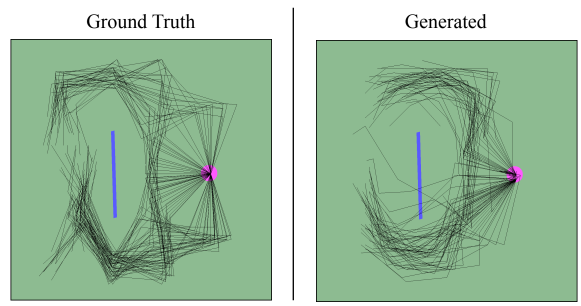

Figure 5: Distribution of trajectories sampled from KEYIN. Each black line denotes one of 100 trajectories of the manipulated object. The barrier is shown in blue, initial position in pink. The model covers both modes of the distribution.

6.2. Keyframe-based video modeling

We verify that KEYIN can represent complex data distributions in terms of discovered keyframes and attains diversity and visual quality comparable to state-of-the-art prediction models. We show sample generations on the Pushing and Gridworld datasets on the supplementary website. Fig. 5 visualizes multiple sampled Pushing sequences from our model conditioned on the same start position, showing that KEYIN is able to cover both modes of the demonstration distribution. We further show that KEYIN is competitive with prior approaches on video prediction metrics for sequence modeling and outperforms prior approaches in terms of keyframe modeling in Tables 8 & 9 in the supplement.

6.3. Hierarchical keyframe-based planning

We test whether the inferred keyframes can be used as subgoals for hierarchical planning in the pushing environment. We follow the planning procedure detailed in Sec. 5. We design a simple cost function for the pushing domain based on detected centroids of the puck in both the goal image and the predicted keyframes (more details in supplement, Sec. E). After finding a plan of subgoals, a low-level controller reaches each subgoal via model predictive control using ground truth dynamics, employing CEM for optimization of the action trajectory.

We find that KEYIN is able to plan coherent subgoal paths towards the final goal that often lead to successful task execution (executions are shown on the supplementary website[5] and in the supplement, Fig. 11). To quantitatively evaluate the keyframes discovered, we compare to alternative subgoal selection schemes: fixed time offset (*Jumpy*, similar to Buesing et al. (2018)), a method that determines points of peak surprise (*Surprise*, see Sec. 6.1), a bottleneck-based subgoal predictor (time-agnostic prediction or TAP, Jayaraman et al. (2019)), and subgoals selected at fixed intervals from sequences generated by CIGAN, an alternative sequence modeling approach (Wang et al. (2019)). We additionally compare to an approach that plans directly towards the final goal using the low-level controller (*Flat*). We evaluate all methods with the shortest path between the target and the actual position of the object after the plan is executed. As the goal of this experiment is to evaluate the quality of predicted subgoals, all methods use the same low-level controller.

As shown in Table 2, our method outperforms prior approaches. TAP shows only a moderate increase in performance over the Flat planner, which is likely because it fails to predict good subgoals and often simply predicts the final image as the bottleneck. We think this is due to the large stochasticity of our dataset and the absence of the clear bottlenecks that TAP is designed to find. Our method outperforms the planners that use Jumpy and Surprise subgoals. This further confirms that KEYIN is able to produce informative keyframes, such that it is easier for the low-level controller to follow them.

Table 2: Planning performance on the Pushing task.

METHOD	POSITION ERROR	SUCCESS RATE
INTITIAL	1.32 ± 0.06	-
RANDOM	1.32 ± 0.07	-
FLAT	0.90 ± 0.14	15.0 %
TAP	0.80 ± 0.16	23.3 %
SURPRISE	0.64 ± 0.28	50.8 %
JUMPY	0.62 ± 0.33	58.8 %
JUMPY - CIGAN	0.99 ± 0.19	15.8 %
KEYIN (OURS)	$\mathbf{0.50 \pm 0.26}$	**64.2 %**

7. Conclusion

We introduced KEYIN, a method for representing a sequence through its informative keyframes by jointly keyframing and inpainting. KEYIN first generates the keyframes of a sequence and their temporal placement and then produces the full sequence by inpainting the frames in between. We showed that KEYIN discovers informative keyframes on several datasets with stochastic dynamics. Furthermore, by using the keyframes for planning, we showed our method outperforms several other hierarchical planning schemes.

ACKNOWLEDGEMENTS

We thank the members of the GRASP laboratory at Penn, CLVR laboratory at USC, and RAIL at UC Berkeley for many fruitful discussions. We thank Marvin Zhang, Leonard Hasenclever and Shao-Hua Sun for helpful comments on the manuscript, Dinesh Jayaraman for help with the TAP baseline, and Frederik Ebert for advice on CEM planning. We are grateful for support through the following grants: NSF-IIP-1439681 (I/UCRC), NSF-IIS-1703319, NSF MRI 1626008, ARL RCTA W911NF-10-2-0016, ONR N00014-17-1-2093, ARL DCIST CRA W911NF-17-2-0181, the DARPA-SRC C-BRIC, and by Honda Research Institute. K.G.D. is supported by a Canadian NSERC Discovery grant. K.G.D. contributed to this work in his personal capacity as an Associate Professor at Ryerson University.

5. https://sites.google.com/view/keyin

References

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a broken ELBO. In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

Lars Buesing, Theophane Weber, Sébastien Racanière, S. M. Ali Eslami, Danilo Jimenez Rezende, David P. Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, and Daan Wierstra. Learning and querying fast generative models for reinforcement learning. *arXiv:1802.03006*, 2018.

Arunkumar Byravan, Felix Leeb, Franziska Meier, and Dieter Fox. Se3-pose-nets: Structured deep dynamics models for visuomotor planning and control. *Proceedings of IEEE International Conference on Robotics and Automation*, 2017.

Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. *arXiv preprint arXiv:1904.12165*, 2019.

Silvia Chiappa, Sébastien Racanière, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2015.

E. Denton and R. Fergus. Stochastic video generation with a learned prior. In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv:1812.00568*, 2018.

Kuan Fang, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Dynamics learning with cascaded variational inference for multi-step manipulation. *CoRL*, 2019.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2016.

Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, and Theophane Weber. Temporal difference variational auto-encoder. 2019.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Jiawei He, Andreas Lehrmann, Joseph Marino, Greg Mori, and Leonid Sigal. Probabilistic video generation using holistic attribute control. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

D. Jayaraman, F. Ebert, A. A. Efros, and S. Levine. Time-agnostic prediction: Predicting predictable video frames. *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

Nal Kalchbrenner, Aäron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1771–1779. JMLR. org, 2017.

Taesup Kim, Sungjin Ahn, and Yoshua Bengio. Variational temporal abstraction. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compositional imitation learning: Explaining and executing one task at a time. *Proceedings of International Conference on Machine Learning (ICML)*, 2019.

A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv:1804.01523*, abs/1804.01523, 2018.

Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*, 2018.

Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *ICLR*, 2020.

Alexander Neitz, Giambattista Parascandolo, Stefan Bauer, and Bernhard Schölkopf. Adaptive skip intervals: Temporal abstraction for recurrent dynamical models. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2018.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2015.

Scott Reed, Aäron van den Oord, Nal Kalchbrenner, Sergio Gómez Colmenarejo, Ziyu Wang, Yutian Chen, Dan Belov, and Nando de Freitas. Parallel multiscale autoregressive density estimation. In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of International Conference on Machine Learning (ICML)*, 2014.

Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer-Verlag New York, 2004.

Wenling Shang, Alex Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Learning world graphs to accelerate hierarchical reinforcement learning. *arXiv preprint arXiv:1907.00664*, 2019.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V. Le, and Honglak Lee. High fidelity video prediction with large neural nets. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2019.

Angelina Wang, Thanard Kurutach, Kara Liu, Pieter Abbeel, and Aviv Tamar. Learning robotic manipulation through visual planning and acting. *arXiv preprint arXiv:1905.04411*, 2019.