Language-aided Video Prediction from a Single Image

Zheyu Wen zheyw@umich.edu
Mu Chen muchen@umich.edu
Yuanfeng Wu yuanfenw@umich.edu
Chao Chen joecc@umich.edu
Mingshuo Shao mingshuo@umich.edu

Abstract

Our objective is to do video prediction based on only single input image. To this end, we propose a method of 'Language aided Video Prediction from Single image' (LVPS). We extract language features from VisualCOMET to predict the context information given single image. Combined with the video generation backbone (Ordinary Differential Equation and 3D convolutional VAE), we generate sharper and temporally coherent videos from a single shot of the image compared to the methods without language information. We explore two different ways to associate language and video features during video prediction, in which we found Noise Contrastive Estimation (NCE) contrastive loss plays important role in predicting a high quality video with less uncertainty that is introduced by single input image. Then, we compare different video prediction methods both qualitatively and quantitatively.

1. Introduction

Given a still image, people could imagine several possible future images over time. However, the exact motion is often unpredictable due to an intrinsic ambiguity. There are two main problems in video prediction. One is uncertainty and another is the low quality of video generation especially in a long prediction. Uncertainty is introduced by insufficient temporal information. Low quality is introduced by the improper model and training strategy. We are going to decrease the uncertainty by combining language features with video features and improve generation quality by using new backbones and novel contrastive approach.

Language is seldom used in video prediction, the question is what new information that language can provide to improve the video prediction. Video can be predicted from a single sentence which has been proved in most Language-to-Video research[12, 31, 10]. As language provides the focus of the image, like the region of interest and the logic, people can extract static and dynamic information from a text. In this paper, VisualCOMET[28] provides the chance of combining language into video prediction since it predicts what happen after the given image.

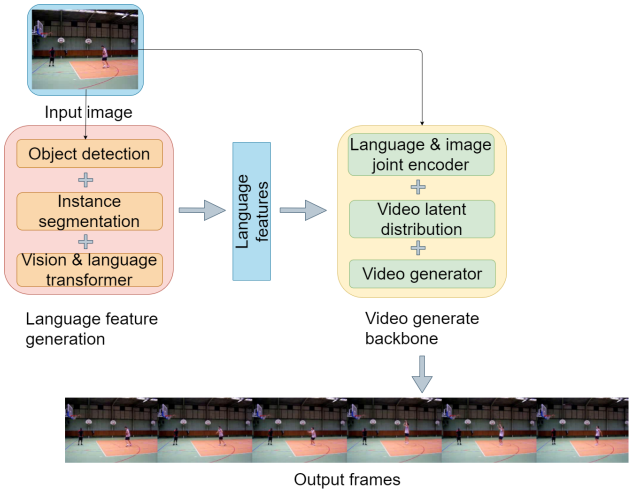


Figure 1: **Model Overview.** The language features were captured to be jointly encoded with vision features. Here we use different backbones of joint encoder to study the influence of using language features in video prediction.

Intuitively, the language information from VisualCOMET provides an extra supervising signals to the model during training and inference, which may help the model to make better predictions. More precisely, VisualCOMET is trained to be a deterministic system combining the temporal information of video in language modality, so it has potential to decrease the uncertainty in video prediction by giving a bias to the video latent distribution. Many previous video prediction works [32, 36, 4, 22] build latent space of video information in a distribution while using input single image as context to approximate posterior probability of video frames. Single image alone is far from representing the dynamic information and will result in low quality generation.

Language and video information, however, are information from two different modalities and are inherently distributed very differently. To effectively generate video using both language and video information, we need to come up with methods to bridge the gap between the two modalities. Our workflow is showed in Fig.1. The input of the generation model is a single image and the output will be a series of frames which can be transformed to a continuous video. For prediction, language features will be captured from a pretrained model to be jointly encoded with vision features.

We extensively analyze this general method by experimenting with multiple generator backbones and methods for combining language and vision. We use different backbones for joint encoder by VAE-ODE [11] or 3D-CVAE [23]. Neural Ordinary Differential Equation is a method that can generate future prediction based only on one single input initial condition.

We propose two methods for language integration. One is the naive method that simply multiplies and adds the encoded language features to the video embedding, leaving the task of cross-modality adaptation entirely to the language encoder. The second one is inspired by the Noise Contrastive Estimation (NCE) method [26, 27] which jointly trains the video and language encoders to embed the video and language features into a common feature space where matching features have higher similarity.

We tested our thought in two backbones and methods of combining language in experiment. Our experiments showed that all methods combining language are able to improve the quality of the prediction and NCE consistently outperforms the naive method. We showcase the effectiveness of latent code of video by adding one linear layer to predict the video action label.

2. Related Work

Video generation A video prediction can be generated from a single image [23, 3, 36, 31] or even from one sentence[31, 10]. The common approach to interpret the video from a still image is to sample temporal dynamics of the video in a variational probabilistic manner, conditioned on the normalized motion vector, or encoded image and motion feature. In language-to-video research, people usually extract static and dynamic information from a text using variational autoencoders (VAE) and generate the video prediction using generative adversarial network. In our model, we combine these two modalities to predict future video frames from single image and language feature associated with the scene in the image. Recently, other novel approaches has been invented such as the time-agnostic method introduced in [18] which predicting the video by limiting the solution space using bottleneck events.

Language model Currently, transformer-based language models such as GPT-2[29], GPT-3[8] and BERT[14] have been widely used in many contexts. Many works have also applied those models on the downstream task of language reasoning [34, 28], in which VisualCOMET makes it possible for combining video prediction with language model.

Variational AutoEncoder Variational AutoEncoder (VAE) [20] and Generative Adversarial Network (GAN) [17] are two popular generative models. Variational AutoEncoder is easier to train and has very clear probability explanation based on its prior assumption. Besides standard VAE, there are some variants that explore more properties of the latent space. In [9], beta-VAE is introduced which adds a beta parameter in front of KL divergence in order to further disentangle the unknown factors. [30] also proposed a method to introduce hierarchy in latent space, which is in contrast to rotational invariant standard VAE method. In this work, we tried to model the language and video features in a joint latent space so language can indicate the bias of sampling in the established video feature space.

Some work tried to use dynamic VAE for video prediction. There is a critical assumption difference between standard VAE and dynamic VAE. Signals in trajectory are not assumed i.i.d.. [6, 5, 13, 21, 25] provide many dynamic VAE models which compute the transform probability between different latent codes in different timestamps. These works are summarized in [16].

3D-CVAE model [22] is also applicable in video generation and especially useful when only single image is available as context. 3D-CVAE could model the 1D latent distribution for the whole video trajectory in inference mode. In generative mode, it draw conditional probability from known single image context.

Neural Ordinary Differential Equation Neural ODE was first introduced in [11]. It introduces a new encoder-decoder framework to compress the trajectory by LSTM and decode by ODE. It was first used in some toy model like modeling spiral and do extrapolation for unseen data during training. Later on, more machine learning methods related to ODE appear, including Augmented ODE introduced in [15], which was used to solve the problem of crossing line in the trajectory. This method enabled ODE to be a more useful tool. In this work, we adopt this useful tool to decode the future frames of video. Second-order ODE was introduced in [33]. Stochastic Differential Equation (SDE) in [19] provides stochastic methods to model the temporal relationship of given data, which explains the uncertainty in the trajectory and allows more real life applications.

Contrastive loss NCE has been used to associate between

features from different modalities as in [27]. Inspired by DeepCluster, [1] introduced three clustering-based approaches to utilize multi-modal data in training video models. The key observation in their work is that cluster assignments learned from one modality can be used as pseudo-labels to refine the representation of another modality. [35] introduced deep clustering method for unsupervised representation learning. [2] introduced cross-modal audio-video clustering by self-supervised learning. Contrastive loss enable us to put language feature and video feature into a joint latent space so the language feature can shed light on the direction of the trajectory in prediction.

3. Method

3.1. Definition

During training time, the model is given a starting frame X_0 as an image and the future frames as a sequence of images $\{X_i\}_{i\in[m]}$. Our model intends to encode the starting frame into the image feature x_0 using an image encoder, and encode the all future video sequence $\{X_i\}, \quad i = 1, ..., m$ into a single latent distribution with Gaussian prior assumption and z as latent code for the video trajectory. Language feature l is generated from starting image X_0 using VisualCOMET. x_0, z and l will be passed to a decoder which aims to reconstruct the input sequence $\{X_i\}$.

During test time, the model is only given the starting frame X_0. It samples the video trajectory z from either a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ if not using language information or from a conditional Gaussian distribution $\mathcal{N}(\mu(l), \sigma^2(l))$ with $\mu(l)$ and $\sigma^2(l)$ as function of input language features l if incorporated. The decoder will generate future frames based on the feature of the input starting frame x_0, and the sampled latent code of trajectory z.

3.2. Language feature generation

VisualCOMET is a novel framework that takes a single input image to predict events that might happen next as a set of GPT-2 features. It is a signal stream transformer model that encodes language and visual representations. The model uses the architecture as GPT-2 conditioned on visual embedding $\mathcal{V}$, ground-truth event description e, scene's location p and inference type r. To train the model, Park et al. [28] use Visual Commonsense Graphs that consist of over 1.4 million textual descriptions of visual commonsense inferences annotated over 59,000 images, each paired with short summaries of before and after. Each person in the dataset is identified by a special tag. Region of Interest Align features from Faster RCNN is the visual embedding $\mathcal{V}$, which is passed through a non-linear layer to get the final representation for each detected person. Each person's visual representation is summed with the word embedding of the token to refer the person in text. This approach is referred as "Person Grounding" input. Then for each inference sentence s and specific type r, the objective is to maximize $P(s^r|\mathcal{V}, e, p, r)$. Before the final output layer of language inference, we extract language feature. Suppose our language model is denoted as $L_\nu(X_0)$ with ν as model parameter and X_0 as single image input. Then the language feature could be obtained by

$$l = L_\nu(X_0) \tag{1}$$

where l is language feature.

VisualCOMET highly depends on Detectron Network for segmentation and boxing. Thus, initially, the image needs to be fed into the Detectron Network to be segmented and labeled. The output from the Detection Network contains the image name, its original resource, segmentation, bounding box information, and image categories. Moreover, VisualCOMET can only infer the human behaviors, thus if the image does not contain a single person or the Detectron Network misses all human targets in the pictures, VisualCOMET would not generate the inference sentence. To avoid this problem, we pre-process the Detectron Network's output and its corresponding pictures before fed into VisualCOMET Network. The images are filtered based on the labels. By checking whether the label contains human, if not, the image would be disregarded. Our video prediction models are only tested on the sampled frames from UCF101 in which the Detectron network has detected the existence of human.

3.3. Video prediction backbone

Reconstructing video from a single image frame is challenging, in which a single image doesn't include the temporal information which introduces uncertainty in the prediction. Besides, the quality of reconstruction is hard to guarantee which always results in vague results in long time prediction. Our solution against those challenges is in two folds. First, we model the temporal uncertainties using VAE-based architectures. Second, we condition the sampling of future trajectories on the corresponding language information as context cues which restricts the sampling space into a more temporally coherent subspace of video trajectories. For the backbone architecture, we tested VAE-ODE and 3D-CVAE.

3.3.1 VAE-ODE

Our Neural ODE based VAE-ODE model is illustrated in Fig. 2. We input the whole trajectory $X_0, X_1, \cdots, X_m$ and compute latent of the trajectory. Image encoder helps to encode X_i into latent feature x_i. We initialize h_0, c_0 for LSTM network and update hidden state h_i until the last h_m is obtained. To establish the latent distribution of the whole trajectory, a linear layer helps to transfer h_m into mean

and log variance of distribution which are denoted as μ and σ^2. In a word, we obtain posterior approximate probability $p(z|X_0, X_1, \cdots, X_m)$ with Gaussian Normal distribution as prior assumption. Then we sample from the posterior distribution denoted as z.

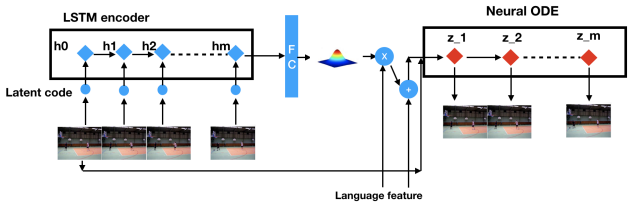


Figure 2: model architecture using ODE as backbone.

We adapt Neural ODE to reconstruct the whole trajectory from the visual feature of given image, the visual latent code sampled from video latent space and language feature l from section 3.2.

$$[x_1, x_2, \cdots, x_m] = f_\theta(x_0, z, l) \tag{2}$$

$$\hat{X}_i = g_\gamma(x_i) \tag{3}$$

where θ is the parameter of the ODE network. By decoding x_i, following Eqn. 3, we obtain the reconstruction of the original video frame $\hat{X}_i$. The whole loss function measure the reconstruction quality and approximate posterior probability of video latent.

$$\mathcal{L} = \sum_i ||\hat{X}_i - X_i|| + D_{KL}(q(z|X_0, X_1, \cdots, X_m)||p(z)) \tag{4}$$

where D_{KL} is Kullback–Leibler divergence to measure the distance between two input distribution, and $p(z)$ is assumed prior of z, and here we use Gaussian Normal distribution.

3.3.2 3D-CVAE

Besides using VAE-ODE as the video generation backbone, we have also implemented and experimented with 3D-CVAE which has been used in previous work [24]. 3D-CVAE is modified from a conventional convolutional CVAE architecture but uses 3D convolution and 3D transposed convolution for the encoder and generator of the video frames. 3D convolution extracts hierarchical representations not only within the spatial dimensions of the individual video frames but also along the temporal dimension of the video. It is based on a reasonable assumption that long and complex actions are composed of shorter and simpler actions. The architecture of 3D-CVAE is illustrated in Fig. 3.

Since both of our backbones are based on the VAE-framework, the input and output schemes of the 3D-CVAE, as well as the loss function, are similar to the ones in VAE-ODE introduced in the previous section with f_θ and g_γ denoted as parameter of encoder and decoder for 3D-CNN.

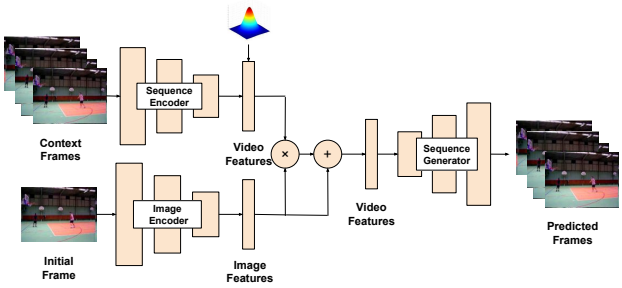


Figure 3: The 3D-CVAE backbone architecture

3.4. Language Integration

Inspired by the recent work [27] that uses Noise Contrastive Estimation (NCE) to associate features from different modalities, we designed a novel NCE-based encoding framework that encodes the video and language features into a joint feature space with desirable properties, where matching pairs of video and language features have higher similarities in the joint space than arbitrarily sampled pairs of video and language features. We observed that information of language and video features were effectively integrated into each other within the joint feature space, and the results are significantly improved on the qualities of the predicted video frames.

NCE Loss The NCE [26] models the log-odds ratio between the *true distribution* P and the *noise distribution* Q. P is the real distribution that the neural network is trying to predict, and Q is the distribution we will use to generate negative samples. We are free to specify any Q we want, depending on the context of our problem.

$$\text{logits} = \log(\frac{P}{Q}) = \log(P) - \log(Q) \tag{5}$$

NCE has been used to associate between features from different modalities as in [27]. For our problem, we can define P as the distribution of the distances of the positive pairs of language and video features and Q as the distribution of the distances of the negative pairs of features. We define a similarity metric s, video sequence embedding network E_{vid} and language embedding network E_{lan}. The NCE Loss can be written as

$$\mathcal{L}(\mathcal{P}_i, \mathcal{N}_i) = \log(\frac{s(E_{vid}(x), E_{lan}(y))}{\sum_{(x',y')\in\mathcal{N}_i} s(E_{vid}(x'), E_{lan}(y'))}) \tag{6}$$

To sample positive pair (x, y), we simply use a matching pair of video and language features. To sample negative pairs $\mathcal{N}_i$, we randomly sample bunch of images x' and pair them with language feature y' of some other potentially unrelated videos.

Our objective is to maximize $\mathcal{L}$ - the ratio of true distribution over noise, so that the networks E_{vid} and E_{lan} will learn to map video and textual features to a language-video joint semantic space, where matching features have higher similarities than irrelevant features.

The resulting video embedding $E_{vid}(x)$ is trained to associate to the corresponding language feature so it is "language-aware". E_{vid} or E_{lan} alone can be directly used as the feature extractor for our downstream task, such as generation of future frames.

Video generation with NCE Our language-video joint encoder adopts a simplified version of the NCE-loss formulation above. To compute the contrastive loss, We only use one positive and negative pair of video and language features in each training iteration. Such design choice may result in a less stable convergence but significantly speeds up the training of the model, and we have verified in our experiments that this simplified NCE loss is still able to produce good results in practice.

Our language-video encoder consists of two networks E_{vid} and E_{lan} that respectively encode raw video sequence x and extracted language features y. To model the uncertainty of the temporal dynamics in the generated video, we map the encoded language features $E_{lan}(y)$ to language latent codes l by sampling from a Normal distribution that is conditioned on the language feature. We use the simplified NCE loss to establish the joint embedding space of $E_{vid}(x)$ and l. Since in our method the subspace of language latent code and video features are trained to assimilate each other. During generation stage, we only use the sampled language latent code to generate the future frames.

4. Experiments

We train our model by modeling the latent distribution of the video to infer its parameter. Single starting image is used in both training and testing time. During test time, our model was only given the first frame of a video sequence and sampled video dynamic information from Gaussian distribution. We adapted two backbones and two ways of combining video latent distribution with language. We evaluated the MSE, SSIM, PSNR and LPIPS between the predicted frames and the ground truth. Overall, our results show that incorporating language into video prediction model helps improve the quality of the prediction.

4.1. Datasets

UCF-101 dataset contains 13320 videos within 101 action categories. These realistic action videos containing camera motion and cluttered background were collected from YouTube. All clips have a fixed frame rate as 25 FPS with a resolution of 320*240. We trained and tested our models on subset of UCF101 Dataset under the Basketball, categories. For our experiments, we chose the Basketball, archer and push-up category for training and testing which contains 134 videos. We used 100 videos for training and the rest 34 videos for testing. Each image in the video sequence was evenly sampled and obtained 10 frames to construct the training data.

4.2. Implementation

For VAE-ODE and its variants, we used a three layer Bayesian Neural Network introduced in [33] with 128 hidden units to model the gradient of the ordinary differential equation. The RNN video encoder is a LSTM with 256 hidden units. One fully connected layer is added following the output of LSTM to construct the video latent distribution. For 3D-CVAE , we use the exact parameters reported in [24] for image encoder, video sequence encoder and video sequence generator. The optimizer we used for both models are Adam optimizer with beta1 $= 0.9$, beta2 $= 0.99$. All variants of VAE-ODE used learning rate of 0.001 and trained for 5000 epochs with batch size 1. All variants of 3D-CVAE were trained with learning rate of 0.001 for 1000 epochs with batch size 5.

4.3. Evaluation Metrics

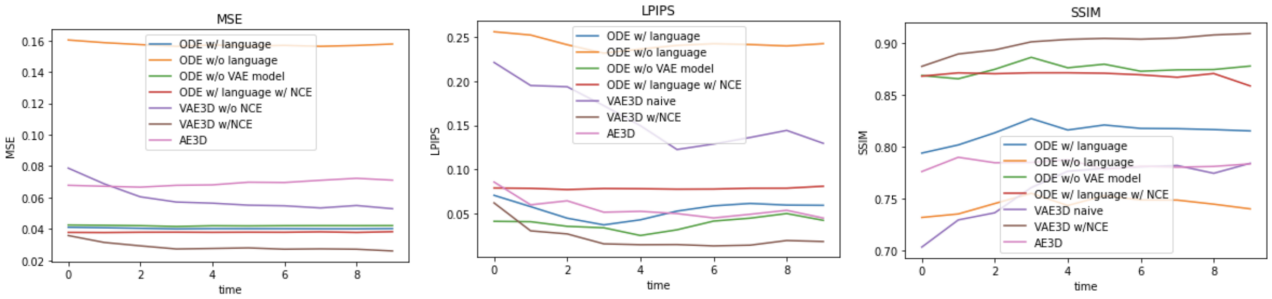
Peak Signal to Noise Ratio (PSNR): PSNR[7] measures the quality of the prediction by calculating the ratio between the maximum possible pixel value and Mean-squared error between the ground truth and the prediction in logarithmic decibel scale. Higher number suggests better result.

Structural Similarity Index Measure (SSIM): SSIM[38] is a method for measuring the similarity between two images from luminance, contrast and structure. The resultant SSIM index is a decimal value between -1 and 1, the higher number means higher similarity.

Learned Perceptual Image Patch Similarity (LPIPS): PSNR and SSIM prefer blurry prediction rather than sharp but imperfect result. Hence, we also include LPIPS[37], which has been shown to correlate better with human perception than SSIM and PSNR. It's a metric based on the linearly weighted cosine distance of visual features. Lower LPIPS distances suggest better performance.

Method & Architecture	MSE	SSIM	PSNR	LPIPS
VAE-ODE	0.16	0.76	27	0.23
3D-CVAE	0.06	0.76	24.61	0.15
VAE-ODE + language(naive)	0.04	0.81	30.21	0.09
3D-CVAE + language(naive)	0.06	0.78	23.21	0.05
VAE-ODE + language(NCE)	**0.03**	0.87	30.05	0.07
3D-CVAE + language(NCE)	0.03	**0.89**	**30.89**	**0.02**
Deterministic-ODE	0.04	0.89	30.54	0.06

Table 1: We quantitatively evaluated the results of different architectures and language integration methods. The first two rows are the metrics evaluation of our two backbone architectures without language. Row 3-4 show the performance of the models using the naive language integration methods. Row 5-6 show the performance of the two backbones that uses our novel NCE-based language integration method. Those results are compared to the results from deterministic video reconstruction. The results show that incorporating language improves the quality of prediction, and our novel NCE-based language integration method consistently outperforms than the naive method for language integration.



(a) MSE. (b) LPIPS. (c) SSIM.

Figure 4: MSE, LPIPS and SSIM in video generation from single image were evaluated for different experiment setting. x-axis is time stamp of video frames and y-axis is metric value. We take the model without language input as baseline. The effect of adding language and NCE loss were compared to baseline in different backbones. The experiment shows that language and NCE loss play important role in generating high quality video prediction.

4.4. Ablation study

We tested multiple different design choices as listed below and evaluated how each of them affects the performance of the model. Our main results for ablation studies are shown in Table 1 and Fig. 4.

Variation To find out how effective language features are in supplementing video predictors with temporal information, we compare the quality of frames generated by the language-aided models with the frames reconstructed from the original video. Here, the reconstruction model has access to the ground truth temporal information from its video input, and if the results of our language-aided model is comparable to the results of reconstruction, it clearly shows that our language embedding learns the essential temporal information from the language features. For both ODE-based and 3D-CVAE based models, we trained them with and without the variational sampling of the video latent features and compared their performances.

Architecture ODE is easier to explain for video generation from single image in its mathematical sense. However, 3D VAE is also a common way to compress the spatial temporal information in video prediction. ODE and 3D-VAE have good prediction quality in three metrics. So the key of success is language and NCE loss. Actually 3D CVAE has a slightly better output than ODE though lack the explainable trajectory.

Language integration method As comparison, we implemented the naive and the NCE-based method to integrate language features with encoded video features. The naive method directly multiplies and adds language features to the video features and treats the result as the latent code for generation. The NCE-based method is as described in section

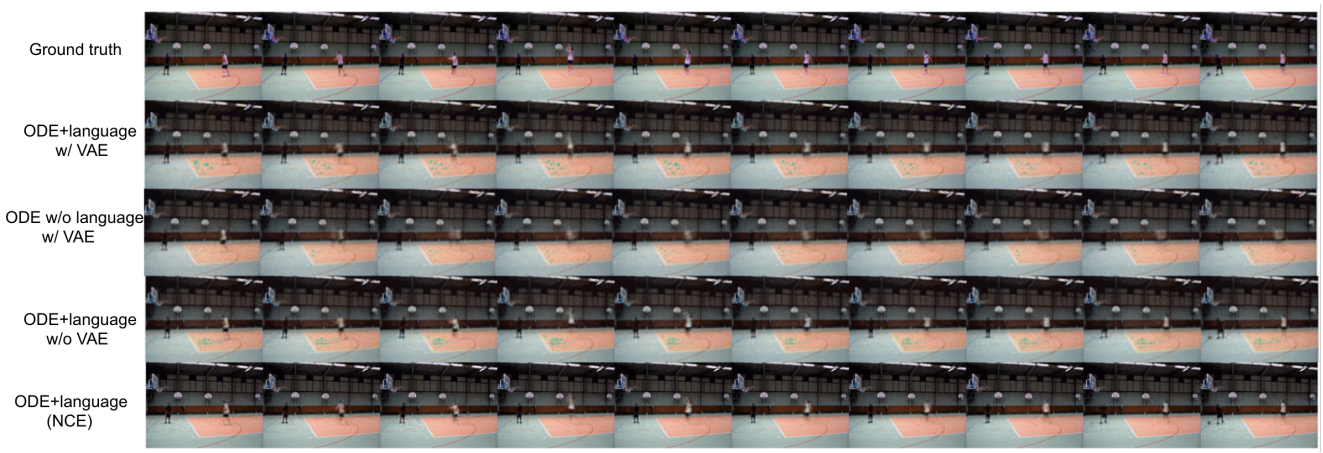


Figure 5: ODE results

Figure 6: 3D-CVAE results

4.3. We found that under the 3D-CVAE framework, NCE-based method consistently performs better than the naive method.

We summarize metric evaluation of two main backbones and their ablation study results in Table 1. Here are several observations. For Mean Squared Error, VAE-ODE+language and 3D-CVAE+language(NCE) achieved best and comparable result. Especially NCE played an important role in discriminating language features in different image setting. 3D-CVAE also achieved a higher luminance similarity compared with other setting. Both ODE and 3D-CVAE+language(NCE) achieved comparable Peak SNR which means ODE itself will ensure the high quality generation while 3D-CVAE must combine with NCE to produce comparable result. ODE+language is better than simple 3D-CVAE result. Also, our best result is comparable to the result of reconstructing the video deterministically, which justifies our hypothesis that the language features from VisualCOMET model contains correlated temporal information about the future dynamics, and that our language integration methods can extract those information so our models can effectively utilize them.

In Figure 4, we compare the individual frames along the generated trajectory with the ground truth. For the comparison, we selected three metrics which are MSE, SSIM and LPIPS. The X axis represents the time-stamp of the generated frames, and the Y axis represents the values of the metrics. One clear pattern we can found from the plots is that the models without language performs worse than models with language inputs. Moreover, we observe that, without variational sampling, the prediction will be more precise but the models are deterministic systems and need video as input which is reconstruction rather than prediction. Again, our results show that the models aided by language feature are preferred, and the models that use NCE achieved best quality in generation.

4.5. Action label prediction

Another way to test that our latent video embedding and language embedding indeed capture temporal information from the video, we trained a 2-layer feed-foward neural network on the joint embedding of 3D-CVAE + NCE model to predict the action label of the corresponding video sequence as a downstream task. In all of the versions of our model,

we fixed the weights of the sequence encoder and uses the outputs of the sequence encoder as the video embeddings. The training set for action label prediction is extracted from 3 actions (Archery, Basketball, Push-up) in the UCF101 Dataset. The predictor trained on latent joint embedding from model that incorporates both video and language information achieved test accuracy of 47%, which is higher than the accuracy of random guess. The result proves that the joint embedding contains useful information about the original videos.

4.6. Qualitative analysis

Generation quality We compare two different backbones in different training settings and visualize the results in Fig. 5 and Fig. 6. Visualization of the generation quality of using language as feature in ODE backbone is shown in Fig. 5. The first line is ground truth in UCF101 basketball video frames. The second line is the generation of video frame with video latent code randomly sampled from Gaussian distribution conditioned on language features input. The third line is generation with video latent code drawn from prior Gaussian distribution without language as model input. The fourth line is model without using VAE but adding language as input.

Fig. 6 is the visualization of the generation quality of 3D-CVAE using different methods for language integration. The first row is the ground truth images from UCF101. The second row is the outputs of 3D-CVAE without any language information. The third row is the outputs of the model that uses NCE-based language integration. The last row is the results using the naive language integration method. We can observe that model that incorporates language using the NCE method predicts frames with the highest visual quality.

Overall, ODE + language(NCE) and 3D-CVAE + language(NCE) give the best results visually. Their counterparts that use the naive language integration method achieves comparable results but are slightly less sharp. Other methods can't resolve the uncertainty in generation or have low pixel quality compared with ground truth.

Latent space visualization To see how language features help predict video more accurately, we plot the latent distribution of ground truth, which is obtained from encoding video frames in training session. Language-aided method and method without language also provide latent codes in testing time. We plot the two latent variables among 128 latent variable and visualize the relative position in the whole trajectory prediction. Language-aided method do make the latent space closer to ground truth as shown in Fig. 7.

Feature map To visualize the convolutional feature map, we plot the output of each convolutional layer along the trajectory as shown in Fig. 8. The first three layers are more like a downsampling of original image while the last convolutional layer gives some abstract feature.

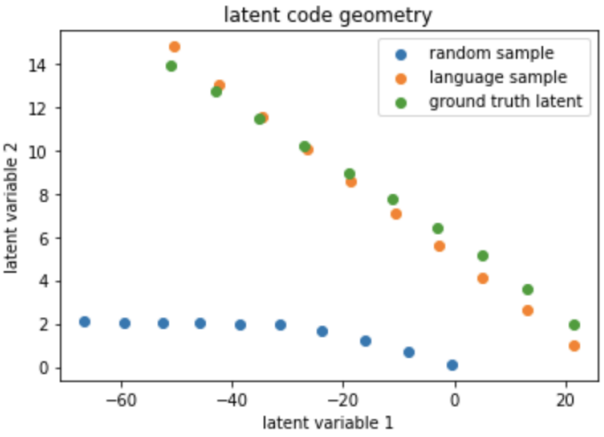


Figure 7: latent for different input cases.

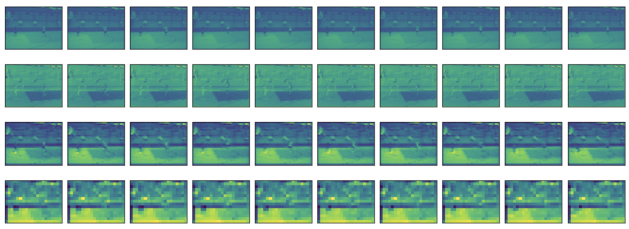


Figure 8: feature map.

5. Conclusion

In this paper, our target is to generate a high quality video prediction with lower uncertainty from a single given image. Language and NCE loss play important roles in correcting the prediction of video. With language as model input, we tried two different backbones to deal with the video generation part. ODE is a innovative way to deal with this problem since ODE only has to know the initial condition and Neural Network can be the dynamic function. This model even has potential to do interpolation andd extrapolation. For another backbone 3D-VAE, it's trained faster and also obtain a good reasonable result with NCE loss, but it couldn't do extrapolation and this method rely on contrastive loss to make sure its success.

We also analyze the video latent code and find out the success of this model in sense of latent space. This model could also do downstream task to predict video action label from our video latent code.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *Advances in Neural Information Processing Systems*, 33, 2020.

[2] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering, 2020.

[3] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy Campbell, and Sergey Levine. 2018. 6th International Conference on Learning Representations, ICLR 2018 ; Conference date: 30-04-2018 Through 03-05-2018.

[4] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction, 2018.

[5] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks, 2015.

[6] Riddhish Bhalodia, Shireen Y. Elhabian, Ladislav Kavan, and Ross T. Whitaker. Deepssm: A deep learning framework for statistical shape modeling from raw images, 2018.

[7] Alan C. Bovik. *The Essential Guide to Video Processing*. Academic Press, Inc., USA, 2nd edition, 2009.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[9] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018.

[10] Q. Chen, Q. Wu, J. Chen, Q. Wu, A. van den Hengel, and M. Tan. Scripted video generation with a bottom-up generative adversarial network. *IEEE Transactions on Image Processing*, 29:7454–7467, 2020.

[11] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[12] Carl Vondrick Kevin Murphy Cordelia Schmid Chen Sun, Austin Myers. Videobert: A joint model for video and language representation learning, 2019.

[13] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data, 2016.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[15] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In *Advances in Neural Information Processing Systems*, pages 3140–3150, 2019.

[16] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review, 2020.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[18] Dinesh Jayaraman, Frederik Ebert, Alexei A. Efros, and Sergey Levine. Time-agnostic prediction: Predicting predictable video frames, 2018.

[19] Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. In *Advances in Neural Information Processing Systems*, pages 9847–9858, 2019.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[21] Simon Leglaive, Xavier Alameda-Pineda, Laurent Girin, and Radu Horaud. A recurrent variational autoencoder for speech enhancement, 2020.

[22] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images, 2018.

[23] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *European Conference on Computer Vision*, 2018.

[24] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 600–615, 2018.

[25] Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder, 2018.

[26] Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812*, 2018.

[27] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[28] Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. Visualcomet: Reasoning about the dynamic context of a still image, 2020.

[29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

[30] Jan Stuehmer, Richard Turner, and Sebastian Nowozin. Independent subspace analysis for unsupervised learning of disentangled representations. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1200–1210, Online, 26–28 Aug 2020. PMLR.

[31] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances In Neural Information Processing Systems*, 2016.

[32] Tianfan Xue, Jiajun Wu, Katherine L. Bouman, and William T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks, 2016.

[33] Cagatay Yildiz, Markus Heinonen, and Harri Lahdesmaki. Ode2vae: Deep generative second order odes with bayesian neural networks. In *Advances in Neural Information Processing Systems*, pages 13412–13421, 2019.

[34] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning, 2019.

[35] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning, 2020.

[36] Jiangning Zhang, Chao Xu, Liang Liu, Mengmeng Wang, Xia Wu, Yong Liu, and Yunliang Jiang. Dtvnet: Dynamic time-lapse video generation via single still image, 2020.

[37] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018.

[38] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.