CAGE: Unsupervised Visual Composition and Animation for Controllable Video Generation

Aram Davtyan[1], **Sepehr Sameni**[1], **Björn Ommer**[2], **Paolo Favaro**[1]

[1]Computer Vision Group, Institute of Informatics, University of Bern, Switzerland
[2]CompVis @ LMU Munich and MCML, Germany
aram.davtyan@unibe.ch, sepehr.sameni@unibe.ch, b.ommer@lmu.de, paolo.favaro@unibe.ch

Abstract

The field of video generation has expanded significantly in recent years, with controllable and compositional video generation garnering considerable interest. Most methods rely on leveraging annotations such as text, objects' bounding boxes, and motion cues, which require substantial human effort and thus limit their scalability. In contrast, we address the challenge of controllable and compositional video generation without any annotations by introducing a novel unsupervised approach. Our model is trained from scratch on a dataset of unannotated videos. At inference time, it can compose plausible novel scenes and animate objects by placing object parts at the desired locations in space and time. The core innovation of our method lies in the unified control format and the training process, where video generation is conditioned on a randomly selected subset of pre-trained self-supervised local features. This conditioning compels the model to learn how to inpaint the missing information in the video both spatially and temporally, thereby learning the inherent compositionality of a scene and the dynamics of moving objects. The abstraction level and the imposed invariance of the conditioning input to minor visual perturbations enable control over object motion by simply using the same features at all the desired future locations. We call our model CAGE, which stands for visual Composition and Animation for video GEneration. We conduct extensive experiments to validate the effectiveness of CAGE across various scenarios, demonstrating its capability to accurately follow the control and to generate high-quality videos that exhibit coherent scene composition and realistic animation.

Project website — https://araachie.github.io/cage

Introduction

Video generation has gained significant attention in recent years, offering a transformative approach to various domains, ranging from content creation (Bar-Tal et al. 2024) to robotics (Guo et al. 2023a), autonomous driving[1] (Hu et al. 2023; Zhang et al. 2024; Gao et al. 2024) and video games (Menapace et al. 2024). Controllable video generation models are of particular interest, as they enable users to simulate the outcomes of desired changes in the environment, such as editing scene composition or assigning specific actions (e.g., moving the agent or other objects in a certain direction). Unlike traditional simulators, which require extensive manual effort to design and maintain synthetic environments, controllable video generation models can learn directly from real-world data, capturing interactions within complex scenes. This capability allows for the creation of realistic and dynamic environments that can be used to train and evaluate intelligent agents in a more flexible manner (Ha and Schmidhuber 2018; Mendonca, Bahl, and Pathak 2023).

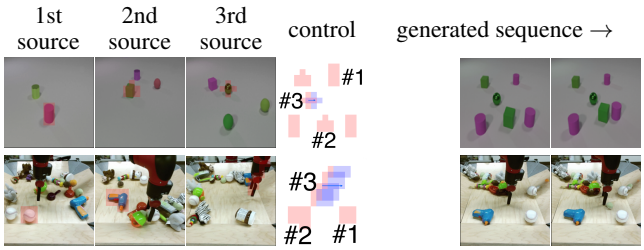

Figure 1: Scene composition and animation with CAGE on the CLEVRER and the BAIR datasets. CAGE is able to combine multiple object features from different source images and use them to compose and animate the scene in a controllable way. The selected features are shown as overlaying red patches. Blue patches in the controls correspond to the intended future locations of the objects. Notice the ability of the model to carefully adjust the appearances (e.g., sizes, shadows and lights) of the objects based on their location in the target layout. Use the Acrobat Reader to play the first images in the generated sequences as videos.

However, integrating control presents challenges due to the unavailability of information about ongoing actions in real-world video data. Existing models often rely heavily on large-scale and expensive supervision, such as text-annotations (Hu, Luo, and Chen 2021), objects' bounding boxes (Wang et al. 2024; Li et al. 2024), segmentation (Han et al. 2022), and motion cues (Shi et al. 2024). These requirements not only limit the scalability and flexibility of these models but also constrain their applicability to new domains where such annotated data is scarce or nonexistent.

To address this limitation, recent advancements have pro-

[1]For example, GAIA by Wayve (https://wayve.ai) and the world models developed by Waabi (https://waabi.ai)

posed unsupervised learning methods that build controllable video generation models only from real videos (i.e., without any information about the actions or the action space) (Menapace et al. 2021; Blattmann et al. 2021; Davtyan and Favaro 2022; Bruce et al. 2024). In these models, control is often defined as a separate input from visual data, ranging from motion encodings (Blattmann et al. 2021; Davtyan and Favaro 2022) to general embeddings (Menapace et al. 2021, 2022; Bruce et al. 2024), which are learned directly from the visual data. However, these control signal choices impose limitations on the tasks the model can perform. For instance, they allow to specify objects' motion, but not how to compose a scene, which we aim to address in this work.

We propose CAGE, short for *visual Composition and Animation for video GEneration*, a generative model capable of creating environments and animating objects within them (see Fig. 1). A key innovation in CAGE is that its control is specified directly through a set of "visual tokens" rather than embeddings from a separate action space. These visual tokens provide information regarding the identity of objects (*what*), or parts thereof, and their spatio-temporal placement provides information about *where* the objects should appear in certain frames in the future. As visual tokens we use DINOv2 spatial features (Oquab et al. 2023). Our experiments demonstrate that leveraging DINOv2 features enables our model to mitigate overfitting to the control signal and facilitates zero-shot transfer from other image domains. During training, we extract DINOv2 features from future video frames and utilize a sparse subset of these features as the control signal. By using this partial information, the model is trained to inpaint the surrounding context, both spatially and temporally, and thus learns to generate future frames consistent with the specified control signal. In summary, CAGE produces a video output that encapsulates the desired composite scene, along with animations depicting the objects within it, including their interactions. Because we train CAGE in a fully unsupervised manner, it can be readily scaled to accommodate large datasets. Our main contributions are summarized as follows:

- CAGE is a novel controllable video generation model trained without any human supervision, capable of scene composition and object animation;
- we introduce a unified control format that simultaneously can specify how to compose and animate a scene through a sparse set of visual tokens. This format allows to describe a wider range of prediction tasks than motion-based controls. In particular, it allows to compose scenes. Moreover, our control allows zero-shot transfer from frames not in our training set;
- CAGE generates more realistic videos than prior work (according to the commonly used metrics) and its controllability has been validated experimentally.

Prior Work

Video Generation. Recent advancements in the field of video generation have been remarkable, driven predominantly by the impressive capabilities of diffusion and flow-based models. These models have significantly enhanced the generation of both images and videos from text descriptions, setting a new benchmark in the domain. The foundational work in text-to-image generation (Rombach et al. 2022) has paved the way for its extension to the text-to-video arena (Guo et al. 2023b; Ho et al. 2022a,b). However, as highlighted by Blattmann *et al.* (Blattmann et al. 2023) and Zhang *et al.* (Zhang et al. 2023a), despite these advancements, current text-to-video models often produce videos with objects whose motion trajectories appear relatively random. Moreover, specifying precise motion dynamics with a lone text prompt poses significant challenges.

Supervised Controllable Video Generation. A first approach that can exploit more accurate motion specifications in the text prompt is MAGE (Hu, Luo, and Chen 2021). This approach aligns well with the text-to-video generation pipeline, albeit necessitating increased supervision. An alternative strategy is to use ControlNet (Zhang, Rao, and Agrawala 2023) for videos (Zhang et al. 2023b; Chen et al. 2023; Ma et al. 2023). This is particularly effective when the future frame structure is known, as it offers a predefined control mechanism. Models such as Boximator (Wang et al. 2024) and Motion-I2V (Shi et al. 2024) enhance controllability by employing bounding boxes or motion flows as more intuitive control mechanisms. These models rely on supervised learning paradigms and in particular on text descriptions. While effective, this reliance poses inherent limitations on scalability and adaptability, as supervised methods require large, labeled datasets, which are resource-intensive to produce and may reduce their applicability scope.

Unsupervised Controllable Video Generation. Unsupervised approaches in video generation have been gaining traction, primarily focusing on leveraging implicit signals such as masks (Huang et al. 2022), learned actions (Menapace et al. 2021; Bruce et al. 2024; Davtyan and Favaro 2022) and optical flow (Blattmann et al. 2021). These methods offer a promising avenue by minimizing reliance on extensive labeled datasets, thus addressing the scalability and adaptability challenges inherent in supervised models. The closest to our work, YODA (Davtyan and Favaro 2024) conditions the generation on a sparse set of optical flow vectors and achieves remarkable capabilities in generating out-of-distribution and counterfactual scenarios. However, despite the steep progress, these models often struggle to achieve cross-scene generalization, as their generative capabilities are tightly coupled with the specificity of the conditioning signals used during training. Moreover, none of these models is capable of simultaneously composing and animating the scene with the same control signal. In contrast, our model leverages sparse "visual tokens" for local conditioning, offering an innovative, intuitive and flexible mechanism that expands the control space available in the prior work and hence advances video generation.

Inside the CAGE

The goal of controllable video generation is to learn the following conditional distribution

$$p(x^{n+1:n+k} \mid x^{1:n}, a^{n+1:n+k}), \tag{1}$$

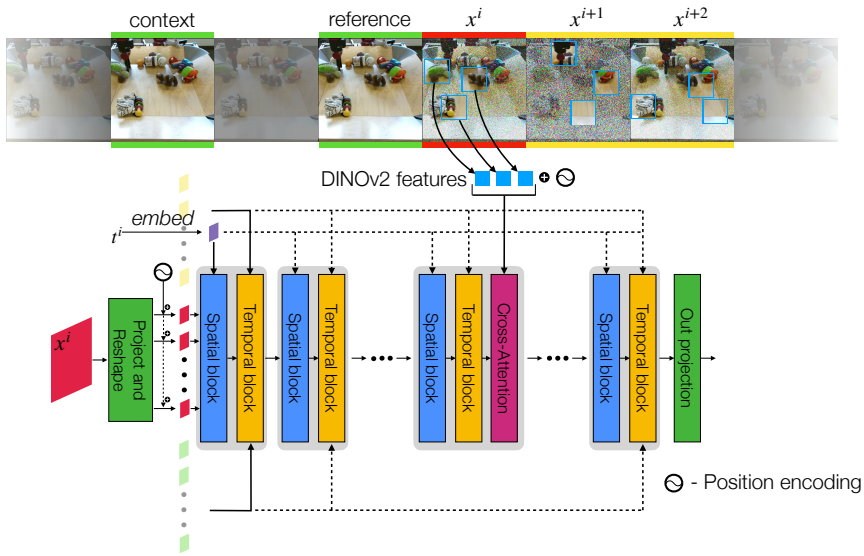

Figure 2: Overall pipeline of CAGE. The model takes all the colored frames and processes them equally and in parallel. The pipeline for a single frame (x^i, in red) is illustrated. CAGE is trained to predict the denoising direction for the future frames ($x^{i:i+2}$) in the CFM (Lipman et al. 2022) framework conditioned on the past frames (*context* and *reference*) and sparse random sets of DINOv2 (Oquab et al. 2023) features. The frames communicate with each other via the Temporal Blocks while being separately processed by the Spatial Blocks. The controls are incorporated through Cross-Attention.

where $x^i \in \mathbb{R}^{3\times H\times W}$, $i = 1,\dots,n+k$ is a sequence of RGB video frames, a^i is the control at time i, and $x^{i:j}$ ($a^{i:j}$) denotes the set of consecutive video frames (controls) between times i and $j \geq i$. Following (Davtyan, Sameni, and Favaro 2023) we model the distribution in eq. 1 as a denoising process in the conditional flow matching (CFM) formulation of diffusion (Lipman et al. 2022). In CFM, the model $v_t(x_t^{n+1:n+k} \mid x^{1:n}, a^{n+1:n+k}, \theta)$ with parameters θ is trained to approximate the direction of the straight line that connects independently sampled noise and data points:

$$\theta^* = \arg\min_\theta \mathbb{E}\left\| v_t - x_1^{n+1:n+k} + (1-\sigma_m)x_0^{n+1:n+k}\right\|_2^2. \tag{2}$$

Here the expectation is with respect to t, x_0, x_1, and a, where t is a random timestamp sampled from $U[0,1]$, $x_0^i \sim \mathcal{N}(0,I)$, $x^{1:n+k}$ and $a^{n+1:n+k}$ are sampled from the dataset, and $x_t^i = tx^i + (1-(1-\sigma_m)t)x_0^i$ with a small $\sigma_m \approx 10^{-7}$. It can be shown that integrating from $t=0$ to $t=1$ the following ODE

$$\begin{aligned} \dot{X}_t &= v_t(X_t \mid x^{1:n}, a^{n+1:n+k}, \theta^*), \\ p(X_0) &= \mathcal{N}(0,I), \end{aligned} \tag{3}$$

where X_t denotes the window of noisy future frames $x_t^{n+1:n+k}$, leads to $p(X_1) \approx p(x^{n+1:n+k} \mid x^{1:n}, a^{n+1:n+k})$. That is, to sample from the probability density function in eq. (1) one can sample Gaussian noise at time 0 and gradually denoise it by following the trajectories of eq. (3) till time $t=1$. For more details, please refer to (Lipman et al. 2022).

As suggested in (Davtyan, Sameni, and Favaro 2023) we relax the computational complexity of conditioning on the past frames by distributing the conditioning over the flow integration steps. This allows to effectively condition each step only on two past observations: the previous one x^n (the reference) and one uniformly sampled from the past x^c (the context). Thus, we work with the model $v_t(x_t^{n+1:n+k} \mid x^n, x^c, a^{n+1:n+k}, \theta)$, $c \sim U\{1,\dots,n-1\}$. At inference, c is also randomized at each integration step of the ODE. This allows the model to observe all the past frames, while keeping the computational costs constant.

We also observed that decoupling the noise levels of separate frames during training is beneficial to the quality of the generated frames. *I.e.*, we set $t = (t_1,\dots,t_k)$ with independently sampled t_i that are not necessarily equal to each other and $X_t = x_t^{n+1:n+k} = (x_{t_1}^{n+1},\dots,x_{t_k}^{n+k})$. This allows the noisy frames to exploit shared information (e.g., the background) in the less noisy frames and leads to improved training. Besides this, the model with decoupled t is more flexible at inference, where the denoising can be done within a sliding window of growing noise levels. However, here we leave exploring this capability to future work and instead focus on the model's controllability. A similar technique was proposed for motion synthesis (Zhang et al. 2023c) and for video diffusion (Ruhe et al. 2024).

Lastly, before we proceed to the definition of our control format, we would like to point out an important requirement for our model. To enable the synthesis of a scene given only the controls (also referred to as "*from scratch*"), we drop the conditioning on the context frame for 50% of the training. Moreover, we drop the reference frame 20% of the time during which we drop the context frame. By simultaneously training our model under different conditioning settings, we endow it with the ability to both predict videos from some

initial frames and to generate them from scratch.

Controlling Scene Composition over Time via Sparse Features

Our goal is to build a video generation model in which the controls can serve both as motion guidance for the video prior and as scene generation specifications. The latter control means that when we omit the conditioning on the past, the model has to generate and animate the scene from scratch. The control input describes the scene in terms of objects (or parts thereof) and their positions in space and time. In contrast to prior work that is either limited to only scene generation (Epstein et al. 2022; Hudson and Zitnick 2021) or uses supervision (Huang et al. 2023; Ma, Lewis, and Kleijn 2023), instead of leveraging composite separate solutions for scene generation and animation (Bruce et al. 2024; Wang et al. 2024), we seek for a unified control that can simultaneously solve both problems in an unsupervised way.

To this end, we propose to use a sparse set of DINOv2 (Oquab et al. 2023) spatial tokens from the last l ViT (Dosovitskiy et al. 2020) layers as the controls a^i. More precisely, for each future frame x^i, each spatial token from the 16×16 grid $f^i \in \mathbb{R}^{d \times 16 \times 16}$ of its DINOv2 encodings is assigned a probability $\pi \in [0, 1]$ to be selected for control. Then, a random 16×16 mask $m^i \in \mathbb{R}^{16 \times 16}$ is sampled from the corresponding Bernoulli distribution. This mask consists of 0s and 1s, where 1s stand for the selected tokens. The control is then calculated as

$$a^i = m^i \cdot f^i + (1 - m^i) \cdot [\text{MSK}], \tag{4}$$

where $[\text{MSK}] \in \mathbb{R}^{d \times 1 \times 1}$ is a trainable token.

By conditioning on sparse DINOv2 features and training for video prediction (generation) we gain two major advantages. 1) The controls are unified and data-agnostic. This means that, during testing, a scene can be constructed by positioning DINOv2 features extracted from unseen images. These images can even belong to domains not present in the training set, as we later demonstrate. Object motion can be specified by simply adjusting the position of the same features in future frames. 2) The controls are abstract enough to enable the prediction of changes in the appearance and location of moving objects across multiple frames by using the same features. In contrast, if RGB patches at the pixel level were used as controls, the model could overfit by relying on the texture details of specific object instances to recover their exact positions. Indeed, in the ablations we show that the controllability suffers from too much information in the control tokens. This motivates us to use features rather than raw color patches. By default, unless otherwise stated, we leverage the ViT-S/14 version of DINOv2.

Note that our model is general enough to leverage other feature extractors that satisfy the properties mentioned above, such as CLIP (Radford et al. 2021). However, we empirically find DINOv2 to perform better. Moreover, we favor the use of DINOv2, in contrast to CLIP, because it was trained in an unsupervised manner, which makes CAGE fully unsupervised. We ablate the choice of DINOv2 as the feature extractor in the supplementary material.

Scale and Position Invariance

Even though DINOv2 features are quite abstract, they preserve information about the original position of the corresponding patches in the input images (Yang et al. 2023). Naively conditioning on those features would lead to overfitting to such positional information and hence achieve limited controllability over the locations of the objects (see Sec. and the supplementary material). In order to mitigate this issue, we propose to feed the features in such a way that the identities of the objects would be preserved, but their positional information would be destroyed. To do so, we calculate the features on random crops instead of whole images. This approach ensures that the position of objects within the crop varies throughout training, encouraging the model to disregard positional information. This procedure is illustrated in Fig. 3.

Out of Distribution Controls

YODA (Davtyan and Favaro 2024) demonstrated the ability to generalize to out of distribution (o.o.d.) control. In fact, YODA can move background objects in the BAIR dataset (Ebert et al. 2017). In the training set, background objects only move when pushed by the robotic arm. This is accomplished by balancing two objectives during training: 1) learning the video prior, which defines how objects should move when no controls are specified, and 2) learning to accurately follow the provided controls. The right balance is the outcome of tuning many components, among which the most important ones, according to Davtyan and Favaro (2024), are the number of the controls and where they are sampled. In contrast, we propose to keep the training simple and rely on the compositionality. Our method has no restrictions on where the controls can be sampled from. Instead, we propose to find the balance at test time by utilizing classifier-free guidance (Ho and Salimans 2022) to allow generalization to the o.o.d. controls. More precisely, we modify the estimated vector field with

$$v_t = (1 + w) \cdot v_t(X_t \mid x^{1:n}, a^{n+1:n+k}, \theta^*) \tag{5}$$

$$- w \cdot v_t(X_t \mid x^{1:n}, \varnothing, \theta^*), \tag{6}$$

where $w \geq 0$ is the guidance strength. The larger w the more the model relies on the control (to move background objects) rather than on the video prior (not to move background objects). Fig. 4 shows some examples of o.o.d. controls in BAIR.

Training Details and Architecture

For computational efficiency, as proposed in (Rombach et al. 2022), CAGE works in the latent space of a pre-trained VQGAN (Esser, Rombach, and Ommer 2021). All frames $x^{n:n+k}$ as well as x^c are separately encoded to latents. Each latent is a feature map of shape $c \times h \times w$. These latents are first reshaped to $(h \cdot w) \times c$. Then a random set of tokens (but the same across time) is dropped from each of the latent codes, leaving $m = \lfloor (1 - r) \cdot h \cdot w \rfloor$ tokens per latent. Here r is the masking ratio that is typically equal to 0.4 in our experiments. The remaining tokens are then concatenated in the first dimension to form a single sequence of $(k + 2) \cdot m$

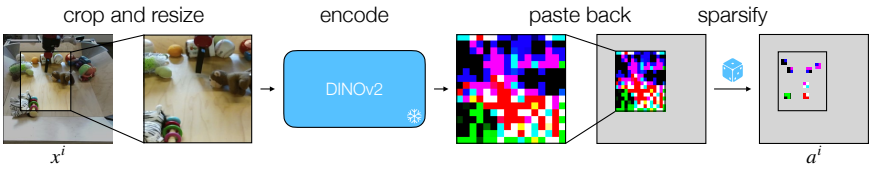

Figure 3: The process of selecting controls for conditioning. The image is first cropped and resized to 224×224 resolution that can be fed to DINOv2 to obtain the spatial tokens. Those are then pasted back to the original location of the crop and sparsified. This is done to prevent overfitting to the position information that is present in DINOv2 features. Besides this, calculating the features on the crops of the image makes the model scale invariant. That is, at inference we are able to copy objects from the background and paste them to the foreground and vice versa. The model should be able to automatically figure out how to scale the objects according to their target position in the scene as well as how to add other position-related textures (e.g. shadows).

tokens. This sequence is then passed to v_t, which we model as a Transformer (Vaswani et al. 2017) that consists of 9 blocks of alternating spatial and temporal attention layers. Spatial layers are allowed to only attend to the tokens within a given frame, while temporal layers observe all the tokens. As suggested in (Hatamizadeh et al. 2023), t is embedded in the architecture as a time-depended bias that is added to the queries, keys and values in all attention layers. We add learnable spatial position encodings to all the tokens before feeding them to v_t. The same position encodings are later added to the control tokens $a^{n+1:n+k}$. For the video indices temporal layers employ relative position encodings (Wu et al. 2021). 5 middle blocks of the Transformer additionally incorporate cross-attention layers, where tokens from the i-th frame attend to a^i. The overall pipeline of CAGE is depicted in Fig. 2.

Experiments

In this section we conduct a series of experiments on several datasets to highlight different capabilities of CAGE and demonstrate its superiority over the prior work.

Data

We test our model on 3 datasets. Ablations are conducted on the **CLEVRER** (Yi et al. 2019) dataset. This dataset consists of 10k training videos capturing multiple synthetic objects colliding into each other and interacting on a flat surface. This dataset combines simplicity of the objects with quite interesting interactions and supports compositionality making it a great field for experimentation. We further test CAGE on the **BAIR** (Ebert et al. 2017) dataset, which is a dataset containing 44k clips of a real robotic arm manipulating diverse set of objects on a table. This dataset provides more complex objects and interactions and is suitable for demonstrating generalization to o.o.d. controls. Finally, we test our model on real egocentric videos from the **EPIC-KITCHENS** (Damen et al. 2021) dataset. For making the training more resource efficient, we restrict ourselves to a single person/kitchen from the dataset, namely P03.

Ablations

We start by ablating different components of CAGE on the CLEVRER (Yi et al. 2019) dataset. For ablations we train CAGE with a smaller batch size of 16 samples. However, later we show that the performance of the model scales accordingly with larger batch sizes.

In order to select the best configuration of the model, we assess its controllability in terms of scene composition. To this end, we have annotated 128 images from the test set of the CLEVRER dataset by selecting 4-6 random patches of the same object in an image. Then, the model is fed as input a real frame and the selected patches shifted to a random target location, and outputs a generated frame. The quality of the generated frame is measured with a combination of two metrics: FID (Heusel et al. 2017) for image realism and CTRL, which we propose for measuring control accuracy (i.e., how much the generated frame follows the input control). CTRL is the ratio of S over D. S is high when the object in the real image has moved to the desired target destination. It is measured as the cosine distance between normalized DINOv2 features at the initial position in the real frame and those at the target location in the generated frame. D instead is high when the object does not move. It is measured as the cosine distance between normalized DINOv2 features at the initial position in the real frame and those at the same initial location but in the generated frame. Finally, CTRL is high when the model is highly controllable and low otherwise. Empirically we observe that both S and D are lower bounded by some positive number, which makes CTRL a valid metric. We test different configurations of the model and demonstrate that the variant with 1 DINOv2 layer, $\pi = 0.1$, $k = 3$, with randomized t_i and scale/position invariance performs the best on CLEVRER (see Table 1). We provide visual examples to highlight the limited control in the ablated models in the supplementary material.

Quantitative Results

CLEVRER. Employing the optimal settings identified in the ablation studies, we train CAGE with the full batch size of 64 samples and report the results in Tab. 2. Besides the scene composition metrics studied in the ablations, following Davtyan and Favaro (2024), we reconstruct 15 frames of a video from a single initial one conditioned on the control for the first generated frame. We report LPIPS (Zhang et al. 2018), PSNR, SSIM (Wang et al. 2004) and FVD (Unterthiner et al. 2018). Table 2 shows that CAGE generates videos of better quality compared to prior work.

BAIR. Following prior work by Menapace *et al.* (Menapace

l	π	k	random t_i	scale/pos inv.	CTRL↑	FID↓
1	0.1	3	✓		1.306	12.88
1	0.1	1	✓	✓	1.529	9.85
1	0.1	3		✓	<u>1.596</u>	5.82
1	0.9	3	✓	✓	1.464	6.19
1	0.5	3	✓	✓	1.566	5.19
1	0.01	3	✓	✓	0.782	6.11
4	0.1	3	✓	✓	1.084	5.24
2	0.1	3	✓	✓	1.500	**4.95**
1	0.1	3	✓	✓	**1.615**	<u>4.98</u>

Table 1: Ablations of model components with respect to compositional scene generation quality. The controllability score (CTRL) and the image quality (FID) are evaluated. The row corresponding to the full model is highlighted in gray. For the CTRL metric the average of 5 runs is reported.

Method	LPIPS↓	PSNR↑	SSIM↑	FID↓	FVD↓	CTRL↑
YODA	**0.126**	**30.07**	0.93	5.7	70	-
CAGE *(ours)*	0.166	30.02	**0.98**	**4.3**	**62**	1.667

Table 2: Comparison with YODA (Davtyan and Favaro 2024) on the *CLEVRER* dataset. 15 frames generated from a single one. Control is provided for the first generated frame.

Method	LPIPS↓	FID↓	FVD↓
CADDY (Menapace et al. 2021)	0.202	35.9	423
Huang et al. (Huang et al. 2022)			
positional	0.202	28.5	333
affine	0.201	30.1	292
non-param	0.176	29.3	293
GLASS (Davtyan and Favaro 2022)	0.118	18.7	411
YODA (Davtyan and Favaro 2024)			
5 controls	<u>0.112</u>	18.2	264
1 control	0.142	19.2	339
CAGE *(ours)*			
10% controls	**0.107**	**6.4**	**136**
1% controls	0.149	<u>7.2</u>	<u>169</u>
tracking	0.194	9.1	214

Table 3: Evaluation on the *BAIR* dataset.

Method	GT	from GT dist.	o.o.d. S	o.o.d. L
CADDY (Menapace et al. 2021)	6.29	-	-	-
GLASS (Davtyan and Favaro 2022)	1.75	25.00	25.07	30.06
YODA (Davtyan and Favaro 2024)	1.41	1.77	2.01	3.37
CAGE *(ours)*	2.74	2.93	3.57	5.45
CAGE *(ours)* w/ pos. interpolation	2.29	2.33	2.85	4.77

Table 4: Optical flow error in pixels of the control applied to the robotic arm in the *BAIR* dataset. Average of 5 runs.

et al. 2021), we trained CAGE on the BAIR dataset (Ebert et al. 2017). We autoregressively generated 29 frames starting from a given initial frame and employing various controls. For feature extraction on BAIR we found that it is optimal to use the 2 last layers of ViT-B/14 variant of DINOv2. The experiment was conducted on the test videos from BAIR, utilizing an Euler solver with 50 steps as our ODE solver to ensure precise temporal evolution. We assessed the quality of the generated frames by reporting LPIPS (Zhang et al. 2018) scores (between the generated images and the ground truth frames) and FID (Heusel et al. 2017) scores. More importantly, we utilized FVD (Unterthiner et al. 2018) to quantify the quality of the generated motions and the consistency of the frames. To verify the generalization capability of our conditioning scheme, we evaluated the same model under three distinct settings (results shown in Tab. 3). In the 10% control setting, we encoded future frames using the same DINOv2 model employed during training and conditioned the model on 10% of these features, randomly sampled with $\pi = 0.1$, which mirrors the training setting. To assess the robustness of our model, we also evaluated it with only 1% of the future features (i.e. by setting $\pi = 0.01$, resulting in 1-2 features per frame on average), achieving superior FID and FVD scores compared to other models. Lastly, unlike the previous two settings, which were non-causal (i.e., conditioned on the features of the ground truth future frames), we experimented with a setting where features from the first frame were propagated to subsequent frames using the flow determined by assessing the cosine similarity between future patch features and the first frame's features. We applied a high cosine similarity threshold of 0.95 to ensure the accuracy of the flows and maintain control percentage comparability with the training setting. This setting approximates the use of the model at inference, and the experiment shows that CAGE performs better than the prior work even under this domain gap between the training and the test times.

Even though methods with different controls are technically not comparable, other than for video quality, and different controllability metrics incorporate biases towards favoring certain controls, for completeness, we measure the controllability score proposed in (Davtyan and Favaro 2024) and report it in Tab. 4. This score measures the discrepancy in pixels between the intended control (which in (Davtyan and Favaro 2024) is an optical flow vector that is applied to an object) and the optical flow estimated with a pre-trained optical flow network (RAFT (Teed and Deng 2020) in this case) between the initial and the generated frames. As done in the ablations, we annotate 128 frames from the BAIR dataset with the locations of the robot patches and apply random shifts to the patches to assess the controllability. Following (Davtyan and Favaro 2024), we report the controllability score in 4 different settings: 1) the shift is estimated from the ground truth future frame 2) the shift is generated with a uniform direction and the norm sampled from the ground truth distribution, which is $\mathcal{N}(7.19, 5.12)$ 3) o.o.d. S - the same as 2), but the norm comes from $\mathcal{N}(10, 0.1)$ 4) o.o.d. L - the same as 2), but the norm comes from $\mathcal{N}(20, 0.1)$. CAGE performs slightly worse than (Davtyan

Method	LPIPS↓	PSNR↑	FID↓	FVD↓
YODA (Davtyan and Favaro 2024)	0.436	16.22	152	664
CAGE *(ours)*	**0.283**	**22.25**	**95**	**393**

Table 5: Evaluation of the generated videos on the *EPIC-KITCHENS* dataset.

$w = 0.0$ $w = 0.0$ $w = 7.0$

Figure 4: The effect of CFG on the generalization to out of distribution controls on the BAIR dataset. While the robotic arm can be controlled with no guidance ($w = 0.0$), with larger w the model is also able to move the background objects not moving on their own in the training data. Click on the images to play them as videos in Acrobat Reader.

and Favaro 2024) in this experiment. However, it is important to note that the metric used here is measured in pixels, while our method operates with patches arranged on a fixed grid. To overcome the limitation of grid-aligned patches and capture shifts smaller than the grid step, we propose interpolating the position encodings added to features before using them as controls. This technique is implemented in a zero-shot use of our pre-trained model and demonstrates better controllability compared to the vanilla implementation. Finally, the highest score of CAGE, which is 4.77, is less than half the patch size, which is 16×16. Based on that, we can claim the controllability of our method.

EPIC-KITCHENS. We perform the same reconstruction experiment as before for the EPIC-KITCHENS dataset to show that our model works in even more realistic settings. We use the model configuration from BAIR and autoregressively generated 15 frames from 1 conditioned on controls estimated from the future ground truth frames. We also trained YODA (Davtyan and Favaro 2024) on the same data using the official codebase [2] and evaluated it the same way. The results are in Tab. 5.

Qualitative Results

In this section, we show some qualitative results to visually illustrate various capabilities of CAGE (more, including videos, can be found in the supplementary material as well as on the project's website).

The main contribution of our model is the ability to both compose and animate scenes using a unified control. This is demonstrated in Fig. 1. With the trained model one is able to select object patches from different images and compose them into a novel scene. This is achieved by extracting the DINOv2 features from the selected patches and placing them in the indicated locations to form the control that is fed to the model. Additionally, one may also specify the motion of the objects by moving the patches in the future frames. CAGE carefully adjusts the appearance of the objects to the locations of the features in the control map, while preserving the objects' identities. Fig. 4 depicts some out of distribution controls on BAIR. As previously discussed, using larger w in classifier-free guidance allows us to adapt to o.o.d. control, such as moving background objects in BAIR. Finally, in Fig. 5 we demonstrate the ability of the model to generalize to cross-domain scenarios. We select the features for control from images from a different domain and transfer them to the data domain the model was trained on (e.g. real objects to CLEVRER objects). This is possible because of our specific choice of conditioning the generation on DINOv2 features that were trained on a large open image dataset and hence are abstract and data-agnostic.

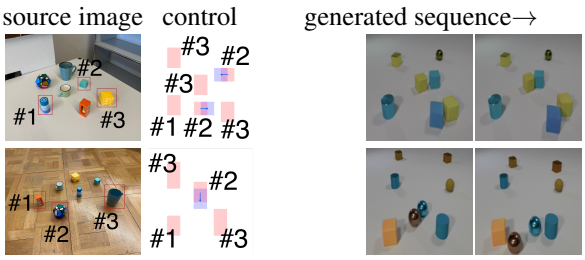

Figure 5: Examples of cross-domain transfer. The features of the objects from images in the first column are borrowed to compose and animate the scenes in the CLEVRER dataset. Notice how CAGE resolves the domain gap and performs a reasonable transfer of the objects (in terms of shapes and colors). However some objects with irregular shape and texture, such as the Rubik's dodecahedron, may turn to multiple objects when transfered. Click on the first images in the generated sequences to play them as videos in Acrobat Reader.

Conclusion and Limitations

We introduced CAGE, an unsupervised method for controllable video generation capable of composing and animating scenes using objects from other frames or out-of-domain images. This is achieved by conditioning the video generation on sparse DINOv2 spatial tokens describing object appearance and spatiotemporal location. We demonstrated our method's capabilities through various experiments. Due to limited computing resources, our experiments were conducted on relatively small datasets, precluding direct comparison with state-of-the-art large-scale video generation models trained on extensive annotated data. However, we believe that the training pipeline, the architecture, and the unsupervised nature of the considered problem enable the scalability of our model and hope that the ideas explored in our work will pave the path towards large-scale unsupervised foundation models for controllable video generation.

[2] https://github.com/araachie/yoda

Acknowledgements

This work was supported by grant 188690 of the Swiss National Science Foundation.

References

Bar-Tal, O.; Chefer, H.; Tov, O.; Herrmann, C.; Paiss, R.; Zada, S.; Ephrat, A.; Hur, J.; Li, Y.; Michaeli, T.; et al. 2024. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*.

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

Blattmann, A.; Milbich, T.; Dorkenwald, M.; and Ommer, B. 2021. iPOKE: Poking a Still Image for Controlled Stochastic Video Synthesis. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 14687–14697.

Bruce, J.; Dennis, M.; Edwards, A.; Parker-Holder, J.; Shi, Y.; Hughes, E.; Lai, M.; Mavalankar, A.; Steigerwald, R.; Apps, C.; et al. 2024. Genie: Generative Interactive Environments. *arXiv preprint arXiv:2402.15391*.

Chen, W.; Wu, J.; Xie, P.; Wu, H.; Li, J.; Xia, X.; Xiao, X.; and Lin, L.-J. 2023. Control-A-Video: Controllable Text-to-Video Generation with Diffusion Models. *ArXiv*, abs/2305.13840.

Damen, D.; Doughty, H.; Farinella, G. M.; Fidler, S.; Furnari, A.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; and Wray, M. 2021. The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(11): 4125–4141.

Davtyan, A.; and Favaro, P. 2022. Controllable Video Generation Through Global and Local Motion Dynamics. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 68–84. Cham: Springer Nature Switzerland. ISBN 978-3-031-19790-1.

Davtyan, A.; and Favaro, P. 2024. Learn the Force We Can: Enabling Sparse Motion Control in Multi-Object Video Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10): 11722–11730.

Davtyan, A.; Sameni, S.; and Favaro, P. 2023. Efficient Video Prediction via Sparsely Conditioned Flow Matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 23263–23274.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ArXiv*, abs/2010.11929.

Ebert, F.; Finn, C.; Lee, A. X.; and Levine, S. 2017. Self-Supervised Visual Planning with Temporal Skip Connections. In *CoRL*.

Epstein, D.; Park, T.; Zhang, R.; Shechtman, E.; and Efros, A. A. 2022. BlobGAN: Spatially Disentangled Scene Representations. *ArXiv*, abs/2205.02837.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12873–12883.

Gao, S.; Yang, J.; Chen, L.; Chitta, K.; Qiu, Y.; Geiger, A.; Zhang, J.; and Li, H. 2024. Vista: A Generalizable Driving World Model with High Fidelity and Versatile Controllability. *arXiv preprint arXiv:2405.17398*.

Guo, H.; Wu, F.; Qin, Y.; Li, R.; Li, K.; and Li, K. 2023a. Recent Trends in Task and Motion Planning for Robotics: A Survey. *ACM Comput. Surv.*, 55(13s).

Guo, Y.; Yang, C.; Rao, A.; Wang, Y.; Qiao, Y.; Lin, D.; and Dai, B. 2023b. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*.

Ha, D.; and Schmidhuber, J. 2018. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems 31*, 2451–2463. Curran Associates, Inc. https://worldmodels.github.io.

Han, L.; Ren, J.; Lee, H.-Y.; Barbieri, F.; Olszewski, K.; Minaee, S.; Metaxas, D.; and Tulyakov, S. 2022. Show Me What and Tell Me How: Video Synthesis via Multimodal Conditioning. *arXiv preprint arXiv:2203.02573*.

Hatamizadeh, A.; Song, J.; Liu, G.; Kautz, J.; and Vahdat, A. 2023. DiffiT: Diffusion Vision Transformers for Image Generation. *ArXiv*, abs/2312.02139.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D. P.; Poole, B.; Norouzi, M.; Fleet, D. J.; et al. 2022a. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022b. Video diffusion models. *arXiv preprint arXiv:2204.03458*.

Hu, A.; Russell, L.; Yeo, H.; Murez, Z.; Fedoseev, G.; Kendall, A.; Shotton, J.; and Corrado, G. 2023. GAIA-1: A Generative World Model for Autonomous Driving. arXiv:2309.17080.

Hu, Y.; Luo, C.; and Chen, Z. 2021. Make It Move: Controllable Image-to-Video Generation with Text Descriptions. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18198–18207.

Huang, H.-P.; Su, Y.-C.; Sun, D.; Jiang, L.; Jia, X.; Zhu, Y.; and Yang, M.-H. 2023. Fine-grained Controllable Video Generation via Object Appearance and Context. *ArXiv*, abs/2312.02919.

Huang, J.; Jin, Y.; Yi, K. M.; and Sigal, L. 2022. Layered Controllable Video Generation. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 546–564. Cham: Springer Nature Switzerland. ISBN 978-3-031-19787-1.

Hudson, D. A.; and Zitnick, C. L. 2021. Compositional Transformers for Scene Generation. In *Neural Information Processing Systems*.

Li, M.; Wan, B.; Moens, M.-F.; and Tuytelaars, T. 2024. Animate Your Motion: Turning Still Images into Dynamic Videos. *arXiv preprint arXiv:2403.10179*.

Lipman, Y.; Chen, R. T.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2022. Flow Matching for Generative Modeling. *arXiv preprint arXiv:2210.02747*.

Ma, W.-D. K.; Lewis, J. P.; and Kleijn, W. 2023. TrailBlazer: Trajectory Control for Diffusion-Based Video Generation. *ArXiv*, abs/2401.00896.

Ma, Y.; Cun, X.; He, Y.-Y.; Qi, C.; Wang, X.; Shan, Y.; Li, X.; and Chen, Q. 2023. MagicStick: Controllable Video Editing via Control Handle Transformations. *ArXiv*, abs/2312.03047.

Menapace, W.; Lathuiliere, S.; Siarohin, A.; Theobalt, C.; Tulyakov, S.; Golyanik, V.; and Ricci, E. 2022. Playable environments: Video manipulation in space and time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3584–3593.

Menapace, W.; Lathuilière, S.; Tulyakov, S.; Siarohin, A.; and Ricci, E. 2021. Playable video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10061–10070.

Menapace, W.; Siarohin, A.; Lathuilière, S.; Achlioptas, P.; Golyanik, V.; Tulyakov, S.; and Ricci, E. 2024. Promptable game models: Text-guided game simulation via masked diffusion models. *ACM Transactions on Graphics*, 43(2): 1–16.

Mendonca, R.; Bahl, S.; and Pathak, D. 2023. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H. Q.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Assran, M.; Ballas, N.; Galuba, W.; Howes, R.; Huang, P.-Y. B.; Li, S.-W.; Misra, I.; Rabbat, M. G.; Sharma, V.; Synnaeve, G.; Xu, H.; Jégou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2023. DINOv2: Learning Robust Visual Features without Supervision. *ArXiv*, abs/2304.07193.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Ruhe, D.; Heek, J.; Salimans, T.; and Hoogeboom, E. 2024. Rolling Diffusion Models. *arXiv preprint arXiv:2402.09470*.

Shi, X.; Huang, Z.; Wang, F.-Y.; Bian, W.; Li, D.; Zhang, Y.; Zhang, M.; Cheung, K. C.; See, S.; Qin, H.; Da, J.; and Li, H. 2024. Motion-I2V: Consistent and Controllable Image-to-Video Generation with Explicit Motion Modeling. *ArXiv*, abs/2401.15977.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 402–419. Springer.

Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2018. Towards Accurate Generative Models of Video: A New Metric & Challenges. *ArXiv*, abs/1812.01717.

Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Neural Information Processing Systems*.

Wang, J.; Zhang, Y.; Zou, J.; Zeng, Y.; Wei, G.; Yuan, L.; and Li, H. 2024. Boximator: Generating Rich and Controllable Motions for Video Synthesis. *ArXiv*, abs/2402.01566.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wu, K.; Peng, H.; Chen, M.; Fu, J.; and Chao, H. 2021. Rethinking and Improving Relative Position Encoding for Vision Transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 10013–10021.

Yang, J.; Ivanovic, B.; Litany, O.; Weng, X.; Kim, S. W.; Li, B.; Che, T.; Xu, D.; Fidler, S.; Pavone, M.; and Wang, Y. 2023. EmerNeRF: Emergent Spatial-Temporal Scene Decomposition via Self-Supervision. *ArXiv*, abs/2311.02077.

Yi, K.; Gan, C.; Li, Y.; Kohli, P.; Wu, J.; Torralba, A.; and Tenenbaum, J. B. 2019. CLEVRER: CoLlision Events for Video REpresentation and Reasoning. *ArXiv*, abs/1910.01442.

Zhang, D. J.; Wu, J. Z.; Liu, J.-W.; Zhao, R.; Ran, L.; Gu, Y.; Gao, D.; and Shou, M. Z. 2023a. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3813–3824.

Zhang, L.; Xiong, Y.; Yang, Z.; Casas, S.; Hu, R.; and Urtasun, R. 2024. Learning Unsupervised World Models for Autonomous Driving via Discrete Diffusion. In *The Twelfth International Conference on Learning Representations*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhang, Y.; Wei, Y.; Jiang, D.; Zhang, X.; Zuo, W.; and Tian, Q. 2023b. ControlVideo: Training-free Controllable Text-to-Video Generation. *ArXiv*, abs/2305.13077.

Zhang, Z.; Liu, R.; Aberman, K.; and Hanocka, R. 2023c. TEDi: Temporally-entangled diffusion for long-term motion synthesis. *arXiv preprint arXiv:2307.15042*.