Boximator: Generating Rich and Controllable Motions for Video Synthesis

(ICML 2025)

2024.12.03

Limitations of text-based motion control (1)

- Imperfect model is not always able to comply to all text prompts

Text : **"Adding wine to a glass."**

ours

Pika

Gen-2

Limitations of text-based motion control (2)

- Position, shape, size, trajectory are **not easy to express** in text

Text : **"A handsome man is taking out a rose from his pocket with his right hand and looking at the rose."**

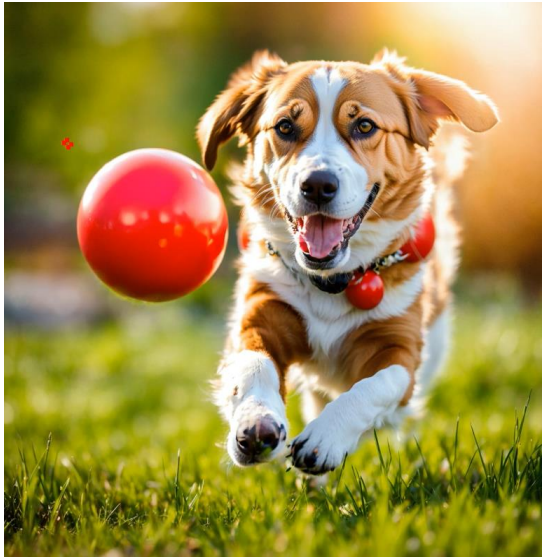
ours

Pika

Gen-2

Suggestion : **Boximator**

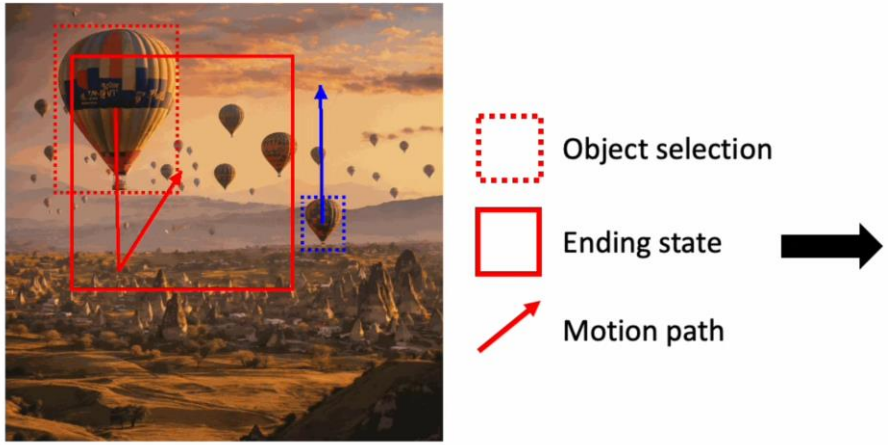
"A girl in red is covering her face with a skull."



"A dog is chasing a red ball."

"The character made of pixels is dancing."

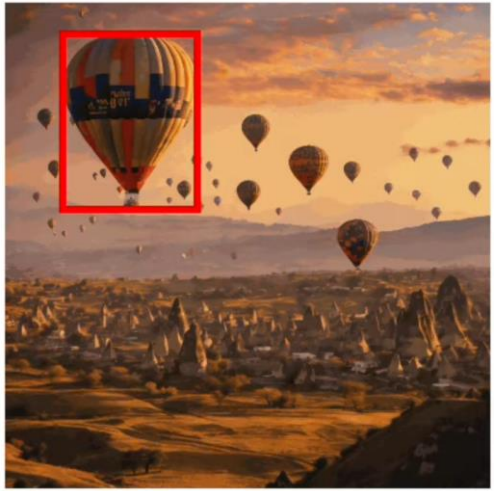
Intro. Boximator : How to move?



https://www.youtube.com/watch?v=reto_TYsYyQL

Intro. Boximator : Two type constraints

Hard box : Object



- For object selection
- For <u>rigorously</u> define final state

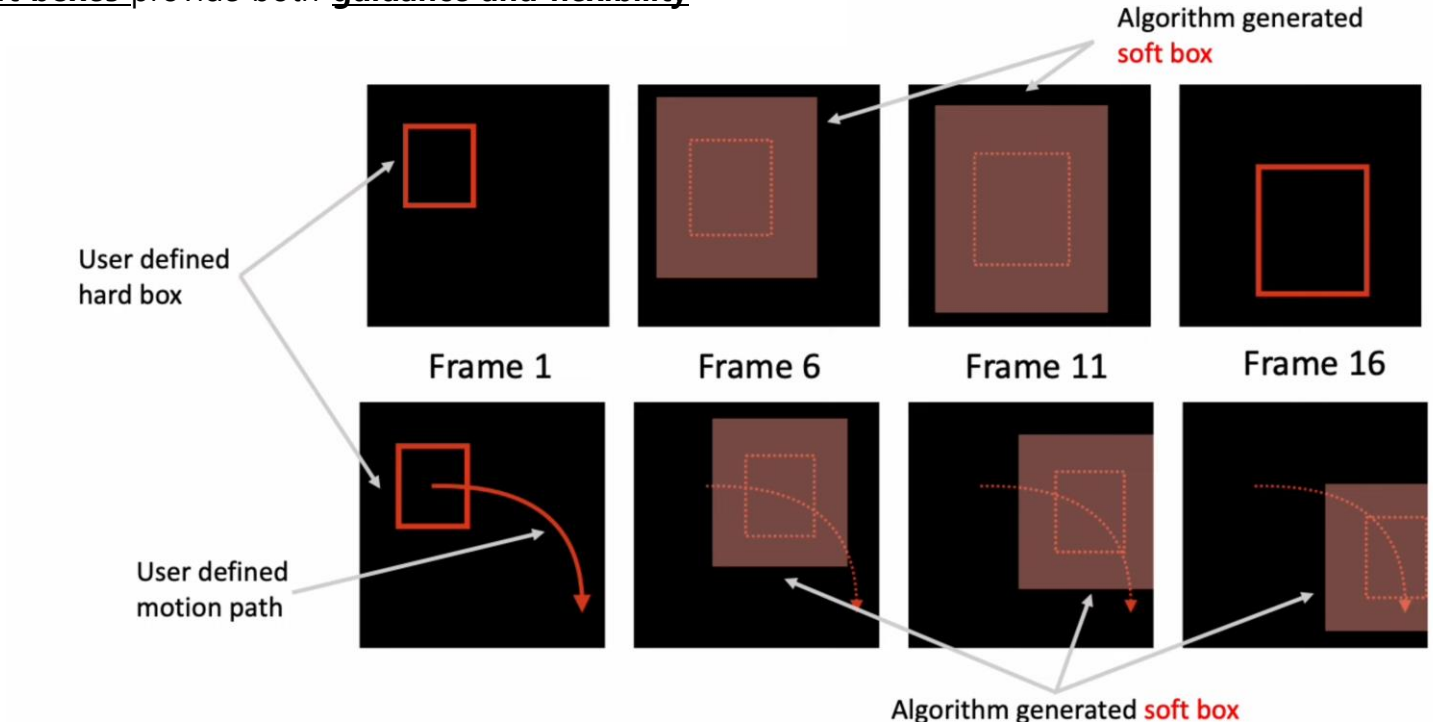
Soft box : Region

- For <u>roughly</u> define object's final state
- For <u>roughly</u> define moving trajectory

https://www.youtube.com/watch?v=reto_TYsYyQL

Inference

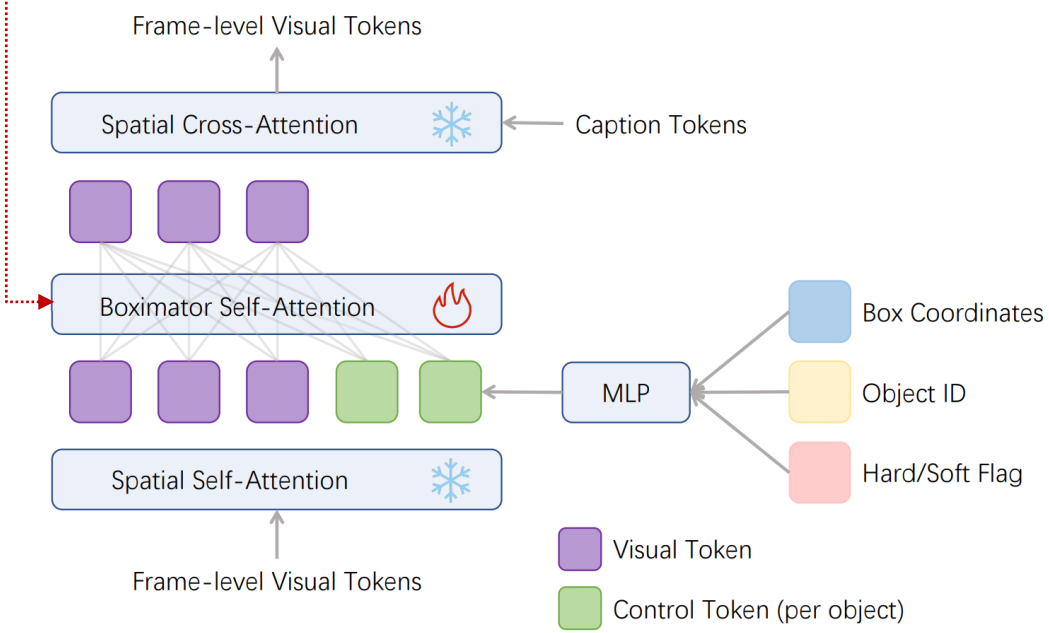
- **User** draw boxes or motion path in first and last frames
- **Algorithm** automatically inserts soft boxes in intermediary frames to achieve better control
- **Soft boxes** provide both **guidance and flexibility**



https://www.youtube.com/watch?v=reto_TYsYyQL

Method : Model architecture

- we **freeze the original model parameters** and solely focus on training the **newly incorporated** **<u>motion control module</u>**



$$v = v + \mathrm{SelfAttn}(v)$$
$$v = v + \mathrm{TS}(\mathrm{SelfAttn}([v, h_{\mathrm{box}}]))$$
$$v = v + \mathrm{CrossAttn}(v, h_{\mathrm{text}})$$

- TS(·) is a <u>token selection operation</u> that exclusively considers visual tokens
- h_{box} is a sequence of <u>control tokens</u>
- Each token represents a box and is defined by:

$$\boldsymbol{t}_b = \mathrm{MLP}(\mathrm{Fourier}([b_{\mathrm{loc}}, b_{\mathrm{id}}, b_{\mathrm{flag}}]))$$

Method : Self-tracking & Multi-Stage training

- A significant **challenge** in video motion control lies in **associating box coordinates with objects** and **maintaining temporal consistency** across frames

self-tracking

(1) generating a bounding box for each object with the right color
(2) aligning these boxes with the Boximator constraints in every frame

Figure 4. Self-tracking: train the model to track every constrained object. This figure shows 3 frames where the black horse and the yellow box surrounding it are generated together.

Multi-stage training

(1) Stage 1, the model is **trained** using all the provided ground truth bounding boxes as **hard box constraints**
(2) Stage 2, we substitute 80% of these **hard boxes with soft boxes**
(3) Stage 3, we continue the Stage 2 **training** but **without self-tracking**

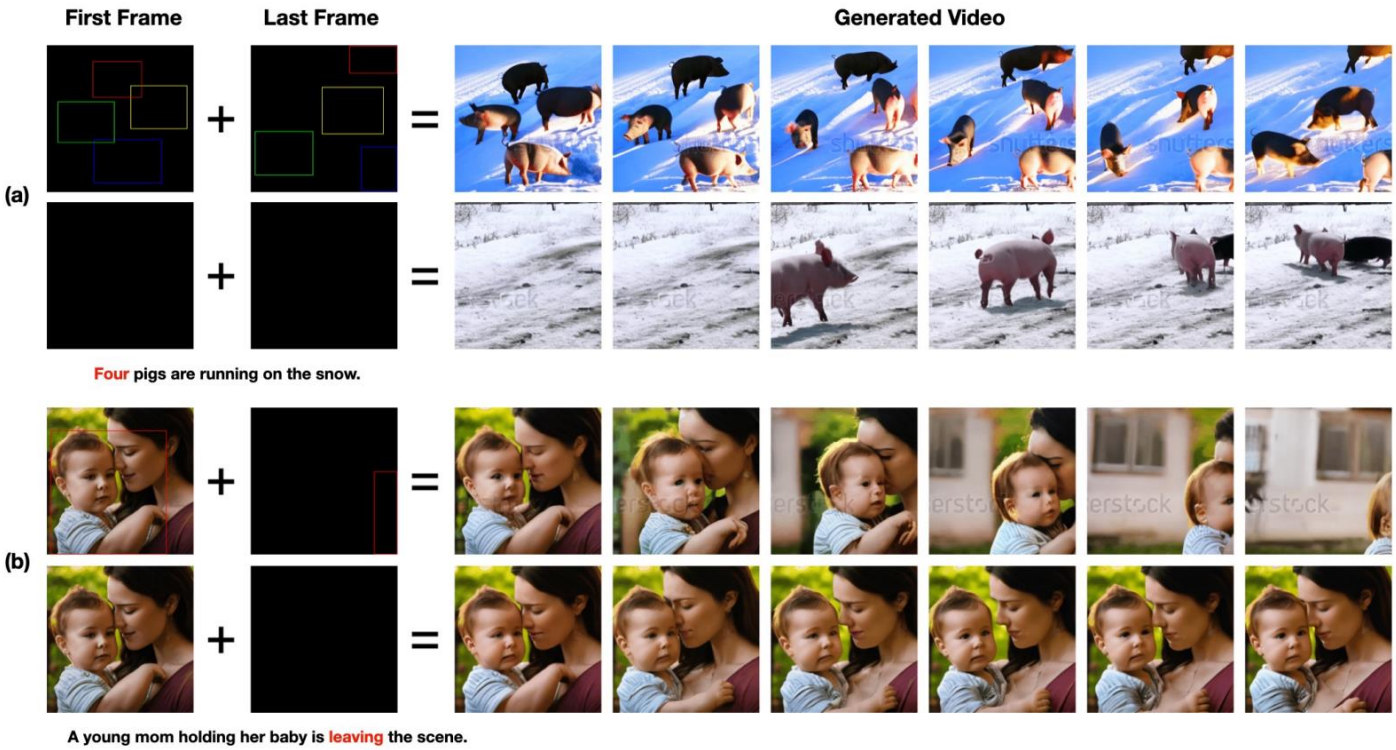
→ Although these boxes are no longer visually present, their internal representation persists, enabling the model to continue aligning with Boximator constraints.

Experiments : quantitative measurement

- **Fre´chet Video Distance (FVD** : evaluate the quality and temporal motion dynamics of generated videos
- **CLIP** : similarity between text and generated video frames
- **AP** : alignment accuracy between the objects generated by the video generation model and the bounding boxes

Models	Extra Input	FVD(↓)	CLIPSIM(↑)	mAP/AP_{50}/AP_{75}(↑)
MagicVideo [43]	-	1290	-	-
LVDM [12]	-	742	0.2381	-
ModelScope [31]	-	550	0.2930	-
Show-1 [42]	-	538	0.3072	-
PixelDance [41]	-	381	**0.3125**	-
Phenaki [30]	-	384	0.2870	-
FACTOR-traj [15]	Box	317	0.2787	0.290*/-/-
PixelDance + Boximator	-	**237**	0.3039	0.094/0.193/0.076
	Box	174	0.2947	0.349/0.479/0.359
	F0	113	0.2890	0.194/0.330/0.177
	F0 + Box	102	0.2874	0.365/0.521/0.384
ModelScope + Boximator	-	239	0.3013	0.096/0.195/0.084
	Box	216	0.2948	0.312/0.470/0.309
	F0	142	0.2865	0.141/0.260/0.126
	F0 + Box	132	0.2852	0.300/0.456/0.299

Experiments : qualitative measurement



Thank you

Q&A

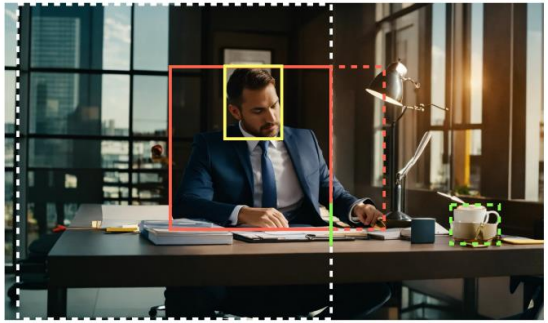


Figure 3. Training data: all bounding boxes are projected to the cropped region (white dashed box).