# BAgger: Backwards Aggregation for Mitigating Drift in Autoregressive Video Diffusion Models

Supplementary Material

## A. Video Results

**Supplementary Website.** Because static figures in the main manuscript cannot fully convey our video results, we provide corresponding videos for each main result on the supplementary website. These can be accessed via the attached HTML file. Due to the 200MB submission size limit and the length of our videos, many clips have been compressed and may appear lower quality than the original generations.

## B. Implementation Details

### B.1. Training Configurations

**Model.** We primarily build our method upon the Wan2.1 [63] and Self Forcing [30] codebases. We primarily work with the Wan2.1 1.3B Text-to-Video model as our base model, fine-tuning it into a block-causal model using FlexAttention [16]. As Wan2.1 1.3B supports bidirectional generation of 21 latent frames, we follow prior works [30, 76] and use a chunk size of 3 frames per chunk, resulting in 7 total chunks. This configuration provides a reasonable balance between parallelism and causal granularity. We apply the same chunking and causal masking scheme to all autoregressive baselines to ensure a fair architectural comparison.

**Data.** We use a subset of the Pexels video dataset [1] for causal fine-tuning and generation of corrective trajectories. Specifically, we use the Pexels subset from the MiraData [34] dataset, and use the "dense prompt" field as the corresponding text for each corresponding video clip. Due to varying FPS and clip lengths in the dataset, we subsample videos that have higher than 16FPS to match the frame rate of our video model (16FPS) and randomly sample a 5 seconds clip from each data sample. For generation of corrective trajectories, we take the initial 9 frames = 3 latent frames = 1 chunk, from a chosen video clip and perform rollouts conditioned on an initial ground truth chunk and the corresponding text prompt.

**Compute and Hyperparameters.** We perform all of our experiments on 16 NVIDIA H100 GPUs (80GB VRAM per GPU). For each round of training, we perform 16,000 training iterations at a batch size of 96, using a constant learning rate of $4e^{-6}$. We follow the original Wan2.1 flow matching [41] objective, and adopt a time step shift in the form, $t'(k,t) = (kt/1000)/(1+(k-1)(t/1000)) \cdot 1000$ and a shift factor $k = 8$.

| Model | Smooth. | Dynamic | Aesthetic | Imaging |
|---|---|---|---|---|
| MAGI-1 [3] | 99.35 | 46.88 | 55.69 | 58.43 |
| SkyReels-V2 [13] | 99.20 | 44.53 | 55.60 | 57.84 |
| Self Forcing [30] | 98.64 | 49.22 | 58.33 | 71.29 |
| Ours | 98.61 | 76.56 | 55.35 | 63.41 |

Table 3. Extended results comparing our method against relevant open baseline models. Our model shows much better motion dynamics while maintaining simialr levels of motion smoothness. For per-frame quality metrics, on average, our method is only second to Self Forcing which is distilled from a 14B base model. Our method outperforms other AR diffusion models (MAGI-1 and SkyReels) in imaging quality, and has similar aesthetic performance, despite being fine-tuned on a much smaller dataset.

### B.2. Inference Configurations

We perform inference for all multi-step AR diffusion models (ours and baselines) at 20 inference steps per chunk. For our model, we perform inference with a classifier-free guidance scale of $w = 6$, and a time step shift coefficient of $k = 8$. For corrective trajectory generations, the same inference configurations are used. For long video generations with sliding windows, we perform KV-cache reinitialization at every sliding window step to eliminate any potential issues caused by OOD KV-cache values as mentioned in Xun et al. [30]. Each sliding window instance, we keep 12 latent frames from the tail end of the previous generation as the initial context of the next set of generations. For multi-prompt generations, we simply change the text-prompt conditioning during the sliding window change.

## C. Evaluations

### C.1. Baseline Details

**History Guidance.** We implement a version of History Guidance resembling Song et al. [57], performing joint guidance between the text-conditioned prediction and fractional history guidance as defined in [57]. The resulting velocity prediction used takes the form,

$$\begin{aligned} v_{\text{hg}}(x_t^i, t, c) = v(x^{<i}, t, \emptyset) \\ + w_{\text{text}}\big[v(x^{<i}, t, c) - v(x^{<i}, t, \emptyset)\big] \\ + w_{\text{hg}}\big[v(x_p^{<i}, t, \emptyset) - v(\emptyset^{<i}, t, \emptyset)\big]. \end{aligned} \quad (4)$$

Where $w_{\text{text}}$, and $w_{\text{hg}}$ are the guidance scale for the text conditioned and history conditioned terms, $p = 800$ is a partial noise level chosen for fractional history guidance, $c$ is the

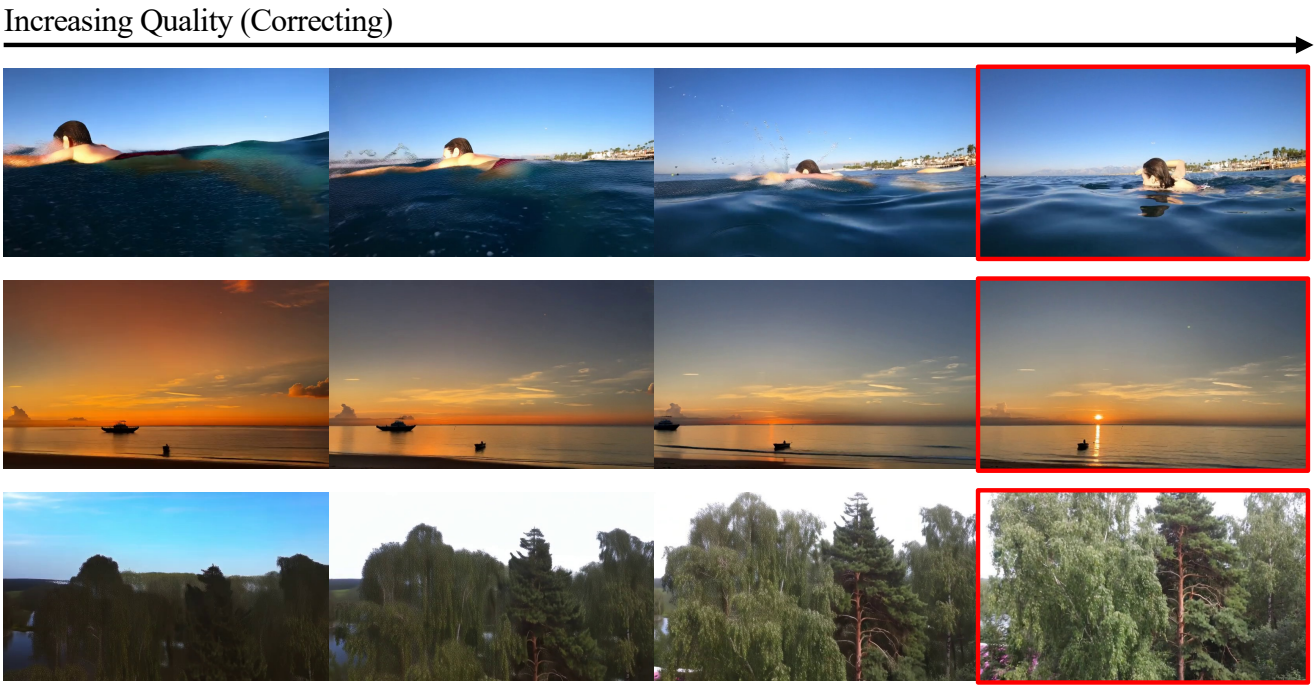


Figure 7. More examples of corrective trajectories gathered from first round of BAgger. Ground truth frames are highlighted in red. Note that the above sequences show generated frames in reverse order, therefore the order above is the order in which the model is trained on.

text prompt, and $\emptyset$ represents the empty string, and $\emptyset^{<i}$ randomly sampled noise in the same shape as the context. In our evaluations we use $w_{\text{text}} = 6$ and $w_{\text{hg}} = 1.2$.

### C.2. Evaluations

**Metric Details.** We base our evaluations on the custom evaluation suite of VBench [31, 32, 81], which spans the 6 dimensions. As mentioned for Tab. 1, we also report drifting metrics which measures the drop in per-frame quality metrics between the first 20% of all frames and the average across all frames. We also report the full metrics for Tab. 2 in Tab. 3. Note that the results in Tab. 2 were calculated by taking the averages between the two motion metrics, and two frame-wise quality metrics respectively.

## D. Additional Results

**Rollout Examples.** We provide additional examples of corrective/drifting trajectories generated during the BAgger algorithm. Our model takes in a set of ground truth frames and generated a set of (potentially) drifted extensions of these frames by rolling out the AR model trained from the previous round.

**Comparison with Self-Forcing (1.3B Teacher).** We provide additional comparisons between our method and Self-Forcing trained with a 1.3B teacher in Tab. 4.

| Method | Global Metrics ↑ | | | | | | Drift ↓ | |
|---|---|---|---|---|---|---|---|---|
| | Sub. | Bkg. | Sm. | Dyn. | Aes. | Img. | $\Delta^{\text{Aes}}$ | $\Delta^{\text{Img}}$ |
| Self-Forcing (1.3B) | 83.82 | 88.50 | 98.43 | 63.28 | **56.53** | 62.76 | 5.40 | 5.41 |
| BAgger (1.3B) | **84.05** | **89.58** | **98.61** | **76.56** | 55.35 | **63.41** | **3.29** | **3.57** |

Table 4. **BAgger vs Self-Forcing (1.3B teacher) on 50s videos**

## E. Ethical Considerations.

Video generative models inherently carry substantial potential for misuse. High-fidelity outputs can enable convincing deepfakes that are increasingly difficult to distinguish from real footage, posing risks for misinformation, manipulation, and privacy. Without careful governance and responsible deployment, these models may also amplify harmful societal biases present in their training data. At the same time, they offer meaningful benefits, such as powering world models for robotics, enhancing accessibility, and supporting safer real-world decision-making. We acknowledge this duality and emphasize the importance of continued research, policy development, and safeguards that mitigate negative impacts while enabling socially beneficial uses.