

Supplementary Material for FB-4D: Spatial-Temporal Coherent Dynamic 3D Content Generation with Feature Banks

Anonymous WACV Algorithms Track submission

Paper ID 1229

001 1. Project Link

002 <https://anonymous.4open.science/r/FB-4D-C766>

003 2. More Experiments

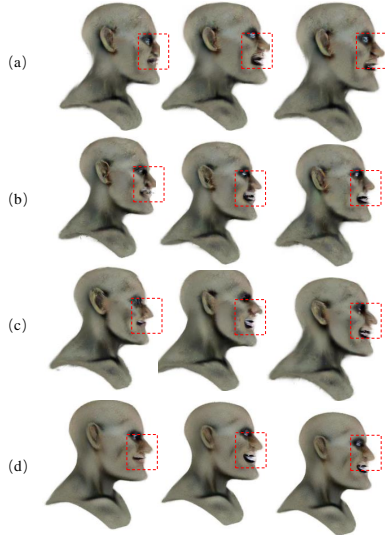


Figure 1. **Ablation study validating the feature bank.** (a) No feature bank is used. (b) Only the key-value feature bank is utilized. (c) Only the output feature bank is used. (d) Our full implementation with both key-value and output feature banks. (see supplementary videos for better comparisons)

004 Ablation study on the utilization of our feature bank

005 To better visualize the role of the key-value feature bank
006 and the output feature bank, we conducted a comprehensive
007 controlled experiment, as shown in Fig. 1. Our full imple-
008 mentation, which incorporates both the key-value and output
009 feature banks, achieves the best consistency.

010 Ablation study on the feature bank updating method.

011 We conducted a detailed comparison of different feature bank
012 updating methods. As shown in Figure 2 (a), using a queue
013 with a window size of 1 for updates may result in insufficient
014 utilization of past information, leading to inconsistencies in

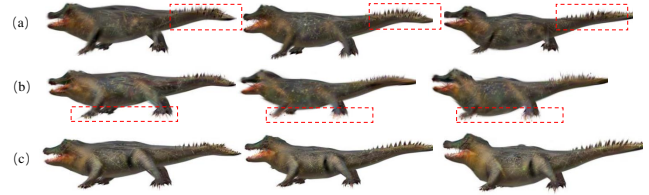


Figure 2. **Ablation study on the feature bank updating method.** (a) Updating the feature bank using a queue with a length of 1. (b) Updating the feature bank using a queue with a length of 2. (c) Dynamically updating the feature bank (our approach).

local details over time. In contrast, Figure 2 (b) demonstrates
that simply increasing the window size can lead to suboptimal
fusion of information, where excessive and redundant data
introduce confusion, ultimately degrading the quality of the
output. Our proposed method, illustrated in Figure 2 (c),
effectively integrates past information, enhancing the amount
of useful information and thereby improving the overall quality
of the generated results.

Ablation study on using the feature bank at different network blocks.

We examine the impact of incorporating a feature bank at
different network blocks. As shown in Figure 3, using feature
bank across all blocks (e) yields the best performance with
minimal artifacts, demonstrating its effectiveness in preserving
essential features. Applying it to specific blocks—downsam-
pling (b), middle (c), or upper (d)—still provides benefits
but to a lesser extent. Without feature bank (a), the model
struggles to retain rich information, leading to weaker repre-
sentations and more artifacts.

Ablation study on the progressively iterations and our usage of the feature bank during multiple iterations.

We analyze the impact of progressive iterations and feature
bank interactions using four configurations. As shown in Figure
5, randomly selecting viewpoints without feature bank inter-
action (a) leads to the poorest performance due to a lack of
accumulated information. Feature bank interaction alone (b)
improves quality but lacks progressive refinement. Progress-
ive iterations (c) enhance stability but underutilize historical

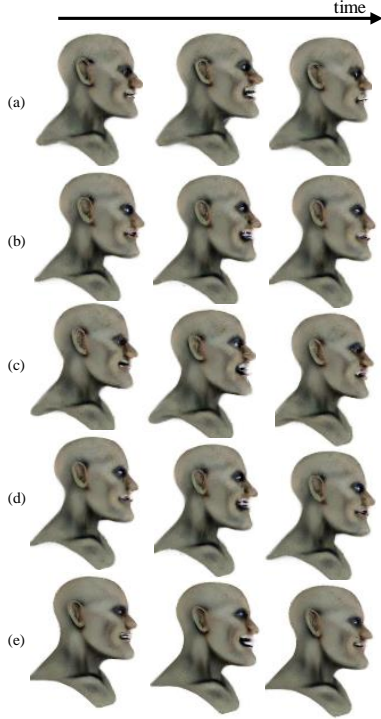


Figure 3. Ablation study on using the feature bank at different network blocks. (a) Without using the feature bank. (b) Using the feature bank at downsampling blocks. (c) Using the feature bank at middle blocks. (d) Using the feature bank at upper blocks. (e) Using the feature bank at all blocks.

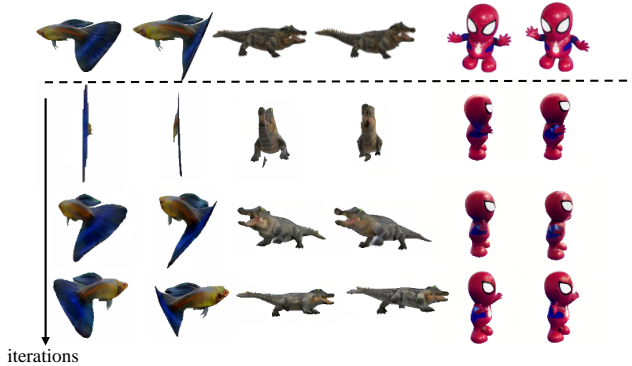


Figure 4. Ablation study on the number of iterations

information. Our method (d), combining both, achieves the best performance by balancing temporal consistency and information integration.

Study on the number of iterations. To better illustrate the benefits of multiple iterations, we visualize the multi-view sequences outputted by stage 1 in the pipeline. As shown in the figure 4, during the first iteration, information from side and other views is noticeably incomplete. However, as the number of iterations increases, we progressively

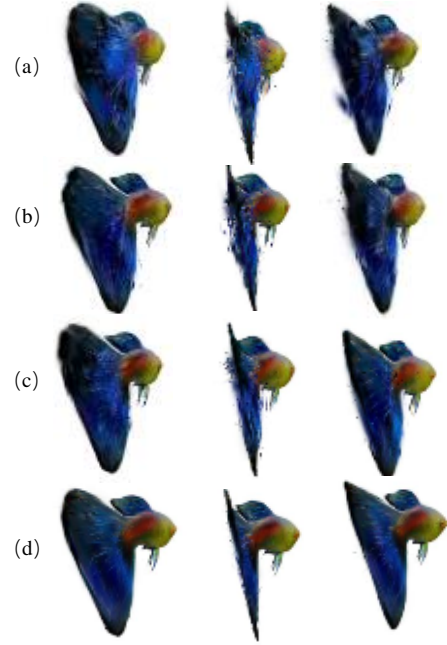


Figure 5. Ablation study on the progressively iterations and our usage of feature bank during multiple iterations. (a) Randomly selecting viewpoints without feature bank interaction across multiple iterations. (b) Randomly selecting viewpoints with feature bank interaction across multiple iterations. (c) Progressively iterating without feature bank interaction across multiple iterations. (d) Our proposed approach, which incorporates both.

generate additional views that are similar to the missing viewpoints, effectively compensating for the lack of information. Furthermore, with each iteration, the generated images maintain improved spatial consistency compared to previous iterations.

3. Computational cost

To optimize memory, we alternate tensor operations between CPU and GPU, maintaining 14GB usage—slightly higher than STAG4D’s 10GB [1], but enabling better historical information retention. The tradeoff is efficiency: 75-step inference takes 60s compared to STAG4D’s 10s. Incorporating CLIP similarity adds about 15 minutes per iteration, resulting in 150 minutes for three iterations over 32 frames. With the sparse CLIP guidance strategy, this is reduced to around 110 minutes. For longer sequences, GPU memory remains stable, while runtime scales approximately linearly with the number of frames due to the frame-by-frame generation process.

References

- [1] Yifei Zeng, Yanqin Jiang, Siyu Zhu, Yuanxun Lu, Youtian Lin, Hao Zhu, Weiming Hu, Xun Cao, and Yao Yao. Stag4d:

072 Spatial-temporal anchored generative 4d gaussians. In *Eu-*
073 *ropean Conference on Computer Vision*, pages 163–179.
074 Springer, 2025. 2