On the Resilience of Text-to-Video Diffusion Models to Hardware Faults

Zachary Coalson [1] **A M Aahad** [1] **Stella Doehring** [1] **Zane Ma** [1] **Sanghyun Hong** [1]

Abstract

We present the first systematic study of the resilience of text-to-video (T2V) diffusion models under random hardware-level faults. While T2V models are widely used for automated video generation due to their ability to produce high-quality, temporally coherent, and realistic videos, their iterative denoising process and spatiotemporal dependencies introduce unique failure modes. We perform an extensive fault-injection study covering both computational and memory faults across three T2V models and a representative benchmark. Our results show that (1) a single fault can degrade overall performance by up to 3.7%, with semantic correctness more affected than perceptual quality; (2) memory faults are more damaging than computational faults, high-order exponent bits are particularly vulnerable, and the widely-used bfloat16 is more susceptible than alternative formats; and (3) 7–28% of faults cause visible artifacts, including semantic changes such as added objects, suggesting that single faults are sufficient to alter output semantics. Our findings reveal reliability risks in deployed T2V systems and motivate further research on improving fault resilience. Code: https://github.com/ztcoalson/T2V-Resilience.

1 Introduction

Text-to-video (T2V) diffusion models (Zheng et al., 2024; Ma et al., 2025; Yang et al., 2025) generate high-quality, coherent videos from text prompts, enabling diverse applications such as physical-world simulation (Brooks et al., 2024), educational content generation (Yang et al., 2024), automated animation creation (Guo et al., 2024; He et al., 2023), and even movie production (Zhu et al., 2023). At the core of these models is a latent Diffusion Transformer (DiT) backbone (Peebles & Xie, 2023), which uses billions of learned parameters to iteratively denoise latent representations into prompt-aligned videos. This process incurs substantial compute and memory overhead, driving deployment onto GPU infrastructure as models continue to scale.

Large-scale GPU infrastructure, however, is susceptible to *random bitwise faults* (Oles et al., 2024; Zhu et al., 2025; Tiwari et al., 2015): system-level soft errors caused by external factors such as radiation (Jain et al., 2022) and electromagnetic interference (Mutlu, 2015; Lin et al., 2025). Although error-correcting codes (ECC) can correct single-bit memory errors, multi-bit faults and computational errors often remain uncorrected (Oles et al., 2024) and can propagate into the weights or activations of T2V models during inference.

Despite these concerns, the resilience of T2V diffusion models to random bitwise faults remains unexplored. Prior fault resilience studies have focused primarily on classification networks (Reagen et al., 2018; Li et al., 2017; Roquet et al., 2024), with more recent work extending to large language models (Agarwal et al., 2023; Sun et al., 2025b). However, these findings do not directly transfer to T2V: the iterative, diffusion-based inference process, spatiotemporal output dependencies, and absence of discrete correctness criteria all distinguish them from previously studied architectures.

Contributions. In this work, we present the first systematic study of the resilience of T2V diffusion models to random bitwise hardware-level faults. T2V diffusion models are particularly vulnerable to such faults due to three factors: (1) iterative reuse of backbone parameters, which can amplify errors across denoising steps; (2) spatiotemporal dependencies across frames (Ma et al., 2025), where small perturbations may cause motion inconsistencies or degraded coherence; and (3) the lack of a well-defined reference for output quality, making fault impacts difficult to quantify.

We design a fault-injection framework that simulates memory and computational faults within the DiT backbone, and evaluate their impact on generation quality, temporal consistency, and semantic alignment using 16 individual and 3 aggregate metrics from a comprehensive T2V benchmark. We apply this framework to three representative T2V diffusion models, injecting more than 300,000 faults across 550 prompts. Our analysis yields the following key findings:

- Random bitwise faults cause a ∼0.3–3.7% drop in overall performance, while semantic correctness degrades up to

[1]Oregon State University, Corvallis, OR, USA. Correspondence to: Zachary Coalson <coalsonz@oregonstate.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

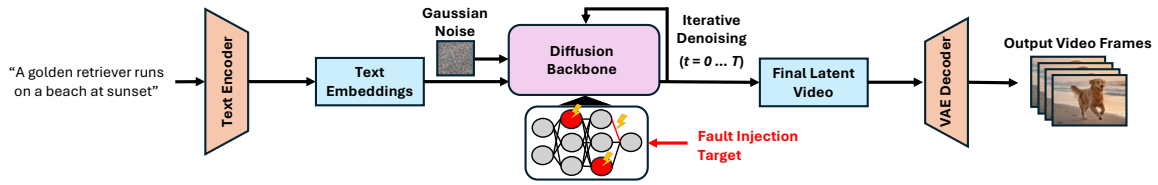

Figure 1. **High-level architecture of a text-to-video diffusion model.** The DiT backbone operates in latent space and is reused across denoising steps. In our study, we assess resilience by injecting random bitwise faults into the backbone's *weights* and *activations*.

6.6× more than visual quality.

- 2-bit memory faults are 2.4–8.6× more impactful than 1-bit computational faults, due to their persistent impact.
- The most significant exponent bit is disproportionately vulnerable, while sensitivity is largely uniform across Transformer blocks and layer types.
- The widely used bfloat16 format is more susceptible to faults than alternative numerical representations.
- 7–28% of faults produce visible semantic alterations or severe distortions, suggesting that individual faults can substantially alter output semantics.

2 Background and Related Work

2.1 Text-to-Video Diffusion Models

Text-to-video (T2V) diffusion models extend image diffusion techniques (Saharia et al., 2022; Zhang et al., 2024) to the spatiotemporal domain, generating coherent video sequences through an iterative denoising process. These models build on denoising diffusion probabilistic models (Ho et al., 2020) and the latent diffusion framework of Rombach et al. (2022), improving efficiency by operating in a compressed latent space rather than pixel space. Recent architectures implement diffusion backbones with Transformers, using their latent representations to improve the scalability and quality of video synthesis (Zheng et al., 2024; Ma et al., 2025; Yang et al., 2025; Wan et al., 2025).

As shown in Figure 1, a typical T2V architecture consists of three primary components: (1) a text encoder that produces semantic embeddings of the input prompt, (2) a spatiotemporal Diffusion Transformer (DiT) backbone that denoises latent representations conditioned on these embeddings, and (3) a variational autoencoder (VAE) decoder that maps the final representation back to pixel space. During inference, the DiT iteratively denoises a sampled latent over T steps, reusing the same backbone weights at each step, before the VAE decodes the result into frames. This weight reuse makes the DiT the central component governing generation and our primary target for fault-injection analysis.

2.2 Fault Resilience of Machine Learning Models

Fault-resilience studies investigate how machine learning models behave under system-level soft errors which manifest as bit-flips that corrupt weights or activations. Early work evaluated the impact of random bitwise faults on classification networks, primarily measuring degradation of top-1 accuracy (Ibrahim et al., 2020; Li et al., 2017; Hong et al., 2019; Reagen et al., 2018). Recent studies have extended resilience analysis to Transformer architectures, including image classification (Roquet et al., 2024) and general language modeling (Agarwal et al., 2023). Most notably, Sun et al. (2025b) conduct a resilience study of large language model inference, evaluating both memory faults and computational faults across multiple tasks and models. However, these findings do not transfer to T2V due to the architectural differences noted in §1; to our knowledge, no prior work has evaluated the fault resilience of T2V diffusion models.

3 Our Fault Injection Framework

3.1 Fault Models

We study fault models characterizing the origin, target operations, and severity (number of corrupted bits) of hardware faults. Following recent work (Sun et al., 2025b), we consider two fault sources: (1) GPU memory faults that corrupt model *weights* and (2) computational faults in hardware components (e.g., ALUs) that corrupt *activations*. For memory faults, we assume a 2-bit fault model—the smallest fault class that escapes SECDED (Single Error Correction, Double Error Detection) correction in modern GPUs (Oles et al., 2024; Zhu et al., 2025; NVIDIA, 2023)—with both flips occurring in the same weight to remain agnostic to ECC implementations. For computational faults, we assume a 1-bit fault model, as single-bit faults are widely studied (Sangchoolie et al., 2017) and are not corrected by hardware.

3.2 Fault Injection Methodology

Fault injection setting. Following prior work (Sun et al., 2025b; Agarwal et al., 2023), we assume that a T2V diffusion model experiences faults at inference time (rather than during training) to quantify the expected impact on downstream users once the model is deployed. We further assume that a fault occurs *once* per inference (i.e., per video generation), since multiple faults within a single inference window are extremely unlikely; this assumption is consistent with prior machine learning resilience studies (Sun et al., 2025b; Agarwal et al., 2023; Reagen et al., 2018; Li et al., 2017). We inject faults only into the DiT's generation process, ex-

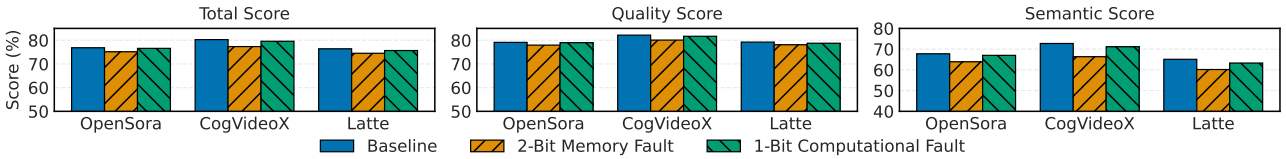

Figure 2. **Impact of hardware faults on overall video quality and semantics.** Average Total, Quality, and Semantic Scores over 100 trials for the fault-free (baseline), 2-bit memory fault, and 1-bit computational fault models. 95% CIs are within ±0.23 across all bars.

cluding the text encoder and VAE, to isolate effects on the latent video generation process. Finally, we assume models are deployed using bfloat16 (Kalamkar et al., 2019), the standard format for modern model deployments (HuggingFace; Wang & Kanwar, 2019); we test alternatives in §4.4.

Fault injection procedure. Following prior work (Sun et al., 2025b), we emulate hardware faults by altering values in PyTorch tensors prior to inference. For *memory faults*, we uniformly sample a DiT weight (by layer and tensor index), uniformly sample two bit positions in [0, 15], and flip the corresponding bits; we then generate one video and revert the weight. For *computational faults*, we sample a layer weighted by parameter count (reflecting fault likelihood), then uniformly sample a diffusion timestep, an output neuron, and one bit position in [0, 15]. A PyTorch hook flips the selected bit in the layer's output tensor at the specified timestep; the hook is removed after generating one video.

3.3 Models and Benchmark

Models. We consider three representative T2V models: (1) OpenSora (Zheng et al., 2024), (2) CogVideoX-2B (Yang et al., 2025), and (3) Latte (Ma et al., 2025). All models employ variants of the T5 text encoder (Raffel et al., 2020), along with model-specific VAEs and DiTs, whose backbone sizes range from 1B–2B parameters. These models have been widely adopted in recent T2V studies (Sun et al., 2025a; Adnan et al., 2025; Guan et al., 2025; Di et al., 2025; Gao et al., 2025), making them well-suited for our analysis.

Benchmark. We use VBench (Huang et al., 2024), a comprehensive benchmark for assessing generated video quality. VBench provides both an evaluation framework with the metrics described below and a set of prompts spanning 11 evaluation categories (e.g., human action and color). Following recent work (Adnan et al., 2025), we sample 50 prompts per category, yielding 550 evaluation prompts in total.

Metrics. We evaluate generated videos using 16 metrics from VBench (Huang et al., 2024), spanning visual quality and semantic fidelity. We focus on VBench's three aggregate scores: a *Quality Score* (weighted average of 7 metrics, e.g., subject consistency and aesthetic quality), a *Semantic Score* (weighted average of 9 metrics, e.g., object class, human action, and spatial relationship), and a *Total Score* that combines both. Full metric definitions are in Appendix D.

4 Empirical Evaluation

4.1 Experimental Setup

Models and video generation. We implement all models using VideoSys (VideoSys Team, 2024). Following recent work (Adnan et al., 2025), all generated videos are 2 seconds long. We use each model's default generation settings.

Metric computation. We conduct 100 trials for each fault model. In each trial, a random fault is sampled per video, resulting in *55,000* generations per fault model. We then compute VBench metrics per trial and report averages across trials to capture the expected impact of each fault model.

4.2 Main Resilience Results

We present results for the three aggregate metrics—Total, Quality, and Semantic Score—in Figure 2; results for the 16 individual metrics are reported in Appendix B. Across fault models and T2V models, the Total Score decreases by 0.3–3.7%, indicating a modest performance degradation. Performance drops are similar across T2V models (within ∼1%), suggesting that resilience is largely determined by the DiT backbone rather than model-specific differences. Examining individual components of performance, we find that video semantics are affected up to 6.6× more than quality: the Semantic Score drops by 1.1–8.8%, whereas the Quality Score drops by only 0.2–2.5%. This indicates that bit-flips are more likely to alter the attributes and relationships of entities in videos rather than their visual rendering, which may lower the barrier to exploitation via targeted *bit-flip attacks* (Rakin et al., 2019; Coalson et al., 2025).

Consistent with Sun et al. (2025b), 2-bit memory faults are 2.4–8.6× more impactful than 1-bit computational faults (1.4–8.8% vs. 0.2–2.8% drops). This gap likely arises because memory faults persist across all denoising steps, whereas computational faults affect a single operation; weight faults also influence many more downstream computations through matrix multiplication (Sun et al., 2025b).

4.3 Resilience Analysis

Model-level components. We analyze the sensitivity of specific model components under the 2-bit memory fault model: bit positions, Transformer block indices, and layer types. For each fault, we record the percentage change

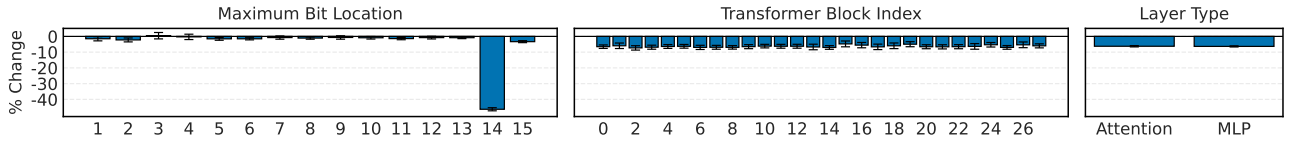

Figure 3. **Resilience of model-level components to 2-bit memory faults** for Latte. The average change across individual VBench metrics, categorized by maximum bit location, Transformer block index, and layer type.

across all 16 individual metrics, averaged by component; for bit location, we group by the *maximum* bit position. Figure 3 shows results for Latte; full results are in Appendix C.

Consistent with prior work (Sun et al., 2025b; Agarwal et al., 2023), we find that the most significant exponent bit (bit 14) is disproportionately vulnerable, causing an average performance drop of 46.3%, compared to at most 3.4% for all other bit positions. In contrast, vulnerability is largely uniform across other components: performance drops range from 4.8–7.3% across Transformer blocks and are nearly identical between MLP (6.4%) and attention (6.3%) layers. The uniformity across blocks is consistent with findings on other Transformer architectures (Agarwal et al., 2023), though it contrasts with earlier CNN studies that found earlier layers more vulnerable (Li et al., 2017).

Diffusion timestep. We next measure the resilience of each diffusion timestep, i.e., the step at which the fault is injected. As weight faults are always injected prior to inference, we instead consider the 1-bit computational fault model. For each time step (normalized to $[0, 1]$ for consistency across models), we aggregate all faults injected during that step and record the average percentage change in all individual metrics. Results for all three models are shown in Figure 4.

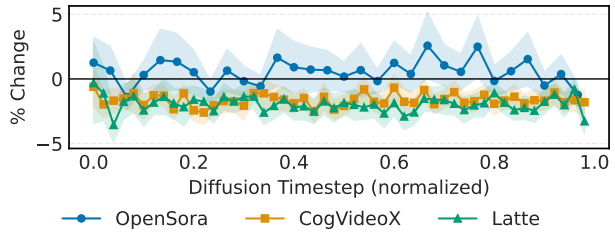

Figure 4. **Impact of computational faults across diffusion timesteps**. The average change in individual VBench metrics for each (normalized) diffusion step at which faults were injected.

Computational faults appear to have a relatively uniform impact across diffusion timesteps, with metric changes ranging from -3.6% to +2.6% and no clear trend as timesteps increase. While we may expect faults at earlier timesteps to matter more, since the perturbed latent passes through more of the remaining denoising process and thus has a greater opportunity to be amplified, we observe no clear dependence on timestep. Taken together with the bit-position results above, this suggests that fault magnitude is the primary driver of degradation, rather than its location or timing.

4.4 Impact of Numerical Representation

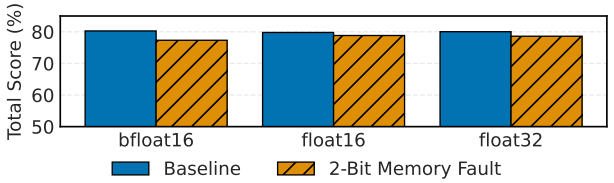

Figure 5. **Impact of numerical representation** for CogVideoX. The average Total Score under the fault-free (baseline) and 2-bit memory fault models. 95% CIs are within ±0.08 for all bars.

We examine how numerical representation affects resilience by comparing bfloat16, float16, and float32 under the 2-bit memory fault model on CogVideoX; for float32, we sample bits in [0, 31]. Results are shown in Figure 5. We find that bfloat16 is the most vulnerable format, with its Total Score dropping by 3.1%, compared to 1.8% for float32 and 1.2% for float16. This is likely because bfloat16 allocates the most bits to the exponent (8/16) while supporting the same range as float32, making large perturbations more probable.

4.5 Qualitative Analysis

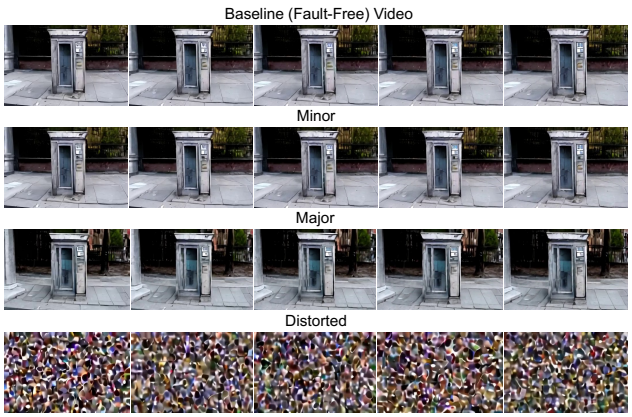

Figure 6. **Example of qualitative fault outcomes for representative videos** generated from OpenSora. We show 5 evenly spaced frames from each video. Prompt: "a dilapidated phone booth stood as a relic of a bygone era on the sidewalk, frozen in time."

We qualitatively analyze fault-induced visual changes by taxonomizing them into three categories: **Minor** (only minor or imperceptible changes from the original), **Major** (at least one substantial difference, e.g., a changed background), and **Distorted** (significant corruption, e.g., pure static). Fig-

ure 6 illustrates each category under 2-bit memory faults: the major fault adds a tree and alters ground tiling, while the distorted fault produces random static across all frames. To measure the prevalence of these outcomes, we further randomly select 50 fault-injected videos per T2V model–fault model pair (300 total) and manually assign each to a category. Figure 7 shows the distribution by fault model.

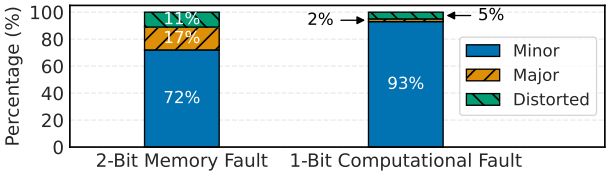

Figure 7. **Distribution of qualitative fault outcomes** for the 2-bit memory and 1-bit computational fault models (150 samples each).

We find that while most faults result in only minor changes (72–93%), a nontrivial fraction causes major alterations (2–17%) or even complete distortion of the generated video (5–11%). 2-bit memory faults produce substantially more non-minor changes than 1-bit computational faults (28% vs. 7%), suggesting that persistent corruption of the diffusion process is necessary to produce noticeable visual changes. These results contrast with the modest quantitative degradation in §4.2, indicating that standard performance metrics fail to capture the distinct failure modes introduced by faults. Moreover, the frequency of major changes shows that a few bit-level modifications are sufficient to substantially alter output semantics, motivating future work on whether such changes can be induced in a targeted manner.

5 Discussion

Potential countermeasures. Many machine learning defenses against hardware faults exist; we discuss those that appear most promising based on our findings or that are intrinsic to diffusion-based generation. Since high-order exponent bits account for the majority of degradation (§4.3), defenses should prioritize protecting them. General redundancy (e.g., full ECC or triplication) corrects such faults but is costly at the scale of billion-parameter DiT backbones, motivating selective protection of only the most vulnerable bits. Several works have proposed such schemes for other DNN architectures (Catalán et al., 2025; Xie et al., 2025; Fuengfusin et al., 2023; Zheng et al., 2025), and adapting them to T2V diffusion models is a promising direction for future work. An alternative is to correct faults on-the-fly: because the most damaging flips produce large-magnitude perturbations to otherwise bounded weights, range-based clipping (Chen et al., 2021; Sun et al., 2025c; Sha et al., 2024) can suppress out-of-distribution activations and mitigate their impact, at the cost of modest compute overhead and the risk of clipping legitimate in-distribution values. Such methods typically require careful, architecture-specific tuning (e.g., (Sun et al., 2025c)), implying comparable effort would be needed for T2V. Finally, as faults perturb the iterative denoising trajectory, stochastic sampling methods that contract accumulated errors, such as restart sampling (Xu et al., 2023), may dampen their impact at inference.

Limitations and future work. Our study considers two fault models, 2-bit memory and 1-bit computational faults, chosen to reflect uncorrected faults in modern GPUs (§3.2); however, other fault models are also practical (Dos Santos et al., 2022; Beigi et al., 2023) and merit investigation. We further focus on random faults, as is standard in resilience studies (Sun et al., 2025b; Agarwal et al., 2023; Roquet et al., 2024). While this characterizes expected-case behavior under naturally occurring faults, several of our findings—particularly the frequency of major semantic alterations (§4.5)—suggest faults could be exploited adversarially; confirming whether such effects can be induced in a *targeted* manner is a key open direction. Finally, our quantitative conclusions rely on VBench metrics, which may not fully capture the impact of faults: as our qualitative analysis (§4.5) shows, the scores understate visible failure modes, motivating metrics better suited to quantifying fault impact.

6 Conclusion

As T2V diffusion models are increasingly deployed on large-scale GPU infrastructure, understanding their resilience to hardware faults is critical to ensuring their reliability. To this end, we present the first systematic fault-injection study of T2V diffusion model inference, evaluating the impact of memory and computational faults on three representative models across 16 video-quality and semantic-correctness metrics. Our results show that while aggregate performance degradation is modest (at most 3.7%), this obscures a more complicated picture: semantic correctness is affected more than visual quality, memory faults are significantly more damaging than computational faults, and up to 28% of faults produce visible alterations—such as added objects or distortion—that standard metrics fail to capture. These findings demonstrate that T2V diffusion models are *not* robust to random bitwise faults, and that their failure modes differ from those observed in previously studied architectures.

To facilitate further study of T2V resilience and the evaluation of future defenses, we open-source our fault-injection framework at https://github.com/ztcoalson/T2V-Resilience.

Acknowledgment

We thank the anonymous reviewers for their valuable feedback. This work in part is supported by the Samsung Strategic Alliance for Research and Technology (START) program. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Adnan, M., Kurella, N., Arunkumar, A., and Nair, P. J. Foresight: Adaptive layer reuse for accelerated and high-quality text-to-video generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=q39uZC6RSo.

Agarwal, U. K., Chan, A., and Pattabiraman, K. Resilience assessment of large language models under transient hardware faults. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 659–670. IEEE, 2023.

Beigi, M. V., Cao, Y., Gurumurthi, S., Recchia, C., Walton, A., and Sridharan, V. A systematic study of ddr4 dram faults in the field. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 991–1002. IEEE, 2023.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., and Ramesh, A. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Catalán, I., Flich, J., and Hernández, C. Exploiting neural networks bit-level redundancy to mitigate the impact of faults at inference: I. catalán et al. *The Journal of Supercomputing*, 81(1):183, 2025.

Chen, Z., Li, G., and Pattabiraman, K. A low-cost fault corrector for deep neural networks through range restriction. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 1–13. IEEE, 2021.

Coalson, Z., Woo, J., Lin, C. S., Qu, J., Sun, Y., Chen, S., Yang, L., Saileshwar, G., Nair, P., Fang, B., and Hong, S. Prisonbreak: Jailbreaking large language models with at most twenty-five targeted bit-flips, 2025. URL https://arxiv.org/abs/2412.07192.

Di, D., Feng, H., Sun, W., Ma, Y., Li, H., Chen, W., Fan, L., Su, T., and Yang, X. Dh-facevid-1k: A large-scale high-quality dataset for face video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12124–12134, 2025.

Dos Santos, F. F., Kritikakou, A., Condia, J. E. R., Guerrero-Balaguera, J.-D., Reorda, M. S., Sentieys, O., and Rech, P. Characterizing a neutron-induced fault model for deep neural networks. *IEEE Transactions on Nuclear Science*, 70(4):370–380, 2022.

Fuengfusin, N., Tamukoh, H., Tanaka, Y., Nomura, O., and Morie, T. Efficient repetition coding for deep learning towards implementation using emerging non-volatile memory with write-errors. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE, 2023.

Gao, H., Chen, P., Shi, F., Tan, C., Liu, Z., Zhao, F., Wang, K., and Lian, S. Lemica: Lexicographic minimax path caching for efficient diffusion-based video generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=QIXdI207nq.

Guan, K., Lai, Z., Sun, Y., Zhang, P., Liu, W., Liu, K., Cao, M., and Song, R. Etva: Evaluation of text-to-video alignment via fine-grained question generation and answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21299–21309, 2025.

Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., and Dai, B. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *International Conference on Learning Representations*, 2024.

He, Y., Xia, M., Chen, H., Cun, X., Gong, Y., Xing, J., Zhang, Y., Wang, X., Weng, C., Shan, Y., et al. Animate-a-story: Storytelling with retrieval-augmented video generation. *arXiv preprint arXiv:2307.06940*, 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Hong, S., Frigo, P., Kaya, Y., Giuffrida, C., and Dumitras, T. Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 497–514, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL https://www.usenix.org/conference/usenixsecurity19/presentation/hong.

Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

HuggingFace. Optimizing llms for speed and memory. Available at: https://huggingface.co/docs/transformers/en/llm_tutorial_optimization.

Ibrahim, Y., Wang, H., Bai, M., Liu, Z., Wang, J., Yang, Z., and Chen, Z. Soft error resilience of deep residual networks for object recognition. *IEEE Access*, 8:19490–19503, 2020. doi: 10.1109/ACCESS.2020.2968129.

Jain, A., Veggetti, A. M., Crippa, D., Benfante, A., Gerardin, S., and Bagatin, M. Radiation tolerant multi-bit flip-flop system with embedded timing pre-error sensing. *IEEE Journal of Solid-State Circuits*, 57(9):2878–2890, 2022. doi: 10.1109/JSSC.2022.3149928.

Kalamkar, D., Mudigere, D., Mellempudi, N., Das, D., Banerjee, K., Avancha, S., Vooturi, D. T., Jammalamadaka, N., Huang, J., Yuen, H., Yang, J., Park, J., Heinecke, A., Georganas, E., Srinivasan, S., Kundu, A., Smelyanskiy, M., Kaul, B., and Dubey, P. A study of bfloat16 for deep learning training, 2019. URL https://arxiv.org/abs/1905.12322.

Li, G., Hari, S. K. S., Sullivan, M., Tsai, T., Pattabiraman, K., Emer, J., and Keckler, S. W. Understanding error propagation in deep learning neural network (dnn) accelerators and applications. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pp. 1–12, 2017.

Lin, C. S., Qu, J., and Saileshwar, G. Gpuhammer: Rowhammer attacks on gpu memories are practical. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, USA, 2025. USENIX Association.

Ma, X., Wang, Y., Chen, X., Jia, G., Liu, Z., Li, Y.-F., Chen, C., and Qiao, Y. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025.

Mutlu, O. The rowhammer problem and future directions. *IEEE Computer Architecture Letters*, 2015.

NVIDIA. Nvidia gpu memory error management, 2023. Available at: https://docs.nvidia.com/deploy/pdf/nvidia-gpu-mem-error-mgmt.pdf.

Oles, V., Schmedding, A., Ostrouchov, G., Shin, W., Smirni, E., and Engelmann, C. Understanding gpu memory corruption at extreme scale: The summit case study. In *Proceedings of the 38th ACM International Conference on Supercomputing*, pp. 188–200, 2024.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Rakin, A., He, Z., and Fan, D. Bit-flip attack: Crushing neural network with progressive bit search. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1211–1220, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society. doi: 10.1109/ICCV.2019.00130. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00130.

Reagen, B., Gupta, U., Pentecost, L., Whatmough, P., Lee, S. K., Mulholland, N., Brooks, D., and Wei, G.-Y. Ares: A framework for quantifying the resilience of deep neural networks. In *Proceedings of the 55th Annual Design Automation Conference*, pp. 1–6, 2018.

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

Roquet, L., Fernandes dos Santos, F., Rech, P., Traiola, M., Sentieys, O., and Kritikakou, A. Cross-layer reliability evaluation and efficient hardening of large vision transformers models. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pp. 1–6, 2024.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.

Sangchoolie, B., Pattabiraman, K., and Karlsson, J. One bit is (not) enough: An empirical study of the impact of single and multiple bit-flip errors. In *2017 47th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pp. 97–108. IEEE, 2017.

Sha, Q. S., Paulitsch, M., Pattabiraman, K., Hagn, K., Oboril, F., Buerkle, C., Scholl, K.-U., Hinz, G., and Knoll, A. Global clipper: Enhancing safety and reliability of transformer-based object detection models. *arXiv preprint arXiv:2406.03229*, 2024.

Sun, K., Huang, K., Liu, X., Wu, Y., Xu, Z., Li, Z., and Liu, X. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8406–8416, 2025a.

Sun, Y., Coalson, Z., Chen, S., Liu, H., Zhang, Z., Hong, S., Fang, B., and Yang, L. Demystifying the resilience of large language model inference: An end-to-end perspective. In *Proceedings of the International*

Conference for High Performance Computing, Networking, Storage and Analysis, SC '25, pp. 1127–1144, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400714665. doi: 10.1145/3712285.3759803. URL https://doi.org/10.1145/3712285.3759803.

Sun, Y., Zhu, Z., Mulpuru, C., Gioiosa, R., Zhang, Z., Fang, B., and Yang, L. Ft2: First-token-inspired online fault tolerance on critical layers for generative large language models. In *Proceedings of the 34th International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '25, New York, NY, USA, 2025c. Association for Computing Machinery. ISBN 9798400718694. doi: 10.1145/3731545.3731570. URL https://doi.org/10.1145/3731545.3731570.

Tiwari, D., Gupta, S., Gallarno, G., Rogers, J., and Maxwell, D. Reliability lessons learned from gpu experience with the titan supercomputer at oak ridge leadership computing facility. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '15, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450337236. doi: 10.1145/2807591.2807666. URL https://doi.org/10.1145/2807591.2807666.

VideoSys Team. Videosys: An easy and efficient system for video generation, 2024. URL https://github.com/NUS-HPC-AI-Lab/VideoSys.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., et al. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Wang, S. and Kanwar, P. Bfloat16: The secret to high performance on cloud tpus, 2019.

Xie, R., Haq, A. U., Fang, Y., Ma, L., Sen, S., Venkataramani, S., Liu, L., and Zhang, T. Breaking the hbm bit cost barrier: Domain-specific ecc for ai inference infrastructure. *IEEE Computer Architecture Letters*, 2025.

Xu, Y., Deng, M., Cheng, X., Tian, Y., Liu, Z., and Jaakkola, T. Restart sampling for improving generative processes. *Advances in Neural Information Processing Systems*, 36: 76806–76838, 2023.

Yang, S., Du, Y., Dai, B., Schuurmans, D., Tenenbaum, J. B., and Abbeel, P. Probabilistic adaptation of black-box text-to-video models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pjtIEgscE3.

Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LQzN6TRFg9.

Zhang, C., Zhang, C., Zhang, M., Kweon, I. S., and Kim, J. Text-to-image diffusion models in generative ai: A survey, 2024. URL https://arxiv.org/abs/2303.07909.

Zheng, W., Xu, B., Gu, J., and Chen, H. SAVE: Software-implemented fault tolerance for model inference against GPU memory bit flips. In *2025 USENIX Annual Technical Conference (USENIX ATC 25)*, pp. 1585–1604, 2025.

Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., and You, Y. Open-sora: Democratizing efficient video production for all, 2024. URL https://arxiv.org/abs/2412.20404.

Zhu, J., Yang, H., He, H., Wang, W., Tuo, Z., Cheng, W.-H., Gao, L., Song, J., and Fu, J. Moviefactory: Automatic movie creation from text using large generative models for language and images. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pp. 9313–9319, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3612707. URL https://doi.org/10.1145/3581783.3612707.

Zhu, Z., Sun, Y., Parakal, D., Fang, B., Farrell, S., Bauer, G. H., Bode, B., Foster, I. T., Papka, M. E., Gropp, W., et al. Understanding the landscape of ampere gpu memory errors. *arXiv preprint arXiv:2508.03513*, 2025.

A Detailed Experimental Setup

We use Python 3.10.19 with PyTorch 2.2.2 for all experiments, and the official VBench repository[1] to evaluate videos. We run these experiments on a system with a 48-core Intel Xeon Processor, 768GB of memory and 8 NVIDIA A40 GPUs.

B Individual Metric Results

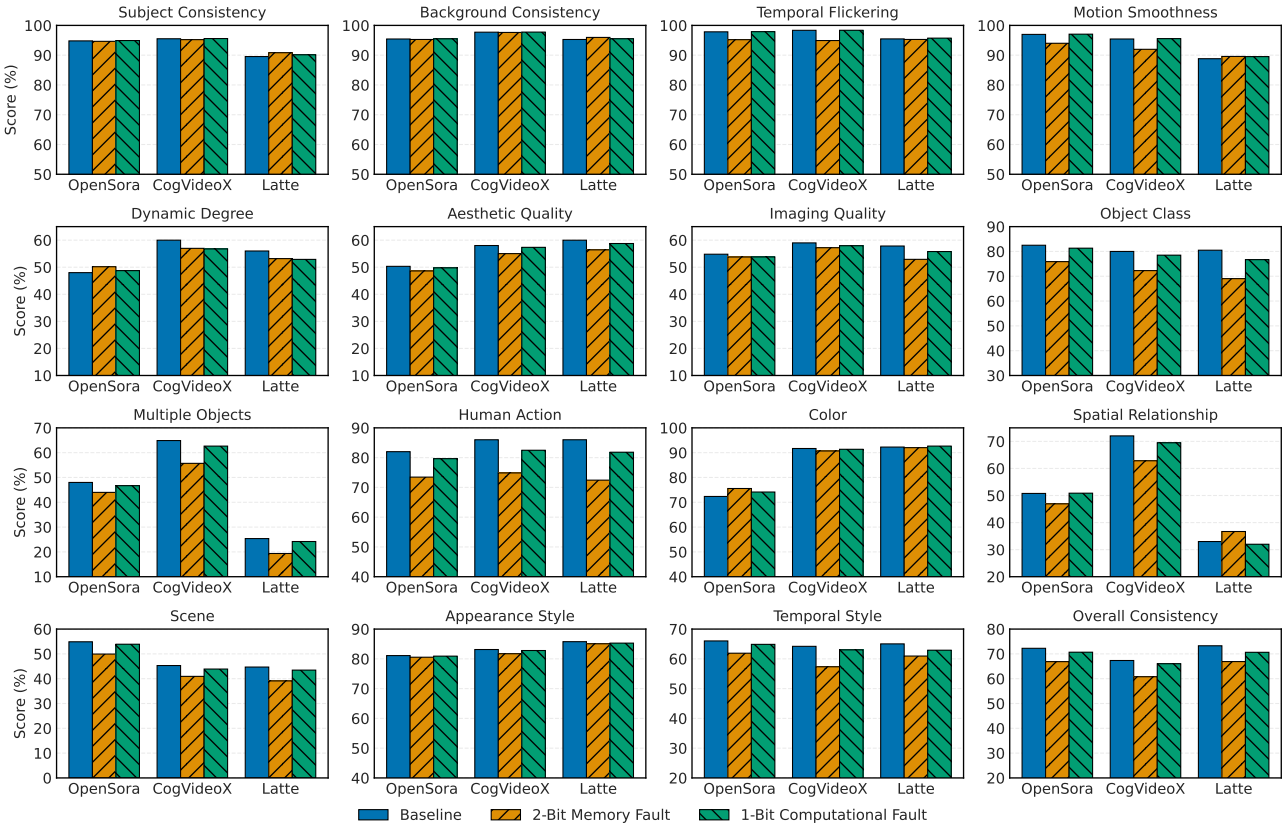

Figure 8. **Impact of hardware faults on individual video quality and semantics metrics.** For each metric, we report the average score over 100 trials for the fault-free (baseline), 2-bit memory fault, and 1-bit computational fault models. 95% CIs are within ± 0.95.

We examine changes in performance across the 16 individual metrics computed by VBench; results are shown in Figure 8. Resilience varies substantially across metrics: metrics such as Subject and Background Consistency are minimally affected, with changes ranging from -0.3% to +1.4%, whereas others, such as Multiple Objects, Human Action, and Object Class, suffer much larger degradations of 14.2–23.7%. Consistent with our aggregate findings, quality metrics are generally more resilient (changes of -8.5% to +4.5%) than semantic metrics (changes of -23.7% to +11.2%), though it varies considerably within each category. We attribute these differences to the types of visual perturbations induced by faults. Specifically, our qualitative analysis (§4.5) reveals that non-imperceptible fault-induced outcomes typically manifest as either semantic alterations (e.g., an added object) or complete distortion (e.g., a black screen). Such changes likely compromise semantic correctness more than visual quality.

C Full Resilience Analysis Results

Here, we complement the resilience analysis on Latte in §4.3 with additional results for OpenSora and CogVideoX, shown in Figures 9 and 10, respectively. The results are largely consistent with Latte: the most significant exponent bit is disproportionately vulnerable, while vulnerability is largely uniform across Transformer blocks and layer types. However, OpenSora exhibits two noticeable discrepancies. First, all bit positions except the most significant exponent bit show a

[1] https://github.com/Vchitect/VBench

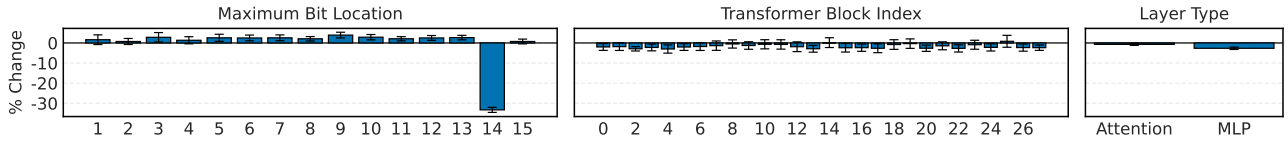

Figure 9. **Resilience of model-level components to 2-bit memory faults** for OpenSora. The average change across individual VBench metrics, categorized by maximum bit location, Transformer block index, and layer type.

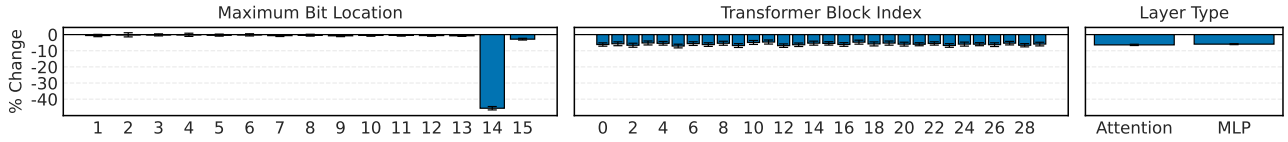

Figure 10. **Resilience of model-level components to 2-bit memory faults** for CogVideoX. The average change across individual VBench metrics, categorized by maximum bit location, Transformer block index, and layer type.

positive average change in individual metrics (up to +3.9%). We attribute this primarily to OpenSora exhibiting substantially higher per-trial variance than the other models, as reflected in its wider confidence intervals, which makes the sign of these averages less reliable as an indicator of a systematic effect. Second, unlike CogVideoX and Latte, attention layers are moderately more resilient than MLPs, with an average change of -0.7% compared to -2.7%, suggesting that MLP layers in this model are slightly more sensitive to random weight-level perturbations.

D VBench Metric Definitions

We provide brief definitions of all VBench metrics and refer readers to the original work (Huang et al., 2024) for full details:

- **Subject Consistency:** Measures whether subjects (e.g., a person, vehicle, or animal) maintain a consistent appearance across all frames of the video.
- **Background Consistency:** Evaluates the temporal stability of the background, i.e., that the environment remains coherent and does not change unexpectedly over time.
- **Temporal Flickering:** Quantifies high-frequency visual instability between adjacent frames, capturing undesired flicker or jitter artifacts in static regions.
- **Motion Smoothness:** Assesses whether object and camera motions are temporally smooth and physically plausible, without abrupt or unnatural transitions.
- **Dynamic Degree:** Measures the overall amount of motion present in the video, capturing whether the generated content exhibits meaningful dynamics rather than static.
- **Aesthetic Quality:** Evaluates the perceptual appeal of individual frames, including photographic composition, color harmony, and artistic quality.
- **Imaging Quality:** Assesses low-level fidelity of frames, focusing on distortions such as blur, noise, or exposure.
- **Object Class:** Measures whether the generated video correctly depicts the objects specified in the prompt.
- **Multiple Objects:** Evaluates the model's ability to generate and correctly compose multiple distinct objects within the same scene as described by the prompt.
- **Human Action:** Assesses whether human subjects in the video perform the actions indicated in the prompt.
- **Color:** Measures the accuracy with which object colors in the generated video match prompt specifications.
- **Spatial Relationship:** Evaluates whether the spatial relationships between objects (e.g., left/right, above/below) are consistent with the prompt description.
- **Scene:** Measures whether the overall scene (e.g., ocean, city, forest) aligns with the scene described in the prompt.
- **Appearance Style:** Assesses consistency between the visual appearance of frames and the requested artistic or visual style (e.g., watercolor, oil painting).
- **Temporal Style:** Evaluates whether temporal characteristics such as camera motion or cinematic style align with the prompt's temporal style description.

- **Overall Consistency:** Measures holistic alignment between the generated video and the text prompt, capturing both semantic correctness and stylistic coherence.

All metrics are computed independently for each generated video and normalized per dimension using empirical minimum and maximum reference values, yielding scores in $[0, 1]$ (which we report as percentages). In addition, we define the three aggregate metrics that summarize overall generation quality across the benchmark:

- **Quality Score:** Aggregates overall video quality as a weighted average of the *first seven* metrics listed above.
- **Semantic Score:** Aggregates prompt-level correctness as a weighted average of the *remaining nine* above metrics.
- **Total Score:** Combines quality and semantic correctness as a weighted average of the Quality and Semantic Scores, placing a $4\times$ greater emphasis on quality.

These metrics enable both coarse- and fine-grained analysis of how faults impact diverse aspects of video generation.