УДК 004.9

Д. Н. ГАВРИК

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ОПТИМИЗАЦИИ НЕЙРОННЫХ СЕТЕЙ

Белорусский национальный технический университет
ООО «ФронтПоинт»
г. Минск, Республика Беларусь

Аннотация. *Проведено прикладное сравнительное тестирование ускорений вывода Stable Video Diffusion (image-to-video). Для всех методов использован фиксированный вход и параметры (1024×576, 25 кадров), базовый вариант FP16/25 шагов. Описано восемь сравниваемых подходов: стандартный запуск SVD в FP16, INT8 weight-only квантование UNet, torch.compile+TF32, снижение шагов, подстановка дистиллированных весов (AnimateLCM), 2:4 разреженность, LCM-режим, а также генерация ключевых кадров с последующей интерполяцией RIFE. Измерялись время и VRAM, качество / плавность оценивались прокси-метриками CLIP similarity, tSSIM, tLPIPS. Ключевые кадры+RIFE дает наибольшее ускорение при сохранении сильной привязки к исходнику, LCM обеспечивает сбалансированное ~2× ускорение; агрессивное снижение шагов ухудшает динамику.*

Ключевые слова: *Stable Video Diffusion (SVD), диффузионные модели, image-to-video, ускорение вывода, уменьшение числа шагов диффузии, INT8 weight-only квантование (UNet), torch.compile, TF32, дистиллированные веса, полуструктурная разреженность 2:4, LCM-режим (scheduler), ключевые кадры, интерполяция кадров RIFE, CLIP similarity, tSSIM, tLPIPS*

Введение

Диффузионные модели стали одним из основных подходов к генерации изображений и видео благодаря устойчивому обучению и высокому качеству синтеза. Однако практическое применение таких моделей ограничено итеративной природой процесса удаления шума: для получения одного результата требуется десятки последовательных прогонов UNet, что напрямую увеличивает задержку и стоимость вычислений [1].

Для задач генерации видео ситуация усложняется дополнительным временным измерением и высокими требованиями к памяти, поскольку модель должна поддерживать согласованность между кадрами и хранить промежуточные тензоры для всего пакета кадров. Stable Video Diffusion (SVD) представляет собой масштабируемую латентную видеодиффузионную модель, способную генерировать высокое разрешение и короткие ролики при условии заданного исходного изображения [2]. Несмотря на оптимизации латентного пространства, вывод SVD остается ресурсоемким и плохо подходит для интерактивных сценариев.

Часть классов ускорений взаимно независима и может комбинироваться: (а) уменьшение числа шагов / замена семплера, (б) дистилляция в модели быстрого вывода (consistency/LCM), (в) оптимизация вычислительного графа и аппаратных настроек, (г) компрессия параметров (квантование, разреженность), (д) переиспользование вычислений и перенос части работы на постобработку (ключевые кадры + интерполяция). На практике выбор подхода определяется компромиссом между скоростью, качеством и требованиями по VRAM.

Цель работы – на одном и том же входе и конфигурации генерации сравнить набор прикладных приемов ускорения вывода SVD и показать, какие из них дают выигрыш без заметного ухудшения визуального результата, а какие требуют дополнительной подготовки весов или иных условий. Задачи исследования: (1) реализовать единое сравнительное тестирование для методов M0–M7, (2) измерить время и потребление памяти, (3) оценить изменения качества с помощью прокси-метрик, (4) сформулировать практические рекомендации для выбора режима генерации в зависимости от требований.

Постановка эксперимента и метрики

Экспериментальная постановка. Для всех методов использовался один и тот же вход (изображение 1024x576) и одинаковая длина результата – 25 кадров. Базовый режим (M0) выполнялся с 25 шагами диффузии. Эксперименты запускались на одной GPU с суммарным объемом видеопамяти 16 ГБ, что соответствует типичной потребительской конфигурации, где VRAM часто является главным ограничением.

Модель и конвейер обработки. В качестве базового решения использовалась модель Stable Video Diffusion (SVD) для генерации видео по входному изображению. Параметры генерации фиксировались: разрешение 1024x576, длина результата – 25 кадров, базовое число шагов диффузии – 25 [2].

Методы ускорения. Сравнивались восемь вариантов (M0–M7). Каждый метод выполнялся в отдельном процессе, что уменьшает влияние состояния CUDA аллокатора и кешей. Все значения времени фиксировались в секундах на уровне вызова генера-

ции (без учета записи видео на диск); для метода с интерполяцией дополнительно учитывалось время постобработки.

Метрики. (1) Время вывода секунд и производные величины: s/frame и ускорение относительно M0. (2) Пиковая память VRAM: allocated (реально занятая под тензоры) и reserved (кеш аллокатора PyTorch). (3) Прокси-метрика привязки к исходному изображению: средняя косинусная близость CLIP-эмбеддингов каждого кадра к эмбеддингу входного изображения [3]. (4) Метрики временной согласованности между соседними кадрами: temporal SSIM (tSSIM) [4] и temporal LPIPS (tLPIPS) [5]. Эти метрики характеризуют плавность / изменчивость внутри одного видео и не являются сравнением с эталоном.

Ограничения. Косвенные метрики не полностью заменяют субъективную оценку человеком: высокие значения tSSIM и низкие tLPIPS могут означать как гладкость без мерцания, так и недостаток движения. Поэтому интерпретация результатов выполнялась совместно по времени, привязке к исходнику и динамике внутри ролика.

Описание сравниваемых режимов (M0–M7)

M0 – базовый вариант: стандартный запуск SVD в FP16 без специальных оптимизаций (25 шагов, 25 кадров).

M1 – квантование + смешанная точность: weight-only INT8 квантование линейных слоев (nn. Linear) внутри UNet с помощью TorchAO, после чего выполняется стандартная генерация [6]. Квантование выполняется перед выводом и не входит в измеряемые секунды.

M2 – оптимизация графа и аппаратных настроек: включение TF32 для матричных операций и компиляция UNet через torch.compile(mode="reduce-overhead") [7]. На первом прогоне возможны существенные накладные расходы компиляции и прогрева графа.

M3 – уменьшение числа шагов диффузии: параметр num_inference_steps снижается до 15, 10 или 4. Подход дает прямое ускорение, но может ухудшать детализацию и приводить к потере движения.

M4 – дистилляция через замену весов: подстановка дистиллированных весов UNet, подготовленных для малошагового вывода в парадигме Latent Consistency Models (например, семейство AnimateLCM) [8]. В данном прогоне веса использовались с обычным числом шагов для оценки эффекта замены без изменения траектории семплирования.

M5 – полуструктурная разреженность 2:4: попытка использовать частично структурированное разрежение (2:4) и ускоренные ядра cuSPARSELt [9]. Без дообучения с учетом разреженности – может приводить к деградации или коллапсу результата, что важно учитывать.

M6 – ключевые кадры + интерполяция: диффузией генерируется уменьшенное число ключевых кадров, затем промежуточные кадры восстанавливаются интерполяцией с помощью RIFE (IFNet) [10]. Такой подход переносит часть вычислений с дорогого процесса удаления шума на более дешевую постобработку.

M7 – латентная консистентность (LCM): использование дистиллированных весов и специализированного планировщика, позволяющего получать результат за малое число шагов (4) при сохранении разумного баланса качества и динамики [8].

Иллюстрация результата

На рисунках 1–2 показаны первый, средний и последний кадры результатов для всех режимов ускорения (M0–M7), а для M3 – все варианты числа шагов (15, 10, 4).

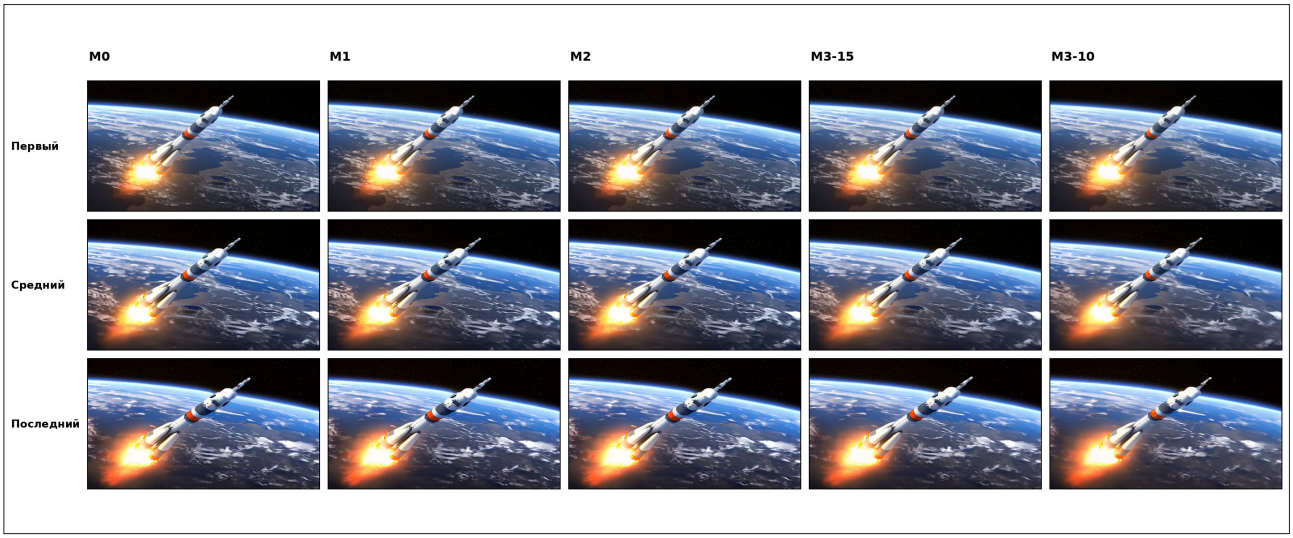

Рисунок 1. Первый, средний и последний кадры (M0, M1, M2, M3-15, M3-10)

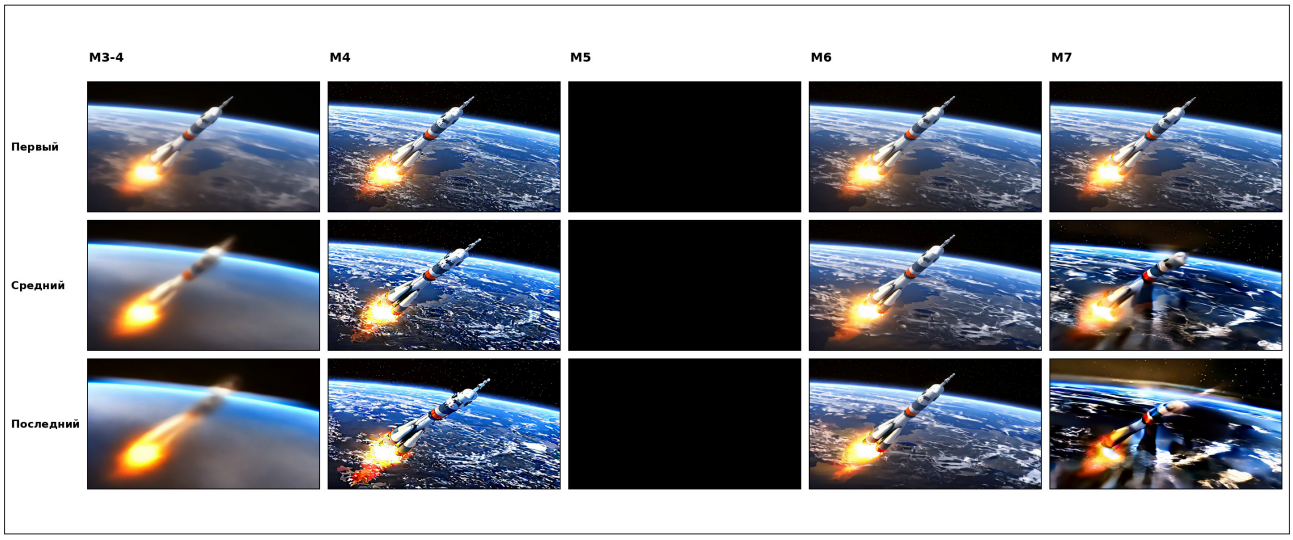

Рисунок 2. Первый, средний и последний кадры (M3-4, M4, M5, M6, M7)

Результаты и обсуждение

Таблица 1 содержит результаты одного прогона для всех методов в компактном виде (столбцы – методы, строки – метрики), включая время, VRAM (allocated/reserved) и пик температуры GPU.

Наиболее быстрое решение в рамках сравнения – M6 (ключевые кадры + RIFE): 99.4 с, что соответствует ускорению 3.45x. При этом CLIP similarity (0.981) остается высокой, то есть кадры хорошо удерживают содержание исходного изображения. Рост tSSIM (0.809) и снижение tLPIPS (0.127) согласуются с тем, что интерполяция делает движение более гладким и подавляет микромерцание, но может также уменьшать мелкие текстурные изменения.

Агрессивное снижение числа шагов (M3, 4 шага) дает ускорение 2.05x, но ценой снижения CLIP similarity до 0.891. Одновременно tSSIM становится очень высоким (0.979), а tLPIPS низким (0.086), что часто соответствует избыточной статичности и недостатку движения. Такой режим может быть приемлемым для быстрых предпросмотров, но рискован для финального результата.

Режим M7 (LCM, 4 шага) обеспечивает ускорение 1.92x при более сбалансированных временных метриках (tSSIM 0.722; tLPIPS 0.167), близких к базовому варианту, и умеренной потере привязки к исходнику (CLIP similarity 0.93). Это делает LCM привлекательным вариантом, когда требуется ускорение без явной деградации динамики.

Квантование (M1) показывает ускорение 1.27x и заметное снижение пикового allocated до 14849 МБ (около 91 % от общей), а также уменьшение VRAM и alloc. Это важно для практики: даже небольшая экономия памяти может позволить поднять разрешение, число кадров или запустить модель на более слабой GPU. Однако в данном тесте прирост скорости умеренный.

Полуструктурная разреженность 2:4 (M5) дала ускорение 1.35x по времени, но привела к коллапсу результата: CLIP similarity падает до 0.611, tSSIM становится 1.0, а tLPIPS – 0.0, что соответствует почти полностью статичному или черному видео. Это подтверждает выводы о том, что 2:4 разреженность должна сопровождаться дообучением с учетом разреженности или использованием заранее подготовленных разреженных весов [9].

Таблица 1. Сводные метрики (25 кадров, 1024x576)

Метрика	M0	M1	M2	M3-15	M3-10	M3-4	M4	M5	M6	M7
Шаги диффузии, шт	25	25	25	15	10	4	25	25	25	4
Время, с	342.5	269.9	363.2	201.1	251.4	166.8	243.4	254.4	99.4	178.8
s/frame, с	13.70	10.80	14.53	8.04	10.05	6.67	9.73	10.18	3.97	7.15
FPS, кадр/с	0.07	0.09	0.07	0.12	0.10	0.15	0.10	0.10	0.25	0.14
Ускорение vs M0, x	1.00	1.27	0.94	1.70	1.36	2.05	1.41	1.35	3.45	1.92
Peak VRAM alloc, МБ	15517	14849	15520	15517	15517	15517	15517	15523	15453	15517
Δ peak VRAM alloc vs M0, МБ	0	-669	3	0	0	0	0	5	-64	0
Peak VRAM reserved, МБ	31130	31558	34808	31130	31130	31130	31130	31130	27216	31130
Δ peak VRAM reserved vs M0, МБ	0	428	3678	0	0	0	0	0	-3914	0
Peak GPU temp, C	59	57	59	54	54	51	58	54	53	52

продолжение таблицы 1

Метрика	M0	M1	M2	M3-15	M3-10	M3-4	M4	M5	M6	M7
Δ peak temp vs M0, C	0	-2	0	-5	-5	-8	-1	-5	-6	-7
CLIP similarity	0.977	0.977	0.977	0.975	0.973	0.891	0.955	0.611	0.981	0.930
tSSIM	0.720	0.718	0.719	0.737	0.767	0.979	0.543	1.000	0.809	0.722
tLPIPS	0.164	0.164	0.164	0.161	0.156	0.086	0.230	0.000	0.127	0.167

Методы представлены в столбцах, метрики – в строках. Для метода M3 приведены отдельные столбцы для 15/10/4 шагов.

Практические рекомендации

Для минимальной задержки при сохранении сильной привязки к исходному изображению наиболее эффективен подход M6 (ключевые кадры + RIFE): он переносит часть вычислений в постобработку и дает наибольшее ускорение.

Если важен компромисс скорости и динамики без явной деградации, практичным выбором является M7 (LCM, 4 шага) с ускорением около 2x. Квантование (M1) полезно для снижения требований к VRAM, а методы, требующие специальной подготовки весов (например, 2:4), следует применять только при корректной настройке, учитывающей разреженность.

Заключение

Проведенное сравнение показало, что в задаче image-to-video для Stable Video Diffusion наибольший выигрыш по задержке дают стратегии, меняющие саму структуру вычислений. На практике наиболее эффективным оказалось вынесение части работы за пределы диффузионного процесса удаления шума: генерация ключевых кадров с последующей интерполяцией обеспечивает максимальное ускорение при сохранении сильной привязки к исходному изображению. В то же время подходы на основе консистентной дистилляции (LCM) дают близкое к двукратному ускорение без «обеднения» движения, оставаясь наиболее универсальным компромиссом.

Отдельно важно, что простое агрессивное уменьшение числа шагов действительно ускоряет расчет, но появляется риск избыточной статичности. По памяти выявлен фактически фиксированный потолок, поэтому ограничения VRAM остаются ключевым фактором при выборе режима; заметный практический резерв дает только weight-only INT8-квантование, тогда как полуструктурная разреженность без специализированной подготовки весов приводит к деградации вплоть до коллапса результата.

Таким образом, полученные результаты формируют прикладную рамку выбора ускорений: для минимальной задержки – перенос вычислений в постобработку (ключевые кадры + интерполяция), для сбалансированного режима – LCM, при дефиците памяти – INT8-квантование. Перспективным развитием работы является расширение тестирования на разнообразный набор входов и добавление субъективной оценки, а также проверка комбинирования совместимых ускорений в едином конвейере обработки.

ЛИТЕРАТУРА / REFERENCES

1. Ho J., Jain A., Abbeel P. Denoising diffusion probabilistic models. arXiv:2006.11239; 2020. https://doi.org/10.48550/arXiv.2006.11239.

2. Blattmann A., Dockhorn T., Kulal S., Mendelevitch D., Kilian M., Lorenz D., et al. Stable video diffusion: scaling latent video diffusion models to large datasets. arXiv:2311.15127; 2023. https://doi.org/10.48550/arXiv.2311.15127.

3. Radford A., Kim J.W., Hallacy C., Ramesh A., Goh G., Agarwal S., et al. Learning transferable visual models from natural language supervision. arXiv:2103.00020; 2021. https://doi.org/10.48550/arXiv.2103.00020.

4. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing (TIP). 2004;13(4):600–612. https://doi.org/10.1109/TIP.2003.819861.

5. Zhang R., Isola P., Efros A.A., Shechtman E., Wang O. The Unreasonable effectiveness of deep features as a perceptual metric. Computer Vision and Pattern Recognition (CVPR); 2018. https://doi.org/10.48550/arXiv.1801.03924.

6. PyTorch. TorchAO Quantization API. [Documentation]. Available at: https://docs.pytorch.org/docs/stable/quantization-support.html (accessed 19 January 2026).

7. PyTorch. torch.compile. [Documentation]. Available at: https://docs.pytorch.org/docs/stable/generated/torch.compile html (accessed 19 January 2026).

8. Luo S., Tan Y., Huang L., Li J., Zhao H. Latent consistency models: synthesizing high-resolution images with few-step inference. arXiv:2310.04378; 2023. https://doi.org/10.48550/arXiv.2310.04378.

9. NVIDIA. cuSPARSELt: A high-performance CUDA library for sparse matrix-matrix multiplication. [Documentation]. Available at: https://docs nvidia.com/cuda/cusparselt/ (accessed 19 January 2026).

10. Huang Z., Zhang T., Heng W., Shi B. Zhou S. RIFE: Real-Time intermediate flow estimation for video frame interpolation. Computer Vision and Pattern Recognition (CVPR); 2022. https://doi.org/10.48550/arXiv.2011.06294.

D. N. HAURYK

COMPARATIVE ANALYSIS OF NEURAL NETWORK OPTIMIZATION METHODS

Belarusian National Technical University
FrontPoint LLC
Minsk, Republic of Belarus

Abstract. *We benchmark practical ways to accelerate Stable Video Diffusion (SVD) inference for image-to-video. All methods use a fixed setup (1024×576 input, 25 frames) with an FP16 baseline at 25 denoising steps. We compare eight techniques: UNet INT8 weight-only quantization, torch.compile+TF32, step reduction, distilled weights (AnimateLCM), semi-structured 2:4 sparsity, LCM mode/scheduler, and keyframe generation with RIFE interpolation as post-processing. We measure latency and peak VRAM, and track quality/motion via CLIP similarity, tSSIM, and tLPIPS. Keyframes+RIFE achieves the highest speedup while preserving strong conditioning to the input. LCM provides a balanced ~2× speedup, whereas aggressive step cuts (and untuned 2:4) can degrade motion.*

Keywords: *Stable Video Diffusion (SVD), diffusion models, image-to-video, inference acceleration, denoising step reduction (num_inference_steps), UNet INT8 weight-only quantization, torch.compile, TF32, distilled weights (AnimateLCM), semi-structured 2:4 sparsity, LCM mode/scheduler, keyframes, RIFE frame interpolation, CLIP similarity, tSSIM, tLPIPS*

Гаврик Дмитрий Николаевич
Место работы: ООО «ФронтПоинт», г. Минск, Республика Беларусь.
Аспирант-соискатель кафедры «Программное обеспечение информационных систем и технологий» Белорусского национального технического университета. Сфера интересов: генеративные модели и диффузионные модели (image-to-video), ускорение и оптимизация вывода нейросетей, компрессия моделей (квантование, разреженность), малошаговые методы (LCM/consistency), компиляция и оптимизация графа (torch.compile), GPU-вычисления и оптимизация потребления VRAM.
Dzmitry N. Hauryk
FrontPoint LLC, Minsk, Republic of Belarus.
PhD applicant of the Department of Software of Information Systems and Technologies at the Belarusian National Technical University. Research interests: generative AI and diffusion models (image-to-video), neural network inference acceleration and optimization, model compression (quantization, sparsity), few-step methods (LCM/consistency), graph compilation and runtime optimization (torch.compile), GPU computing and VRAM optimization.
Tel.: +375293952030
E-mail: povt@bntu.by
E-mail: dr3952030@icloud.com