COMPRESSING A LIFETIME: EXPLOITING LONG-HORIZON TEMPORAL REDUNDANCY FOR PERSISTENT VISUAL MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Persistent visual memory is essential for always-on smart-glass assistants, but its storage cost is prohibitive: recording 2K video at 10 frames per second produces roughly 17 PB of raw RGB data over a 25-year lifetime, far beyond practical storage limits. Existing systems often reduce storage by selecting salient observations, irreversibly discarding visual evidence that may matter for unforeseen future queries. We instead formulate persistent visual memory as a *memory-compression* problem: retaining a compact representation of the complete observation stream rather than replacing it with selected frames or textual summaries. Our key insight is that egocentric video contains substantial long-horizon redundancy, with similar observations recurring across days and months, yet existing video codecs mainly exploit spatial and short-range temporal redundancy. We introduce a system that leverages this recurring context during both storage and reconstruction. A multi-rate neural video codec supports a wide range of compression levels, a novelty-aware rate selector assigns different rates to observations, and a memory-guided generative decoder reconstructs highly compressed content. Across three egocentric benchmarks, our system achieves compression ratios closed to $10000\times$, beyond the practical range of existing baselines, while substantially improving reconstruction quality and downstream video understanding under extreme compression.

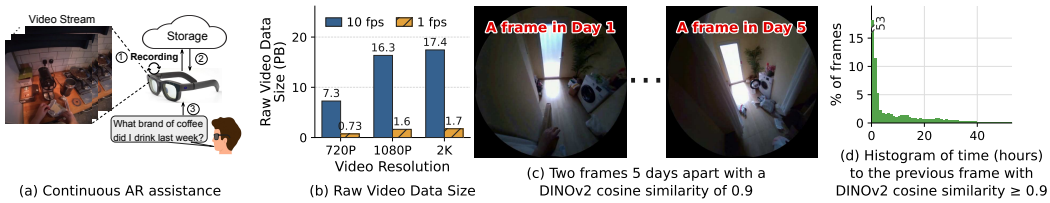
1 INTRODUCTION

Always-on smart glasses enable visual assistants that reason over a user's experience across extended periods of time (Engel et al., 2023; Grauman et al., 2022). Consider the example in Figure 1(a). A user wears the glasses throughout the day while they continuously record video. A week later, the user asks, "What brand of coffee did I drink last week?" The glasses did not know at recording time that this moment would matter, so the assistant can answer only if the video from that day was kept. Such assistance therefore requires recording and storing video continuously, building a persistent visual memory of the user's life.

However, the storage cost of this memory is prohibitive. Figure 1(b) shows the raw size of video recorded 8 hours per day over 25 years. Even at 720P and 1 FPS, the raw data reaches 0.73 PB; at 2K and 10 FPS, it reaches 17.4 PB. These sizes far exceed the 32 GB on-device storage of current smart glasses (Meta, 2025), and even a personal storage budget of a few terabytes. Additionally, storing the video is also not enough: an assistant must process the memory to answer a query. Current video language models already struggle with hour-long videos, and their computation grows with the number of visual tokens in the input (Wang et al., 2024; Zhang et al., 2024; Song et al., 2024). A memory of this size is too large for an assistant to be helpful.

Many existing visual-memory systems reduce storage by retaining only frames, clips, or events judged salient during memory construction (Bärmann & Waibel, 2022; Wang et al., 2023; Ego-R1, 2025; FOCUS, 2025; AdaRD-key, 2025). This filtering is irreversible: a future query such as "Did I leave a charger in that room?" may depend on peripheral evidence that was not considered important when recorded. We therefore treat building a visual memory as a *video compression* problem: retaining a representation of every observation while varying the amount of storage allocated to it.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Continuous AR assistance (b) Raw Video Data Size (c) Two frames 5 days apart with a DINOv2 cosine similarity of 0.9 (d) Histogram of time (hours) to the previous frame with DINOv2 cosine similarity ≥ 0.9

Figure 1: (a) Continuous AR assistance relies on always-on video recording. (b) Raw video data size for recording 8 hours per day over 25 years. (c) Two frames recorded five days apart, with a DINOv2 cosine similarity of 0.9. (d) Histogram of the time gap between each EgoLife frame and its most similar earlier frame with DINOv2 cosine similarity of at least 0.9.

Existing video compression methods mainly exploit *short-range temporal redundancy*: they predict each frame from neighboring frames within a window of a few seconds. This holds for both conventional codecs such as H.265 (Sullivan et al., 2012) and learned codecs such as DCVC (Li et al., 2021; 2023; 2026). At the compression ratios required for a lifetime memory, these methods lose the fine-grained details needed to answer questions (Section 2.2). However, we find that egocentric video also contains substantial redundancy over much longer time spans. Users repeatedly see the same rooms, furniture, appliances, and personal belongings across hours, days, or months. Figure 1(c) shows two nearly identical views recorded five days apart, and Figure 1(d) shows that such repeated views are common: for more than 40% of the EgoLife dataset (Yang et al., 2025) frames that have a similar earlier frame, which was recorded over an hour before. We call this *long-horizon temporal redundancy*. A later observation can therefore be stored at a much lower rate and reconstructed using related observations already in memory.

Based on this insight, we develop a compression system that exploits long-horizon temporal redundancy with three components. First, the *Egocentric Neural Video Codec* (ENVC) represents each frame at one of many compression levels. Second, a *Novelty-aware Rate Selector* (NRS) compares each incoming frame with a bank of high-quality keyframes. Novel observations become new keyframes, and recurring observations are stored at adaptive rates. Third, a *Memory-guided Generative Decoder* (MGD) reconstructs each target frame from its compressed representation, using retrieved keyframes as visual context to restore details lost during compression. Long-horizon temporal redundancy thus determines both the number of bits allocated to each frame and the visual context used for reconstruction.

Compression inevitably degrades video quality, which in turn degrades the quality of downstream assistance. To measure this effect, we build a new VQA benchmark, *EgoCompress*, from seven days of EgoLife recordings (Yang et al., 2025). Its questions focus on fine-grained visual details, such as the color, pattern, material, visible text, or state of an object, which are the first to be lost under aggressive compression. EgoCompress therefore separates compression methods by the quality of their reconstructions far more clearly than existing benchmarks.

We evaluate our system on EgoCompress, EgoSchema, HD-EPIC, and EgoEverything. At extreme compression ratios, our system retains most of the accuracy of uncompressed video and substantially outperforms all baseline methods.

In summary, our contributions are:

- We formulate lifetime visual memory as an extreme-compression problem and identify long-horizon temporal redundancy in egocentric video, a source of redundancy beyond the short-range temporal redundancy exploited by existing compression methods.
- We introduce a system that combines a egocentric neural video codec (ENVC), a novelty-aware rate selector (NRS), and a memory-guided generative decoder (MGD) to exploit long-horizon temporal redundancy during both compression and reconstruction.
- We introduce EgoCompress, a 500-question VQA benchmark on fine-grained visual details that measures how compression affects downstream assistance.



Figure 2: Upper and bottom rows show two examples from EgoSchema, with (a) the original frame at 512×512 resolution; (b) the frame downsampled to 128×128 resolution then coded by H.265, with a total compression ratio of $297\times$; (c) the frame downsampled to 128×128 resolution then coded by DCVC-UF, with a total compression ratio of $1560\times$; (d) the frame as DrVideo represents it using a 50-token text caption retaining no pixels, with a total compression ratio of $3073\times$; (e) our method with a total compression ratio of $3165\times$. (f) EgoSchema VQA accuracy of our method and the baselines across different compression ratios; the dotted line marks the 78.8% accuracy obtained on uncompressed frames.

- Our method operates at compression ratios beyond the practical range of existing compression baselines while substantially outperforming them in both reconstruction quality and downstream video question answering (VQA) accuracy.

2 Background and Related Work

2.1 Long-Horizon Temporal Redundancy in Egocentric Video

Egocentric video is recorded from the user's own viewpoint as they go about daily life. Users repeatedly return to the same rooms and see the same furniture, appliances, artwork, and personal belongings over hours, days, or months. Figure 1(c) shows an example: two frames of the same scene recorded five days apart. To measure how similar they are, we embed each frame with DINOv2 (Oquab et al., 2024) and compute the cosine similarity between the two embeddings. The similarity is above 0.9, even though the frames are five days apart and differ in viewpoint and foreground content.

We then use the same 0.9 threshold to measure how often such repeated views occur in real recordings. We use EgoLife (Yang et al., 2025), which contains seven days of continuous smart-glass recordings. For each frame, we search all earlier frames for those whose DINOv2 cosine similarity with it exceeds 0.9. Among them, we take the most similar frame and record the time gap between the two. Figure 1(d) shows the histogram of these time gaps. More than 40% of the gaps exceed one hour, and the longest gap exceeds 40 hours. In other words, much of what the user sees has already been seen hours or days earlier. We call this *long-horizon temporal redundancy*.

Existing video codecs do not exploit this redundancy. Conventional and learned codecs can use multiple reference frames, but these references come from a local coding window of at most a few seconds (Sullivan et al., 2012; Li et al., 2021; 2023); we refer to the redundancy they exploit as *short-range temporal redundancy*. Memorize-and-recall compression uses stored scene content to encode fixed-camera surveillance video at low bitrates (Memorize-and-Recall, 2020), but relies on a static camera. Visual place-recognition methods retrieve observations of revisited places over long intervals (Arandjelović et al., 2016), but do not use them for compression. Our system exploits long-horizon temporal redundancy for compression: once a scene has been stored at high quality, later observations of it can be stored at a much lower rate and reconstructed using the stored observation as visual context.

2.2 Video Compression

Conventional codecs such as H.265 (Sullivan et al., 2012) and VVC (Bross et al., 2021) reduce spatial and short-range temporal redundancy through intra- and inter-frame prediction. Learned codecs,

including DVC (Lu et al., 2019), scale-space flow (Agustsson et al., 2020), and the DCVC family (Li et al., 2021; 2023; Jia et al., 2025; Li et al., 2026), replace the hand-designed prediction and entropy models with neural networks and now outperform conventional codecs. Both kinds of codecs are designed for pixel fidelity at moderate compression ratios, and predict each frame only from nearby frames.

Figure 2 shows that these codecs cannot reach the compression ratios required for persistent visual memory without a severe loss in quality. To reach high compression, we downsample frames before coding them with H.265 or DCVC-UF (Li et al., 2026). At $297\times$, H.265 already produces heavily blurred frames (Figure 2(b)); at $1560\times$, DCVC-UF frames are smoother but lose most object details and text (Figure 2(c)). The downstream accuracy follows the same trend. Figure 2(f) plots EgoSchema (Mangalam et al., 2023) VQA accuracy against compression ratio: the accuracy of both codecs drops quickly as compression increases, and remains far below the accuracy on uncompressed frames at their highest compression ratios.

Our method exploits long-horizon temporal redundancy in addition to short-range temporal redundancy. At $3165\times$, a higher compression ratio than both codecs, it reconstructs sharp frames that preserve object shapes, colors, and scene layout (Figure 2(e)). Its VQA accuracy is higher than that of both codecs even though it operates at higher compression ratios, and stays close to the uncompressed accuracy beyond $3000\times$ (Figure 2(f)).

2.3 Visual Memory in Other Forms

Long-video understanding systems reduce the input to the language model through temporal sampling, token pruning, retrieval, or recurrent memory. EgoSchema (Mangalam et al., 2023) established long-form egocentric reasoning as a benchmark, while MovieChat+ (Song et al., 2024), MA-LMM (He et al., 2024), and Goldfish (Ataallah et al., 2024) build compact memories or retrieve relevant clips for downstream language models. Egocentric-memory systems such as Grounded (Di & Xie, 2024) and AMEGO (Goletto et al., 2024) organize observations into textual, object-centric, or event-centric records. A common approach is to convert video into text. For example, DrVideo (Ma et al., 2025) converts a long video into a text document of frame captions and answers questions by retrieving from this document.

Text is extremely compact: a 50-token caption represents a frame at about $3073\times$ compression (Figure 2(d)). However, converting video to text loses information. A caption records what the captioning model considered important at storage time, and cannot retain every detail that an unknown future query may require, such as appearance, spatial relationships, object counts, or visible text. Figure 2(f) shows the consequence: although DrVideo reaches a high compression ratio, its VQA accuracy is much lower than that on uncompressed frames. Our method keeps a visual representation of every frame and achieves both a high compression ratio and high VQA accuracy.

3 Method

3.1 System Overview

Our system exploits long-horizon temporal redundancy by storing novel frames as high-quality keyframes and recurring frames as highly compressed non-keyframes. Figure 3 shows the three stages of the system: sensing on the edge device, storage in the cloud, and query in the cloud.

The smart glasses continuously record a video stream. For the current frame I_t, the ENVC encoder compresses it into an ENVC embedding E_t^{qp}. The quantization parameter qp (Li et al., 2023; Jia et al., 2025) sets the compression rate of the encoder, and a larger qp gives higher quality. The ENVC encoder can produce embeddings at several compression rates at once. We denote the highest-quality level by $qp = N$. The edge device produces embeddings $E_t^0, \ldots, E_t^m$ at a set of candidate levels, together with the highest-quality embedding E_t^N, and sends them to the cloud.

In the cloud, the ENVC decoder first decodes E_t^N into a high-quality frame I_t^N, which is passed to the Novelty-aware Rate Selector (NRS). NRS decides whether the current frame is a keyframe or a non-keyframe. It computes the DINOv2 embedding D_t of I_t^N and compares it with the DINOv2 embeddings $D_{0:k-1}$ of the keyframes already stored in memory. If the current frame is a keyframe,

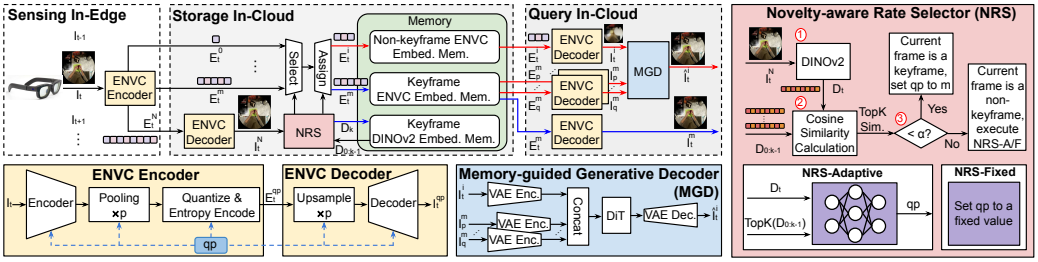


Figure 3: Overview of our persistent visual-memory system. Top: sensing on the edge device, storage in the cloud, and query in the cloud. Red lines show the data path of non-keyframes and blue lines show the data path of keyframes. Bottom and right: the ENVC encoder and decoder, the Memory-guided Generative Decoder (MGD), and the Novelty-aware Rate Selector (NRS).

the memory stores its DINOv2 embedding D_k and its high-quality ENVC embedding E_t^m. Otherwise, NRS selects a qp level i, and the memory stores the ENVC embedding E_t^i at the corresponding compression rate. In Figure 3, blue lines show the data path of keyframes and red lines show that of non-keyframes.

At query time, the stored frames are decompressed. A keyframe is decoded directly by the ENVC decoder into I_t^m. A non-keyframe is stored at a much higher compression rate, so we retrieve its ENVC embedding E_t^i together with the ENVC embeddings $E_p^m, \ldots, E_q^m$ of its matched keyframes, and decode all of them with the ENVC decoder into images I_t^i and $I_p^m, \ldots, I_q^m$. These images are passed to the Memory-guided Generative Decoder (MGD), which uses the keyframes as reference and generatively restores the low-quality, highly compressed non-keyframe into $\hat{I}_t^i$. Frame retrieval is orthogonal to compression; our evaluation assumes that the target frame is known.

3.2 Egocentric Neural Video Codec

ENVC compresses each frame independently into a compact ENVC embedding and decodes it back to an image. It must cover both near-lossless keyframes and non-keyframes stored at several thousand times compression, so a single model supports 64 compression rates, indexed by $qp \in \{0, \ldots, 63\}$, where a larger qp gives higher quality and $N = 63$ is the highest-quality level.

ENVC follows the architecture of DCVC-RT (Jia et al., 2025): an encoder maps the frame to a latent, a quantization and entropy coding module turns the latent into a bitstream using a hyperprior (Ballé et al., 2018), and a decoder maps the latent back to an image. We add a pooling and upsample architecture to support wider compression range. After the encoder, the latent is average-pooled by a factor p determined by qp, and the pooled latent is passed to the quantization and entropy coding module to produce the bitstream E_t^{qp}. At decoding time, the latent recovered from the bitstream is upsampled by the same factor p and passed to the decoder, so one decoder serves every qp. For a 512×512 frame, the encoder outputs a 64×64 latent, and the pooling factor is $p = 16, 8, 4, 2, 1$ for $qp \in [0, 7]$, $[8, 15]$, $[16, 31]$, $[32, 47]$, and $[48, 63]$, respectively. This gives a stored latent grid of $\frac{64}{p} \times \frac{64}{p}$, from 64×64 down to 4×4.

To train ENVC, in each training frame, we sample qp uniformly and minimize

$$\mathcal{L}_{\text{ENVC}} = \left\| I_t - \hat{I}_t^{qp} \right\|_1 + \beta\, \mathcal{L}_{\text{perc}}\left(I_t, \hat{I}_t^{qp}\right) + \lambda_{qp}\, R\left(E_t^{qp}\right), \tag{1}$$

where $\hat{I}_t^{qp}$ is the decoded frame, $\mathcal{L}_{\text{perc}}$ is a perceptual loss with weight $\beta = 0.005$, and R is the number of bits estimated by the entropy model. The rate weight λ_{qp} decreases on a log scale from 8 at $qp = 0$ to 0.001 at $qp = 63$, so each qp settles at a different rate. A single model covers compression ratios from about $200\times$ to nearly $10{,}000\times$ (Table 1).

3.3 Novelty-aware Rate Selector

The Novelty-aware Rate Selector (NRS) decides whether the current frame is a keyframe and, if not, at which qp it is stored. NRS first computes the DINOv2 embedding (Oquab et al., 2024) of the

high-quality decoded frame, $D_t = \mathrm{DINOv2}(I_t^N)$ (step 1). It then computes the cosine similarity between D_t and the embeddings $D_{0:k-1}$ of the k keyframes stored in memory, and keeps the top-K similarity scores together with the corresponding keyframe embeddings (step 2). If the highest similarity is below a threshold α (step 3), the current frame is novel: it differs from every stored keyframe, so it becomes a new keyframe. The memory stores its DINOv2 embedding as D_k and its high-quality ENVC embedding E_t^m. Otherwise, part of the current frame is already contained in the matched keyframes, so the frame is a non-keyframe and NRS selects the $qp = i$ at which it is stored. We use two variants to select i.

First, NRS-Fixed (NRS-F) stores every non-keyframe at the same fixed qp, chosen once for a target storage budget. This ignores the relation between the non-keyframe and its matched keyframes: a frame that the keyframes already explain well costs as many bits as one they barely cover. To improve this, NRS-Adaptive (NRS-A) chooses the qp per frame. It feeds D_t and the embeddings of the top-K matched keyframes ($K = 3$) into a small CNN, which outputs the qp used to compress the current frame. The CNN has four stride-2 convolution layers, each followed by group normalization and SiLU, then global average pooling and a linear layer that scores each candidate qp. The candidates are $qp \in \{0, 8, 16\}$, which store the frame as a 4×4, 8×8, or 16×16 latent grid. Figure 7 in Appendix B shows examples of the qp selected by NRS-A.

We label each training frame of NRS-A offline. We compress it at every candidate qp, reconstruct it with MGD, and answer its VQA question on the reconstruction. The label $qp_t^\star$ is the lowest-rate candidate that gives the same answer as the highest-rate candidate. The two kinds of error are not equally harmful: choosing a lower rate than $qp_t^\star$ changes the answer, while choosing a higher rate only wastes storage. We therefore minimize the expected cost under the predicted distribution $\pi(qp)$, plus a small cross-entropy term:

$$\mathcal{L}_{\mathrm{NRS}} = \sum_{qp} \pi(qp)\, c(qp) - \gamma \log \pi\big(qp_t^\star\big), \tag{2}$$

where $c(qp)$ is a fixed penalty η if compressing at qp changes the answer, and otherwise the extra storage of qp over $qp_t^\star$, normalized by the storage of the highest-rate candidate. We use $\gamma = 0.05$; sweeping η trades accuracy for storage.

3.4 Memory-guided Generative Decoder

A non-keyframe keeps the coarse layout of the scene but loses textures, text, and other fine details. The Memory-guided Generative Decoder (MGD) restores these details with a diffusion model, using the matched keyframes as reference. Its inputs are the decoded non-keyframe I_t^i and the decoded matched keyframes $I_p^m, \ldots, I_q^m$, and its output is the restored frame $\hat{I}_t^i$. The non-keyframe specifies the viewpoint, layout, and transient content of the scene, while the keyframes supply high-quality appearance of the content that recurs.

MGD is built on the FLUX.2 [klein] diffusion transformer (DiT) (Black Forest Labs, 2025). A VAE encoder first encodes the non-keyframe and each keyframe into latents. These latents are concatenated as reference tokens with the latent of the frame being generated, which starts from pure noise. The DiT iteratively denoises the generated latent while attending to the reference tokens, and a VAE decoder maps the final latent back to the image $\hat{I}_t^i$.

We keep the VAE frozen and fine-tune the DiT with LoRA (Hu et al., 2022). Let z_0 be the VAE latent of the original frame I_t and ϵ Gaussian noise. We sample a noise level $\tau \in [0, 1]$, form $z_\tau = (1 - \tau)z_0 + \tau\epsilon$, and train the DiT v_ϕ with the standard flow-matching loss:

$$\mathcal{L}_{\mathrm{MGD}} = \big\| v_\phi\big(z_\tau, \tau;\, I_t^i, I_p^m, \ldots, I_q^m\big) - (\epsilon - z_0) \big\|_2^2. \tag{3}$$

Training examples cover several non-keyframe qp levels and one to three keyframes, so a single MGD handles every operating point that NRS produces.

4 EgoCompress: Measuring How Compression Affects Downstream VQA

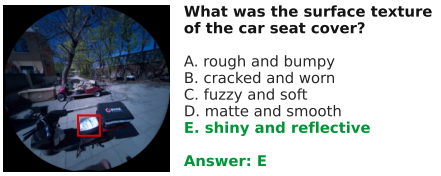


Figure 4: Example in EgoCompress.

Compressing visual memory degrades the stored frames, and this degradation propagates to the quality of downstream assistance. We build EgoCompress to measure this effect directly: it evaluates a compression method by the VQA accuracy obtained on its reconstructed frames. Existing video VQA benchmarks are poorly suited for this purpose. Their questions are often answerable from coarse scene semantics that survive heavy compression, so different compression methods obtain similar accuracy, and their videos are too short to expose visual recurrence across days. EgoCompress instead focuses on fine-grained visual details, which are the first to be lost under aggressive compression. We construct it by chronologically concatenating seven days of EgoLife (Yang et al., 2025) recordings from the same user. Target frames are sampled throughout the timeline; those appearing later can be associated with relevant visual evidence recorded hours or days earlier.

Our benchmark contains 500 object-centric questions. An agent first localizes candidate objects with bounding boxes in each target frame. We randomly select a localized object and generate a question about a fine-grained attribute, such as its color, pattern, shape, material, visible text, or state. A question-generation agent proposes each question and answer, after which a separate review agent checks visual grounding, clarity, and answerability. We use Qwen3-VL-32B-Instruct (Bai et al., 2025) for all agent roles in the pipeline. Finally, we manually review all examples and remove ambiguous, invalid, or duplicate questions. Figure 4 shows an example question in EgoCompress.

5 Experiments

5.1 Experimental Setup

We evaluate on four egocentric VQA benchmarks: EgoCompress (Section 4); the 500-question test set of EgoSchema (Mangalam et al., 2023); and 500 questions randomly sampled from each of HD-EPIC (Perrett et al., 2025) and EgoEverything (Tang et al., 2026). All frames are resized to 512×512 before compression. Each video is sampled at 1 FPS, every sampled frame is compressed and reconstructed by the method under test, and the reconstructed frames are given to the VLM together with the question. We use Qwen3-VL-32B (Bai et al., 2025) and Gemini 3.1 Pro (Google DeepMind, 2026) as VQA models, with identical prompts for every method.

Always-on wearable cameras typically capture about one frame per second (Hodges et al., 2006; Gurrin et al., 2014), and our system also stores the video at 1 FPS. At this rate adjacent frames are about one second apart, so we run both codec baselines in intra (image) mode. H.265 (Sullivan et al., 2012) is the conventional baseline and DCVC-UF (Li et al., 2026), the state-of-the-art neural video codec, is the learned baseline. Each codec is run at its lowest-quality setting (CRF 51 for H.265 and QP 0 for DCVC-UF). We reach higher ratios by average-pooling each frame to 256×256 or 128×128 before coding and upsampling it back to 512×512 after decoding. DrVideo (Ma et al., 2025) converts the video into a document of per-frame captions, and we give this document directly to the VLM instead of frames. Its three operating points use caption budgets of 200, 100, and 50 tokens per frame.

We report VQA accuracy and reconstruction quality in terms of PSNR, MS-SSIM (Wang et al., 2003), and LPIPS (Zhang et al., 2018). The compression ratio is the size of the raw 512×512 RGB frame (786,432 bytes) divided by the average number of stored bytes per frame. For our system with NRS, this is the overall ratio over all frames of a video: keyframes are stored at low compression with DINOv2 embeddings and non-keyframes at high compression. Keyframes are stored at $m = 32$. We use $\alpha = 0.92$ as the keyframe threshold and retrieve $K = 3$ matched keyframes for every non-keyframe. NRS-F stores non-keyframes at $qp = 24$, 16, 8, or 0, giving four operating points; NRS-A gives three operating points by sweeping the penalty η. MGD generates each frame with 20 sampling steps.

Training details of ENVC and MGD are given in Appendix A.

Method	OP	EgoCompress Comp. Ratio	EgoCompress Qwen Acc.	EgoCompress Gemini Acc.	EgoSchema Comp. Ratio	EgoSchema Qwen Acc.	EgoSchema Gemini Acc.	HD-EPIC Comp. Ratio	HD-EPIC Qwen Acc.	HD-EPIC Gemini Acc.	EgoEverything Comp. Ratio	EgoEverything Qwen Acc.
Uncompressed	–	1×	63.1	74.9	1×	78.8	70.5	1×	53.2	64.8	1×	47.6
H.265	1	214×	43.2	58.1	212×	74.0	62.5	235×	33.6	35.0	232×	45.2
	2	270×†	36.4	44.1	272×†	63.0	50.1	284×†	27.2	27.0	283×†	44.0
	3	306×‡	29.2	35.2	304×‡	55.2	44.5	309×‡	23.2	22.8	308×‡	44.4
DCVC-UF	1	126×	46.2	52.3	125×	71.5	62.3	132×	36.8	40.8	131×	43.6
	2	450×†	35.8	42.1	461×†	66.1	54.5	494×†	29.0	28.6	491×†	41.6
	3	1644×‡	27.8	36.2	1658×‡	56.7	44.7	1801×‡	26.0	21.8	1793×‡	42.0
DrVideo	1	878×	42.3	47.0	989×	62.7	52.4	923×	37.4	48.0	919×	43.1
	2	1699×	41.9	45.1	1648×	62.0	50.8	3073×	37.8	51.9	1657×	43.5
	3	3126×	40.2	44.1	3073×	61.5	48.6	3138×	36.0	44.7	3162×	42.8
Ours: ENVC	1	212×	57.2	68.1	191×	76.0	68.9	231×	45.8	54.8	223×	44.8
	2	697×	48.9	55.9	655×	71.5	62.1	758×	35.2	41.4	706×	44.3
	3	974×	44.3	53.4	909×	69.3	61.7	1048×	35.4	37.8	992×	44.2
	4	3026×	35.8	40.7	2816×	59.1	46.7	3243×	28.0	24.6	3045×	43.4
	5	9700×	29.4	38.0	8978×	49.3	42.3	9989×	25.0	23.4	9814×	40.8
Ours: ENVC + NRS-F + MGD	1	667×	64.2	72.1	595×	76.3	69.3	736×	48.6	61.6	638×	47.1
	2	924×	64.0	70.8	786×	75.6	68.1	1002×	46.0	58.4	908×	46.9
	3	2412×	59.1	66.3	1787×	74.4	67.9	2789×	39.4	50.2	2283×	45.3
	4	5229×	49.6	56.3	3062×	72.8	65.5	6556×	34.0	41.8	4636×	43.5
Ours: ENVC + NRS-A + MGD	1	3414×	61.4	71.2	2766×	75.3	69.0	3950×	47.5	54.8	3125×	46.2
	2	3780×	60.8	68.4	2933×	74.5	66.8	4312×	45.2	53.6	3382×	45.5
	3	4419×	58.3	62.7	3065×	74.3	65.8	5871×	39.2	47.5	4359×	44.9

†Frames are average-pooled to 256 × 256 before coding; ‡to 128 × 128.

Table 1: VQA accuracy (Acc.) (%) of Qwen3-VL-32B and Gemini 3.1 Pro on frames reconstructed from each compressed memory, with the compression ratio (Comp. Ratio) of each operating point. OP denotes the operating point.

5.2 Evaluation Results

Table 1 reports the VQA accuracy of all methods. ENVC alone already extends the operating range of image codecs by almost an order of magnitude: with a single model it reaches about 9600× compression on average, compared with at most about 1700× for DCVC-UF. At OP4, ENVC compresses 1.8× more than the highest-compression operating point of DCVC-UF (OP3) while its VQA accuracy is 3.3% higher.

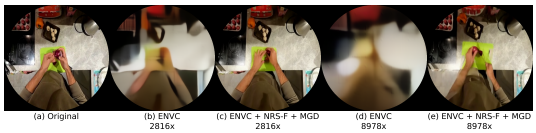


Figure 5: EgoSchema frame: (a) original; (b) ENVC, $qp = 8$; (c) MGD on (b); (d) ENVC, $qp = 0$; (e) MGD on (d).

Adding NRS-F and MGD substantially improves accuracy. Compared with ENVC at OP4, ENVC + NRS-F + MGD at OP4 compresses 1.6× more on average and achieves 12.2 points higher VQA accuracy on average. The gain comes from MGD restoring the details that ENVC discards: Figure 5 shows that at both $qp = 8$ and $qp = 0$, ENVC keeps only the coarse layout of the frame, while MGD restores sharp object boundaries and textures from the matched keyframes. At OP1, ENVC + NRS-F + MGD compresses 659× on average while losing only 2.0 points of accuracy relative to uncompressed frames on average, substantially better than baselines. NRS-A further improves on NRS-F by allocating bits according to the matched keyframes. Compared with NRS-F at OP3, NRS-A at OP2 compresses 1.6× more and still achieves 1.7% higher accuracy on average.

Figure 6 compares reconstruction quality in terms of LPIPS. ENVC + NRS-F + MGD obtains clearly lower LPIPS than all baselines across the whole range. At about 3000×, its LPIPS is 0.25–0.29 on the three benchmarks, whereas DCVC-UF already reaches 0.54–0.58 at about 1800×. Even at about 9000×, its LPIPS of 0.34–0.37 is on par with or lower than that of DCVC-UF at about 130×. The full PSNR, MS-SSIM, and LPIPS curves are in Appendix B.

Decoder	OP1	OP2	OP3	OP4	Avg.
Gemini 3 Pro Image	59.8	59.2	52.4	40.3	52.9
MGD (ours)	64.2	64.0	59.1	49.6	59.2

Table 3: VQA accuracy of Qwen3-VL-32B on EgoCompress when the ENVC + NRS-F outputs at each operating point of Table 1 are decoded by Gemini 3 Pro Image or by MGD.

Ref.	P25	P50	P75	Max
KF1	0.3 min	18.1 min	1.61 hr	51.8 hr
KF2	2.8 min	34.3 min	3.24 hr	51.8 hr
KF3	5.8 min	48.9 min	6.07 hr	51.8 hr

Table 4: Temporal distance from each non-keyframe to its three matched keyframes on EgoCompress.

5.3 Ablation Studies

Impact of keyframe number. Table 2 varies the number of matched keyframes given to MGD from one to three, at OP1 and OP4 of ENVC + NRS-F + MGD in Table 1. We report VQA accuracy with Qwen3-VL-32B on EgoCompress, together with PSNR, LPIPS, and MS-SSIM. More keyframes give better results on every metric at both operating points. We therefore use three keyframes in all other experiments.

OP	# KF	Acc.	PSNR	LPIPS	MS-SSIM
OP1 (667×)	1	60.8	26.71	0.2141	0.8758
	2	62.7	26.98	0.2043	0.8824
	3	64.2	**27.12**	**0.1990**	**0.8860**
OP4 (5229×)	1	47.9	20.62	0.3450	0.6184
	2	48.4	20.71	0.3375	0.6264
	3	49.6	**20.76**	**0.3335**	**0.6307**

Table 2: Effect of the number of matched keyframes (# KF) for ENVC + NRS-F + MGD at OP1 and OP4 of Table 1.

Comparison with an existing image generator. To test whether a general-purpose image generator can replace MGD, we use Gemini 3 Pro Image (Google DeepMind, 2025) as the decoder. It receives the same ENVC + NRS-F outputs as MGD, and is asked to reconstruct the original frame; we tuned its prompt for the best performance. Table 3 shows that MGD achieves higher VQA accuracy at all four operating points. The results show the effectiveness of MGD.

Keyframe statistics. Table 4 reports the temporal distance between each non-keyframe and its three matched keyframes, measured on the seven-day EgoLife video of EgoCompress. The matched keyframes have 75th-percentile distances of 4.1 hours, and matches reach 51.8 hours. MGD therefore reconstructs frames from content recorded hours or days earlier. Keyframes ratios are 1.8% on EgoCompress, 4.4% on EgoSchema, 1.1% on HD-EPIC, and 2.3% on EgoEverything.

6 Conclusion

We formulat persistent visual memory as an extreme-compression problem and identify long-horizon temporal redundancy in egocentric video. Building on this insight, we introduced a system combining an egocentric neural video codec (ENVC), a novelty-aware rate selector (NRS), and a memory-guided generative decoder (MGD). We also developed a long-horizon VQA benchmark to evaluate the impact of compression to downstream video understanding. Our system operates far beyond the compression range of existing baselines while substantially improving both reconstruction quality and downstream performance.

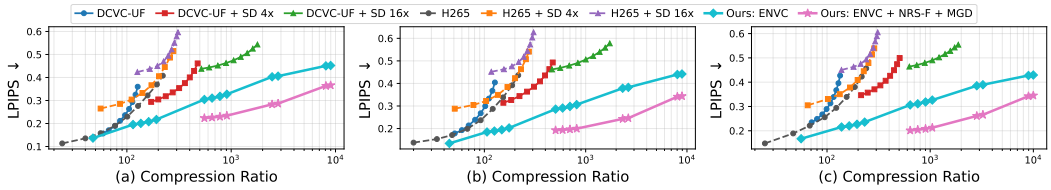


Figure 6: LPIPS versus compression ratio (lower is better) on (a) EgoCompress, (b) EgoSchema, and (c) HD-EPIC.

AI USE STATEMENT

The research ideas and core method are entirely the authors' own, and no AI was used to generate research ideas. Generative AI was used for three purposes: (1) to implement approximately 60% of the code under the authors' direction, (2) to polish the paper's language, and (3) to check mathematical derivations developed by the authors and improve their presentation. All AI-assisted content was reviewed by all authors, and AI-generated code was additionally validated using author-written test cases. The authors take responsibility for the final content of this work, including any text, claims, code, or artifacts produced with the aid of generative AI.

ACKNOWLEDGMENTS

Leave this out for the anonymous submission; fill it in for the camera-ready.

REFERENCES

AdaRD-key, 2025. AdaRD-key: Adaptive relevance-diversity keyframe sampling for long-form video understanding. *arXiv:2510.02778*, 2025.

Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Ballé, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Kirolos Ataallah et al. Goldfish: Vision-language understanding of arbitrarily long videos. In *European Conference on Computer Vision*, 2024. TODO: verify the full author list and page numbers.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

Leonard Bärmann and Alex Waibel. Where did I leave my keys? episodic-memory-based question answering on egocentric videos. In *CVPR Workshops*, pp. 1560–1568, 2022.

Black Forest Labs. FLUX.2: Frontier visual intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. doi: 10.1109/TCSVT.2021.3101953.

Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL `https://openaccess.thecvf.com/content/CVPR2024/html/Di_Grounded_Question-Answering_in_Long_Egocentric_Videos_CVPR_2024_paper.html`.

Ego-R1, 2025. Ego-R1: Chain-of-tool-thought for ultra-long egocentric video reasoning. *arXiv:2506.13654*, 2025.

Jakob Engel et al. Project Aria: A new tool for egocentric multi-modal AI research. *arXiv:2308.13561*, 2023.

FOCUS, 2025. FOCUS: Efficient keyframe selection for long video understanding. *arXiv:2510.27280*, 2025.

Gabriele Goletto, Tushar Nagarajan, Giuseppe Averta, and Dima Damen. AMEGO: Active memory from long egocentric videos. In *European Conference on Computer Vision*. Springer, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/02032.pdf.

Google DeepMind. Gemini 3 pro image. https://deepmind.google/models/gemini-image/pro/, 2025.

Google DeepMind. Gemini 3.1 pro. https://deepmind.google/models/gemini/pro/, 2026.

Kristen Grauman et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. arXiv:2110.07058.

Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. LifeLogging: Personal big data. *Foundations and Trends in Information Retrieval*, 8(1):1–125, 2014.

Bo He et al. MA-LMM: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. TODO: verify the full author list.

Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. SenseCam: A retrospective memory aid. In *International Conference on Ubiquitous Computing*, 2006.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu. Towards practical real-time neural video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18114–18125, 2021.

Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Jiahao Li, Wenxuan Xie, Zhaoyang Jia, Bin Li, Zongyu Guo, Xiaoyi Zhang, and Yan Lu. Ultra-fast neural video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11006–11015, 2019.

Zhaoyang Lv, Nicholas Charron, Pierre Moulon, Alexander Gamino, Cheng Peng, Chris Sweeney, Edward Miller, Huixuan Tang, Jeff Meissner, Jing Dong, et al. Aria Everyday Activities dataset. *arXiv preprint arXiv:2402.13349*, 2024.

Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, et al. Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In *European Conference on Computer Vision*, 2024.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. DrVideo: Document retrieval based long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Memorize-and-Recall, 2020. Placeholder: memorize-and-recall surveillance video compression, 2020.

Meta. Ray-Ban Meta smart glasses: Technical specifications. `https://www.meta.com/ai-glasses/`, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, Jacob Chalk, Zhifan Zhu, Rhodri Guerrier, Fahd Abdelazim, Bin Zhu, Davide Moltisanti, Michael Wray, Hazel Doughty, and Dima Damen. HD-EPIC: A highly-detailed egocentric video dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. MovieChat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*, 2024. doi: 10.48550/arXiv.2404.17176. URL `https://arxiv.org/abs/2404.17176`.

Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

Qiance Tang, Ziqi Wang, Jieyu Lin, Ziyun Li, Barbara De Salvo, and Sai Qian Zhang. EgoEverything: A benchmark for human behavior inspired long context egocentric video understanding in AR environment, 2026. URL `https://arxiv.org/abs/2604.08342`.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. LVBench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024.

Ying Wang, Yanlai Yang, and Mengye Ren. LifelongMemory: Leveraging LLMs for answering queries in long-form egocentric videos. *arXiv:2312.05269*, 2023.

Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems and Computers*, 2003.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. EgoLife: Towards egocentric life assistant. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28885–28900. IEEE, 2025.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

A Implementation Details

ENVC training. We train ENVC on an internal dataset of 1.5M egocentric images at 512×512 with $\mathcal{L}_{\text{ENVC}}$, using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 10^{-4}. Training takes 120 epochs on 64 H200 GPUs.

MGD training. We build the MGD training set from Aria Everyday Activities (Lv et al., 2024), Nymeria (Ma et al., 2024), HD-EPIC, EgoLife (Yang et al., 2025), and Ego4D (Grauman et al., 2022), 1.2M samples in total; the training videos do not overlap with any evaluation video. Each sample consists of an original frame, its ENVC-compressed version at a non-keyframe qp, and one to three matched keyframes selected by NRS. We train MGD with the flow-matching loss $\mathcal{L}_{\text{MGD}}$ (Lipman et al., 2023), applying LoRA with rank 16 and scaling factor 32 to the attention and MLP layers of the DiT. We use AdamW with a learning rate of 10^{-4}, weight decay 0.01, and a cosine schedule, and train for 20 epochs on 32 H200 GPUs.

B Additional Results

Figure 7 shows examples of the qp selected by NRS-A (Section 3.3).



Figure 7: (a) original current frame (b) matched keyframe 1 (c) matched keyframe 2 (d) matched keyframe 3 (e) compressed current-frame with compression ratio selected by NRS-A. The NRS-A selects higher/lower compression ratio when the current-frame is similar/different to keyframes in the upper/bottom row.

Figure 8 reports PSNR, MS-SSIM, and LPIPS versus compression ratio for all methods on EgoCompress, EgoSchema, and HD-EPIC. PSNR is computed inside the circular field of view; MS-SSIM and LPIPS are computed on the full frame. For our methods, the compression ratio is that of the non-keyframes.

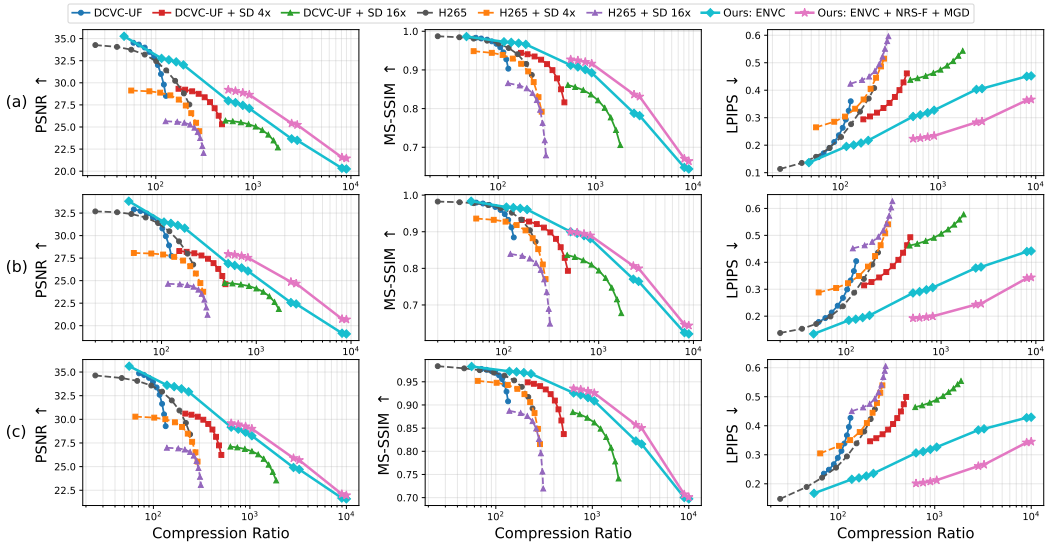


Figure 8: PSNR, MS-SSIM (higher is better), and LPIPS (lower is better) versus compression ratio on (a) EgoCompress, (b) EgoSchema, and (c) HD-EPIC.