DSAD: Dynamic Soft Anisotropic Diagrams for Reduced-Order Video Representation

Zhiyang Dou
MIT CSAIL

Laki Iinbor
Independent Researcher

Wojciech Matusik
MIT CSAIL

Abstract

Explicit video primitives decompose a group of pictures (GoP) into local units that can be rendered, inspected, and edited. This is challenging for implicit neural representations, which typically encode a clip holistically in network weights. Existing GoP-based methods model appearance with explicit primitives such as 2D Gaussians, but their motion network consumes appearance budget and obscures which parameters move which region. Our insight is that in a short GoP, nearby primitives often move together, so the rule that decides which primitive owns a pixel can also decide which primitives share a motion, and a small set of shared motions can efficiently encode a GoP's deformation. We introduce Dynamic Soft Anisotropic Diagrams (DSAD), which apply one soft-distance geometry at two scales: dense sites of a *Soft Anisotropic Diagram* (SAD), a top-K-masked softmax over anisotropic Apollonius scores, own local colors, and coarser *anchors* reuse the same geometry to form a partition-of-unity motion basis, which a *reduced-order model* (ROM) pairs with knot-based temporal coefficients. The ROM uses only 0.24–0.46 of the parameters read by D2GV's MLP, which frees budget for appearance. On Bunny, UVG, and DAVIS, DSAD has the highest PSNR among compared GoP primitives, though D2GV-AR has higher MS-SSIM on UVG. Against D2GV with its original MLP at 1280×720, at matched training length and decoder-read parameters, DSAD gains a mean $+1.92$ dB and wins 223 of 302 GoP–budget pairs, leading on every DAVIS scene while D2GV leads on 3 of 7 UVG scenes and 3 of 32 in a wider DAVIS pool; the margin holds at 1080p. DSAD decodes at 706–$1{,}095$ FPS from a cold start on one RTX 2080 Ti. Project page: https://frank-zy-dou.github.io/projects/DSAD/.

1 Introduction

Decomposing a video into short groups of pictures (GoPs) allows each clip to be fitted and decoded in parallel under a dedicated parameter budget. Within each GoP, representing the signal with explicit local primitives is attractive: each primitive accounts for a local image region, moves coherently with the scene, and exposes a discrete unit that can be inspected, ranked, pruned, or edited. Designing such a representation requires deciding both how primitives partition spatial appearance and how they parameterize temporal motion under a shared budget.

Existing video representations either sacrifice spatial locality or restrict it to appearance. Implicit neural representations (INRs) such as HNeRV [8] and HiNeRV [19] amortize coordinate networks or convolutional decoders over an entire sequence; fitting one network to a full clip yields strong global reconstruction quality, with HiNeRV achieving the highest overall PSNR in our UVG/DAVIS comparisons (Tab. 2), but provides neither block-level streaming nor spatially localized units. Conversely, GoP primitive methods such as D2GV [26] and D2GV-AR [27] model canonical appearance explicitly with 2D Gaussians, yet delegate temporal deformation to an implicit neural module: a $107{,}646$-parameter coordinate MLP in D2GV, or an MLP paired with a neural ODE in D2GV-AR. This hybrid design couples explicit appearance primitives to an opaque motion network.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

For short GoPs, predicting per-primitive trajectories with a dense neural network is also poorly matched to scene motion. Because neighboring primitives often share camera, object, or background motion, the deformation field has a lower intrinsic dimension than the number of primitives it moves. Classical reduced-order modeling (ROM) captures this structure by factoring a field into spatial modes and temporal coefficients. Adopting a reduced-order formulation for video primitives, however, requires a spatial basis that is local, differentiable, and explicit enough to be inspected, ranked, and edited.

Canonical appearance suggests such a basis from another direction. In Gaussian splatting, overlapping elliptical kernels blend across boundaries without assigning clear responsibility for a pixel or edge. Partitioning the canonical image plane with a Soft Anisotropic Diagram (SAD) [15] instead assigns each location sparse, normalized ownership weights over competing generators. Evaluated at a coarser scale over a smaller set of anchors, the same soft-distance rule defines spatial motion modes, so canonical appearance and reduced-order motion share a single geometric foundation.

We instantiate this formulation as Dynamic Soft Anisotropic Diagrams (DSAD). SAD's top-K-masked softmax over anisotropic Apollonius distances defines a differentiable partition of unity on the image plane, and DSAD applies it twice: each pixel takes its color from a few dense sites, and each site takes its motion from a few coarser anchors, much as a deformable mesh follows its handles. The site-to-anchor weights set where motion acts and one-dimensional temporal knots set how it changes over time, so the ROM has no neural layers and its size depends on anchors and knots, not on sites. Sites control appearance detail, anchors the spatial resolution of motion, and knots its temporal bandwidth, and the budget can be split among them directly: adding sites refines appearance without enlarging the motion model, and parameters saved against D2GV's original MLP become additional sites. Since the mask gives distant anchors exactly zero weight, zeroing one anchor's displacement changes a median of 2.2–3.5% of the pixels on 4 tested DAVIS GoPs and leaves the others bit-identical at 8 bits (Sec. 3). Voronoi, Apollonius, power/Laguerre, and anisotropic Apollonius diagrams arise as parameter-tied cases of this family, with anisotropic Apollonius as our default.

We evaluate DSAD through cross-dataset benchmarks and controlled paired experiments. For a fair comparison, we follow the D2GV-AR [27] evaluation protocol on UVG and DAVIS (Tab. 2), where DSAD achieves the highest PSNR among compared GoP primitive methods across all budget tiers. A stricter paired control fits D2GV with its original MLP at matched training length and per-GoP decoder-read parameter budgets (within 0.61% in Tab. 2) to isolate the effect of replacing Gaussian+MLP with SAD appearance and anchor-ROM motion at both 1280×720 and 1920×1080 (Sec. 4.2). Component ablations separate appearance and motion gains, and optical-flow spectral analysis relates scene-level margins to low-rank motion structure. Our contributions are:

- We introduce a soft-distance primitive family whose dense sites decode canonical appearance and whose coarser anchors form a partition-of-unity deformation basis, eliminating neural-network layers from the per-GoP representation;
- We propose an anchor-based reduced-order motion model whose rank is independent of the canonical primitive count and whose localized anchors support direct motion edits (Appendix J);
- We provide controlled evaluations across INR and primitive baselines: at 1280×720, DSAD gains a mean $+1.92$ dB over paired D2GV (223 of 302 GoP–budget pairs won), and $+2.21$ dB at 1920×1080 (266 of 302 pairs won at 60k fine iterations), with SAD appearance supplying the largest Bunny gain and cold-start decoding averaging 706–$1{,}095$ FPS at 720p on one RTX 2080 Ti.

2 Related work

Video representations. Implicit video representations such as NeRV [7], E-NeRV [22], FFNeRV [21], HNeRV [8], and HiNeRV [19] fit video sequences through coordinate networks, frame-conditioned decoders, or learned feature grids. By amortizing parameters across an entire clip, a single network can exploit long-range temporal redundancy; HiNeRV pushes this regime further by pairing multiscale feature grids with a deep convolutional decoder, decoding at CNN rates (Tab. C5) and achieving the highest overall PSNR in our UVG and DAVIS comparisons. Our matched-budget reproduction of HiNeRV (without post-training compression) exceeds per-GoP DSAD by 0.6–3.5 dB across all four UVG and DAVIS splits (Tab. 2); on Bunny, published full-clip fits of HiNeRV (41.14 dB at 3.25 M parameters) and HNeRV-Boost [56] (41.09 dB at 3.08 M) similarly exceed our per-GoP fit of

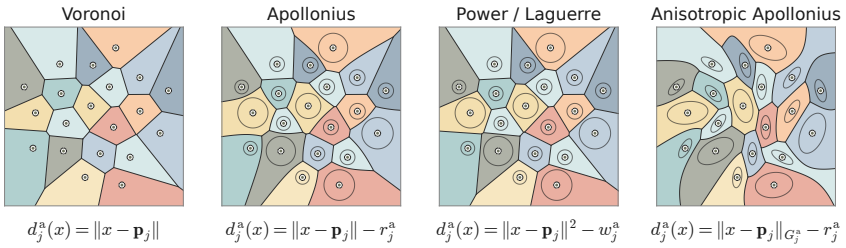


Figure 1: **Soft-distance diagram family.** Illustrative panels share anchors (dots), with cells minimizing Eq. (3). Circles and equal-area ellipses show each anchor's scale and metric. DSAD softens the SAD score with the top-K partition of Eq. (4).

39.20 dB at 3.00 M (Tab. I12; Appendix C discusses the INR codecs CANeRV [42], HANeRV [43], and GIViC [12]). However, full-clip INRs neither support streamable block encoding nor expose spatially localized units for inspecting motion or pixel ownership; our comparative claims are therefore scoped to GoP primitive methods.

Primitive-based representations take the complementary approach of rasterizing explicit local elements. Extending static and deformable Gaussian splatting [17, 55, 54], D2GV [26] and D2GV-AR [27] represent each video GoP with canonical 2D Gaussians animated by a time-conditioned MLP or a neural ODE. While this exposes editable canonical primitives, temporal deformation remains an implicit neural mapping. Contemporaneous STGV [25] drives Gaussian deformation with spatio-temporal hash encodings under a closely related GoP protocol, though public code for a paired comparison is unavailable (Appendix C). Keeping the explicit per-GoP setting, we ask whether canonical appearance support and temporal motion support can be unified within a single explicit coordinate system.

Motion structure. Dynamic NeRF and Gaussian frameworks typically parameterize deformation through coordinate MLPs [37, 34, 46, 23, 28], per-frame primitive states [30], attention modules [16], skeletal priors [33], or feed-forward predictors [24]. Although expressive, these parameterizations entangle where motion occurs in space with how strongly each pattern is activated over time. Classical reduced-order modeling (ROM) separates these axes by factoring a field into spatial modes and temporal coefficients [14, 2, 40]. While standard snapshot bases such as SVD do not directly yield renderable primitives, DSAD brings the ROM factorization into per-GoP primitive fitting: spatial modes are represented by learned local anchors, temporal trajectories by knot-based curves, and modal weights by the same soft-distance geometry used for canonical appearance ownership.

Tessellations. Voronoi [47, 4], Apollonius [11], power (Laguerre) [3], and anisotropic diagrams partition space among competing generators with progressively richer geometry, controlling cell ownership through location, additive radius, anisotropic metric, and softmax temperature (Fig. 1, Appendix G). Such tessellations have a long history in geometric processing, including surface reconstruction, meshing, motion planning, and shape skeletonization [1, 53, 41, 10, 50]. Recent work has relaxed Voronoi-style partitions into differentiable primitives for learning: VoronoiNet [51] uses soft Voronoi cells as function approximators, VoroMesh [31] and VoroLight [29] optimize generators for surface and volumetric representations, Differentiable Surface Triangulation [38] smooths triangulation objectives, Spherical Voronoi [9] partitions the unit sphere for view-dependent appearance, and RadiantFoam [13] employs volumetric-mesh cells for differentiable ray tracing. Most closely related to our motion basis and appearance decoder, SkinCells [20] learns anisotropic Voronoi cells as sparse top-l skinning weights for rigged meshes, and SAD [15] differentiates anisotropic Apollonius diagrams for static image fitting. We extend the SAD soft-distance family to video by using the tessellation as a shared dual-scale coordinate chart from which both pixel-level RGB ownership and primitive-level ROM motion weights are read.

3 Method

DSAD parameterizes canonical appearance and temporal deformation through a shared local ownership geometry in the canonical image plane (Fig. 2). Whereas D2GV [26] pairs N^{GS} canonical 2D Gaussians with a $107{,}646$-parameter coordinate MLP that predicts per-primitive temporal residuals,

Table 1: **Notation.** Sites are indexed by i, anchors by j, and knots by k. Anchor scalars carry a superscript a. Parentheses indicate default values in the main experiments.

Appearance sites, $i = 1, \dots, N$		*Motion anchors*, $j = 1, \dots, M$	
N	sites per GoP	M	anchors per GoP, ROM rank (256; Sec. 3.4)
$\mathbf{s}_i,\ \mathbf{c}_i$	canonical position, RGB color	$\mathbf{p}_j$	canonical position
$r_i,\ G_i,\ \tau_i$	radius, metric, temperature	$r_j^{\mathrm{a}}, w_j^{\mathrm{a}}, G_j^{\mathrm{a}}, \tau_j^{\mathrm{a}}$	radius, power weight, metric, temp.
$d_i(x)$	score $\|x - \mathbf{s}_i\|_{G_i} - r_i$	$d_j^{\mathrm{a}}(x)$	score, Eq. (3)
$\mathcal{O}_{\mathrm{s}}(x)$	site mask, K_{s} sites (8)	$\mathcal{O}_{\mathrm{a}}(x)$	anchor mask, K_{a} anchors (12)
$\psi_i(x)$	ownership weight, Eq. (5)	$\phi_j(x)$	motion-basis weight, Eq. (4)
Time and reduced-order motion, channels $\kappa \in \{\Delta\mathbf{s}, \Delta\mathbf{f}\}$			
$t,\ T$	frame time in $[0, 1]$, frames per GoP	K_{t}	knots per anchor and channel (8)
$\Delta\mathbf{s}_i(t),\ \Delta\mathbf{f}_i(t)$	site displacement, color residual	$t_{j,k}^{\kappa}, \rho_{j,k}^{\kappa}, \mathbf{v}_{j,k}^{\kappa}$	knot time, bandwidth, value
Φ	$N \times M$, $\Phi_{ij} = \phi_j(\mathbf{s}_i)$	$\mathbf{a}_j^{\kappa}(t),\ A^{\kappa}(t)$	anchor coefficient, its stack over j
$\Delta(x, t)$	field $\sum_j \phi_j(x)\, \mathbf{a}_j^{\Delta\mathbf{s}}(t)$		

we replace both components with soft-distance tessellations: dense SAD sites govern pixel-level RGB ownership, and coarser ROM anchors interpolate primitive residuals through the same soft-distance rule. This design defines two controlled replacement axes relative to D2GV: substituting SAD sites for 2D Gaussians in appearance, and substituting an anchor-based ROM for the coordinate MLP in motion. Our paired D2GV baseline retains its original $107{,}646$-parameter MLP; where noted, we also evaluate a $9{,}678$-parameter single-layer MLP that implements the lightweight architecture described in D2GV-AR [27] without its neural-ODE module, as official code for D2GV-AR's MLP is not publicly available (Appendix B).

Formally, each GoP spans T frames at normalized times $\mathcal{T} \subset [0, 1]$, with pixel coordinates $x \in \mathbb{R}^2$. Let $\Theta_i = (\mathbf{s}_i, r_i, G_i, \tau_i, \mathbf{c}_i)$ denote the i-th canonical SAD site and $\mathcal{P} = \{\mathbf{p}_j\}_{j=1}^{M}$ denote the motion anchors, with $M \ll N$. At frame time t, DSAD renders

$$\begin{aligned} \Theta_i(t) &= \big(\mathbf{s}_i + \Delta\mathbf{s}_i(t),\, r_i,\, G_i,\, \tau_i,\, \mathbf{c}_i + \Delta\mathbf{f}_i(t)\big), \qquad i = 1, \dots, N, \\ \hat{I}_t &= \mathcal{R}_{\mathrm{SAD}}\big(\{\Theta_i(t)\}_{i=1}^{N}\big), \qquad \big[\Delta S(t), \Delta F(t)\big] = \Phi\big[A^{\Delta\mathbf{s}}(t), A^{\Delta\mathbf{f}}(t)\big], \end{aligned} \tag{1}$$

where $\mathcal{R}_{\mathrm{SAD}}$ is the SAD renderer of Sec. 3.2 and site displacements $\Delta\mathbf{s}_i(t)$ are applied prior to the coordinate $\tanh$ map (Appendix B). Here $\Delta S(t) \in \mathbb{R}^{N\times 2}$ and $\Delta F(t) \in \mathbb{R}^{N\times 3}$ stack per-site position and color residuals, while $A^{\Delta\mathbf{s}}(t) \in \mathbb{R}^{M\times 2}$ and $A^{\Delta\mathbf{f}}(t) \in \mathbb{R}^{M\times 3}$ stack temporal anchor coefficients (Sec. 3.3). The time-invariant mixing matrix

$$\Phi_{ij} = \phi_j(\mathbf{s}_i), \qquad \Phi\mathbf{1} = \mathbf{1}, \qquad \|\Phi_{i:}\|_0 \le K_{\mathrm{a}} \tag{2}$$

evaluates the anchor weights ϕ_j of Eq. (4) at the canonical site positions, forming a sparse partition of unity with at most K_{a} nonzeros per row kept by the anchor mask of Sec. 3.1. Because canonical geometry fixes Φ, only the low-rank anchor coefficients $A^{\Delta\mathbf{s}}(t)$ and $A^{\Delta\mathbf{f}}(t)$ vary with time, constraining primitive trajectories directly to the subspace spanned by Φ rather than approximating a dense neural field after training. Tab. 1 summarizes our notation.

We use the term *diagram* in the geometric sense of a spatial partition induced by competing generators: for any query point, nearby active generators define sparse, nonnegative coordinates that sum to one and carry explicit spatial support. Applying this coordinate rule at two scales assigns pixel colors to dense sites and primitive trajectories to coarser anchors. Because each anchor has bounded spatial support, individual motion modes can be inspected, ranked by coefficient energy, and edited locally (Appendix J); zeroing a single anchor's displacement coefficients modifies a median of 2.2–3.5% of the pixels across 4 tested DAVIS GoPs while leaving all remaining pixels bit-identical at 8-bit precision (we leave anchor pruning and progressive coding to future work).

3.1 Shared anchor tessellation

Each anchor j is parameterized by a position $\mathbf{p}_j \in \mathbb{R}^2$, a temperature $\tau_j^{\mathrm{a}} > 0$, and optional shape parameters: an additive radius r_j^{a}, a power weight w_j^{a}, or an anisotropic metric $G_j^{\mathrm{a}} = G(\alpha_j^{\mathrm{a}}, \hat{u}_j^{\mathrm{a}})$. Appearance sites similarly use $G_i = G(\alpha_i, \hat{u}_i)$, where $G(\alpha, \hat{u}) = e^{\alpha}\hat{u}\hat{u}^{\top} + e^{-\alpha}(I - \hat{u}\hat{u}^{\top})$ has unit

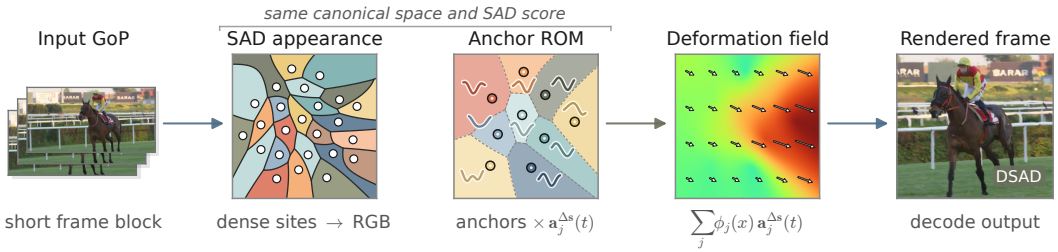


Figure 2: **DSAD representation.** Given observed GoP frames, dense SAD cells govern local RGB ownership while coarser ROM anchors with knot-based temporal coefficients induce the low-rank deformation field that warps canonical sites to render each frame. Appearance and motion share the same soft-distance geometry.

determinant for unit $\hat{u}$ and induces the Mahalanobis norm $\|v\|_G = \sqrt{v^\top G v}$ (Appendix D). Varying only the generator score yields four classical diagram families:

$$d_j^{\mathrm{a}}(x) = \begin{cases} \|x - \mathbf{p}_j\| & \text{Voronoi,} \\ \|x - \mathbf{p}_j\| - r_j^{\mathrm{a}} & \text{Apollonius,} \\ \|x - \mathbf{p}_j\|^2 - w_j^{\mathrm{a}} & \text{Power (Laguerre),} \\ \|x - \mathbf{p}_j\|_{G_j^{\mathrm{a}}} - r_j^{\mathrm{a}} & \text{SAD (anisotropic Apollonius).} \end{cases} \tag{3}$$

A top-K mask retains the generators with the largest logits ($K = K_{\mathrm{a}}$ for anchors and K_{s} for sites). For anchors, the logit is $-d_j^{\mathrm{a}}(x)/(\tau_j^{\mathrm{a}})^\beta$, and the active set forms the *anchor mask* $\mathcal{O}_{\mathrm{a}}(x)$. Applying a restricted softmax over $\mathcal{O}_{\mathrm{a}}(x)$ converts the hard diagram into a local partition of unity:

$$\phi_j(x) = \begin{cases} \dfrac{\exp\!\left(-d_j^{\mathrm{a}}(x)/(\tau_j^{\mathrm{a}})^\beta\right)}{\sum_{j' \in \mathcal{O}_{\mathrm{a}}(x)} \exp\!\left(-d_{j'}^{\mathrm{a}}(x)/(\tau_{j'}^{\mathrm{a}})^\beta\right)} & j \in \mathcal{O}_{\mathrm{a}}(x), \\ 0 & \text{otherwise,} \end{cases} \qquad \beta = \begin{cases} 2 & \text{Power,} \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

The weights in Eq. (4) are continuous everywhere except on the measure-zero locus where the K_{a}-th and $(K_{\mathrm{a}}{+}1)$-th logits tie. We initialize anchors on a sunflower lattice and zero all score-specific scalars so that all four diagram variants start training from the same isotropic Voronoi partition; DSAD uses the SAD score by default.

Unlike an unnormalized radial basis layer, Eq. (4) enforces explicit local competition: generators outside $\mathcal{O}_{\mathrm{a}}(x)$ carry zero weight, preventing distant anchors from consuming capacity, while the score $d_j^{\mathrm{a}}(x)$ controls cell geometry prior to softmax normalization. Voronoi scores yield isotropic cells, Apollonius and power scores adjust support through additive radii or squared-distance weights, and the SAD score further aligns cells with directional structure via G_j^{a}. Because all four scores share the same downstream pipeline, we can ablate cell geometry in isolation.

3.2 SAD appearance decoder

Canonical appearance uses the same SAD geometry at the primitive scale. Each site i scores pixel x via $d_i(x) = \|x - \mathbf{s}_i\|_{G_i} - r_i$, and the *site mask* $\mathcal{O}_{\mathrm{s}}(x)$ retains the $K_{\mathrm{s}} = 8$ sites with the largest logits $-d_i(x)/\tau_i$. An acceleration structure maintains an approximate top-K_{s} candidate set during most of training, whereas the final iterations, evaluation, and decoding use the exact top-K_{s} set computed by our fused CUDA rasterizer (Appendix E). The rendered color at pixel x is

$$\hat{I}(x) \;=\; \sum_{i \in \mathcal{O}_{\mathrm{s}}(x)} \psi_i(x)\, \mathbf{c}_i, \quad \psi_i(x) \;=\; \frac{\exp\!\left(-d_i(x)/\tau_i\right)}{\sum_{i' \in \mathcal{O}_{\mathrm{s}}(x)} \exp\!\left(-d_{i'}(x)/\tau_{i'}\right)}. \tag{5}$$

Each SAD site stores 9 parameters whereas a D2GV Cholesky Gaussian is read as 8 (Appendix B), so matching canonical budgets sets $N = \frac{8}{9} N^{\mathrm{GS}}$. Unlike Gaussian splatting, where boundary pixels blend overlapping elliptical footprints via opacity and radial falloff, Eq. (5) assigns pixel color through normalized local competition among at most K_{s} sites. Although cell boundaries need not coincide with image edges, we expect this normalized competition to sharpen high-contrast transitions and

give the motion module cleaner canonical locations: in our paired comparisons, DSAD leads D2GV by $+2.77$ dB within 2 px of Canny edges versus $+1.71$ dB elsewhere, achieving a larger near-edge gain on 20 of 26 GoP–budget pairs (Appendix J).

3.3 Anchor-ROM temporal model

The anchor tessellation provides the spatial modes of our reduced-order deformation model. Each anchor j carries a temporal coefficient curve per residual channel $\kappa \in \{\Delta\mathbf{s}, \Delta\mathbf{f}\}$, parameterized by K_t learnable knots with times $t^\kappa_{j,k}$, bandwidths $\rho^\kappa_{j,k}$, and values $\mathbf{v}^\kappa_{j,k}$ ($\mathbb{R}^2$ for $\Delta\mathbf{s}$ and $\mathbb{R}^3$ for $\Delta\mathbf{f}$):

$$\mathbf{a}^\kappa_j(t) = \sum_{k=1}^{K_\mathrm{t}} \omega^\kappa_{j,k}(t)\, \mathbf{v}^\kappa_{j,k}, \qquad \omega^\kappa_{j,k}(t) = \mathrm{softmax}_k\Big(-\tfrac{(t-t^\kappa_{j,k})^2}{2(\rho^\kappa_{j,k})^2}\Big), \tag{6}$$

where the softmax normalizes across all K_t knots of anchor j, and zero-initializing $\mathbf{v}^\kappa_{j,k}$ starts training from identity motion. Stacking $\mathbf{a}^\kappa_j(t)$ across anchors into $A^\kappa(t)$ yields the primitive residuals

$$\big(\Delta S(t),\, \Delta F(t)\big) = \big(\Phi\, A^{\Delta\mathbf{s}}(t),\, \Phi\, A^{\Delta\mathbf{f}}(t)\big), \qquad \Phi \in \mathbb{R}^{N\times M}, \tag{7}$$

with Φ of Eq. (2), as in Eq. (1); equivalently, $\Delta\mathbf{s}_i(t) = \Delta(\mathbf{s}_i, t)$ samples the continuous displacement field $\Delta(x,t) = \sum_j \phi_j(x)\, \mathbf{a}^{\Delta\mathbf{s}}_j(t)$ at the canonical sites. Following D2GV's renderer, which ignores opacity offsets during rasterization, both methods omit opacity residuals (Appendix F).

With 3-channel RGB feature residuals enabled, the ROM motion head contains

$$P_\mathrm{ROM} \;=\; \underbrace{M(3+\nu)}_{\text{spatial basis}} \;+\; \underbrace{4\,MK_\mathrm{t}}_{\Delta\mathbf{s}:\, t+\rho+\mathbf{v}\in\mathbb{R}^2} \;+\; \underbrace{5\,MK_\mathrm{t}}_{\Delta\mathbf{f}:\, t+\rho+\mathbf{v}\in\mathbb{R}^3} \tag{8}$$

parameters, where 3 counts anchor position and temperature and ν counts score-specific scalars per anchor (Voronoi $\nu=0$, Apollonius and Power $\nu=1$, SAD $\nu=4$). Because site shapes (G_i, r_i) remain fixed in the canonical frame, all temporal deformation acts through the precomputed mixing matrix Φ, which is evaluated once from the canonical diagram and reused across all frames; extensions to per-site affine warps or time-varying metrics are costed in Tab. G9 and left for future work.

Equation (7) makes the reduced-order assumption operational: temporal degrees of freedom scale with modal rank M rather than primitive count N, so increasing N refines canonical appearance without expanding the motion head. At our default $M=256$ and $K_\mathrm{t}=8$, P_ROM totals 19,200–20,224 parameters, which is roughly $4.1\times$ smaller than the original D2GV MLP (0.24–0.46 of the MLP's decoder-read count across the budgets of Tab. 2; Appendix B). On Bunny and in our paired UVG and DAVIS comparisons against D2GV with its original MLP, these saved deformation parameters are reallocated directly to additional canonical appearance sites. Conversely, although the ROM at $M=256$ is $2.1\times$ the size of the 9,678-parameter lightweight MLP, DSAD achieves a larger PSNR margin over lightweight D2GV than over original-MLP D2GV under the 60k protocol (Tab. J23), so the gain does not rely on parameter reallocation alone. Structurally, each row of Φ specifies which anchors govern a primitive and each row of $A^\kappa(t)$ specifies how those anchors move over time, making the deformation model a sparse partition-of-unity interpolant in which M controls spatial modal resolution, K_t controls temporal bandwidth, and N controls appearance detail.

3.4 Training objective

For a fair comparison, training follows D2GV's coarse/fine protocol: 10k canonical coarse iterations followed by joint fine iterations (120k iterations in Tab. 2 following D2GV-AR, and 60k iterations elsewhere unless stated) under the L2 reconstruction objective

$$\min_{\Theta,\,\mathcal{P},\,\mathcal{V}} \sum_{t\in\mathcal{T}} \|\hat{I}_t - I_t\|_2^2 \tag{9}$$

over site parameters $\Theta = \{\Theta_i\}$, anchor parameters $\mathcal{P}$, and knot parameters $\mathcal{V} = \{t^\kappa_{j,k}, \rho^\kappa_{j,k}, \mathbf{v}^\kappa_{j,k}\}$, where I_t is the observed frame. Canonical sites are initialized from the GoP's temporal-mean frame, half uniformly and half weighted by image-gradient magnitude, and each training pass iterates over frames in a fresh deterministically seeded random permutation; the final 6k fine iterations replace approximate candidate tracking with exact top-8 site queries. The anisotropic axis of G_i is optimized

as a 2-vector and stored as a single angle (9 numbers per site), with results under the earlier 10-number storage convention also reported (Tab. 2, Appendix B). We set $M = 256$ by default and scale anchor count at larger UVG and DAVIS budgets via $M = 32\lfloor 8B/B_{\mathrm{S}} + \frac{1}{2} \rfloor$ for GoP budget B and 720p Small budget B_{S}, re-matching N to the budget target (Appendix J). Auxiliary movement, temporal-smoothness, and as-rigid-as-possible penalties (Eq. (11), Appendix F) are disabled (weight zero), as enabling them alters Bunny PSNR by only -0.05 dB under the 60k protocol.

4 Experiments

4.1 Datasets and metrics

For a fair comparison, we follow the 1280×720 GoP evaluation protocol of D2GV-AR [27] on UVG [32] and DAVIS [35, 36] at Small (3.5/1.8 M) and Large (5.8/3.3 M) per-video budgets, and use Big Buck Bunny [5] for ablations (Sec. 4.3). Reconstruction quality is measured by frame-averaged PSNR and MS-SSIM [49] on frames decoded from stored parameters, with decode FPS timed on a single RTX 2080 Ti. Following D2GV-AR's E-NeRV [22] convention, we subsample every fourth frame from 120 fps sequences into 10-frame GoPs, with a few scene-specific exceptions; D2GV-AR's settings table lists 5-frame GoPs for four UVG scenes, and evaluations on consecutive-frame UVG GoPs are labeled. In Tab. 2, rows for D-3DGS, D2GV-AR, and all INRs except HiNeRV are transcribed from D2GV-AR, while HiNeRV is our reproduction (Tab. B4).

Unless noted, DSAD uses the training settings of Sec. 3.4. Our primary primitive evaluation is strictly paired: DSAD and D2GV [26] fit identical GoPs with Adan [52], an L2 loss, and 120k fine iterations, each using its own rasterizer and learning-rate schedules stretched to 120k. Per-video budgets count parameters read by the decoder: 9 scalars per site plus the ROM for DSAD, and 8 per Gaussian plus 82,997 from the original MLP for D2GV (Appendix B), keeping DSAD within 0.61% above and 0.46% below D2GV's total. D2GV otherwise retains its original setup, whose MLP outperformed the lightweight MLP by $+2.49$ dB at equal budget on 60k UVG GoP$_1$ tests (Appendix J).

4.2 Main comparison

Tab. 2 separates methods by representation regime: full-clip INRs amortize one network over the entire video for global context, whereas GoP primitive methods fit independent, streamable blocks. Across all four UVG and DAVIS budget columns, DSAD achieves the highest PSNR among compared GoP primitives, exceeding D2GV-AR's reported rows by $+0.47$–$+4.15$ dB. The two regimes are ranked separately, with ties sharing a rank: **1st**, 2nd, 3rd.

Fig. 4 compares reconstructions on a UVG-Small GoP chosen before the comparison, cropped to the region of largest ground-truth motion. DSAD preserves the gate's thin structures more sharply than D2GV, gaining $+2.42$ dB on this GoP against a UVG-Small median margin of $+0.58$ dB, while full-clip HiNeRV remains sharpest overall.

On UVG, D2GV-AR attains higher MS-SSIM; its training loss includes an SSIM term, whereas DSAD and D2GV optimize L2 alone; adding an untuned SSIM term to DSAD lowered both metrics (Appendix J). Among full-clip INRs, which fit a single network per video outside our GoP primitive scope, our matched-budget HiNeRV [19] reproduction leads DSAD by 0.6–3.5 dB across all four splits; on UVG, transcribed HNeRV also leads by 0.09 and 0.75 dB in PSNR, and E-NeRV, FFNeRV, and HNeRV (Large) lead in MS-SSIM.

In the paired 1280×720 control, DSAD improves on 223 of 302 GoP–budget pairs with a mean gain of $+1.92$ dB, winning 103 of 106 DAVIS pairs and 120 of 196 UVG pairs when GoPs are weighted equally; the bottom row of Tab. 2 gives per-video split means and win counts, all positive. Under the earlier ten-scalar-per-site count, which reduces site count by 10% (Appendix B), DSAD retains a $+1.73$ dB mean margin ($217/302$) and still leads all compared GoP primitives in PSNR across all four columns of Tab. 2. Two additional benchmarks under the 60k protocol show consistent gains: across all 343 GoPs (24 clips) of MCL-JCV [48] at the UVG-Small budget, DSAD gains a mean $+2.64$ dB and wins 327 pairs (Appendix J), and across a wider pool of 70 DAVIS GoPs from 32 scenes at the Bunny-ablation budget, it gains a mean $+2.96$ dB (Tab. J17).

Higher resolution. At 1920×1080 under the 60k protocol and Tab. 2's budgets, DSAD achieves a paired mean gain of $+2.21$ dB (266 of 302 pairs won), spanning $+1.15$–$+4.16$ dB across splits and

Table 2: **Main cross-dataset comparison under the D2GV-AR budgets.** Per-video PSNR (dB) and MS-SSIM (higher is better). Italics denote results transcribed from [27]; gray numbers indicate full-clip INRs outside the GoP primitive class, with the two classes ranked separately. The unranked second DSAD row uses the ten-scalar-per-site storage count (10% fewer sites; Appendix B).

Method	Source	UVG–Small PSNR	UVG–Small MS-SSIM	UVG–Large PSNR	UVG–Large MS-SSIM	DAVIS–Small PSNR	DAVIS–Small MS-SSIM	DAVIS–Large PSNR	DAVIS–Large MS-SSIM
Full-clip INR video representations									
NeRV [7]	from [27]	*33.04*	*0.9486*	*33.40*	*0.9521*	*26.08*	*0.8699*	*26.82*	*0.8902*
E-NeRV [22]	from [27]	*33.48*	*0.9675*	*35.02*	*0.9712*	*27.12*	*0.9044*	*28.55*	*0.9232*
FFNeRV [21]	from [27]	*35.17*	*0.9701*	*35.73*	*0.9720*	*28.20*	*0.9242*	*29.73*	*0.9484*
HNeRV [8]	from [27]	*35.59*	*0.9607*	*37.18*	*0.9765*	*29.31*	*0.9115*	*31.53*	*0.9459*
HiNeRV [19] (matched budget)	reproduced	**38.49**	**0.9854**	**39.91**	**0.9897**	**34.52**	**0.9848**	**36.93**	**0.9915**
GoP primitives: baselines									
D-3DGS [54]	from [27]	*26.27*	*0.8868*	*28.54*	*0.9124*	*24.54*	*0.8385*	*26.48*	*0.8808*
D2GV-AR [27]	from [27]	*35.03*	***0.9663***	*35.83*	***0.9728***	*29.82*	*0.9457*	*31.87*	*0.9668*
D2GV [26] (original MLP, paired)	reproduced	34.80	0.9592	35.61	0.9636	29.82	0.9361	32.48	0.9673
GoP primitives: ours									
DSAD	ours	**35.50**	0.9635	**36.43**	0.9704	**33.97**	**0.9745**	**35.93**	**0.9826**
DSAD (10 numbers per site)	ours	35.36	0.9624	36.33	0.9698	33.63	0.9726	35.69	0.9821
DSAD − D2GV (paired), ΔPSNR	ours	+0.81 (65/98)		+0.97 (55/98)		+4.18 (53/53)		+3.47 (50/53)	

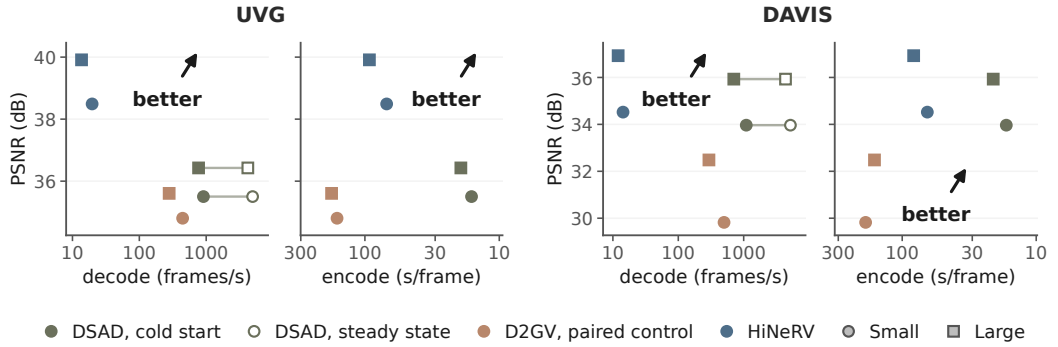


Figure 3: **Quality and speed at** 1280×720**.** Small/Large budgets of Tab. 2 (up and right is better). Lines join cold-start decoding (rebuilding axes and exact candidates) to steady-state decoding (reusing candidates). Note that lightweight D2GV (428–597 FPS, 66.4–89.6 s/frame at 60k) uses a different schedule and budget and is reported in Tab. J15.

winning every DAVIS pair at both budgets (53/53 each). At D2GV's larger native 1080p budgets, the mean gain is +2.28 dB (271 of 302 pairs won; Appendix J).

On Bunny at D2GV-AR's ∼3 M target under the 60k protocol, DSAD reaches 39.20 dB, surpassing both D2GV-AR's 120k report (37.64 dB) and our paired D2GV baseline at matched decoder-read parameters (36.42 dB, a gain of +2.77 dB with 13 of 13 GoPs won; Tab. I12).

Speed. Measured end to end on GoP_1 of every scene (Fig. 3, Tab. J15), DSAD decodes at 706–1,095 FPS from a cold start, which rebuilds site axes and exact top-K_s candidates from stored parameters, and reaches 4,196–5,187 FPS at steady state when reusing cached candidates. By comparison, paired D2GV reaches 279–501 FPS and lightweight D2GV 428–597 FPS when precomputing GPU frame times once per GoP with bit-identical output. DSAD's cold-start rate is at least 1.5× lightweight D2GV's on split means, down to 0.60× per GoP, while full-clip HiNeRV decodes at 12–20 FPS. For training, our fused CUDA kernels make DSAD's 120k per-frame fit at least 3.1× faster than lightweight D2GV's 60k fit, 7.6× faster than paired D2GV's 120k fit, and 3.8× faster than HiNeRV's 300-epoch fit, with baseline fit times estimated from timed windows; without kernel fusion, eager PyTorch trains 9.6–20.9× slower and decodes at 290–520 FPS from a cold start (Appendix J). Baselines use their original trainers and renderers; Fig. 3 plots paired 120k D2GV and omits lightweight D2GV due to its distinct 60k schedule and budgets.

Where the gains appear. The aggregate gains in Tab. 2 are not uniform across scenes. While every DAVIS scene favors DSAD at both budgets, UVG-Small scene margins span from −0.65 dB on

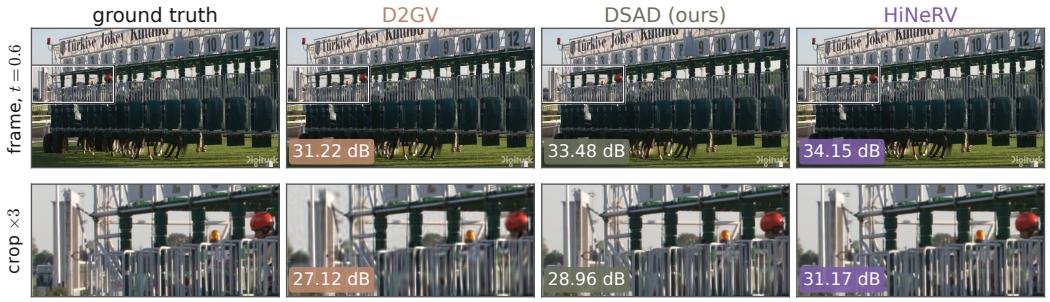


Figure 4: **UVG *ReadySetGo* GoP$_1$ reconstruction.** Fits from Tab. 2 at 1280×720 and the UVG-Small budget (HiNeRV [19] fits the full clip). Top: full frame; bottom: zoomed crop with per-frame PSNR (dB).

ShakeNDry to $+2.87$ dB on Bosphorus (Tab. H10, Appendix H), with D2GV leading on ShakeNDry, Beauty (-0.18 dB), and Jockey (-0.11 dB) at Small and on the same three scenes at Large (-1.13, -0.25, and -0.39 dB). Because Bosphorus supplies the largest UVG margin at both budgets, excluding it lowers the UVG scene-mean margin from $+0.71$ to $+0.35$ dB at Small ($50/83$ pairs won) and from $+0.83$ to $+0.05$ dB at Large ($40/83$). At the GoP level, every split has a paired t-statistic of at least 4.2; however, because GoPs within a video are correlated, testing across scene means is statistically significant on DAVIS ($t = 8.0$ and 5.3) but not across the 7 UVG scenes ($t = 1.6$ and 1.0), and simple optical-flow displacement or texture metrics do not account for this inter-scene spread (Appendix H).

Flow rank and the margin. To examine whether low-rank motion structure accounts for the UVG spread, we extract RAFT [44] optical flow from each GoP's first frame to its next four frames and compute the leading singular value of the stacked fields (Appendix H). Across the 7 UVG-Small scenes, this top-1 flow-energy fraction correlates with the paired PSNR margin at $r_s = 0.893$ (exact $p = 0.012$, and $r_s = 0.500$ over all frames of each GoP). We specified this measure in advance, and three earlier 60k comparison variants likewise yielded positive correlations; given the seven-scene UVG sample and the absence of a significant correlation on DAVIS, we treat this relationship as suggestive rather than conclusive.

4.3 Ablation study

Appearance versus motion. We ablate components on Bunny (132 frames, 13 GoPs, 1280×720), which admits our tightest reproduction of D2GV's published setup. Whereas D2GV with its original MLP reads 287,645 parameters per GoP ($N^{\mathrm{GS}} = 25{,}581$ Gaussians; Appendix B), both DSAD (263,647) and DSAD–GS (263,648) sit 8.3% *below* that total, with matched-budget ablations capped at 263,654. Replacing only the MLP with the ROM (DSAD–GS, 30,428 Gaussians) cuts read motion parameters by $4.1\times$ and reallocates them to canonical Gaussians, adding $+0.34$ dB ($12/13$ GoPs won). Replacing Gaussians with SAD sites at the same ROM, budget, and schedule adds a further $+0.96$ dB ($13/13$ won; Tab. 3), the largest single step in the decomposition, and adopting Sec. 3.4's training recipe contributes an additional $+0.77$ dB ($13/13$ won). Overall, full DSAD wins 13 of 13 Bunny GoPs with a mean gain of $+2.08 \pm 0.63$ dB over D2GV.

Table 3: **Cross-GoP aggregate on Bunny.** Mean ± sample standard deviation over 13 GoPs under the 60k protocol (higher is better), with decoder-read parameter counts. The DSAD rows read 8.3% fewer parameters (Tab. I14).

Method	#GoPs	PSNR ↑	MS-SSIM ↑	Total params
D2GV [26] (GS appearance + MLP motion, original MLP)	13	37.54 ± 1.36	0.9888 ± 0.0041	287,645
DSAD–GS (GS appearance + ROM motion, ablation)	13	37.89 ± 1.50	0.9891 ± 0.0045	263,648
DSAD (SAD appearance + ROM motion, without the recipe)	13	38.85 ± 0.81	0.9920 ± 0.0018	263,647
DSAD (SAD + ROM motion, with the recipe, ours)	13	39.62 ± 0.98	0.9930 ± 0.0019	263,647

Motion basis. We next evaluate two ROM design choices on Bunny: the spatial diagram score (Tab. G7) and the rank–bandwidth allocation (M, K_t) (Tab. G8). Among the four parameter-tied

diagram families, the anisotropic-Apollonius (SAD) score leads on the Bunny mean and on 13 of 13 GoPs; its four anisotropic scalars per anchor add only 5.1% to the Voronoi ROM size, whereas reducing SAD to Apollonius, Voronoi, or power scores changes PSNR by -0.11, -0.18, and -0.32 dB.

At fixed site count N on Bunny, PSNR grows with M and varies by at most 0.07 dB with K_t. Under a matched total budget, however, larger ROM configurations trade off canonical sites: relative to our default $(M, K_\mathrm{t}) = (256, 8)$, increasing M to 384 or 512 changes Bunny PSNR by -0.02 and -0.07 dB, while setting $K_\mathrm{t} = 16$ gives -0.21 dB and $K_\mathrm{t} = 4$ frees sites to give $+0.04$ dB. On DAVIS, by contrast, allocating more budget to motion can be beneficial: on two DAVIS GoPs the K_t trade-off reverses ($+0.16$/$+0.02$ dB at $K_\mathrm{t} = 16$ and -0.36/-0.41 dB at $K_\mathrm{t} = 4$; Appendix G), and across four DAVIS GoPs at matched budget, PSNR continues to rise up to $M = 1024$ on three sequences ($+1.32$, $+1.20$, and $+0.79$ dB at 768 over 256) while peaking at $M = 256$ on camel (Tab. J19), indicating that Bunny's knee near $M \approx 256$ is not universal.

Further analyses. Appendix J extends our evaluation across additional benchmarks, resolutions, and downstream tasks. Under the 120k protocol on GoP_1 across all 26 scene–budget pairs of Tab. 2, DSAD improves PSNR by $+2.77$ dB inside dilated Canny [6] luma edge bands ($21/26$ GoPs won) and by $+1.71$ dB outside them ($21/26$ won), with a larger in-band margin on $20/26$. Under the 60k protocol, DSAD maintains paired margins of $+3.80$, $+7.43$, and $+8.93$ dB at GoP lengths $T = 10$, 20, and 30 (Tab. J18) and decodes at 465–575 FPS cold and 2,531–3,212 FPS steady at 1920×1080 (Tab. J16). Moreover, the explicit primitive structure directly supports spatial inpainting ($+10.12$ dB over copying on moderate motion; Tab. J20), temporal interpolation (Tabs. J21 and J22), localized single-anchor editing, and post-hoc quantization with LZMA compression (0.30–0.45 dB loss at one-fifth of float32 storage; Tab. J24).

5 Conclusion

We introduced DSAD, an explicit per-GoP video representation that unifies canonical appearance ownership and reduced-order motion within a single soft-distance coordinate system. Dense SAD sites decode pixel appearance through local competition, while coarser anchors reuse the same soft-distance geometry to form a sparse partition-of-unity deformation basis, eliminating neural network layers from the per-GoP model.

Under the D2GV-AR GoP protocol, DSAD achieves the highest PSNR among compared GoP primitives across all UVG and DAVIS budgets. In our paired 1280×720 control against D2GV with its original MLP at matched decoder-read budgets and training length, DSAD improves PSNR on 223 of 302 GoP–budget pairs by a mean of $+1.92$ dB. Bunny ablations show that SAD appearance supplies the largest component gain at equal budget (Tab. 3), and UVG-Small flow spectra suggest that the largest gains arise when GoP motion is close to low-rank.

Limitations and operating range. Our results characterize a matched-budget trade-off within the per-GoP regime: replacing only D2GV's MLP with the ROM (DSAD–GS) improves PSNR on Bunny where freed parameters add Gaussians, whereas on the DAVIS pool it averages -0.29 dB ($37/70$ won; Tab. J17), and only Bunny isolates SAD appearance (Tab. 3). While DSAD leads overall across UVG, the 70-GoP DAVIS pool ($+2.96$ dB, $64/70$ won), and 1080p (Tab. J16), D2GV remains ahead on several scenes (3 of 7 UVG and 3 of 32 DAVIS-pool scenes; Tabs. H10 and J17); more anchors close 70–73% of the gap on the one scanned GoP where D2GV leads (Tab. J19), and extending training $4\times$ narrows the UVG margin (Appendix J). Moreover, the link between PSNR margins and low-rank optical flow on UVG-Small (Sec. 4.2) remains correlational, weakly textured deforming regions can leave anchors sparsely sampled, and our budgets count decoder-read parameters rather than coded bits (trailing full-clip HNeRV-Boost by 1.42–1.53 dB on Bunny under post-hoc compression; Tab. J25).

Looking ahead, coupling DSAD with a learned entropy model, joint rate–distortion optimization, and progressive anchor pruning could narrow the bitstream gap with full-clip INR codecs while preserving streamable per-GoP random access and fast decoding. Enriching the translation-only ROM basis with content-adaptive anchor placement, per-anchor affine warps, or time-varying site metrics $G_i(t)$ (Tab. G9) could capture high-rank non-rigid deformations, rotating boundaries, and low-texture motion without reintroducing a coordinate MLP.

Acknowledgments and Disclosure of Funding

We thank the teaching team of MIT 6.8300 for discussions. Zhiyang Dou is partially supported by the Hasso Plattner Fellowship 2026–2027. The authors declare no competing interests.

References

[1] Nina Amenta and Marshall Bern. Surface reconstruction by Voronoi filtering. In *Proceedings of the fourteenth annual symposium on Computational geometry*, pages 39–48, 1998.

[2] Amine Ammar, Bechir Mokdad, Francisco Chinesta, and Roland Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics*, 139(3):153–176, 2006.

[3] Franz Aurenhammer. Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[4] Franz Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[5] Blender Foundation. Big Buck Bunny. Open-source short film, `https://peach.blender.org`, 2008. (c) Blender Foundation, licensed CC-BY 3.0.

[6] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[7] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. NeRV: Neural representations for videos. In *Advances in Neural Information Processing Systems*, volume 34, pages 21557–21568, 2021.

[8] Hao Chen, Matthew Gwilliam, Ser-Nam Lim, and Abhinav Shrivastava. HNeRV: A hybrid neural representation for videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10270–10279, 2023.

[9] Francesco Di Sario, Daniel Rebain, Dor Verbin, Marco Grangetto, and Andrea Tagliasacchi. Spherical Voronoi: Directional appearance as a differentiable partition of the sphere. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22529–22538, 2026.

[10] Zhiyang Dou, Cheng Lin, Rui Xu, Lei Yang, Shiqing Xin, Taku Komura, and Wenping Wang. Coverage axis: Inner point selection for 3D shape skeletonization. *Computer Graphics Forum*, 41(2):419–432, 2022.

[11] Ioannis Z. Emiris and Menelaos I. Karavelas. The predicates of the Apollonius diagram: Algorithmic analysis and implementation. *Computational Geometry*, 33(1–2):18–57, 2006.

[12] Ge Gao, Siyue Teng, Tianhao Peng, Fan Zhang, and David Bull. GIViC: Generative implicit video compression. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17356–17367, 2025.

[13] Shrisudhan Govindarajan, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Radiant foam: Real-time differentiable ray tracing. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4135–4145, 2025.

[14] Philip Holmes, John L. Lumley, Gahl Berkooz, and Clarence W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 2nd edition, 2012.

[15] Laki Iinbor, Zhiyang Dou, and Wojciech Matusik. Soft anisotropic diagrams for differentiable image representation. *ACM Transactions on Graphics*, 45(4):158:1–158:19, 2026. doi: 10.1145/3811372. SIGGRAPH 2026.

[16] Dadong Jiang, Zhi Hou, Zhihui Ke, Xianghui Yang, Xiaobo Zhou, and Tie Qiu. TimeFormer: Capturing temporal relationships of deformable 3D Gaussians for robust reconstruction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8721–8732, 2025.

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[19] Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull. HiNeRV: Video compression with hierarchical encoding-based neural representation. In *Advances in Neural Information Processing Systems*, volume 36, pages 72692–72704, 2023. Code at `https://github.com/hmkx/HiNeRV`.

[20] Egor Larionov, Igor Santesteban, Hsiao-yu Chen, Gene Wei-Chin Lin, Philipp Herholz, Ryan Goldade, Ladislav Kavan, Doug Roble, and Tuur Stuyck. SkinCells: Sparse skinning using Voronoi cells. *Computer Graphics Forum*, 45(2):e70381, 2026. doi: 10.1111/cgf.70381.

[21] Joo Chan Lee, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. FFNeRV: Flow-guided frame-wise neural representations for videos. In *ACM International Conference on Multimedia (ACM MM)*, pages 7859–7870, 2023.

[22] Zizhang Li, Mengmeng Wang, Huaijin Pi, Kechun Xu, Jianbiao Mei, and Yong Liu. E-NeRV: Expedite neural video representation with disentangled spatial–temporal context. In *European Conference on Computer Vision (ECCV)*, pages 267–284, 2022.

[23] Yiqing Liang, Numair Khan, Zhengqin Li, Thu H Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. GauFRe: Gaussian deformation fields for real-time dynamic novel view synthesis. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 2642–2652, 2025.

[24] Chieh Hubert Lin, Zhaoyang Lv, Songyin Wu, Zhen Xu, Thu Nguyen-Phuoc, Hung-Yu Tseng, Julian Straub, Numair Khan, Lei Xiao, Ming-Hsuan Yang, Yuheng Ren, Richard Newcombe, Zhao Dong, and Zhengqin Li. DGS-LRM: Real-time deformable 3D Gaussian reconstruction from monocular videos. In *Advances in Neural Information Processing Systems*, volume 38, pages 46915–46936, 2025.

[25] Jierun Lin, Jiacong Chen, Qingyu Mao, Shuai Liu, Xiandong Meng, Fanyang Meng, and Yongsheng Liang. STGV: Spatio-temporal hash encoding for Gaussian-based video representation. *arXiv preprint arXiv:2604.10910*, 2026.

[26] Mufan Liu, Qi Yang, Miaoran Zhao, He Huang, Le Yang, Zhu Li, and Yiling Xu. D2GV: Deformable 2D Gaussian splatting for video representation in 400FPS. arXiv preprint arXiv:2503.05600v1; code at `https://github.com/Evan-sudo/D2GV`, 2025. Earlier version of [27] by the same authors.

[27] Mufan Liu, Qi Yang, Miaoran Zhao, He Huang, Le Yang, Zhu Li, and Yiling Xu. Progressively deformable 2D Gaussian splatting for video representation at arbitrary resolutions. *arXiv preprint arXiv:2503.05600v2*, 2026. Later arXiv version of [26] by the same authors.

[28] Qingming Liu, Yuan Liu, Jiepeng Wang, Xianqiang Lyv, Peng Wang, Wenping Wang, and Junhui Hou. MoDGS: Dynamic Gaussian splatting from casually-captured monocular videos with depth priors. In *International Conference on Learning Representations (ICLR)*, 2025.

[29] Jiayin Lu, Ying Jiang, Yin Yang, and Chenfanfu Jiang. VoroLight: Learning quality volumetric Voronoi meshes from general inputs. *arXiv preprint arXiv:2512.12984v1*, 2025.

[30] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision (3DV)*, pages 800–809, 2024.

[31] Nissim Maruani, Roman Klokov, Maks Ovsjanikov, Pierre Alliez, and Mathieu Desbrun. VoroMesh: Learning watertight surface meshes with Voronoi diagrams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14565–14574, 2023.

[32] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. UVG dataset: 50/120 fps 4K sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 297–302, 2020.

[33] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. ASH: Animatable Gaussian splats for efficient and photoreal human rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1165–1175, 2024.

[34] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5865–5874, 2021.

[35] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016.

[36] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[37] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10318–10327, 2021.

[38] Marie-Julie Rakotosaona, Noam Aigerman, Niloy J Mitra, Maks Ovsjanikov, and Paul Guerrero. Differentiable surface triangulation. *ACM Transactions on Graphics*, 40(6):267:1–267:13, 2021. doi: 10.1145/3478513.3480554.

[39] Guodong Rong and Tiow-Seng Tan. Jump flooding in GPU with applications to Voronoi diagram and distance transform. In *Symposium on Interactive 3D Graphics and Games (I3D)*, pages 109–116, 2006.

[40] Gianluigi Rozza, Dinh B. P. Huynh, and Anthony T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[41] Osamu Takahashi and Robert J Schilling. Motion planning in a plane using generalized Voronoi diagrams. *IEEE Transactions on Robotics and Automation*, 5(2):143–150, 1989.

[42] Lv Tang, Jun Zhu, Xinfeng Zhang, Li Zhang, Siwei Ma, and Qingming Huang. CANeRV: Content adaptive neural representation for video compression. *arXiv preprint arXiv:2502.06181*, 2025.

[43] Lv Tang, Jun Zhu, Xinfeng Zhang, Li Zhang, Siwei Ma, and Qingming Huang. HANeRV: Hierarchically adaptive neural representation for video compression. *IEEE Transactions on Image Processing*, 35: 5773–5788, 2026.

[44] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419, 2020.

[45] TorchVision maintainers and contributors. TorchVision: PyTorch's computer vision library. `https://github.com/pytorch/vision`, 2016.

[46] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12959–12970, 2021.

[47] Georges Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *Journal für die Reine und Angewandte Mathematik*, 134:198–287, 1908.

[48] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C.-C. Jay Kuo. MCL-JCV: A JND-based H.264/AVC video quality assessment dataset. In *IEEE International Conference on Image Processing (ICIP)*, pages 1509–1513, 2016.

[49] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems & Computers*, volume 2, pages 1398–1402, 2003.

[50] Zimeng Wang, Zhiyang Dou, Rui Xu, Cheng Lin, Yuan Liu, Xiaoxiao Long, Shiqing Xin, Taku Komura, Xiaoming Yuan, and Wenping Wang. Coverage axis++: Efficient inner point selection for 3D shape skeletonization. *Computer Graphics Forum*, 43(5):e15143, 2024.

[51] Francis Williams, Jerome Parent-Levesque, Derek Nowrouzezahrai, Daniele Panozzo, Kwang Moo Yi, and Andrea Tagliasacchi. VoronoiNet: General functional approximators with local support. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1069–1073, 2020.

[52] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive Nesterov momentum algorithm for faster optimizing deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9508–9520, 2024.

[53] Dong-Ming Yan, Wenping Wang, Bruno Lévy, and Yang Liu. Efficient computation of clipped Voronoi diagram for mesh generation. *Computer-Aided Design*, 45(4):843–852, 2013.

[54] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20331–20341, 2024.

[55] Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. GaussianImage: 1000 FPS image representation and compression by 2D Gaussian splatting. In *European Conference on Computer Vision (ECCV)*, pages 327–345, 2024.

[56] Xinjie Zhang, Ren Yang, Dailan He, Xingtong Ge, Tongda Xu, Yan Wang, Hongwei Qin, and Jun Zhang. Boosting neural representations for videos with a conditional decoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2556–2566, 2024.

Appendix Contents

A. More qualitative results of DSAD 15
B. Per-method parameter calculation 17
C. GoP primitives versus full-clip INRs and INR codecs 17
D. Anisotropic norm under $\det G = 1$ 19
E. SAD rasterizer and native gradients 19
F. Residual channels and regularizers 20
G. Motion-basis design 20
H. UVG paired per-scene results and motion-rank evidence 21
I. Bunny: comparison at D2GV-AR's budget and parameter split 22
J. Additional analyses 23
K. Broader impact 29

A More qualitative results of DSAD

Fig. A5 visualizes the learned anchor basis ($M = 256$, $K_t = 8$, $K_a = 12$, SAD mode) and image-space displacement $d(x,t) = \tanh(\operatorname{atanh} x + \Delta(x,t)) - x$ at the first ($t = 0.1$) and last ($t = 1.0$) frames across three UVG GoPs (Appendix B), showing that anchors form scene-adaptive cells rather than a fixed grid. Fig. A6 presents DSAD reconstructions alongside ground-truth frames on five UVG and DAVIS scenes (Jockey, YachtRide, Bosphorus, bmx-trees, and camel) across each GoP horizon, preserving sharp thin structures and articulated limbs.

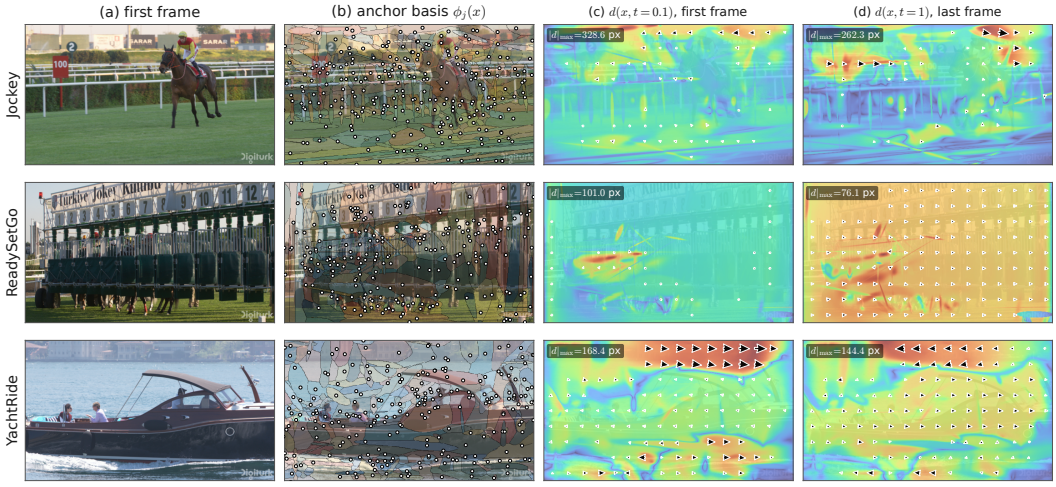


Figure A5: **Learned deformation on UVG.** Rows show Jockey, ReadySetGo and YachtRide GoP_1 at the UVG-Small budget under the 60k protocol. Panels show the first frame (a), dominant ownership with white anchors (b), and first/last-frame displacement (c,d). Magnitudes are normalized per panel.

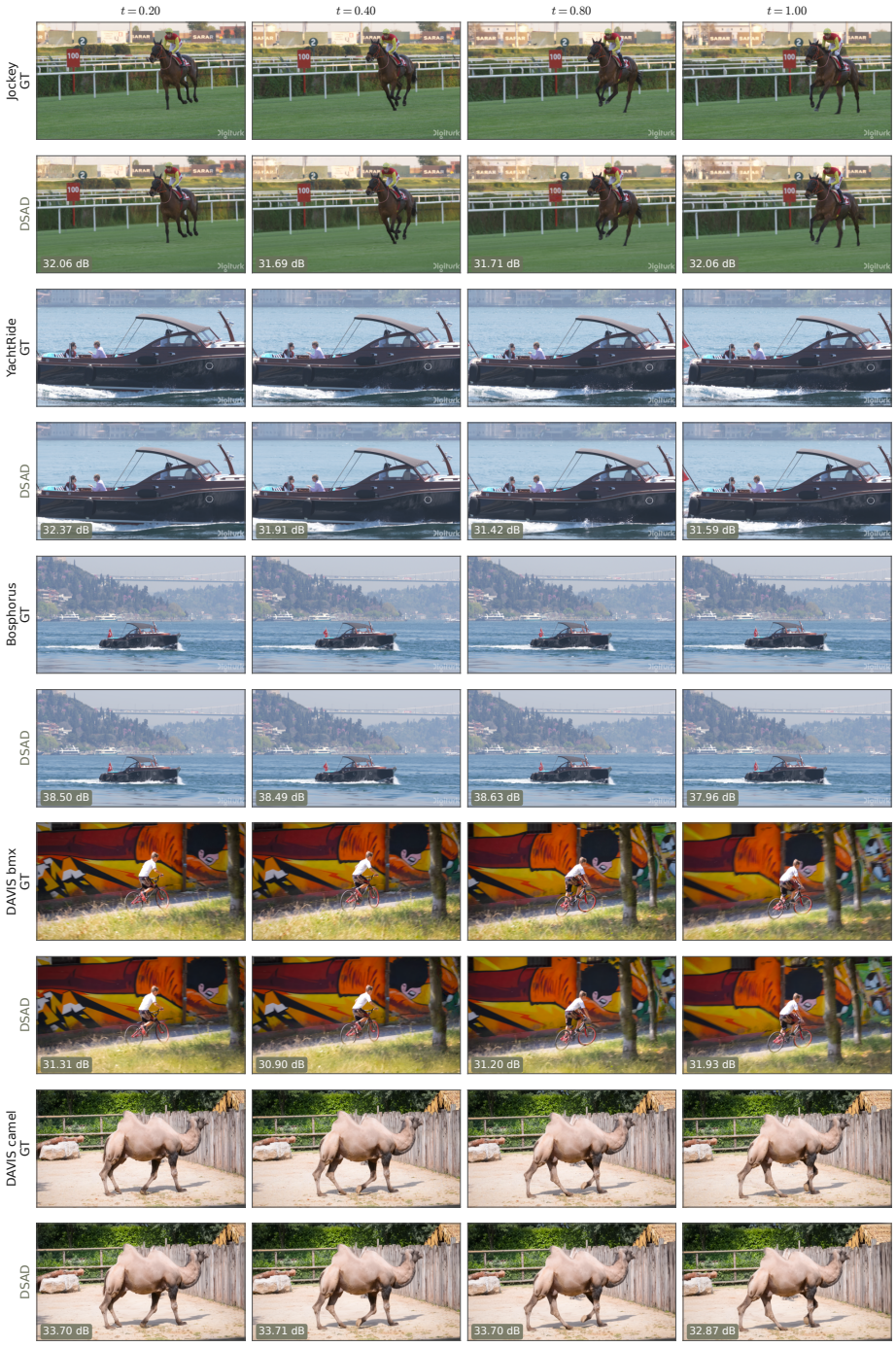


Figure A6: **UVG and DAVIS reconstructions.** Five scenes use GoP_1 at 1280×720 and Small budgets. Ground truth is above DSAD decoded from stored parameters under the 60k protocol at $t=0.2$, 0.4, 0.8 and 1. Each reconstruction shows PSNR (dB).

B Per-method parameter calculation

We summarize the per-video parameter formulas for Tab. 2 under the 1280×720 GoP protocol (1080p budgets are given in Appendix J). Unless noted otherwise, all budgets count the parameters a decoder reads to render frames.

Transcribed rows. The NeRV, E-NeRV, FFNeRV, HNeRV, D-3DGS, and D2GV-AR rows in Tab. 2 are transcribed from D2GV-AR [27] (its Tab. 1) at its nominal Small and Large per-video budgets (3.5/5.8 M on UVG and 1.8/3.3 M on DAVIS, with $P_{\text{per-video}} = P_{\text{model}}$ for full-clip INRs).

D2GV (paired). Each D2GV GoP stores N^{GS} Cholesky-Gaussian primitives (9 scalars each: 2 mean, 3 Cholesky covariance, 3 color, and an unused pruning logit m) plus a deformation MLP of 107,646 parameters [26]. Because our reproduction uses no pruning stage and renders with fixed opacity, the decoder reads 8 scalars per Gaussian and omits the MLP's 24,649-parameter opacity branch (82,997 read MLP parameters):

$$P^{\text{D2GV}}_{\text{per-GoP}} = 8\,N^{\text{GS}} + 82{,}997.$$

To match the target budget in read parameters, we size N^{rel} from stored parameters ($9\,N^{\text{rel}} + 107{,}646 \approx \text{budget}/\#\text{GoPs}$) and reallocate the unread parameters as additional Gaussians, $N^{\text{GS}} = \lfloor (9\,N^{\text{rel}} + 24{,}649)/8 \rfloor$ (e.g., $N^{\text{GS}} = 18{,}959$ rather than 14,114 on UVG-Beauty Small for 234,669 read parameters per GoP; $N^{\text{GS}} = 25{,}581$ for 287,645 on the Bunny ablations and DAVIS pool; and $N^{\text{GS}} = 18{,}471$ for 230,765 at Bunny's 3 M target). "D2GV with D2GV-AR's lightweight MLP" replaces this MLP with our reproduction of the single-layer MLP described in [27] (9,678 parameters), sized by stored parameters.

DSAD (ours). Each GoP stores N SAD sites at 9 scalars each: 2 for unconstrained position z_i (mapped via $\mathbf{s}_i(t) = \tanh(z_i + \Delta z_i(t)) \in [-1, 1]^2$, written as $\mathbf{s}_i + \Delta\mathbf{s}_i(t)$ in Sec. 3), 2 for the metric $G_i(\alpha_i, \hat{u}_i)$ (log-aspect α_i and axis angle of $\hat{u}_i$), 1 radius r_i, 1 temperature τ_i, and 3 RGB color $\mathbf{c}_i$. With $M = 256$ anchors and $K_{\text{t}} = 8$ knots, the ROM contributes $P_{\text{ROM}} = M(3{+}\nu) + 9MK_{\text{t}}$ parameters (Eq. (8)): 20,224 for SAD ($\nu = 4$), 19,200 for Voronoi ($\nu = 0$), and 19,456 for Apollonius or Power ($\nu = 1$). At larger budgets, M scales per Sec. 3.4 to 416 and 480 ($P_{\text{ROM}} = 32{,}864$ and 37,920) for UVG and DAVIS Large in Tab. 2, and to 256–672 (20,224–53,088) at native 1080p budgets:

$$P^{\text{DSAD}}_{\text{per-GoP}} = 9\,N + P_{\text{ROM}}, \qquad N = \big\lfloor \big(\text{budget}/\#\text{GoPs} - P_{\text{ROM}}\big)/9 \big\rfloor.$$

On UVG-Beauty Small, $N = 23{,}714$ gives $P_{\text{per-GoP}} = 233{,}650$ (0.4% below paired D2GV's 234,669). Across the 302 GoP–budget pairs of Tab. 2, DSAD's budget is within $+0.61\%$ (19 pairs) and -0.46% of paired D2GV, allocating a smaller motion share (0.24–0.46 of D2GV's 82,997) and a larger appearance share (213,426 vs. 151,672 on UVG-Beauty Small). On Bunny, DSAD uses $N = 23{,}381$ ($P_{\text{per-GoP}} = 230{,}653$, 3.00 M/video) at the 3 M target (Tab. I12) and $N = 27{,}047$ (263,647, 8.3% below D2GV's 287,645) in the ablations (Tabs. 3 and I13), where DSAD–GS uses 30,428 Gaussians (263,648).

The axis angle. During training, each site's axis $\hat{u}_i$ is optimized as an unconstrained 2-vector (10 numbers per site). Because the metric G_i depends only on the normalized direction $(\hat{u}_i \cdot d)^2$, we store the angle of $\hat{u}_i$ in $[0, \pi)$ as a single 32-bit float (9 numbers per site) and reconstruct $(\cos\theta_i, \sin\theta_i)$ at decoding, which changes decoded PSNR by a median of only 3.4×10^{-6} dB across all 2,445 fits. Note that under the earlier 10-number-per-site count, DSAD holds 10% fewer sites at the same total budget (second DSAD row of Tab. 2; Appendix J).

Training settings. Tab. B4 lists the training hyperparameters for DSAD, paired D2GV, and HiNeRV. DSAD and D2GV share the coarse/fine protocol, the Adan optimizer and step schedule for canonical primitives, and the frame-level MSE objective (Eq. (9)); at 120k fine iterations, primitive halving periods (20k to 40k) and D2GV MLP decay horizons (50k to 100k) scale $2\times$, while DSAD's ROM learning rates decay over the full fine stage. HiNeRV is trained once per clip using its 1280×720 configuration without pruning or quantization.

C GoP primitives versus full-clip INRs and INR codecs

Tab. C5 summarizes the operating-point differences between the two regimes: full-clip INRs require a complete-sequence pre-pass and emit no spatially local handles, while GoP primitives keep the

Table B4: **Training settings.** Columns show DSAD, paired D2GV (original MLP) and our reproduction of HiNeRV. Cells give values for the 60k protocol and the 120k fits of Tab. 2. Learning rates are absolute, and a dash marks an inapplicable setting.

Setting	DSAD	D2GV (paired control)	HiNeRV
Fitted unit, length	one GoP, 10k coarse iterations (sites only), then 60k or 120k fine iterations (sites and ROM)	one GoP, 10k coarse iterations (Gaussians only), then 60k or 120k fine iterations (Gaussians and MLP)	one clip, 300 epochs, no pruning or quantization epochs
Batch	one full frame per iteration	one full frame per iteration	128 patches of 80×80 pixels
Frame order	a new seeded permutation of the GoP's frames in every pass, in both stages	in order, cyclically	patches shuffled in every epoch
Loss	mean squared error on the render clamped to $[0, 1]$, movement, temporal and ARAP weights 0	mean squared error on the clamped render	$0.7\,\ell_1 + 0.3\,(1-\text{MS-SSIM})$, 5×5 window
Optimizer	Adan for sites, $\beta = (0.98, 0.92, 0.99)$, $\epsilon = 10^{-8}$, no weight decay, no clipping	the same Adan for Gaussians	Adam [18], $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, no weight decay, gradient norm clipped at 1
Learning rate	position $5{\cdot}10^{-3}$, color, temperature and radius 10^{-2}, orientation $2.5{\cdot}10^{-3}$, anisotropy $2.5{\cdot}10^{-2}$	$5{\cdot}10^{-3}$ for every Gaussian parameter	$1.78{\cdot}10^{-3}$ (the configured $2{\cdot}10^{-3}$ times $8/9$, the batch's share of a frame)
Schedule	halved every S iterations from the start of the coarse stage, $S = 20$k for 60k fits and $S = 40$k for 120k fits	the same, $S = 20$k for 60k fits and $S = 40$k for 120k fits	linear ramp from $1.78{\cdot}10^{-5}$ over 30 epochs, then cosine over the 300 epochs to $1.78{\cdot}10^{-5}$, updated every step
Motion model	ROM with SAD anchors, position and color channels, $M = 256$ (416 and 480 at the UVG and DAVIS Large budgets of Tab. 2), $K_\text{t} = 8$, $K_\text{a} = 12$	original MLP, width 156, 107,646 parameters	—
optimizer	Adam, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-15}$, no weight decay, fine stage only	the same Adam	—
learning rate	anchors $5{\cdot}10^{-3}$, knots $2{\cdot}10^{-3}$	$8{\cdot}10^{-4}$	—
schedule	exponential decay to 1% of the initial rate at the last fine iteration (60k and 120k alike)	exponential decay to $1.6{\cdot}10^{-5}$ at fine iteration D, constant afterwards, $D = 50$k for 60k fits and $D = 100$k for 120k fits	—
Site candidates	$K_\text{s} = 8$ per pixel, kept per frame: a jump-flood build with 16 refinement passes at a frame's first use and at every 1,000th use, one pass at every second use, 4 injected candidates per pass, exact top-8 in the last 6k fine iterations	—	—
Initialization	sites drawn from pixels of the GoP's mean frame, half uniformly and half by luminance-gradient magnitude, with that frame's colors. Anchors on a sunflower lattice, temperature 0.25. Knots uniform in time, bandwidth 0.15, zero values	uniform random positions, shapes and colors	—
Precision	32-bit floating point	32-bit floating point	16-bit mixed precision
Seed	1	1	0

per-clip cost bounded and expose primitives that can be inspected, edited, or rendered before the full clip is seen.

INR codecs and concurrent primitive work. Because NeRV, Boosting-NeRV, and HiNeRV publish UVG results on full-length 1080p sequences (or 1920×960 crops for HNeRV) rather than the $4\times$ temporally subsampled 720p clips of the D2GV-AR protocol, Tab. 2 reports D2GV-AR's transcribed INR rows alongside our matched HiNeRV reproduction (Sec. 2). On Bunny, full-clip HNeRV-Boost [56] reports 41.09 dB (0.9934 MS-SSIM) at 3.08 M (1.89 dB above per-GoP DSAD's 39.20 dB at 3.00 M; Tab. I12), with 13 h 51 min encoding per 1080p video and 13.15 FPS decoding on a V100. CANeRV [42] and HANeRV [43] report coded-rate results, including BD-rate against H.266/VVC, rather than fixed-budget quality, and GIViC [12] relies on a pretrained 225.9 M-parameter receiver decoder (9.79 FPS decode and 1.78 h per 32-frame 1080p GoP encode on an A100, vs. DSAD's standalone 465–575 FPS cold-start decode (GoP_1) and median 22.6 s/frame encode across the 604 1080p fits of Tab. J16 on one RTX 2080 Ti). Concurrent STGV [25] drives Gaussian deformation with spatio-temporal hash encodings under a related 720p GoP protocol (1280×640 on Bunny) without public code.

Table C5: **GoP primitives versus full-clip INRs.** Deployment properties motivate the paired GoP protocol. The first four rows are properties by construction, with no measurement involved.

Property	INR (full-clip)	GoP primitives
Streamable training	✗	✓
Online (incremental) training	✗	✓
Local primitive ownership	✗	✓
Per-clip pre-pass	required	not required
PSNR at a matched budget	HiNeRV higher by 0.6–3.5 dB (Tab. 2)	lower
Decode FPS at 720p	12–20 (HiNeRV)	706–1,095 cold start, 4,196–5,187 steady state (DSAD, fused kernels, Tab. J15)

D Anisotropic norm under $\det G = 1$

For a unit vector $\hat{u} \in \mathbb{R}^2$ and a scalar α, $G(\alpha, \hat{u}) = e^{\alpha}\hat{u}\hat{u}^{\top} + e^{-\alpha}(I - \hat{u}\hat{u}^{\top})$ is symmetric positive definite with $\det G = 1$ and condition number $e^{2|\alpha|}$: it has eigenvalue e^{α} along $\hat{u}$ and $e^{-\alpha}$ across it, so lengths along $\hat{u}$ are scaled by $e^{\alpha/2}$ and, for $\alpha > 0$, cells are compressed along $\hat{u}$. For a generator at $\mathbf{q}$ (a site position $\mathbf{s}_i$ or an anchor position $\mathbf{p}_j$), letting $\xi = (x - \mathbf{q})^{\top}\hat{u}$,

$$\|x - \mathbf{q}\|_G^2 = e^{\alpha}\xi^2 + e^{-\alpha}\big(\|x - \mathbf{q}\|^2 - \xi^2\big). \tag{10}$$

Sites use $G_i = G(\alpha_i, \hat{u}_i)$ and anchors $G_j^{\mathrm{a}} = G(\alpha_j^{\mathrm{a}}, \hat{u}_j^{\mathrm{a}})$. The unit-determinant constraint decouples the temperature from the metric, so the SAD anchor score adds only four scalars per anchor (r_j^{a}, α_j^{a}, and the two components of $\hat{u}_j^{\mathrm{a}}$).

E SAD rasterizer and native gradients

The site mask $\mathcal{O}_{\mathrm{s}}(x)$ in Eq. (5) is read from a per-pixel candidate field $C : \Omega \to \{1, \dots, N\}^{K_{\mathrm{s}}}$ approximating the K_{s} sites with the largest logits $-d_i(x)/\tau_i$. During training, C is refreshed incrementally via jump flooding (JFA) [39] and Voronoi particle tracking (VPT) following the single-image SAD rasterizer [15]. During the final 6k fine iterations, C is rebuilt as the exact top-K_{s} set on every second use of each frame; evaluation and decoding likewise use the exact top-K_{s} set built directly from stored parameters (cached across passes in steady-state decoding).

Per-frame persistent candidate buffers. A single shared C that chases the T deformed configurations of the site set simultaneously yields a stale site mask at every frame. We instead maintain T independent buffers $\{C^{(t)}\}_{t=1}^{T}$ ($K_{\mathrm{s}} = 8$ indices per pixel, ≈295 MB per 10-frame 720p GoP) and activate the current frame's buffer before each render call. Each buffer is seeded on its frame's first call by jump flooding with initial VPT passes, refined by VPT every second call, and re-flooded every 1,000 calls so that $C^{(t)}$ tracks only the slow drift of canonical sites. When decoding from

stored parameters without training buffers, the decoder builds each frame's exact top-8 candidates in 0.7–1.2 ms per 720p frame (split means; Tab. J15).

Native analytic backward. The closed-form derivative of Eq. (5) through the masked softmax weights, site colors, and geometric parameters is evaluated in a fused CUDA backward kernel. All reported DSAD fits execute the forward pass, analytic backward, and optimizer updates through these fused CUDA kernels, which match the reference PyTorch step up to floating-point accumulation order while speeding up training by 9.6–20.9× (16.1–20.9 s/frame training and 4,196–5,187 FPS steady-state decoding at 720p on one RTX 2080 Ti; Tab. J15).

F Residual channels and regularizers

The optional motion penalties of Sec. 3.4 add $\lambda_{\text{move}}\mathcal{L}_{\text{move}} + \lambda_{\text{tmp}}\mathcal{L}_{\text{tmp}} + \lambda_{\text{arap}}\mathcal{L}_{\text{arap}}$ to Eq. (9), with

$$\begin{aligned} \mathcal{L}_{\text{move}} &= \frac{1}{|\mathcal{T}|N}\sum_{t,i}\|\Delta\mathbf{s}_i(t)\|_2^2, \qquad \mathcal{L}_{\text{tmp}} = \frac{1}{2|\mathcal{U}|M}\sum_{t\in\mathcal{U},j}\|D_t^2\,\mathbf{a}_j^{\Delta\mathbf{s}}(t)\|_2^2, \\ \mathcal{L}_{\text{arap}} &= \frac{1}{|\mathcal{T}||\mathcal{E}|}\sum_{t,(i,\ell)\in\mathcal{E}}\big(\|\mathbf{s}_i(t)-\mathbf{s}_\ell(t)\|_2 - \|\mathbf{s}_i-\mathbf{s}_\ell\|_2\big)^2, \end{aligned} \tag{11}$$

where $\mathbf{s}_i(t) = \mathbf{s}_i + \Delta\mathbf{s}_i(t)$ is the deformed site position, $\mathcal{U}$ holds the 30 interior points of a uniform 32-point time grid on $[0,1]$ with second-order finite difference D_t^2, and $\mathcal{E}$ is the canonical 6-NN graph over sites ($\lambda_{\text{move}} = \lambda_{\text{tmp}} = \lambda_{\text{arap}} = 0$ in the final configuration).

Tab. F6 ablates the deformation channels and motion penalties across all 13 Bunny GoPs under the 60k protocol. Removing the color residual ($\Delta\mathbf{s}$ only) saves 10,240 parameters ($5MK_{\text{t}}$) but changes PSNR by -0.23 dB, whereas enabling the movement, temporal, and ARAP penalties ($\lambda_{\text{move}} = 0.01$, $\lambda_{\text{tmp}} = 0.001$, $\lambda_{\text{arap}} = 0.005$) changes PSNR by -0.05 dB (3/13 GoPs higher). An opacity residual (+6,144 parameters, 0.00 dB) is inactive because both DSAD and paired D2GV render with fixed opacity.

Table F6: **Bunny residuals and regularizers.** PSNR/MS-SSIM average 13 GoPs with the recipe of Sec. 3.4, $N = 27{,}047$, $M = 256$, $K_{\text{t}} = 8$ and $K_{\text{a}} = 12$ under the 60k protocol. Δ is the paired PSNR difference to the +feature row without penalties. Higher quality is better.

Variant	PSNR ↑	MS-SSIM ↑	Deform params	ΔPSNR
xyz only	39.39	0.9926	9,984	−0.23
+ feature residual	39.62	0.9930	20,224	–
+ feature + opacity residual	39.62	0.9930	26,368	+0.00
feature, + move/temporal/ARAP penalties	39.57	0.9929	20,224	−0.05

G Motion-basis design

Fig. 1 illustrates the soft-distance diagram family (Eq. (3)), Tab. G7 compares the four anchor scores on Bunny, and Tab. G8 scans the anchor count M and knot count K_{t} under the 60k protocol ($K_{\text{a}} = 12$, SAD basis, feature residuals on; matched-budget rows adjust N by 9 parameters per site to stay within 263,654 parameters per GoP). At a fixed site count on two DAVIS GoPs, the SAD basis leads on camel$_1$ by 0.44–0.79 dB and sits 0.07 dB behind Apollonius on bmx-trees$_1$.

Rank versus bandwidth. Across both Bunny (Tab. G8) and two DAVIS GoPs (bmx-trees$_1$ and camel$_1$), temporal bandwidth saturates rapidly: on Bunny, $K_{\text{t}} = 8$ outperforms $K_{\text{t}} = 16$ by +0.21 dB at a matched budget, and on bmx-trees$_1$ and camel$_1$, increasing K_{t} from 8 to 16 changes PSNR by +0.41 and +0.12 dB at fixed N (−0.23 and −0.15 dB from $K_{\text{t}} = 16$ to 32; +0.16 and +0.02 dB at matched budget, vs. −0.36 and −0.41 dB at $K_{\text{t}} = 4$). By contrast, the required spatial rank M depends on the content rather than transferring from Bunny's $M \approx 256$ knee.

Cost of richer motion models. Tab. G9 lists the parameter cost of extending translation knots with per-anchor affine maps (+8,192 parameters, equivalent to ≈104 additional anchors or ≈3.6 knots per anchor) or time-varying site metrics $G_i(t)$ (+4,096 parameters, or +6,144 with time-varying radii $r_i(t)$, sharing the $\Delta\mathbf{s}$ knot times and bandwidths).

Table G7: **Bunny tessellation ablation.** Rows vary only the motion anchor score $d_j^{\mathrm{a}}(x)$ and share SAD appearance and the recipe of Sec. 3.4. PSNR/MS-SSIM average 13 GoPs under the 60k protocol, with Δ relative to SAD. Higher quality is better.

Motion basis	PSNR ↑	MS-SSIM ↑	ΔPSNR	Extra per-anchor parameters
Voronoi	39.44	0.9926	-0.18	–
Apollonius	39.51	0.9928	-0.11	r_j^{a}
Power (Laguerre)	39.30	0.9923	-0.32	w_j^{a}
SAD (anisotropic Apollonius)	**39.62**	**0.9930**	–	$r_j^{\mathrm{a}}, \alpha_j^{\mathrm{a}}, \hat{u}_j^{\mathrm{a}}$

Table G8: **Bunny rank and knot scans.** Values are mean PSNR over 13 GoPs under the 60k protocol (higher is better). Fixed N keeps 27,047 SAD sites, so totals grow with the ROM. Matched budgets change the site count N, shown beside PSNR.

M	Fixed N	ROM params	Matched	N
128	39.38	10,112	39.48	28,171
256	39.62	20,224	39.62	27,047
384	39.73	30,336	39.60	25,924
512	39.80	40,448	39.55	24,800

K_{t}	Fixed N	ROM params	Matched	N
4	39.55	11,008	39.66	28,071
8	39.62	20,224	39.62	27,047
16	39.63	38,656	39.41	24,999
32	39.64	75,520	38.86	20,903

Table G9: **Cost of richer ROM motion.** Each row adds one component to translation knots at $M=256$, $K_{\mathrm{t}}=8$, with SAD scores and feature residuals. Ratios compare ROM totals with D2GV's original MLP, counting its 82,997 read or 107,646 stored parameters. Smaller totals cost less storage.

ROM variant	Extra parameters	Total	MLP read / ROM	MLP stored / ROM
Translation knots (ours)	–	20,224	4.10×	5.32×
+ per-anchor affine motion	+8,192 ($4MK_{\mathrm{t}}$)	28,416	2.92×	3.79×
+ time-varying site metric G_i	+4,096 ($2MK_{\mathrm{t}}$)	24,320	3.41×	4.43×

H UVG paired per-scene results and motion-rank evidence

Tab. H10 reports the per-scene PSNR breakdown for the UVG columns of Tab. 2 (the last column averages over frames per video, as in Tab. 2). At 120k, UVG-Small margins range from -0.65 dB on ShakeNDry to $+2.87$ dB on Bosphorus, with D2GV leading on ShakeNDry, Beauty, and Jockey at both budgets. Because Bosphorus carries the largest margin at both budgets ($+2.87$ and $+5.50$ dB), excluding it leaves a mean scene margin of $+0.35$ dB at Small and $+0.05$ dB at Large (per-GoP paired margin $+0.44$ and $+0.15$ dB; 50 and 40 of 83 pairs won). Alongside per-GoP paired t-statistics (6.1, 4.3, 21.5, and 14.5 across the four splits), two-sided one-sample t-tests over scene means give $+0.71 \pm 1.18$ dB ($t=1.6$, $p=0.16$, $n=7$) on UVG-Small, $+0.83 \pm 2.19$ dB ($t=1.0$, $p=0.35$) on UVG-Large, $+4.14 \pm 1.26$ dB ($t=8.0$, $p<0.001$, $n=6$) on DAVIS-Small, and $+3.41 \pm 1.58$ dB ($t=5.3$, $p=0.0032$) on DAVIS-Large. Note that at the scene level the margin is statistically significant on DAVIS and not on UVG.

Table H10: **UVG per-scene PSNR.** Small rows are above Large rows for the paired 120k fits of Tab. 2, with D2GV's original MLP. Scene values are GoP means in dB, ordered by the Small margin Δ. Bold marks the higher PSNR.

	Bosphorus	ReadySetGo	HoneyBee	YachtRide	Jockey	Beauty	ShakeNDry	Mean, 98 GoPs	Mean, 7 videos
D2GV (Small)	35.73	28.04	40.46	30.42	**35.26**	**36.38**	**37.56**	34.64	34.80
DSAD (Small)	**38.60**	**29.20**	**41.59**	**31.18**	35.14	36.20	36.91	**35.45**	**35.50**
Δ (Small)	+2.87	+1.16	+1.13	+0.76	−0.11	−0.18	−0.65	+0.81	+0.70
D2GV (Large)	34.18	29.50	42.12	31.43	**36.61**	**36.89**	**38.75**	35.42	35.61
DSAD (Large)	**39.68**	**30.55**	**42.32**	**32.27**	36.23	36.63	37.62	**36.39**	**36.43**
Δ (Large)	+5.50	+1.06	+0.19	+0.84	−0.39	−0.25	−1.13	+0.97	+0.82

Motion-rank protocol. For every GoP of each UVG-Small scene, we extract pretrained RAFT [44] (torchvision [45]) optical flow from the GoP's first frame to its next four frames, resample each displacement field (u, v) to a 32×32 grid, center rows, and compute the energy fraction in the leading singular component (Fig. H7, where solid curves show DSAD site displacements $\tanh(\operatorname{atanh} x + \Delta(x,t))$ on GoP_1 across its 9 non-zero frame steps and dotted curves show the scene-averaged 4-step RAFT flow). Averaged per scene, this top-1 fraction correlates with the paired 120k PSNR margin at $r_s = 0.893$ ($p = 0.012$; $r_s = 0.500$ over all frames of each GoP; Sec. 4.2), and remained positive under the earlier 60k protocol ($r_s = 0.786$, $p = 0.048$; $r_s = 0.536$, $p = 0.236$, against lightweight D2GV). Note that on DAVIS-Small the correlation is not statistically significant ($r_s = +0.37$, $p = 0.497$ across scenes; $+0.20$ across 51 GoPs).

Scene motion and texture. Tab. H11 reports mean RAFT flow magnitude from frame 0 to later frames (D_{far}), between consecutive frames (D_{step}), and mean luma Sobel gradient magnitude per UVG-Small scene (HoneyBee and ShakeNDry have the lowest displacement and Beauty the lowest texture; on the DAVIS pool, classic-car (133 and 42 px) and motocross-jump (140 and 91 px) of the 3 scenes with negative mean margins exceed the pool median displacement of 43/9 px across 32 scenes, while dancing (22 and 6 px) does not).

Table H11: **UVG-Small motion and texture.** Values average each scene's protocol GoPs, with RAFT at 1280×720. Scenes are ordered by D_{far}, alongside the paired margin of Tab. H10.

Scene	D_{far} (px)	D_{step} (px)	Sobel ($\times10^{-2}$)	ΔPSNR (dB)
HoneyBee	0.6	0.2	1.62	$+1.13$
ShakeNDry	4.8	1.3	2.37	-0.65
Bosphorus	33.0	6.6	1.47	$+2.87$
Beauty	49.8	36.7	0.91	-0.18
ReadySetGo	96.7	28.6	2.84	$+1.16$
YachtRide	99.3	14.3	2.06	$+0.76$
Jockey	174.4	108.6	1.20	-0.11

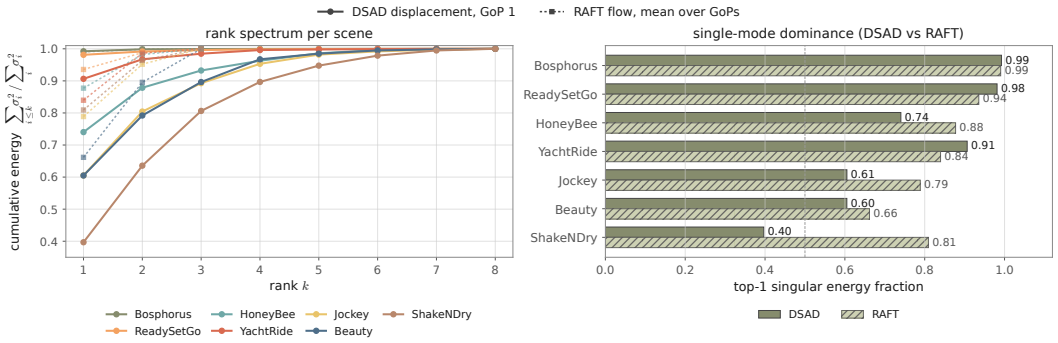


Figure H7: **UVG-Small motion spectra.** Left shows cumulative singular energy for DSAD (solid) and RAFT [44] (dotted). Right shows top-1 energy per scene, ordered by decreasing paired margin. The dashed line marks one half.

I Bunny: comparison at D2GV-AR's budget and parameter split

Tab. I12 compares DSAD and D2GV on Bunny (13 GoPs, 132 frames at 1280×720) at D2GV-AR's ~ 3 M per-video budget (Sec. 4.2). Our paired 60k fits match 3.00 M read parameters per video (23,381 SAD sites for DSAD vs. $N^{\text{GS}} = 18{,}471$ Gaussians with the original MLP for D2GV; Appendix B), while lightweight-MLP D2GV with 24,566 Gaussians (earlier stored count) reaches 34.65 dB / 0.9764.

Tab. I13 decomposes the canonical and motion parameter counts of the Bunny ablation rows in Tab. 3 (Sec. 4.3) under the 60k protocol. All three rows share the same ROM ($K_t = 8$, no motion penalties) and match decoder-read parameters (30,428 Gaussians at 8 scalars each for DSAD–GS versus 27,047 SAD sites at 9 scalars each for DSAD, sitting 8.3% below the D2GV reference's

Table I12: **Bunny at D2GV-AR's budget.** The 132 native 1280×720 frames form 13 GoPs at the ∼3 M per-video target. Italics mark transcribed results, and upright rows are our paired fits under the 60k protocol. Shading marks the highest GoP-primitive PSNR, with higher values better.

Method	PSNR ↑	MS-SSIM ↑	Per-video Params
Full-clip INRs (as reported by their papers)			
HiNeRV [19]	*41.14*	–	*3.25 M*
HNeRV-Boost [56]	*41.09*	*0.9934*	*3.08 M*
GoP primitives			
D2GV-AR (Ours-SR) at 3 M [27]	*37.64*	–	*3 M (nominal, its Tab. 13)*
D2GV [26] (paired, original MLP)	36.42	0.9854	3.00 M
DSAD (60k, ours)	39.20	0.9923	3.00 M

287,645 read parameters at $N^{\mathrm{GS}} = 25{,}581$). The first two rows vary only the canonical appearance representation under cyclic frame order, while the third row adds the training recipe of Sec. 3.4. Tab. I14 supplements Tab. 3 with reference rows sized under the earlier stored-parameter count.

Table I13: **Bunny appearance-head ablation.** PSNR/MS-SSIM average 13 GoPs at 1280×720 under the 60k protocol, with higher values better. Rows share the ROM and per-GoP budget. The first two vary only canonical appearance, and the last adds the training recipe.

Method	PSNR ↑	MS-SSIM ↑	Canonical params	Motion params	Total params
DSAD–GS (GS + ROM), $N^{\mathrm{GS}} = 30{,}428$	37.89	0.9891	243,424	20,224	263,648
DSAD (SAD + ROM), 27,047 SAD sites, without the recipe	38.85	0.9920	243,423	20,224	263,647
DSAD (SAD + ROM, **ours**), with the recipe	**39.62**	**0.9930**	243,423	20,224	263,647

Table I14: **Bunny aggregate with stored-count references.** Rows extend Tab. 3 with D2GV and DSAD–GS references under the 60k protocol and earlier count. PSNR/MS-SSIM are mean ± sample standard deviation over 13 GoPs, with higher values better.

Method	#GoPs	PSNR ↑	MS-SSIM ↑	Total params
D2GV [26] (GS appearance + MLP motion, original MLP; parameters read)	13	37.54 ± 1.36	0.9888 ± 0.0041	287,645
D2GV [26] (the same at its stored count; reference only)	13	36.75 ± 1.21	0.9865 ± 0.0044	287,646
DSAD–GS (GS appearance + ROM motion, ablation; parameters read)	13	37.89 ± 1.50	0.9891 ± 0.0045	263,648
DSAD–GS (the same at its stored count; reference only)	13	37.53 ± 1.46	0.9882 ± 0.0047	263,647
DSAD (SAD appearance + ROM motion, without the recipe)	13	38.85 ± 0.81	0.9920 ± 0.0018	263,647
DSAD (SAD + ROM motion, with the recipe, ours)	13	39.62 ± 0.98	0.9930 ± 0.0019	263,647

J Additional analyses

Unless noted otherwise, the following analyses use the 60k protocol with the final training recipe of Sec. 3.4 ($M = 256$, 9 scalars per site for DSAD, and D2GV sized by decoder-read parameters; Appendix B) and evaluate frames decoded from stored parameters. Earlier experiments that store 10 scalars per site and size D2GV by total stored parameters are marked "earlier count."

UVG frames. Following D2GV-AR [27], we subsample every fourth frame from the first (150 frames per sequence and 75 for ShakeNDry across 98 GoPs of 10 frames, with shorter final GoPs where sequence length is not a multiple of 10). Under the earlier 60k protocol on the first 150 consecutive frames (80 for ShakeNDry; 408 GoP–budget pairs including DAVIS), DSAD achieves a paired margin of +4.78 dB over lightweight D2GV (408/408 pairs won).

Anchor count. The 1080p fits at Tab. 2's Large budgets use the 720p Large anchor count; all other studies keep $M = 256$ unless they vary M.

Stored site axes. At a matched budget, storing each site's axis as a scalar angle yields 11% more sites than the 10-number-per-site count and improves mean PSNR by +0.19 dB across Tab. 2 (277/302 pairs higher), while the 10-number row still ranks first among compared GoP primitives in every PSNR column.

Timing data and protocol. Tab. J15 reports decoding and encoding throughput on GoP_1 of every scene across the four splits of Tab. 2 at 1280×720 on a single idle RTX 2080 Ti (HiNeRV on full videos; D2GV and HiNeRV encode times scaled from timed windows, HiNeRV from one video per split). For DSAD, cold-start decoding starts from stored parameters on the GPU and reconstructs site axes (0.134–0.142 ms per GoP) and exact top-8 per-frame candidates, whereas steady-state decoding reuses cached candidates; in both modes Φ of Eq. (2) is evaluated once per GoP pass. Paired D2GV (120k, original MLP sized by read parameters) is timed both with its unread opacity branch omitted and GPU-built frame timestamps (yielding bit-identical frames, 7–12% and 10–24% faster) and in its unmodified original implementation (224–376 FPS); lightweight D2GV (60k protocol of Tab. J23) is timed similarly (365–466 FPS unmodified). At 1920×1080 (GoP_1 across the 26 scene–budget pairs of Tab. J16's first budget set), DSAD decodes at 465–575 FPS from a cold start and 2,531–3,212 FPS at steady state. Without our fused CUDA kernels, eager PyTorch motion evaluation decodes at 290–520 FPS cold (463–857 FPS steady) and trains at 149–382 s/frame (estimated on four GoP_1 fits, 9.6–20.9× slower than fused CUDA).

Table J15: **Decode and encode speed.** Split means use 1280×720 inputs on one idle RTX 2080 Ti. Higher decode FPS and lower encode seconds per frame are better. Lightweight D2GV uses the 60k protocol of Tab. J23. Fig. 3 plots the cold-start, steady-state, paired-control and HiNeRV rows.

Method	UVG-Small	UVG-Large	DAVIS-Small	DAVIS-Large
Full-clip INR				
HiNeRV [19] (our reproduction), decode FPS	20	14	14	12
HiNeRV, encode s/frame (300 epochs)	68.9	92.6	64.9	82.1
GoP primitives				
D2GV [26], paired control, decode FPS	444	279	501	294
original implementation, decode FPS	336	224	376	238
D2GV, lightweight MLP (60k), decode FPS	587	428	597	452
DSAD, decode FPS, cold start	909	769	1,095	706
DSAD, decode FPS, steady state	4,946	4,196	5,187	4,361
D2GV, paired control, encode s/frame (120k)	161.2	177.0	187.5	161.0
D2GV, lightweight MLP, encode s/frame (60k)	78.5	66.5	89.6	66.4
DSAD, encode s/frame (120k)	16.1	19.3	16.7	20.9

MCL-JCV. On all 24 clips of MCL-JCV [48] resized to 1280×720 (343 GoPs: 336 of 10 frames and 7 trailing 5-frame GoPs) at the UVG-Small per-GoP budget under the 60k protocol (23,714 SAD sites / 233,650 parameters for DSAD vs. 18,959 Gaussians / 234,669 read parameters for paired original-MLP D2GV), DSAD averages 37.43 dB vs. 34.79 dB, achieving a paired margin of $+2.64 \pm 1.69$ dB ($t = 28.9$, $327/343$ GoPs won; D2GV leads on $1/24$ clips: videoSRC04).

1080p repeat. Tab. J16 repeats the paired comparison at native 1920×1080 (1600×900 on DAVIS boat) under the 60k protocol at both Tab. 2's per-GoP budgets and D2GV's native 1080p budgets (0.57–1.16% higher total for DSAD). Across the four splits, DSAD achieves paired margins of $+1.15$–$+4.16$ dB at the first budget set ($266/302$ pairs won, mean $+2.21$ dB; D2GV leads on UVG-Large ShakeNDry (-0.54 dB, $1/8$), UVG-Small ShakeNDry (-0.16 dB, $1/8$) and UVG-Large Beauty (-0.005 dB, $9/15$), with 36 pairs below by at most 0.90 dB) and $+1.41$–$+3.98$ dB at the native 1080p budgets ($271/302$ pairs won, mean $+2.28$ dB; D2GV leads on UVG-Large ShakeNDry (-0.36 dB, $1/8$), with 31 pairs below by at most 0.71 dB and smallest scene margins on UVG-Large ShakeNDry (-0.36 dB, $1/8$) and UVG-Small Beauty ($+0.004$ dB, $9/15$)).

DAVIS pool. Across 70 ten-frame GoPs from 32 DAVIS scenes at the Bunny-ablation budget under the 60k protocol (Tab. J17; 27,047 sites / 263,647 parameters for DSAD, 30,428 Gaussians / 263,648 for DSAD–GS (Tab. 3 configuration), and 25,581 Gaussians / 287,645 read parameters for original-MLP D2GV), full DSAD gains $+2.96 \pm 1.93$ dB ($t = 12.8$, $64/70$ GoPs won) and motion-only DSAD–GS gains -0.29 dB ($37/70$ won). Note that DSAD trails on 6 GoPs concentrated in 3 scenes (classic-car, dancing and motocross-jump).

GoP length. Partitioning frames 1–60 of Bunny and DAVIS camel and cows into GoPs of length T at fixed per-GoP budgets under the 60k protocol (Tab. J18), DSAD outperforms original-MLP D2GV (25,581 Gaussians) by $+3.80$, $+7.43$, and $+8.93$ dB at $T = 10$, 20, and 30, winning all 45 GoPs.

Anchor count on four DAVIS GoPs. Scanning $M \in [128, 1024]$ at a matched per-GoP budget of 263,654 under the 60k protocol (Tab. J19), PSNR increases up to $M = 1024$ on bmx-trees_1, motocross-jump_1, and motocross-jump_2 (closing 70% and 73% of the gap to D2GV on motocross-

Table J16: **1080p repeat.** Per-video PSNR and paired per-GoP margins use the 60k protocol at Tab. 2's and D2GV's native 1080p budgets. D2GV uses its original MLP, sized by parameters read, with higher PSNR better. Native-budget margins are smallest on UVG-Large ShakeNDry (-0.36 dB, $1/8$) and UVG-Small Beauty ($+0.004$ dB, $9/15$).

Split	D2GV	DSAD	ΔPSNR	wins
Tab. 2 budgets				
UVG-Small	32.28	33.33	+1.15	79/98
UVG-Large	32.92	34.11	+1.32	81/98
DAVIS-Small	27.81	31.95	+4.16	53/53
DAVIS-Large	29.84	33.70	+3.86	53/53
D2GV-native budgets				
UVG-Small	32.55	33.87	+1.42	85/98
UVG-Large	33.18	34.46	+1.41	80/98
DAVIS-Small	29.32	33.30	+3.98	53/53
DAVIS-Large	30.62	34.41	+3.80	53/53

Table J17: **DAVIS pool.** Per-scene PSNR differences compare DSAD–GS and the full method ($\approx 263{,}650$ parameters each) with D2GV's original MLP ($287{,}645$ read parameters per GoP). Positive differences favor DSAD, with GoP wins alongside. Fits use the 60k protocol.

Scene	GoPs	DSAD–GS	wins	DSAD	wins
bear	2	+1.21	2/2	+3.99	2/2
bike-packing	2	−0.68	0/2	+1.85	2/2
blackswan	3	+0.87	3/3	+3.69	3/3
bmx-bumps	2	−2.76	0/2	+1.66	2/2
bmx-trees	3	−3.32	0/3	+1.35	3/3
boat	3	+0.50	3/3	+3.25	3/3
boxing-fisheye	2	−0.85	0/2	+1.92	2/2
breakdance	2	−1.75	0/2	+0.90	2/2
breakdance-flare	2	+0.22	2/2	+1.80	2/2
bus	2	+1.06	2/2	+3.40	2/2
camel	3	+2.93	3/3	+5.98	3/3
car-roundabout	2	−0.26	1/2	+4.32	2/2
car-shadow	2	−2.21	0/2	+6.17	2/2
car-turn	2	+1.04	2/2	+3.39	2/2
cat-girl	2	−0.63	0/2	+2.67	2/2
classic-car	2	−3.02	0/2	−0.63	0/2
color-run	2	−1.35	0/2	+3.39	2/2
cows	3	+1.04	3/3	+3.21	3/3
crossing	2	+0.34	2/2	+4.45	2/2
dance-jump	2	+0.67	2/2	+3.52	2/2
dance-twirl	2	+0.97	2/2	+3.58	2/2
dancing	2	−2.57	0/2	−0.47	0/2
disc-jockey	2	−1.22	0/2	+0.63	2/2
dog	2	+0.97	2/2	+2.00	2/2
dog-agility	2	−1.53	0/2	+1.35	2/2
drift-chicane	2	−0.07	0/2	+4.49	2/2
horsejump-high	2	+0.76	1/2	+5.98	2/2
motocross-jump	2	−4.55	0/2	−0.60	1/2
parkour	2	+0.13	1/2	+3.77	2/2
scooter-black	2	−0.46	1/2	+2.39	1/2
soapbox	2	+1.38	2/2	+5.10	2/2
train	3	+1.41	3/3	+4.17	3/3
All	70	−0.29	37/70	+2.96	64/70

Table J18: **GoP-length sweep.** Mean PSNR compares DSAD and D2GV (original MLP), with paired differences at fixed per-GoP budgets. Fits use the 60k protocol. Higher PSNR is better.

Data	T	GoPs	DSAD	D2GV	ΔPSNR	wins
Bunny	5	12	40.04	38.36	$+1.68 \pm 0.15$	12/12
Bunny	10	6	38.78	36.37	$+2.41 \pm 0.80$	6/6
Bunny	20	3	37.04	32.48	$+4.55 \pm 2.32$	3/3
Bunny	30	2	35.63	26.75	$+8.87 \pm 1.19$	2/2
DAVIS	10	12	31.95	27.45	$+4.50 \pm 1.41$	12/12
DAVIS	20	6	30.24	21.37	$+8.86 \pm 1.18$	6/6
DAVIS	30	4	28.83	19.88	$+8.95 \pm 1.39$	4/4

jump_2 at $M = 768$ and 1024) and peaks at $M = 256$ on camel_1 (within 0.73 dB above 256). Holding $N = 27{,}047$ fixed likewise improves PSNR up to $M = 1024$ on the first three GoPs (5.7–12.8% above D2GV's budget from $M = 768$ upward).

Table J19: **DAVIS anchor-count scan.** Per-GoP totals are at most 263,654, with site counts in the last row. Bold marks DSAD's highest PSNR per GoP. The last column gives paired D2GV with its original MLP and 287,645 read parameters under the 60k protocol.

GoP	128	256	384	512	768	1024	D2GV
bmx-trees_1	30.49	30.85	31.43	31.90	32.18	**32.24**	30.70
camel_1	32.75	**32.92**	32.69	32.67	32.63	32.20	25.62
motocross-jump_1	33.40	34.79	35.24	35.40	35.59	**35.76**	34.28
motocross-jump_2	30.86	32.37	32.98	33.10	33.57	**33.62**	34.09
Sites / Gaussians	28,171	27,047	25,924	24,800	22,553	20,306	25,581

Inpainting. Holding out four random rectangular masks (≈20% of pixels per frame) across five DAVIS GoPs under the 60k protocol (Tab. J20), DSAD improves hole PSNR over nearest-observed-frame copying by $+10.12$ dB on average ($+6.30$–$+14.00$ dB) on GoPs labeled moderate by inspection (4.69 dB below full-frame supervision) and by $+0.85$ dB on GoPs labeled fast (14.89 dB below full supervision).

Table J20: **DAVIS inpainting.** Hole PSNR compares the fit, a copy from the nearest observed frame and a control that saw every pixel. The last column gives the fit's observed-pixel PSNR. Fits use the 60k protocol. Higher PSNR is better.

GoP	Motion	Fit	Copy	Gain	Control	Observed
camel_1	moderate	23.98	17.68	+6.30	34.54	31.98
cows_2	moderate	28.40	18.33	+10.07	30.95	31.19
soapbox_1	moderate	36.83	22.83	+14.00	37.81	36.07
bmx-trees_1	fast	15.72	15.08	+0.63	30.77	31.46
motocross-jump_1	fast	19.51	18.45	+1.06	34.23	35.18

Temporal interpolation. Supervising only even frames $(0, 2, \dots)$ under the 60k protocol and decoding at held-out odd timestamps (Tab. J21), DSAD outperforms linear neighbor blending on the DAVIS pool by $+1.54 \pm 2.06$ dB ($50/70$ GoPs won; $+1.75$ dB on interior odd frames excluding the final extrapolated frame, with 22.41 vs. 33.35 dB under full supervision). Note that on slow, smooth Bunny scenes, DSAD trails neighbor blending (-0.39 ± 2.78 dB overall, $+0.48$ dB interior, $5/13$ won; 33.57 vs. 39.20 dB full supervision).

Interpolation against HNeRV-Boost. Under HNeRV-Boost's Bunny interpolation protocol (130 frames at 1280×720, odd frames supervised and 65 even frames held out; Tab. J22), per-GoP DSAD (60k protocol, 3.00 M) achieves 33.68 dB on held-out frames compared with 29.32 dB for full-clip HNeRV-Boost [56] (300 epochs, 3.08 M; $+4.36$ dB), whereas HNeRV-Boost reaches higher PSNR on supervised frames (41.21 vs. 39.45 dB) and linear neighbor blending reaches 34.05 dB.

Edge-band metric. On GoP_1 across all 26 scene–budget pairs of Tab. 2 (120k protocol), we dilate Canny [6] luma edges (Sobel 90th-percentile high threshold, low at half) by 2 px to form an edge

Table J21: **Temporal interpolation.** Mean PSNR on held-out odd frames compares the even-frame fit, frame blending and full supervision. Higher PSNR is better. Fits use the 60k protocol.

Data	GoPs	Fit	Blend	Gain	wins	Control
DAVIS pool	70	22.41	20.87	+1.54	50/70	33.35
Bunny (∼3 M)	13	33.57	33.96	−0.39	5/13	39.20

Table J22: **Bunny interpolation against HNeRV-Boost.** Both methods hold out even-numbered frames f00002, f00004, … among frames 1–130 at 1280×720. PSNR averages each set's frames, with higher values better. DSAD uses the 60k protocol.

Method	Params	Supervised frames	Held-out frames
HNeRV-Boost [56] (our reproduction)	3.08 M	41.21	29.32
DSAD (13 GoP fits)	3.00 M	39.45	33.68
Linear blend of the supervised neighbors (per GoP)	–	–	**34.05**

band covering 20–54% of pixels. DSAD outperforms paired D2GV by +2.77 dB inside the edge band (21/26 GoPs won) and +1.71 dB outside it (21/26 won), with a larger in-band margin on 20/26 GoPs.

Decoding at a higher resolution. Although site positions and the SAD metric are normalized in units of $\max(W, H)$ (with additive radii rescaled proportionally), decoding 720p fits directly at 1920×1080 trails bicubic upsampling of the 720p decode across all 151 UVG-Small and DAVIS-Small GoPs (0.53 and 0.46 dB on average), as sharp cell boundaries between 720p pixel centers are not constrained during 720p training. Note that we therefore make no super-resolution claim for DSAD.

Training with exact candidates and SSIM. Using exact top-8 candidates during the final 6k fine iterations aligns training with exact decoding; an untuned SSIM loss term at D2GV-AR's weight lowered both PSNR and MS-SSIM for DSAD on six 120k GoPs (earlier count).

The earlier 60k protocol. Tab. J23 reports the paired comparison at 10k coarse and 60k fine iterations (D2GV's original training length), with D2GV sized by all stored parameters and DSAD at 10 numbers per site. Relative to Tab. J23, the main 120k protocol of Tab. 2 stretches both schedules 2× (which benefits D2GV more than DSAD, +0.51 vs. +0.23 dB across 48 earlier-count pairs), sizes D2GV by decoder-read parameters (20–35% more Gaussians on UVG-Small), and stores 9 numbers per site for DSAD (11% more sites), resulting in smaller margins in Tab. 2 than in Tab. J23 across every split.

Table J23: **Earlier 60k protocol.** Both methods use 60k fine iterations with their own schedules and, except the nine-number row, the earlier count. Quality is per-video PSNR (dB)/MS-SSIM, with higher values better. Paired margins give DSAD minus D2GV in dB, alongside GoP–budget pairs won.

	UVG–Small		**UVG–Large**		**DAVIS–Small**		**DAVIS–Large**	
Method	PSNR	MS-SSIM	PSNR	MS-SSIM	PSNR	MS-SSIM	PSNR	MS-SSIM
D2GV [26], original MLP	33.65	0.9475	34.84	0.9568	28.30	0.9047	30.92	0.9519
D2GV [26], lightweight MLP	31.58	0.9162	32.18	0.9220	27.75	0.8899	28.47	0.9057
DSAD (10 numbers per site)	34.98	0.9589	35.95	0.9668	33.26	0.9697	35.29	0.9799
DSAD (9 numbers per site; no paired margin)	35.10	0.9598	36.05	0.9674	33.58	0.9716	35.51	0.9804
DSAD (10 numbers per site) − original MLP (wins)	+1.46 (79/98)		+1.24 (74/98)		+5.02 (53/53)		+4.41 (53/53)	
DSAD (10 numbers per site) − lightweight MLP (wins)	+3.46 (98/98)		+3.90 (98/98)		+5.53 (53/53)		+6.87 (53/53)	

D2GV capacity and training length. On GoP_1 across the scene–budget pairs of Tab. 2 under the 60k protocol (earlier count for DSAD): (1) replacing the 9,678-parameter lightweight MLP with D2GV's original 107,646-parameter MLP at equal total budget increases D2GV PSNR by +2.49 dB on average across all 14 UVG GoPs (up to +5.50 dB); (2) sizing original-MLP D2GV by decoder-read parameters rather than stored parameters adds a further +0.48 dB across all 26 pairs; and (3) extending fine-stage training 4× to 240k iterations raises original-MLP D2GV by +0.88 dB

and DSAD by $+0.30$ dB across 14 UVG GoPs, narrowing the mean margin from $+1.48$ to $+0.90$ dB ($+0.49$ dB with DSAD's own schedule).

Anchor editing. Setting a single anchor's displacement coefficients to zero removes one localized motion mode from the decoded video: across 4 DAVIS GoPs under the 60k protocol, the median single-anchor edit modifies 2.2–3.5% of pixels (8-bit values) while leaving all remaining pixels bit-identical.

Rate–distortion with standard compression. We compress trained Bunny parameters (60k protocol, 13 GoPs, 132 frames) post hoc without retraining via uniform scalar quantization (13 bits for site positions and anchor parameters, 11 for displacement knots, 9 for colors, 8 for radii, orientations, anisotropies, knot times/bandwidths, and feature knots, and 7 for temperatures), Morton site ordering, and LZMA, counting every compressed byte. This fixed bit allocation outperforms a uniform bit width across all ten parameter groups by 8.32 dB at equal rate (Tab. J26) and reduces storage to roughly one-fifth of float32 at a cost of 0.30–0.45 dB (Tab. J24). Interpolated to HNeRV-Boost's three operating bitrates, per-GoP DSAD sits 1.42–1.53 dB below full-clip HNeRV-Boost (Tab. J25).

Table J24: **Bunny rate ladder.** The 13 GoPs contain 132 frames at 1280×720, with rungs indexed by sites N, anchors M and knots K_t. Bpp counts every compressed byte, and Δ is compressed minus float32 PSNR. Fits use the 60k protocol. Higher quality at lower bpp is better.

(N, M, K_t)	Params/GoP	fp32 bpp	fp32 PSNR	bpp	PSNR	MS-SSIM	Δ
2,000, 32, 8	20,528	0.070	29.25	0.0166	28.94	0.8975	−0.30
5,000, 64, 8	50,056	0.171	32.75	0.0391	32.40	0.9573	−0.35
10,000, 128, 8	100,112	0.342	35.88	0.0757	35.51	0.9805	−0.37
23,381, 256, 8	230,653	0.789	39.20	0.1682	38.81	0.9915	−0.39
27,047, 256, 8	263,647	0.902	39.63	0.1914	39.21	0.9923	−0.41
41,843, 256, 8	396,811	1.357	40.71	0.2781	40.26	0.9938	−0.45

Table J25: **Matched-rate Bunny comparison.** PSNR/MS-SSIM use HNeRV-Boost's frames 1–130 protocol, with our bytes covering all 132 frames. HNeRV-Boost's bpp counts Huffman code lengths and quantizer ranges, omitting tables and tensor shapes. DSAD uses the 60k protocol. Higher quality is better.

bpp	HNeRV-Boost PSNR	DSAD PSNR	ΔPSNR	HNeRV-Boost MS-SSIM	DSAD MS-SSIM
0.0491	34.83	33.41	−1.42	0.9735	0.9648
0.1013	38.19	36.66	−1.53	0.9874	0.9843
0.2094	40.87	39.43	−1.44	0.9931	0.9926

Table J26: **Shared bit width versus fixed allocation.** The base rung of Tab. J24 covers 13 Bunny GoPs and 132 frames, with every byte counted. Rows quantize all ten groups at one width with the remaining procedure unchanged, followed by the fixed allocation. Fits use the 60k protocol. Higher PSNR/MS-SSIM at lower bpp is better.

Bits per number	bpp	PSNR	MS-SSIM
7	0.1245	23.94	0.7600
8	0.1502	26.84	0.8866
9	0.1934	31.06	0.9573
10	0.2177	35.12	0.9826
11	0.2442	37.88	0.9902
12	0.2695	39.10	0.9923
13	0.2953	39.48	0.9928
fixed allocation	0.1914	39.21	0.9923

K Broader impact

DSAD is a per-GoP reconstruction representation rather than a generative model: it refits the frames it is given and invents no content beyond them (the interpolation and inpainting studies decode held-out frames and holes from the fitted primitives). As an inspectable primitive representation, it could be useful for forensic analysis (per-anchor motion energies expose where motion lives), for video editing (locality of anchors maps to local edits), and for compression research. The same locality, however, can lower the cost of *producing* edited or partially synthetic content from a real clip. We do not view this as a new capability beyond existing primitive video codecs. Downstream systems that combine DSAD with generative components should still be assessed for manipulation risk. The data we use are public, depict scenes rather than identifiable individuals at sensitive granularity, and do not target any decision-making setting.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the per-GoP video-representation setting, matched-budget scope, main contributions, and code availability. The main experiments in Sec. 4 and the scoped limitations in Sec. 5 support and qualify these claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Sec. 5 includes a dedicated discussion of limitations and operating range, covering the matched-budget per-GoP scope, scene-level performance variations across UVG and DAVIS (Appendix H and Appendix J), spatial rank scaling and motion-basis ablations (Tabs. G8 and G7), behavior on low-texture deforming surfaces and under 1080p resolution or extended training schedules, and post-hoc rate–distortion trade-offs relative to full-clip INRs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper does not include theorem/proof-style theoretical results. It contains representation formulas, parameter-count expressions, and a derivation of the unit-determinant anisotropic norm in Appendix D, but no formal theorems whose proofs would need to be provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Secs. 3–4 specify the representation, loss, training protocol, datasets, metrics, budgets, and paired comparison design. Appendices E and B give implementation and parameter-count details. Code and logs will be made available under the MIT license at `https://github.com/Frank-ZY-Dou/Dynamics-Modeling/tree/main/DSAD` and via the project page, `https://frank-zy-dou.github.io/projects/DSAD/`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Code, configurations, per-GoP logs, and reproduction scripts will be made available under the MIT license at `https://github.com/Frank-ZY-Dou/Dynamics-Modeling/tree/main/DSAD` and via the project page, `https://frank-zy-dou.github.io/projects/DSAD/`. The repository will cover the full method (SAD appearance and anchor ROM), the DSAD–GS ablation, and the scripts behind the reported experiments, with instructions for reproducing the main tables. The procedures are also described in Sec. 3, Sec. 4, and Appendix B. The experiments use the public UVG, DAVIS, MCL-JCV, and Big Buck Bunny datasets.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Secs. 3.4–4 specify the D2GV-style GoP protocol, matched parameter budgets, objective, iteration counts, datasets, and metrics. Tab. B4 (Appendix B) lists optimizers, absolute learning rates, schedules, initialization and seeds for DSAD, the paired D2GV and our reproduction of HiNeRV. Appendix F documents residual channels and regularizers.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Sec. 4.2 reports paired win rates and per-split paired t-tests across all 302 UVG and DAVIS GoP–budget pairs, supplemented by scene-level t-tests in Sec. 4.2 and Appendix H. Tab. 3 reports means and standard deviations across all 13 Bunny GoPs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Tab. J15, Sec. 4.2, and Appendix J report per-frame encoding and decoding times on a single NVIDIA RTX 2080 Ti GPU, and Appendix E reports candidate-buffer memory usage. All experiments were conducted on RTX 2080 Ti GPUs. The camera-ready experiments total approximately 2,700 job-hours, including ablations and repeats (GPU-hours are fewer, as up to three D2GV jobs shared a GPU); the submission and rebuttal experiments are outside this total.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes].

Justification: The work uses public video benchmarks and open research code, and does not collect personal data, conduct user studies, or deploy a decision-making system. We are not aware of any deviation from the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes].

Justification: Appendix K frames the work as efficient per-GoP reconstruction with local, inspectable primitives and notes that downstream forensic or editing risk depends on the larger synthesis system in which such representations are used.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The work does not provide a high-risk foundation model, general-purpose image generator, scraped dataset, or human-data asset requiring controlled access.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The paper cites every asset where it is used. The data are UVG (sequences owned by Digiturk, CC BY-NC 3.0 Unported), Big Buck Bunny (CC BY 3.0), and DAVIS 2017 and MCL-JCV under the research terms of use their authors publish. The code includes SAD (MIT), HiNeRV (MIT), Boosting-NeRV for HNeRV-Boost (Apache-2.0) and RAFT through torchvision (BSD-3-Clause). D2GV's repository has no license file, so the package will keep it as an unmodified submodule with our changes as a separate patch. The package will preserve upstream license notices and terms of use for code, data, and trained models.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The paper introduces a new codebase. The code will be made available under the MIT license at `https://github.com/Frank-ZY-Dou/Dynamics-Modeling/tree/main/DSAD` and via the project page, `https://frank-zy-dou.github.io/projects/DSAD/`, including configuration files, logs, parameter-count scripts, and documentation for reproducing the reported tables.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: This work does not involve crowdsourcing, user studies, or human-subject data collection.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: This work does not involve human subjects or crowdsourcing, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes].

Justification: LLMs were used for language editing, drafting parts of the text, manuscript consistency checks, and as a coding assistant, including implementing and testing code and running and monitoring experiments. All such work was reviewed and validated by the authors. LLMs were not part of the proposed model, training objective, or inference procedure, and the method and research direction are the authors'.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.