PHYSGRAPH: POINT GRAPHS FOR PHYSICAL SCENE UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world videos mix rigid bodies, cloth, and liquids – yet existing dynamic-reconstruction methods commit to a single motion model. We present *PhysGraph*, a temporal interpolation and extrapolation pipeline that uses a one-shot, foundation-model-predicted material phase as a *prior* for selecting one of three classical motion models on a shared per-object kNN point graph: Procrustes-style SE(3) for solids, thin-plate-spline residuals for deformables, and optimal transport for liquids. The graph is reused across branches as a computational primitive – for neighborhood-aware correspondence filtering (rigid, elastic), as the elastic-kernel subgraph (TPS), and as the rewireable bipartite source-to-target structure that OT solves over (liquid). Composed with bidirectional Gaussian splatting, the resulting motions synthesize novel frames up to $26\times$ faster than per-scene-optimized baselines. On six synthetic scenes PhysGraph reduces depth error by $2.5\times$ over linear 3D interpolation; on 90 DAVIS sequences it has the highest dynamic-region recall at every stride with zero failures (*vs.* up to 69% for per-scene optimizers), and it leads dynamic-region recall in extrapolation as well. PhysGraph targets large temporal gaps, where per-scene optimizers become ill-conditioned and fail. The source code will be made public.

1 INTRODUCTION

Real-world videos are physically heterogeneous: a single clip may contain a static background, tumbling rigid objects, draping cloth, and pouring liquids, each governed by different physics. Humans effortlessly recognize this (a coffee mug is rigid, a curtain deforms, water flows) whereas reconstruction and motion prediction methods typically commit to a single motion model: rigid SE(3), a learned deformation field, or a generic spring system. Rigid estimation via RANSAC (Fischler & Bolles, 1981; Barath & Matas, 2018; Barath et al., 2020; 2022) cannot model non-rigid deformation; deformation graphs (Sumner et al., 2007; Sorkine & Alexa, 2007) and neural deformation fields (Park et al., 2021a;b; Bozic et al., 2021) struggle with textureless surfaces and liquid motion; and physics-based simulation (Müller et al., 2003; Macklin et al., 2014) requires explicit scene specification. Dynamic reconstruction via NeRFs (Mildenhall et al., 2020; Pumarola et al., 2021; Cao & Johnson, 2023; Fridovich-Keil et al., 2023) or 4D Gaussian splatting (Wu et al., 2024; Yang et al., 2024b; Luiten et al., 2024; Huang et al., 2024) requires per-scene optimization, dense temporal supervision, and, crucially, makes no attempt to identify *what* materials are present.

We argue that *material identity should select the motion model*: rather than fitting a single generic deformation field to an entire scene, PhysGraph first asks *what* each object is made of, then routes it to a model whose inductive bias matches that material's motion. The pipeline (Fig. 1) is modular – off-the-shelf models provide 3D geometry, segmentation, and dense correspondences; our contribution is the per-object motion layer on top. We position PhysGraph as *material-conditional motion-model selection over a shared point graph.* The branches use classical priors (SE(3) Procrustes; TPS bending-energy on the rigid residual; Sinkhorn OT with viscosity-shaped easing) over one per-segment kNN graph, reused as a neighborhood-consistency check for rigid, as outlier rejection plus the elastic-kernel subgraph for TPS, and as a bipartite source-to-target graph that *rewires* between frames for OT; the phase controls the graph's rigidity under motion. Our contributions are:

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

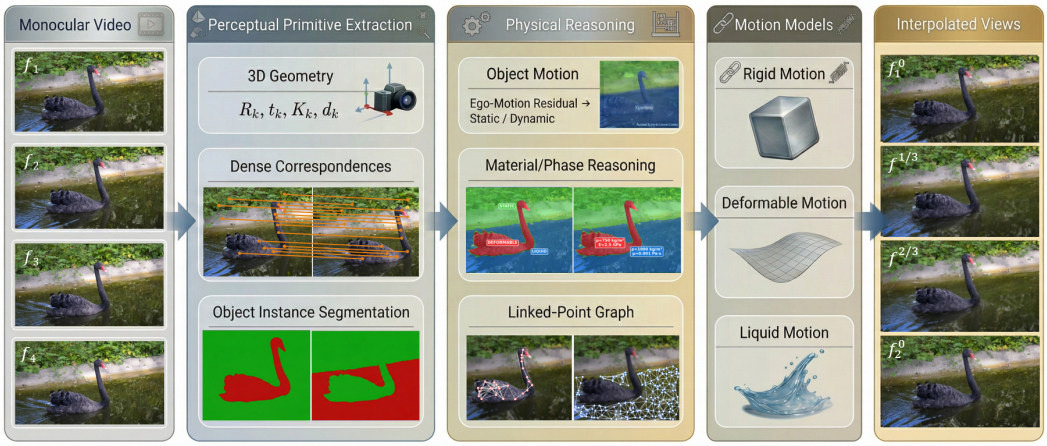


Figure 1: **Pipeline**. Off-the-shelf models extract per-frame depth, camera poses, correspondences, and instance segmentation from a monocular video. A foundation model classifies each segment's material phase (solid, deformable, or liquid), which selects, per object, one of three classical motion models on a shared kNN point graph: SE(3) Procrustes for solids, TPS residuals for deformables, and Sinkhorn-OT for liquids. The recovered motions are composed with bidirectional Gaussian splatting to synthesize novel frames for temporal interpolation or extrapolation.

- A **material-phase-dispatch motion framework**: a per-segment kNN point graph reused across three motion branches (rigid SE(3), elastic TPS, reassignable OT), with the discrete material phase selecting which solver to apply and how the graph behaves under motion.
- A **foundation-model-driven phase classifier** that selects the dispatch from a single frame, paired with a deformable motion-model ablation isolating the choice of TPS-on-residual against ARAP, Laplacian, neural, and CPD alternatives.

PhysGraph achieves $2.5\times$ lower depth error than per-pixel linear 3D interpolation on six synthetic scenes, the highest dynamic-region recall on all 90 DAVIS sequences with zero failures, up to $26\times$ less compute than optimization-based methods, and extrapolation with explicit 3D output.

2 Related Work

Physics-aware scene representations. PhysGaussian (Xie et al., 2024), PAC-NeRF (Li et al., 2023a), Spring-Gaus (Zhong et al., 2024), Physics3D (Liu et al., 2024), and OmniPhysGS (Lin et al., 2025) parameterize a reconstructed scene with physics and run forward simulation; learning-based simulators (Li et al., 2019; Sanchez-Gonzalez et al., 2020) predict particle dynamics from data; classical frameworks – SPH (Müller et al., 2003), PBD (Müller et al., 2007), unified solvers (Macklin et al., 2014) – require explicit scene specification. These methods *generate* novel dynamics from a constitutive model. PhysGraph instead uses material identity as a *prior* to select a classical motion solver per segment, with no per-scene training. Material recognition progressed from MINC (Bell et al., 2015) classifiers to zero-shot vision-language models (Radford et al., 2021; Bai et al., 2023), whose physical reasoning PhysDreamer (Zhang et al., 2024) and PhysBench (Chow et al., 2025) probe; we use them to dispatch each segment to a motion model.

Dynamic 3D reconstruction. NeRF-based dynamics model time via deformation fields (Park et al., 2021a;b), scene flow (Li et al., 2021; 2023b), or space-time factorization (Cao & Johnson, 2023; Fridovich-Keil et al., 2023). The 3DGS (Kerbl et al., 2023) family – 4DGS (Wu et al., 2024), Deformable 3DGS (Yang et al., 2024b), Dynamic 3DGS (Luiten et al., 2024), SC-GS (Huang et al., 2024), DynMF (Kratimenos et al., 2024), Shape of Motion (Wang et al., 2025), MoSca (Lei et al., 2025), GaussianPrediction (Zhao et al., 2024) – requires per-scene optimization over the full video, taking minutes to hours per sequence; DynamicFusion (Newcombe et al., 2015) pioneered non-rigid tracking via embedded graphs. PhysGraph operates between any keyframe pair in a single pass. A feed-forward line predicts 4D Gaussians directly (Lin et al., 2026; Xu et al., 2025; Hur et al., 2026), removing per-scene optimization at the cost of large-scale training and, where conditioned on input cameras, sensitivity to calibration error. PhysGraph needs no training beyond its off-the-shelf components and exposes an explicit per-object motion model that also extrapolates.

Monocular geometry, correspondence, and non-rigid registration. PhysGraph relies on foundation-model perception as a modular front end: monocular depth (Ranftl et al., 2022; Yin et al., 2023; Yang et al., 2024a; Bochkovskii et al., 2025), joint pointmap+pose estimation (Wang et al., 2024; Leroy et al., 2024; Zhang et al., 2025), dense matching and tracking (Edstedt et al., 2023; 2024; Teed & Deng, 2020; Xu et al., 2022; Doersch et al., 2023; Karaev et al., 2024), and text-prompted segmentation (Kirillov et al., 2023; Carion et al., 2025). Our contribution sits on top: classical non-rigid registration – coherent point drift (Myronenko & Song, 2010), ARAP (Sorkine & Alexa, 2007), embedded deformation graphs (Sumner et al., 2007; Bozic et al., 2020) – is replaced per segment by a material-appropriate model: SE(3) for rigid, TPS (Duchon, 1977) on the rigid residual for elastic, and entropic optimal transport (Cuturi, 2013; Peyré & Cuturi, 2019) with McCann displacement interpolation (McCann, 1997) for liquids.

3 PhysGraph: Material-Conditional Motion-Model Selection

Given a monocular video, PhysGraph recovers per-pixel 3D geometry, temporally consistent instance segmentation, per-object material classification with quantitative physical properties, and material-appropriate motion models. The pipeline is modular and optimization-free: off-the-shelf models provide geometry, segmentation, and dense correspondences; our contribution is the physical reasoning layer that classifies materials and selects per-object motion models.

3.1 Scene Reconstruction and Object Discovery

For each keyframe $k \in \{0, \dots, K-1\}$ at time τ_k, a monocular depth model provides a dense point cloud $\mathbf{P}_k \in \mathbb{R}^{H\times W\times 3}$ in world coordinates, camera-to-world pose $[\mathbf{R}_k \mid \mathbf{t}_k] \in \mathrm{SE}(3)$, intrinsics $\mathbf{K}_k$, and a per-pixel confidence map $c_k \in \mathbb{R}^{H\times W}_{\geq 0}$. For each consecutive pair $(k, k+1)$, a correspondence model yields a dense warp field $\mathbf{w}_{k\to k+1} \in \mathbb{R}^{H\times W\times 2}$ with validity mask $\mathcal{V}_k$; bilinear sampling of $\mathbf{P}_{k+1}$ at the warp targets gives per-pixel 3D correspondences $(\mathbf{P}_k(u,v), \mathbf{q}_k(u,v))$. A video segmentation model provides instance masks tracked across time, *i.e.* a segment map $\mathcal{S}_k$ with temporally consistent identities; each pixel inherits its segment's material classification and motion model. Keyframes are not detected: every s-th frame is a keyframe, mirroring the sparse-capture use case (low frame rate, triggered capture, frame drops), and every method receives the identical keyframe set.

3.2 Motion Analysis

We classify each pixel as Static or Dynamic by comparing observed motion against the camera ego-motion prediction. For pixel $\mathbf{x}$ at frame k, the expected location under pure camera motion is $\hat{\mathbf{u}}_{\mathbf{x}} = \pi\big(\mathbf{K}_{k+1},\ \mathbf{R}_{k+1}^{\top}\big(\mathbf{P}_k(\mathbf{x}) - \mathbf{t}_{k+1}\big)\big)$ with π the pinhole projection, and the ego-motion residual is $r(\mathbf{x}) = \|\mathbf{w}_{k\to k+1}(\mathbf{x}) - \hat{\mathbf{u}}_{\mathbf{x}}\|_2$. A pixel is flagged as dynamic if $r(\mathbf{x}) > \tau_r$ or its depth confidence $c_k(\mathbf{x}) < \tau_c$. Background not covered by any prompt (Section 3.3) forms a static segment that bypasses this test, so low confidence in textureless static regions does not flag them. A segment-level majority vote determines each segment's label; a second vote across all frame intervals enforces temporal consistency and prevents flickering. Dynamic segments are sub-typed by their material phase; when material priors are unavailable, they default to rigid.

3.3 Material Classification and Physical Properties

This is the link between perception and physics: for each dynamic segment we identify its material, determine its phase, and retrieve quantitative properties (qualitative examples in Figures 4 and 6).

Material identification. We define 59 concrete noun prompts (*e.g.* "water", "creature", "vehicle", "cloth", "rock"), each pre-mapped to Liquid, Deformable, Solid, or Static (*e.g.* "road", "wall"; full list in Section A). The prompts are hypernyms that partition the material world at the granularity the dispatch needs, not an object whitelist, and the list can be extended at test time without training. A text-prompted segmentation model (Carion et al., 2025) is queried with every prompt on the first keyframe to discover which objects are present (*discovery*); each detected prompt is then propagated across all keyframes by the model's video predictor (*tracking*). Because prompts are queried independently, a multi-material object (a bottle of water) yields overlapping segments; conflicts are

resolved by a phase priority rule, liquid > deformable > solid > static (higher-scoring masks break ties), and each object's phase is a majority vote across keyframes.

Phase determination. The material noun maps to a phase through a *fixed* noun-to-phase table – no language-model reasoning is involved: Solid (rigid under normal forces), Deformable (bends or stretches), or Liquid (flows freely) – and the phase selects the motion model. Segments matched by no prompt, or with low-confidence predictions, fall back to the geometric classification of Section 3.2 and default to rigid, so unlisted objects are still tracked conservatively.

Physical property estimation. Given the frame with the segment highlighted, a vision–language model estimates its continuous properties – Young's modulus E and dynamic viscosity μ – as ordinal scores mapped to physical units (Section D); we query rather than tabulate because a coarse noun spans instances with very different parameters ("fence": wood or metal). These properties can parameterize the motion model: viscosity shapes the temporal easing of liquids (Section 3.7) and stiffness can set the TPS regularizer λ_{tps} (Section 3.6).

3.4 Linked-Point Representation

We treat each segment as a 3D point graph shared across the three motion branches: segment s is a set of points $\mathcal{P}_s = \{\mathbf{p}_i\}_{i=1}^{N_s}$ with a k-nearest-neighbor graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$, where edge (i, j) stores its length $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$. The phase controls how this graph behaves under motion: **Rigid links**: the graph is fixed and edge lengths are preserved ($\|\mathbf{p}'_i - \mathbf{p}'_j\| = d_{ij}$); edges enter the rigid solver as a neighborhood-consistency check on residuals. **Elastic links**: the graph is fixed but edge lengths relax under a soft elastic energy; edges enter the deformable solver as neighborhood-based outlier rejection and as the fully-connected TPS kernel between control points, whose stiffness λ_{tps} controls. **Reassignable links**: the graph *rewires* between frames; a bipartite source-to-target coupling $\mathbf{T}$ is solved per frame pair on a subsample of the segment, and the source kNN graph propagates the coupled displacements to every vertex. The phase thus controls a single property – the graph's *rigidity under motion*, from locked (solid) through elastically coupled (deformable) to freely reassignable (liquid) – while the same kNN structure feeds correspondence filtering and propagation in all three branches.

3.5 Rigid Links: Solid Body Motion

Given point correspondences $\mathbf{p}_i \leftrightarrow \mathbf{q}_i$ between two observed frames, the rigid-link constraint reduces the motion to a single SE(3) transform $(\mathbf{R}_s, \mathbf{t}_s)$ with $\mathbf{q}_i \approx \mathbf{R}_s\mathbf{p}_i + \mathbf{t}_s$, which we estimate robustly by minimizing the Geman-McClure (Geman & McClure, 1985) objective

$$\min_{\mathbf{R},\mathbf{t}} \sum_i \rho_\sigma\big(\|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|\big), \quad \rho_\sigma(r) = \frac{r^2}{r^2 + \sigma^2}, \tag{1}$$

initialized by SVD-Procrustes and optimized with graduated non-convexity. The kNN graph validates the fit: correspondence i is an inlier only if its post-fit residual is consistent with the median residual of its neighbors $\mathcal{N}(i)$ in $\mathcal{G}_s$ (MAD-based threshold), which rejects isolated bad matches that survive the global objective. Segments with inlier ratio $< 30\%$ or median residual > 0.1 m fall back to linear motion. The transform can be evaluated at any time – interpolating or extrapolating. For $\alpha = (t - \tau_k)/(\tau_{k+1} - \tau_k)$, the pose is obtained by Slerp around the centroid $\bar{\mathbf{p}}_s$:

$$\mathbf{R}_t^{(s)} = \texttt{Slerp}\big(\mathbf{I}, \mathbf{R}_s; \alpha\big), \quad \mathbf{t}_t^{(s)} = (1 - \alpha)\,\bar{\mathbf{p}}_s + \alpha\,(\mathbf{R}_s\bar{\mathbf{p}}_s + \mathbf{t}_s) - \mathbf{R}_t^{(s)}\bar{\mathbf{p}}_s. \tag{2}$$

3.6 Elastic Links: Deformable Object Motion

For deformable objects, edges are soft constraints: nearby points move similarly, but local stretching is permitted. We decompose motion into a global rigid component plus a residual non-rigid deformation. A rigid transform captures the dominant motion; residual correspondences $\tilde{\mathbf{q}}_i = \mathbf{R}_s^\top(\mathbf{q}_i - \mathbf{t}_s)$ isolate the non-rigid part. Each residual is compared against the median residual of its kNN neighborhood in $\mathcal{G}_s$ via a MAD threshold, and isolated outliers are discarded before the solve. We model the residual with a thin-plate spline (TPS) (Duchon, 1977), a radial-basis interpolant whose closed-form solution minimizes the integrated bending energy of a thin shell – a smooth motion regularizer whose bending-energy prior suits cloth-like deformation. From the filtered residuals, N_c control

points are selected by farthest-point sampling on $\mathcal{G}_s$, forming a fully-connected weighted subgraph, and the TPS system

$$\begin{bmatrix} \mathbf{K} + \lambda_{\text{tps}}\mathbf{I} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \tag{3}$$

fits the control-point displacements $\mathbf{d}$, where $K_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$ is the polyharmonic kernel (the edge lengths d_{ij} of the elastic subgraph), $\mathbf{P} = [\mathbf{1} \mid \mathbf{p}_1 \cdots \mathbf{p}_{N_c}]^\top$ the affine basis, $\mathbf{w}, \mathbf{a}$ the radial-basis and affine coefficients, and the Tikhonov weight λ_{tps} – fixed or set from the stiffness E (Section 4) – controls how strongly the field penalizes non-affine bending ($\lambda_{\text{tps}} \to \infty$ removes the radial terms and leaves the affine fit on top of the rigid transform). The fitted displacement field $\mathbf{f}_{\text{tps}}$ is evaluated at every non-control vertex. At fraction α, $\hat{\mathbf{p}}_i(\alpha) = \mathbf{p}_i + \alpha\, \mathbf{f}_{\text{tps}}(\mathbf{p}_i)$ is composed with the interpolated rigid transform, $\mathbf{P}_t(\mathbf{p}) = \mathbf{R}_t^{(s)}\, \hat{\mathbf{p}}(\alpha) + \mathbf{t}_t^{(s)}$.

3.7 Reassignable Links: Liquid Motion via Optimal Transport

Liquids do not preserve point identity across frames: particles mix, surfaces reshape, and optical flow cannot provide reliable correspondences, so the motion model must *discover* how mass moves between two observed point clouds. We frame this as graph rewiring: the fixed source-to-source edges are replaced by a bipartite source-to-target coupling that reassigns each vertex's target. Both clouds are subsampled to N_{OT} points, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N_{\text{OT}} \times 3}$, and entropic optimal transport (Cuturi, 2013) returns a coupling $\mathbf{T} \in \mathbb{R}_{\geq 0}^{N_{\text{OT}} \times N_{\text{OT}}}$ that we read as a weighted bipartite graph:

$$\min_{\mathbf{T}} \sum_{ij} T_{ij}\, \|\mathbf{A}_i - \mathbf{B}_j\|^2 - \varepsilon\, H(\mathbf{T}), \qquad \mathbf{d}_i^{\text{OT}} = \sum\nolimits_j \bar{T}_{ij}\, \mathbf{B}_j - \mathbf{A}_i, \quad \bar{T}_{ij} = T_{ij} / \textstyle\sum_l T_{il}, \tag{4}$$

subject to uniform marginals, with ε controlling the entropic blur of the coupling (smaller ε gives a sparser plan); each subsampled source point's displacement $\mathbf{d}_i^{\text{OT}}$ is the barycentric map of its outgoing edges, propagated to every vertex of the segment by inverse-distance weights over its k=8 nearest subsampled points. The rigid component $(\mathbf{R}_s, \mathbf{t}_s)$ of a liquid segment is the robust SE(3) fit of Section 3.5 when it passes the reliability gate and otherwise the centroid translation with $\mathbf{R}_s$=$\mathbf{I}$; OT is solved on the rigid residual, with the target taken to the source frame, $\mathbf{B} \leftarrow \mathbf{R}_s^\top(\mathbf{B} - \mathbf{t}_s)$.

Viscosity-aware temporal easing. Constant-velocity interpolation looks rigid; viscous fluids are visually slower at the start and end of a pour. As a visual prior we remap the linear fraction α through the regularized incomplete Beta CDF,

$$\alpha_{\text{eff}} = F_{\text{Beta}(a,a)}(\alpha), \quad a = 1 + \tfrac{1}{2}\log_{10}(\mu/\mu_{\text{ref}}), \qquad \mathbf{P}_t(\mathbf{p}) = \mathbf{R}_t^{(s)}\big(\mathbf{p} + \alpha_{\text{eff}}\, \mathbf{d}^{\text{OT}}(\mathbf{p})\big) + \mathbf{t}_t^{(s)}, \tag{5}$$

with $\mu_{\text{ref}} = 10^{-3}$ Pa·s, and compose the scaled OT displacement with the rigid motion via McCann interpolation (McCann, 1997). Water ($a \approx 1$) is linear; honey ($a \approx 2.85$) follows an S-curve.

From motion models to novel frames. For both interpolation (between keyframes A, B) and extrapolation ($\alpha > 1$ beyond the last pair), each segment is transformed by its own motion model (Equations (2), (3) and (5)), camera poses are interpolated by Slerp for rotation and linearly for translation, and forward/backward point clouds are merged and rendered via Gaussian splatting (Kerbl et al., 2023). Fallbacks ensure robustness: unreliable rigid fits or excessive TPS distortion revert to linear interpolation; failed OT reverts to blending (Section 4.4 measures how often they fire). Because each keyframe pair is solved independently and anchored at both observed endpoints, interpolation errors do not accumulate over the sequence; drift applies only to extrapolation.

4 Experiments

Implementation details. **Depth and pose:** MonST3R (Zhang et al., 2025). **Correspondences:** CoTracker (Karaev et al., 2024) (80×80 grid) for all DAVIS experiments; Section E.2 shows it is the most robust of four flow sources at $s \geq 10$. **Segmentation and materials:** SAM 3 (Carion et al., 2025) with the 59 material nouns. **Rendering:** Gaussian splatting (Kerbl et al., 2023) via gsplat, one isotropic Gaussian per pixel of scale $\sigma = 0.5\,|z_{\text{cam}}|/f_{\text{avg}}$. **Rigid:** Geman-McClure with graduated non-convexity (500 iterations, σ=0.05 m). **Deformable:** TPS with up to N_c=100 control points, MAD multiplier 3, and a fixed regularizer λ_{tps}=10^{-4}; a stiffness-adaptive variant mapping E log-linearly to $\lambda_{\text{tps}} \in [10^{-4}, 0.05]$, $N_c \in [30, 150]$, and MAD multiplier $\in [1.5, 4]$ changes PSNR by

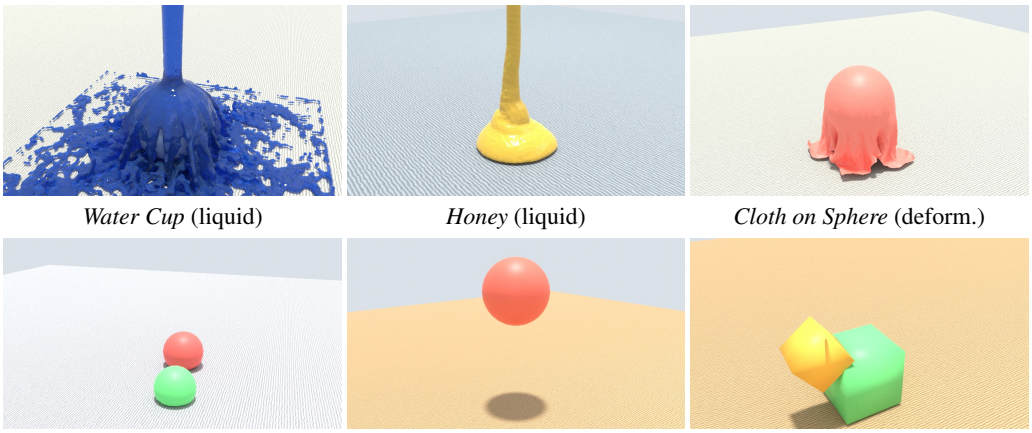


Figure 2: **Synthetic benchmark.** Six Blender scenes spanning three material phases: liquids (water, honey), deformable objects (cloth, elastic balls), rigid bodies (bouncing ball, tumbling cubes). Each scene provides 200 frames at 854×480 with full ground-truth annotations.

Table 1: **Temporal interpolation** at stride s=5, averaged over 6 Blender scenes (2 liquid, 2 deformable, 2 rigid), with GT depth and poses. Centroid error (CE, px), mask IoU, PSNR/SSIM, dynamic-region PSNR (dPSNR), depth MAE/RMSE (m). **Bold**: best per column (splines win full-frame SSIM through the static background; see text).

Method	CE (px)↓	IoU ↑	PSNR (dB)↑	SSIM ↑	dPSNR (dB)↑	Depth (m) MAE↓	Depth (m) RMSE↓
Linear	8.8	0.885	22.91	0.526	24.52	0.0090	0.0417
Rigid body	17.8	0.851	22.43	0.510	24.06	0.0450	0.0827
Spline (cubic)	82.0	0.589	18.08	**0.664**	18.07	0.0201	0.0638
Spline (CR)	82.0	0.590	18.15	**0.664**	18.16	0.0170	0.0564
PhysGraph	**7.9**	**0.903**	**23.53**	0.536	**25.37**	**0.0036**	**0.0152**

<0.01 dB (Section E.3), so we report the fixed setting. **Liquid:** Sinkhorn OT with 2500 (synthetic) / 2000 (DAVIS) subsampled points, ε=10^{-2} relative to the median transport cost, viscosity-aware easing. All remaining thresholds and fallback criteria are listed in Section A. All timings use one RTX 4090.

4.1 Interpolation on Synthetic Scenes

Setup. Six Blender scenes (Figure 2) – two liquids (*Water Cup*, *Honey*), two deformables (*Cloth on Sphere*, *Elastic Balls*), two rigid bodies (*Bouncing Ball*, *Tumbling Cubes*) – each 200 frames at 854×480 with full GT annotations. Every s-th frame ($s \in \{3, 5, 10, 25\}$) is a keyframe and the remaining frames are synthesized, using GT poses and depth to isolate the motion-modeling contribution. Baselines: per-pixel *Linear* 3D interpolation, per-segment *Rigid body* SE(3) interpolation, and *Cubic/Catmull-Rom spline* fitting through multi-keyframe positions.

Results. In Table 1 (stride 5), the most informative comparison is against *Linear* 3D interpolation, a competitive baseline on the same per-segment motion budget and the "no motion model" condition: +0.62 dB PSNR and 2.5× lower depth MAE (0.0036 vs. 0.0090 m). Rigid-body interpolation and 3D splines fall further behind because four of the six scenes are liquid or deformable, regimes a single SE(3) cannot represent and that splines amplify into 80+ px drift. The splines' higher *full-frame* SSIM reflects the static background: they fail on the dynamic object (centroid error 82 px, IoU 0.59), but SSIM is dominated by the ∼90% static background, which splines render sharply from unperturbed keyframe geometry; every object-sensitive metric favors PhysGraph. The advantage over Linear *grows with the temporal gap*, from +0.6 dB at stride 5 to +4.9 dB at stride 25 (per-stride breakdown in Section B), where linear blending cannot capture rigid arcs, cloth drape, or liquid transport.



(a) GT (b) **PhysGraph** (c) MoSca (d) GP (e) SoM

Figure 3: **Qualitative comparison** on four DAVIS sequences (rows: *roundabout*, *crossing*, *cat-girl*, *blackswan*) vs. MoSca, GaussianPrediction (GP), and Shape of Motion (SoM). PhysGraph produces sharper boundaries and more coherent motion; hatched boxes mark failures.

4.2 Interpolation on Real-World Scenes

We evaluate on all 90 sequences of DAVIS 2017 (Pont-Tuset et al., 2017) with no ground-truth supervision – MonST3R depth and poses, CoTracker correspondences, SAM 3 segments and materials – so error propagated through the off-the-shelf chain is included in every number. **Baselines:** three optimization-based dynamic 3DGS methods, each requiring per-scene optimization over the full video: MoSca (Lei et al., 2025), Shape of Motion (Wang et al., 2025), and GaussianPrediction (Zhao et al., 2024); and MoVieS (Lin et al., 2026), a feed-forward 4D Gaussian model that needs no per-scene optimization. Fig. 3 shows qualitative examples. **Protocol:** keyframes are sampled at stride s and every method receives the same keyframe-only input (MoSca/SoM run on the keyframe set; GaussianPrediction, an extrapolator by design, is fed bidirectional keyframes and queried at intermediate times). Because optimization-based methods can fail catastrophically, we report *recall* (% sequences with PSNR $\geq 15/20$ dB, full-frame and dynamic-region) plus failure rate and runtime. MoVieS is conditioned on camera parameters and receives the same MonST3R geometry as PhysGraph, with held-out cameras from the same Slerp interpolation of the bracketing keyframes; it renders at its native 518×294 and the shared evaluator upsamples to DAVIS resolution.

Results. The proposed PhysGraph has the highest dynamic-region recall (dPSNR@15) at every stride (Table 2) – 83.5% at s=3, 15 points above the next best, and 23.8% at s=25 – with two practical advantages. *Robustness*: PhysGraph and GaussianPrediction never fail, while MoSca fails on 11–69% and Shape of Motion on 17–29% of sequences, increasingly at larger strides as per-scene optimization becomes ill-conditioned under the same upstream noise. *Efficiency*: 1.4–3.1 minutes per sequence, $16-26\times$ faster than MoSca, $7-16\times$ faster than GaussianPrediction, and $1.1-2.4\times$ faster than Shape of Motion; only the feed-forward MoVieS is faster still, at 0.1–0.6 minutes. MoSca's higher @20 scores at $s\leq5$ come only from the sequences on which its optimization converges; at s=25 it fails on 69% of sequences and reaches 0% at PSNR@20. MoVieS never fails and is the fastest method, but its dynamic-region recall is the lowest of the four at every stride (41.1 vs. 83.5 at s=3, 15.6 vs. 23.8 at s=25): a single forward pass conditioned on noisy cameras reconstructs the static scene well while the moving object becomes blurry.

4.3 Extrapolation

Because each branch yields an explicit motion model, PhysGraph extrapolates by evaluating it at $\alpha>1$. In Table 3, PhysGraph leads dynamic-region recall over GaussianPrediction, the only competitor without failures, at every horizon (dPSNR@15 36.6 vs. 30.0 at s=3; 29.3 vs. 23.6 at s=5; 25.6 vs. 19.7 at s=10) at $3\times$ lower runtime; MoSca fails on 31–62% of sequences. Unlike interpolation, extrapolation drifts with horizon: recall decays with stride for all methods.

Table 2: **Temporal interpolation on all 90 DAVIS sequences**: PhysGraph vs. three per-scene-optimized dynamic 3DGS methods and the feed-forward MoVieS. Recall: % of sequences with PSNR ≥ 15 / 20 dB, full-frame and on dynamic regions (dPSNR). Fail %: runs that crash or return no output; they count as not reaching any threshold. Runtime is mean per-sequence time on one RTX 4090. **Bold**: best per stride.

Stride	Method	PSNR@15 ↑	@20 ↑	dPSNR@15 ↑	@20 ↑	Fail % ↓	t (min) ↓
3	MoSca (Lei et al., 2025)	85.7	**60.0**	62.9	20.0	11%	52.2
	Shape of Motion (Wang et al., 2025)	78.9	44.4	64.4	11.1	17%	7.0
	GaussianPred. (Zhao et al., 2024)	94.3	48.6	68.6	20.0	**0%**	21.5
	MoVieS (Lin et al., 2026)	86.7	22.2	41.1	7.8	**0%**	**0.6**
	PhysGraph	**97.5**	54.4	**83.5**	**27.8**	**0%**	2.9
5	MoSca (Lei et al., 2025)	85.7	**48.6**	57.1	17.1	11%	48.5
	Shape of Motion (Wang et al., 2025)	71.1	30.0	50.0	4.4	17%	4.5
	GaussianPred. (Zhao et al., 2024)	91.4	28.6	54.3	14.3	**0%**	21.5
	MoVieS (Lin et al., 2026)	84.4	20.0	36.7	6.7	**0%**	**0.4**
	PhysGraph	**92.6**	40.7	**63.0**	**17.3**	**0%**	3.1
10	MoSca (Lei et al., 2025)	60.0	22.9	34.3	**8.6**	37%	46.2
	Shape of Motion (Wang et al., 2025)	61.1	14.4	34.4	3.3	18%	2.7
	GaussianPred. (Zhao et al., 2024)	65.7	17.1	42.9	5.7	**0%**	22.1
	MoVieS (Lin et al., 2026)	76.7	21.1	27.8	5.6	**0%**	**0.2**
	PhysGraph	**81.5**	**25.9**	**43.2**	6.2	**0%**	2.0
25	MoSca (Lei et al., 2025)	28.6	0.0	11.4	0.0	69%	36.1
	Shape of Motion (Wang et al., 2025)	34.4	4.4	13.3	0.0	29%	1.6
	GaussianPred. (Zhao et al., 2024)	50.0	**14.7**	19.0	2.9	**0%**	22.2
	MoVieS (Lin et al., 2026)	62.2	13.3	15.6	2.2	**0%**	**0.1**
	PhysGraph	**62.5**	10.0	**23.8**	**5.0**	**0%**	1.4

Table 3: **Extrapolation on all 90 DAVIS sequences**: given five keyframes at stride s, predict the next s frames. Recall (%) at 15/20 dB, full-frame (PSNR) and dynamic-region (dPSNR), failure rate, and mean per-sequence time. Shape of Motion does not support extrapolation, and MoVieS reconstructs only its input window: it accepts output timesteps beyond that window but scores below simply repeating the last keyframe. **Bold**: best per stride.

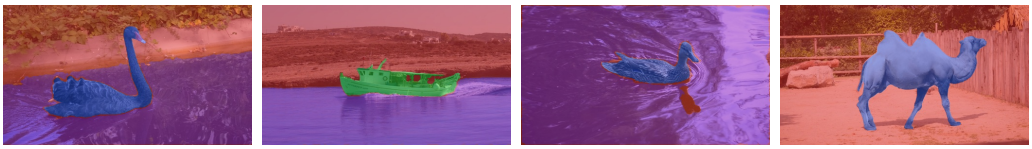
Stride	Method	PSNR@15 ↑	@20 ↑	dPSNR@15 ↑	@20 ↑	Fail % ↓	t (min) ↓
3	MoSca (Lei et al., 2025)	55.6	18.9	30.0	10.0	31%	15.9
	GaussianPred. (Zhao et al., 2024)	84.4	28.9	30.0	5.6	**0%**	4.5
	PhysGraph	**85.4**	**30.5**	**36.6**	**11.0**	**0%**	**1.6**
5	MoSca (Lei et al., 2025)	42.7	15.7	20.2	3.4	47%	16.7
	GaussianPred. (Zhao et al., 2024)	**76.4**	25.8	23.6	**5.6**	**0%**	4.6
	PhysGraph	73.2	**26.8**	**29.3**	4.9	**0%**	**1.6**
10	MoSca (Lei et al., 2025)	28.2	7.0	9.9	1.4	62%	14.2
	GaussianPred. (Zhao et al., 2024)	**62.0**	**15.5**	19.7	1.4	**0%**	4.8
	PhysGraph	61.0	13.4	**25.6**	**3.3**	**0%**	**1.8**

4.4 Ablation Studies

How often do the fallbacks fire, and do they carry the result? On a 10-sequence DAVIS subset, rigid fits essentially never fall back and the liquid branch reverts on 9% / 0% of segment-intervals (s=5 / 25); the deformable fallback covers only 0.4% of the dynamic area at s=5 and 8.1% at s=25, and most of it is segments with too few correspondences for *any* model rather than rejected TPS solves. Disabling the diagnostic gates changes PSNR by -0.01 dB on average (worst -0.08 dB; full study in Section E.4): the fallback is a safety net, not the source of the robustness.

Table 4: **Deformable motion model ablation** on three DAVIS sequences (blackswan, camel, breakdance), averaged over the development strides $s \in \{2, 10, 25\}$. TPS, Laplacian, ARAP, and Neural are within 0.4 dB of one another; TPS has the best PSNR, SSIM, and IoU at a runtime within 2% of the fastest. CE = centroid error; ms/frame = motion fitting and 3D point synthesis per synthesized frame, excluding the perception front-end and rendering.

Method	CE (px)↓	IoU↑	PSNR (dB)↑	SSIM↑	ms/frame↓
TPS (Duchon, 1977)	15.5	**0.287**	**17.99**	**0.457**	**2670**
Laplacian (Sorkine et al., 2004)	16.0	0.283	17.94	0.455	2625
ARAP (Sorkine & Alexa, 2007)	**15.0**	0.283	17.77	0.447	2707
Neural (Park et al., 2021a)	16.1	0.275	17.60	0.444	3904
CPD (Myronenko & Song, 2010)	59.8	0.199	15.49	0.375	2823



blackswan: St+D+L *boat*: St+So+L *mallard-water*: D+L *camel*: St+D

Figure 4: **Automatic material phase classification** (Section 3.3). Segments are classified static (St), solid/rigid (So), deformable (D), or liquid (L). Water is identified as liquid, animals as deformable, boats as rigid solids, and background as static, which selects each object's motion model.

Table 5: **Zero-shot material phase classification** on the MINC-2500 texture benchmark (250 stratified samples from 21 categories). Each MINC category is mapped to our three-phase taxonomy. We compare SAM 3 (text-prompted segmentation with 37 material queries) against five vision-language models, all without fine-tuning.

Method	Overall (%)	Solid (%)	Deform. (%)	Liquid (%)	s/img
SAM 3 (text-prompted)	**78.0**	85.6	67.6	**84.6**	7.6
Llama 3.2-Vision (11B)	73.2	88.6	57.1	46.2	2.2
Qwen3-VL (8B)	69.6	69.7	**72.4**	46.2	3.4
Qwen2.5-VL (7B)	68.4	90.2	43.8	46.2	1.5
LLaVA-OneVision (7B)	65.6	94.7	33.3	30.8	1.6
LLaVA-1.6 (7B)	60.4	**99.2**	17.1	15.4	2.3

Deformable motion model and runtime. TPS-on-residual, Laplacian, and ARAP are within 0.22 dB of one another under an otherwise identical pipeline (ΔPSNR $\leq$ 0.22 dB, ΔCE $\leq$ 1 px); a per-segment neural displacement field (Park et al., 2021a) matches their accuracy but is 46% slower, and CPD (Myronenko & Song, 2010) performs markedly worse (Table 4; details in Section E.1). We use TPS, which has the best PSNR at a runtime within 2% of the fastest. Total runtime drops from 2.9 min (s=3) to 1.4 min (s=25), dominated by SAM 3 and motion fitting.

Material phase classification. Table 5 compares SAM 3 against five vision-language models on a stratified MINC-2500 (Bell et al., 2015) test set of 21 texture categories, using a separate 37-noun material vocabulary that covers MINC's categories (Section A). SAM 3 achieves the highest overall accuracy (78.0%) with the most balanced per-phase scores; Llama-3.2-Vision is a 3$\times$ faster alternative at 73.2% accuracy. Phase classification results on the DAVIS dataset are shown in Fig. 4.

5 Conclusion

PhysGraph uses a foundation-model-predicted material phase to select one of three classical motion solvers (SE(3) Procrustes, TPS-on-residual, Sinkhorn-OT) on a shared per-object kNN graph, composed with bidirectional Gaussian splatting. It achieves zero failures on 90 DAVIS sequences at up to 26$\times$ less compute than per-scene optimizers, 2.5$\times$ lower depth error than linear 3D interpolation, the highest DAVIS dynamic-region recall at every stride, and extrapolation with explicit geometry. The feed-forward MoVieS runs faster still but has lower dynamic-region recall at every stride. The code and synthetic scenes will be made public.

Limitations. PhysGraph's operating range is well characterized (Section C.1). *Articulation*: an articulated body (breakdance) is returned as a single SAM 3 segment, so one TPS field absorbs the relative part motion; part-aware segmentation or a motion-aware router is a natural extension. *Fast non-rigid motion over large gaps* (horsejump-high, s=25) reaches the limit of the TPS regime. *Phase misclassification* degrades gracefully: the TPS residual vanishes for near-rigid motion, so the rigid component still carries the object, and materials outside the prompt list default to the rigid model. *Scope*: TPS does not model plasticity, fracture, or anisotropy; surface-point Sinkhorn does not enforce incompressibility; and motion is solved per keyframe pair rather than globally. Solving these more extreme cases remains a future direction.

AI use statement

A vision–language model (SAM 3 in the headline configuration; Llama-3.2-Vision and Qwen-VL as benchmarked alternatives) is a component of the proposed method: it predicts each segment's material phase and label from a single video frame and, then, continuous physical parameters (Young's modulus, viscosity) via an LLM call, selecting and parameterizing the motion solver. This use is

described in Section 3.3 and quantified in Sections B and D. In addition, we used generative AI assistants for LaTeX editing and formatting, and for assistance in writing experiment scripts and analysis code; all AI-assisted code was reviewed, tested, and verified by the authors, and all AI-assisted text was checked by the authors for correctness. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics Statement

This work concerns temporal interpolation and extrapolation of dynamic scenes from monocular video. All experiments use publicly released datasets (DAVIS-2017, MINC-2500, our own synthetic Blender benchmark) and publicly released model checkpoints (MonST3R, SAM 3, CoTracker, RAFT, GMFlow, gsplat, Llama-3.2-Vision, Qwen-VL) under their respective licenses; no human subjects or personally identifiable information are involved beyond what the public DAVIS videos contain. The method's outputs are synthesized intermediate or extrapolated frames whose fidelity is comparable to or below existing video-synthesis tools, so we see no incremental societal risk beyond that of existing video-synthesis methods.

Reproducibility Statement

All motion models are specified by closed-form equations in Section 3 (Equations (1) to (5)). Every pipeline module and hyperparameter of the headline configuration is listed in the implementation paragraph of Section 4 and in Section A, including the fixed TPS regularizer and the stiffness-adaptive alternative. The evaluation protocol (keyframe strides, keyframe-only input for every baseline, recall metrics, failure accounting) is given in Section 4.2 and Section A; the synthetic benchmark (six Blender scenes with per-frame depth, masks, poses, and vertex trajectories) is described in Section A. The pipeline performs no training. We will release the full pipeline code, the configuration files for every experiment, the Blender source files and rendering scripts of the synthetic benchmark, and the wrappers used to run all baselines.

References

P. G. Agache, C. Monneur, J. L. Leveque, and J. De Rigal. Mechanical properties and Young's modulus of human skin in vivo. *Archives of Dermatological Research*, 269(3):221–232, 1980.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Daniel Barath and Jiri Matas. Graph-cut RANSAC. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Daniel Barath, Luca Cavalli, and Marc Pollefeys. Learning to find good models in RANSAC. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations (ICLR)*, 2025.

Richard H. C. Bonser and Peter P. Purslow. The Young's modulus of feather keratin. *Journal of Experimental Biology*, 198(4):1029–1033, 1995.

Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Ang Cao and Justin Johnson. HexPlane: A fast representation for dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. PhysBench: Benchmarking and enhancing vision-language models for physical world understanding. In *International Conference on Learning Representations (ICLR)*, 2025.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

Braja M. Das and Khaled Sobhan. *Principles of Geotechnical Engineering*. Cengage Learning, 9th edition, 2018.

Carl Doersch, Yi Yang, Mel Večerík, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Jean Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pp. 85–100. Springer, 1977.

Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust dense feature matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Forest Products Laboratory. *Wood Handbook: Wood as an Engineering Material*. Number FPL-GTR-282. U.S. Department of Agriculture, Forest Service, 2021.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit radiance fields in space, time, and appearance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Stuart Geman and Donald E. McClure. Bayesian image analysis: An application to single photon emission tomography. In *Proceedings of the American Statistical Association, Statistical Computing Section*, pp. 12–18, 1985.

Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. SC-GS: Sparse-controlled Gaussian splatting for editable dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Junhwa Hur, Charles Herrmann, Songyou Peng, Philipp Henzler, Zeyu Ma, Todd Zickler, and Deqing Sun. UFO-4D: Unposed feedforward 4D reconstruction from two images. In *International Conference on Learning Representations (ICLR)*, 2026.

Hamed Joodaki and Matthew B. Panzer. Skin mechanical properties and modeling: A review. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 232(4):323–343, 2018.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. DynMF: Neural motion factorization for real-time dynamic view synthesis with 3D Gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

Jiahui Lei, Yijia Weng, Adam W. Harley, Leonidas J. Guibas, and Kostas Daniilidis. MoSca: Dynamic Gaussian fusion from casual videos via 4D motion scaffolds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision (ECCV)*, 2024.

Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. PAC-NeRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *International Conference on Learning Representations (ICLR)*, 2023a.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations (ICLR)*, 2019.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. DyniBaR: Neural dynamic image-based rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b.

Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. MoVieS: Motion-aware 4D dynamic view synthesis in one second. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong Mu. OmniPhysGS: 3D constitutive Gaussians for general physics-based dynamics generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Fangfu Liu, Hanyang Wang, Shunyu Yao, Shengjun Zhang, Jie Zhou, and Yueqi Duan. Physics3D: Learning physical properties of 3D Gaussians via video diffusion. *arXiv preprint arXiv:2406.04338*, 2024.

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)*, 2024.

Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics (SIGGRAPH)*, 33(4), 2014.

Robert J. McCann. A convexity principle for interacting gases. *Advances in Mathematics*, 128(1): 153–179, 1997.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, 2003.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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)*, 2021a.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 40 (6), 2021b.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

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

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)*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2022.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning (ICML)*, 2020.

Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Eurographics Symposium on Geometry Processing (SGP)*, 2007.

Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and Hans-Peter Seidel. Laplacian surface editing. In *Eurographics Symposium on Geometry Processing (SGP)*, pp. 175–184, 2004.

Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):80, 2007.

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

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4D reconstruction from a single video. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D Gaussian splatting for real-time dynamic scene rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-integrated 3D Gaussians for generative dynamics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning optical flow via global matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958, 2023.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4DGT: Learning a 4D Gaussian transformer using real-world monocular videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

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)*, 2024b.

Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3D: Towards zero-shot metric 3D prediction from a single image. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations (ICLR)*, 2025.

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. PhysDreamer: Physics-based interaction with 3D objects via video generation. In *European Conference on Computer Vision (ECCV)*, 2024.

Boming Zhao, Yuan Li, Ziyu Sun, Lin Zeng, Yujun Shen, Rui Ma, Yinda Zhang, Hujun Bao, and Zhaopeng Cui. GaussianPrediction: Dynamic 3D Gaussian prediction for motion extrapolation and free view synthesis. In *ACM SIGGRAPH*, 2024.

Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation of elastic objects with spring-mass 3D Gaussians. In *European Conference on Computer Vision (ECCV)*, 2024.

A Implementation Details and Datasets

Pipeline modules. PhysGraph composes off-the-shelf foundation models with classical motion solvers. The headline DAVIS configuration uses the following components.

Depth and pose: MonST3R (Zhang et al., 2025) provides per-frame dense point clouds, camera extrinsics/intrinsics, and a per-pixel log-confidence map for every keyframe, in the approximately metric scale of its global alignment; no rescaling is applied, and every metric threshold below is expressed in this unit. Camera poses of synthesized frames use Slerp for rotation and linear interpolation for translation and intrinsics; the headline configuration applies no camera refinement.

Dense correspondence: CoTracker (Karaev et al., 2024) with an 80×80 query grid, tracking each keyframe interval natively. The sparse tracks are densified to a per-pixel warp at MonST3R resolution by piecewise-linear (Delaunay) interpolation of the visible tracks' displacements inside their convex hull and nearest-track fill outside it; a pair with fewer than 10 visible tracks yields no correspondences. (Synthetic experiments use ground-truth flow.) The flow ablation in Section E.2 compares CoTracker against RAFT (Teed & Deng, 2020), GMFlow (Xu et al., 2022), and UniMatch (Xu et al., 2023).

Segmentation & materials: SAM 3 (Carion et al., 2025) text-prompted with our fixed list of 59 concrete nouns (Table 7), propagated across keyframes by the model's video predictor; the per-segment material is decided by majority vote. The phase is a fixed noun-to-phase lookup over the prompt list; background covered by no prompt forms its own static segment.

Physical parameters: a vision–language model receives the frame with the segment highlighted (Section A) and returns a stiffness score and a viscosity score (1–10), mapped to Young's modulus E and dynamic viscosity μ as described in Section D. In the headline configuration μ parameterizes the Beta-CDF easing of the liquid branch, and the TPS regularizer is fixed at $\lambda_{\text{tps}}{=}10^{-4}$; the stiffness-adaptive variant is evaluated in Section E.3.

Motion analysis: a pixel is dynamic if its ego-motion residual exceeds $\tau_r{=}3$ px at MonST3R resolution or its log-confidence is below $\tau_c{=}0.3$; a segment is dynamic if more than half of its valid pixels are, and over the sequence if more than half of its intervals are; dynamic segments are dilated by 3 px and the background segment is static. A segment with fewer than 50 correspondences after filtering falls back to linear motion.

Rigid solver: Geman-McClure objective with graduated non-convexity, initialized by SVD-Procrustes and minimized with Adam over axis-angle and translation (learning rate 0.01, 500 iterations, $\sigma{=}0.05$ m annealed exponentially from $\max(\text{median displacement},\ 10\sigma)$); inliers are correspondences with residual below med $+\ 3{\cdot}1.4826$ MAD; fallback to linear at inlier ratio $<30\%$ or median inlier residual >0.1 m.

Elastic solver: TPS with up to $N_c{=}100$ farthest-point control points, polyharmonic kernel $\phi(r){=}r$, Tikhonov regularization $\lambda_{\text{tps}}{=}10^{-4}$; rigid-subtracted displacements above med $+$ $\max(3{\cdot}1.4826\,\text{MAD},\ 0.5\,\text{med})$ are discarded before the solve; the kNN graph uses $k{=}8$ neighbors with bandwidth h equal to $1.5\times$ the distance to the 8th neighbor; a solve whose maximum predicted displacement exceeds $5\times$ the median is rejected (oscillation guard) and the segment reverts to linear motion.

Liquid solver: entropic OT on $N_{\text{OT}}{=}2500$ (synthetic) / 2000 (DAVIS) randomly subsampled points with uniform marginals, Sinkhorn $\varepsilon{=}10^{-2}\times$ the median transport cost, 100 iterations; displacements are propagated to all points by inverse-distance weights over the $k{=}8$ nearest subsampled points, and per-point displacements above med $+\ 3{\cdot}1.4826$ MAD are zeroed; a segment with fewer than 50 liquid points or a failed solve reverts to linear blending; viscosity-aware Beta-CDF easing.

Renderer: 3D Gaussian splatting via gsplat (Kerbl et al., 2023) with per-pixel isotropic Gaussians of scale $\sigma{=}|z_{\text{cam}}|/f_{\text{avg}}\cdot 1.5$ on DAVIS ($\cdot 1.0$ on the synthetic scenes), bidirectional fusion with voxel-based deduplication at a voxel size equal to the median Gaussian scale of the frame. **Hardware:** one NVIDIA RTX 4090 (24 GB) for all timed experiments; per-sequence runtime is 1.4–2.9 minutes (Table 10). The pipeline involves *no per-scene optimization or training*.

Datasets. We evaluate on (1) a synthetic benchmark of six Blender scenes and (2) the 90 real-world sequences of DAVIS 2017 (Pont-Tuset et al., 2017).

Table 6: **Fallback decomposition for deformable segments** on a 10-sequence DAVIS subset (excluding the articulated breakdance; see text for the totals including it). Each cell: % of deformable segment-intervals / % of deformable dynamic-region area.

Fallback reason	s=5	s=25
TPS solve rejected (oscillation guard)	5.5% / 0.21%	5.3% / 6.42%
Rigid pre-fit rejected	0.8% / 0.04%	1.3% / 1.55%
Too few correspondences for *any* model	19.5% / 0.13%	30.7% / 0.14%
Total	25.8% / 0.4%	37.3% / 8.1%

Synthetic benchmark. Six scenes (Fig. 2) span the three material phases targeted by the framework: liquids (*Water Cup*, *Honey*), deformables (*Cloth on Sphere*, *Elastic Balls*), and rigid bodies (*Bouncing Ball*, *Tumbling Cubes*). Each scene is rendered for 200 frames at 854×480 resolution with full ground-truth annotations: per-frame depth maps, camera poses, instance masks, and per-vertex 3D trajectories. Synthetic evaluation uses these GT annotations directly so the comparison isolates the motion model from the perception backbone.

DAVIS-2017. All 90 sequences are evaluated end-to-end with no GT supervision: MonST3R supplies depth and poses, CoTracker supplies correspondences, and SAM 3 supplies segments and material phases. Resolution is the dataset's native variable resolution.

Evaluation protocol. Keyframes are fixed by the protocol, not detected: for every method, every s-th frame is a keyframe ($s \in \{3, 5, 10, 25\}$ on DAVIS) and the in-between frames are synthesized. Every method receives the same keyframe-only input: MoSca, Shape of Motion, and GaussianPrediction (which nominally consume all frames) are run on the keyframe-only set; GaussianPrediction (an extrapolator by design) is fed bidirectional context and queried at intermediate timestamps. Because optimization-based methods can fail catastrophically (no output, degenerate geometry), we report recall (% sequences achieving PSNR $\geq 15/20$ dB, full-frame and dynamic-region) plus failure rate. Dynamic-region metrics use the DAVIS ground-truth masks.

Noun prompts. Table 7 lists the noun prompts. DAVIS discovery uses 59 object and surface nouns, each mapped to one of four phases; background that no prompt covers forms its own static segment. The MINC-2500 evaluation of Table 5 uses a separate 37-noun material vocabulary that covers MINC's categories.

Table 7: Noun prompts and their phases. Top: the 59 nouns used for SAM 3 discovery on DAVIS. Bottom: the separate 37-noun material vocabulary used for the MINC-2500 evaluation.

DAVIS discovery (59 nouns)	
Liquid (11)	water, liquid, river, ocean, sea, pool, puddle, stream, waterfall, fountain, wave
Deformable (21)	creature, animal, person, human, bird, dog, cat, horse, cow, fish, bear, deer, monkey, elephant, camel, goat, sheep, cloth, fabric, flag, rope
Solid (12)	vehicle, car, truck, bus, motorcycle, bicycle, boat, skateboard, scooter, wheelchair, stroller, ball
Static (15)	ground, floor, road, pavement, sidewalk, wall, building, house, fence, bridge, rock, mountain, hill, sky, ceiling
MINC-2500 evaluation (37 nouns)	
Solid (15)	brick, ceramic, glass, metal, mirror, painted surface, plastic, polished stone, marble, granite, stone, concrete, tile, wood, ice
Deformable (17)	carpet, fabric, cloth, foliage, leaves, plant, grass, food, bread, fruit, hair, fur, leather, paper, skin, wallpaper, rubber
Liquid (5)	water, liquid, river, pool, ocean

VLM prompt. on Each vision–language backend receives the full frame with the segment's interior marked by red dots, its boundary by a green contour, and everything outside darkened, together with the prompt below (sampling temperature 0.3, at most 500 tokens). Stiffness and viscosity scores $s, v \in \{1, \ldots, 10\}$ map to $E = 10^{4+0.7s}$ Pa and $\mu = 10^{-3+0.5v}$ Pa s.

```
Look at this image.  One region is highlighted:  it has RED DOTS
on its interior, a GREEN CONTOUR along its boundary, and the area
OUTSIDE is darkened.  You must ONLY classify the material INSIDE the
highlighted region (where the red dots are).  Ignore everything in
the darkened area.
```

Table 8: **Effect of temporal stride** on synthetic interpolation, averaged over 6 scenes. Larger strides mean sparser keyframes and harder interpolation. **Bold**: best per stride.

	$s = 3$		$s = 5$		$s = 10$		$s = 25$	
Method	IoU	PSNR	IoU	PSNR	IoU	PSNR	IoU	PSNR
Linear	0.957	19.7	0.885	22.9	0.805	18.8	0.649	16.0
Rigid body	0.936	19.4	0.851	22.4	0.769	17.7	0.624	15.2
Spline (cubic)	0.616	18.5	0.589	18.1	0.560	17.5	0.489	15.2
Spline (CR)	0.616	18.5	0.590	18.1	0.562	17.6	0.490	15.2
PhysGraph	**0.966**	**23.3**	**0.903**	**23.5**	**0.827**	**22.6**	**0.676**	**20.9**

```
Step 1 -- Describe what you see at the red dot locations (NOT the
darkened background).  What object or surface are the red dots on?


Step 2 -- Classify that material's physical phase:
- SOLID: Rigid objects that do not deform (metal, plastic, wood,
stone, glass, ice, concrete, buildings, vehicles, ground, roads,
pavement, walls, poles).
- DEFORMABLE: Flexible objects that bend, stretch, or compress
(cloth, fabric, skin, rubber, foam, leaves, trees, bushes,
vegetation, fur, feathers, hair, paper, rope, any animal, any bird,
human body).
- LIQUID: Water surfaces (rivers, pools, lakes, ponds), flowing
liquid, or any fluid with a visible free surface.  If the red dots
are on a water surface, this IS liquid.


NOTE: Things that are near water but not water themselves are NOT
liquid:
- Vegetation next to a river → DEFORMABLE
- Concrete/stone wall next to water → SOLID
- An animal or bird swimming on water → DEFORMABLE
- Reflections, glass, ice, shadows → NOT liquid


Format your response EXACTLY as (nothing else):
MATERIAL: [material name]
PHASE: [SOLID|DEFORMABLE|LIQUID]
PHASE_REASON: [one sentence -- what are the red dots on?]
DENSITY: [number] kg/m3
STIFFNESS: [1-10] (for SOLID/DEFORMABLE, otherwise N/A)
VISCOSITY: [1-10] (for LIQUID only, otherwise N/A)
CONFIDENCE: [0.0-1.0]
```

B Additional Quantitative Results

This section gives the per-stride and per-component breakdowns underlying Section 4.

Synthetic interpolation across all strides. Table 8 reports IoU and PSNR for each baseline at $s \in \{3, 5, 10, 25\}$, averaged over the six synthetic scenes. PhysGraph's lead over Linear widens monotonically with the temporal gap: at s=5 the PSNR margin is $+0.62$ dB; at s=25 it is $+4.9$ dB. Spline baselines collapse already at s=3 (IoU 0.62 vs. 0.97), reflecting the multiplicative noise amplification of fitting per-pixel 3D splines through several keyframes.

Runtime breakdown. Table 10 reports per-component timings averaged over all 90 DAVIS sequences. SAM 3 segmentation and the motion fitting together dominate at dense strides ($\approx 60\%$ of runtime at s=3); rendering time is roughly stride-independent. Total per-sequence runtime drops from 2.9 min (s=3) to 1.4 min (s=25), up to 26× faster than MoSca and GaussianPrediction and 1.1–2.4× faster than Shape of Motion.



Figure 5: **Failure cases** at s=25. Left to right: *breakdance* (articulation absorbed by a single TPS field, distorted limbs), *horsejump-high* (fast large-gap non-rigid motion, ghosting), *train* (rigid object misclassified as deformable; graceful degradation because the TPS residual vanishes for near-rigid motion).

C Qualitative Results

C.1 Failure Cases

Figure 5 shows three representative failure cases at s=25, one per regime named in Section 5: (i) *breakdance* – the dancer's articulated body is returned as one segment and absorbed by a single TPS field, producing distorted limbs mid-interval; (ii) *horsejump-high* – fast non-rigid motion over a 25-frame gap exceeds the TPS regime and produces ghosting (dPSNR 9.3 dB); (iii) *train* – a phase misclassification (the rigid train is labeled deformable) whose output nonetheless degrades gracefully because the TPS residual vanishes for near-rigid motion and leaves the rigid component, illustrating both a dispatch error and the system's graceful-failure property.

C.2 Phase Segmentation on DAVIS

Figure 6 shows qualitative results of our automatic phase segmentation on 20 diverse DAVIS sequences. Each pixel is classified into one of four physical phases: **static** (background), **solid** (rigid objects), **deformable** (non-rigid objects), and **liquid**. These phase labels determine which interpolation model is applied to each segment: ego-motion for static regions, SE(3) rigid body for solids, TPS deformation for deformable objects, and optimal transport for liquids. The segmentation is robust across a wide range of scene types – animals, people, vehicles, and sports – correctly identifying deformable regions (*e.g.* animal bodies, human limbs) and rigid objects (*e.g.* cars, surfboards) even in cluttered scenes with multiple overlapping objects.

C.3 VLM Backend Comparison for Phase Classification

Figure 7 compares the phase segmentation produced by SAM3 (text-prompted) and five Vision-Language Models (VLMs) applied to SAM2 instance masks across 10 diverse DAVIS sequences. SAM3 performs joint segmentation and classification via text prompts, producing its own masks, while the VLM columns all share the same SAM2 instance masks and differ only in phase labels. Qwen2.5-VL-7B provides the most consistent VLM results, correctly identifying deformable animals and rigid vehicles across all scenes. Smaller models (*e.g.* Qwen2.5-VL-3B) occasionally misclassify materials – for instance, labelling water as deformable rather than liquid, or marking animal bodies as solid. LLaVA-OneVision tends to over-predict the solid class, while LLaMA-3.2-Vision

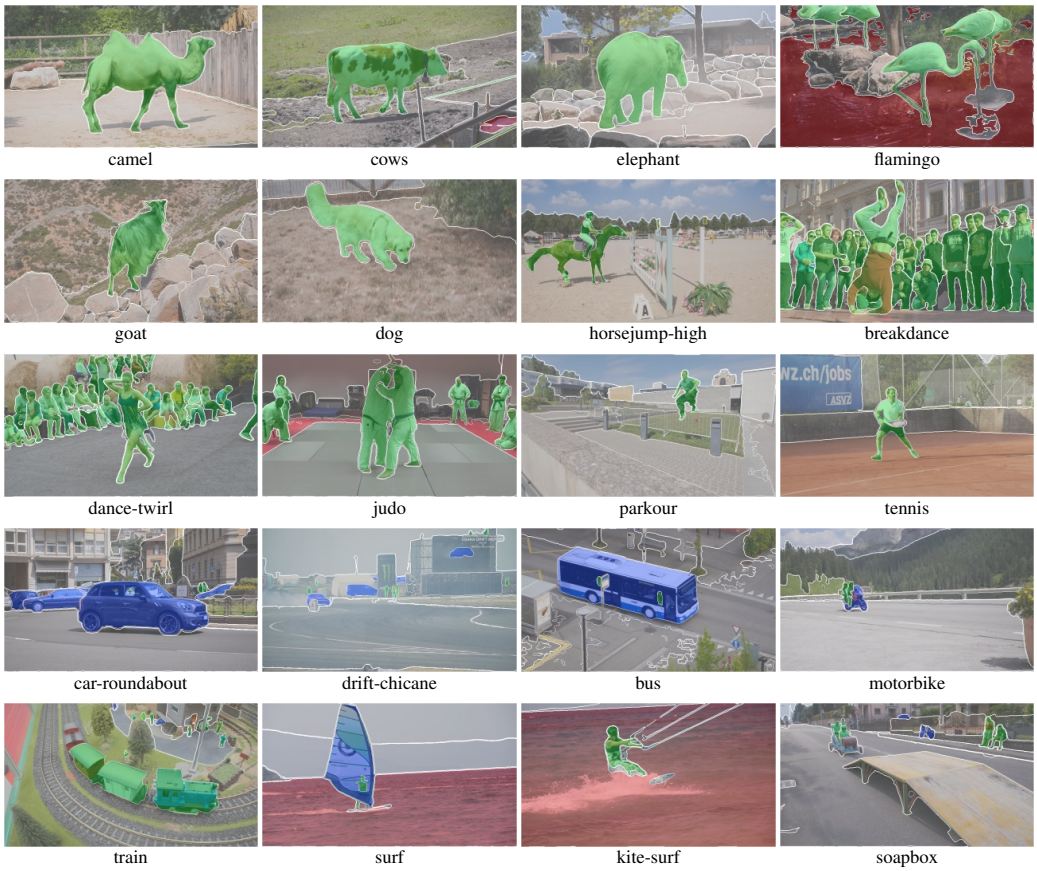


Figure 6: Automatic phase segmentation on 20 DAVIS sequences. Colors denote physical phase: **gray** = static, **blue** = solid (rigid), **green** = deformable, **red** = liquid. The pipeline distinguishes rigid vehicles (*e.g.* bus, train) from deformable bodies (*e.g.* animals, people).

shows reasonable accuracy but occasionally confuses background elements. These differences motivate our default choice of SAM3 (with Qwen2.5-VL-7B as fallback VLM) for the pipeline.

D Accuracy of VLM-Estimated Physical Parameters

Table 9 compares the physical parameters predicted by each VLM backend against reference values for eight material–scene pairs across four DAVIS sequences. The VLM receives the frame with each segment highlighted and returns a stiffness score (1–10) and viscosity score (1–10), which are mapped to Young's modulus via $E = 10^{4+0.7s}$ Pa and viscosity via $\mu = 10^{-3+0.5s}$ Pa s, as well as a direct density estimate in kg/m^3.

Reference values. Water density and viscosity are standard textbook values at 20°C. The feather keratin modulus (2.5 GPa) follows Bonser & Purslow (1995), who report a mean Young's modulus of 2.50 GPa from tensile tests on compact rachis keratin of eight bird species; the density (1300 kg/m^3) is a standard handbook value for β-keratin rather than a measurement from that study. Mammalian skin/muscle composite modulus ($\approx$1 MPa) is from Joodaki & Panzer (2018) (camel) and Agache et al. (1980) (horse), with tissue density $\approx$1050 kg/m^3 from standard physiology references. Wood (softwood fence, $E\approx$10 GPa, $\rho\approx$550 kg/m^3) follows the USDA Wood Handbook (Forest Products Laboratory, 2021). Sand modulus ($\approx$50 MPa, $\rho\approx$1600 kg/m^3) is from typical geotechnical data for dry sand (Das & Sobhan, 2018).

Density. All VLMs predict water density accurately (ρ=1000 vs. reference 998 kg/m^3). For solid materials, Qwen2.5-VL-7B consistently returns plausible values (sand: 1600, wood: 650 kg/m^3), while smaller models occasionally confuse units – LLaVA-OneVision reports sand density as 1.4 kg/m^3 (likely g/cm^3) and LLaMA-3.2 reports feather density as 1.1 kg/m^3. Animal body den-

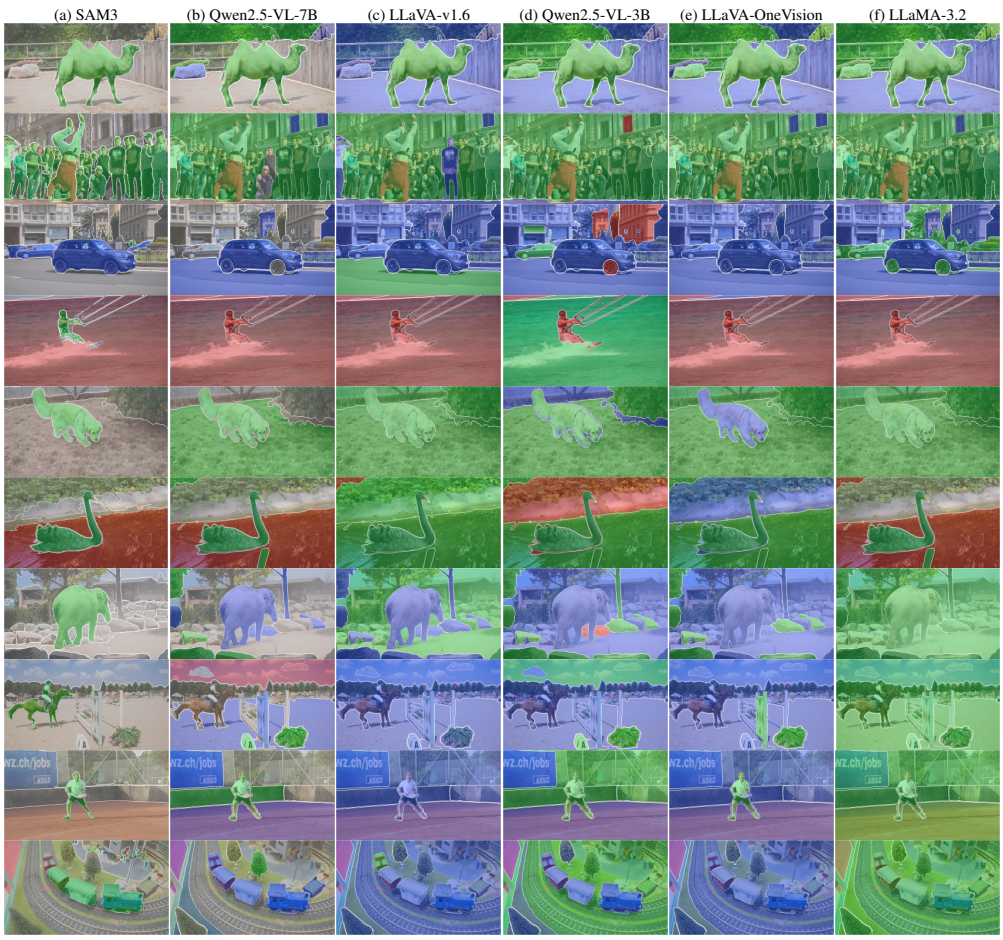


Figure 7: Phase segmentation comparison across SAM3 and five VLM backends on 10 DAVIS sequences. (a) SAM3 uses text-prompted segmentation with its own masks. (b)–(f) SAM2 instance masks classified by different VLMs. Colors denote physical phase: **gray** = static, **blue** = solid, **green** = deformable, **red** = liquid. Rows top to bottom: camel, breakdance, car-roundabout, kite-surf, dog, blackswan, elephant, horsejump-high, tennis, train.

Table 9: VLM-predicted physical parameters *vs.* reference values for eight material–scene pairs across four DAVIS sequences. (a) Density ρ in kg/m^3. (b) Young's modulus E. "–" indicates the VLM failed to identify any segment of that material. Column abbreviations: kite-s. = kite-surf, b.swan = blackswan, h.jump = horsejump-high. Reference sources: water (standard tables), feather keratin (Bonser & Purslow, 1995), mammalian tissue (Joodaki & Panzer, 2018; Agache et al., 1980), wood (Forest Products Laboratory, 2021), sand (Das & Sobhan, 2018).

(a) Density ρ (kg/m^3)

	Water		Feathers	Tissue / Body / Wood			Sand	
Method	kite-s.	b.swan	b.swan	camel	wood	horse	camel	h.jump
Reference	998	998	1300	1050	550	1050	1600	1600
Qwen2.5-VL-7B	1000	1000	325	950	650	1000	1600	1600
LLaVA-v1.6	1000	1000	400	500	500	–	1500	1567
Qwen2.5-VL-3B	–	1000	850	929	930	500	–	2
LLaVA-OneVision	1000	–	750	1000	625	1000	1	1450
LLaMA-3.2	1000	1000	1	1000	900	1000	1600	1200

(b) Young's modulus E (water omitted – always 0)

	Feathers	Tissue / Body / Wood			Sand	
Method	b.swan	camel	wood	horse	camel	h.jump
Reference	2.5 GPa	1 MPa	10 GPa	1 MPa	50 MPa	50 MPa
Qwen2.5-VL-7B	400 MPa	6.3 MPa	2.4 GPa	32 MPa	32 MPa	32 MPa
LLaVA-v1.6	7 MPa	1 MPa	60 GPa	–	32 MPa	32 MPa
Qwen2.5-VL-3B	265 MPa	124 MPa	5.9 GPa	1 MPa	–	159 MPa
LLaVA-OneVision	16 MPa	27 MPa	27 GPa	32 MPa	32 MPa	32 MPa
LLaMA-3.2	1.3 MPa	16 MPa	822 MPa	32 MPa	50 kPa	50 kPa

sity is generally well-estimated at 950–1000 kg/m^3 (reference $\approx$1050), except LLaVA-v1.6 which returns 500 kg/m^3 for the camel.

Young's modulus. The stiffness-to-modulus mapping constrains E to 11 discrete values ($10^{4.7}$ to 10^{11} Pa), limiting fine-grained accuracy but providing order-of-magnitude estimates. Water is correctly assigned E=0 by all VLMs that identify it (Qwen2.5-VL-3B misclassifies ocean water as "Sand" in kite-surf). Wood estimates span 0.8–60 GPa across backends (reference $\approx$10 GPa), with Qwen2.5-VL-3B and LLaVA-OneVision closest. Swan feather keratin is consistently underestimated (1–400 MPa vs. reference 2.5 GPa), which is physically plausible for bulk feather deformation (as opposed to pure keratin). Sand modulus is reasonable at 32–159 MPa (reference $\approx$50 MPa), except LLaMA-3.2 which returns 50 kPa.

Phase classification. Phase errors have the largest impact, since they determine the interpolation model. LLaVA-v1.6 classifies the horse and rider as "solid" (horsejump-high), missing all deformable segments. Qwen2.5-VL-3B misidentifies ocean water as "Sand (deformable)" and LLaVA-OneVision labels lake water as "concrete (solid)" in blackswan. Qwen2.5-VL-7B is the most consistent across all scenes, motivating its use as our default VLM fallback.

E Ablation Studies

E.1 Deformable Motion Model

We compare five deformable motion models on three DAVIS sequences containing prominent deformable objects (blackswan, camel, breakdance), averaged over keyframe strides $s \in \{2, 10, 25\}$. All methods receive identical inputs (SAM3 segmentation, MonST3R geometry, correspondences); only the deformation model applied to dynamic-deformable segments differs.

Thin-Plate Splines (TPS) (Duchon, 1977) fit a smooth radial-basis-function warp through the correspondence control points, minimizing bending energy. We use the polyharmonic kernel $\phi(r) = r$ with Tikhonov regularization λ_{tps}=10^{-4}.

Table 10: **Runtime breakdown** of PhysGraph averaged over all 90 DAVIS sequences on a single NVIDIA RTX 4090. Component times are in seconds; total is in minutes.

Stride	#KF	MonST3R	Flow	SAM3	Physics	Rendering	**Total**
3	29	41.6	23.1	46.2	50.9	11.5	**2.9 min**
5	18	29.2	19.5	39.4	30.7	13.6	**2.2 min**
10	9	18.3	16.6	33.9	18.7	15.0	**1.7 min**
25	5	12.7	15.2	30.9	7.4	15.7	**1.4 min**

Laplacian mesh deformation (Sorkine et al., 2004) preserves local differential coordinates (the Laplacian) of a mesh built over the correspondence points, encouraging detail-preserving deformation while satisfying positional constraints.

ARAP (*As-Rigid-As-Possible*) (Sorkine & Alexa, 2007) seeks per-vertex rigid transformations that best explain the displacement field; the local rotation-preservation prior maintains shape rigidity but can under-fit large non-rigid deformations.

Neural displacement field fits a per-segment MLP (2 hidden layers of 64 units, ReLU) to predict $\Delta\mathbf{p}$ from 3D position, optimized with an ℓ_2 loss on correspondences plus a spatial smoothness penalty (weight 0.1; Adam, learning rate 10^{-3}, 300 full-batch steps), following the deformation-field paradigm of Park et al. (2021a).

CPD (Coherent Point Drift) (Myronenko & Song, 2010) formulates non-rigid registration as a probability density estimation problem, fitting a Gaussian mixture model to the target point set.

TPS achieves the best overall PSNR, SSIM, and IoU at a runtime within 2% of the fastest method (Table 4); the differences to Laplacian and ARAP are small (ΔPSNR ≤ 0.22 dB, ΔSSIM ≤ 0.01, ΔCE $\leq$ 1 px). ARAP has the lowest centroid error but slightly lower PSNR, as its rotation-preserving prior can under-fit large non-rigid deformations. The neural displacement field is competitive in accuracy but 46% slower due to per-segment optimization. CPD performs significantly worse across all metrics, particularly on scenes with large deformations (camel: 85.1 px centroid error *vs.* 7.6 px for TPS); this likely reflects insufficient per-segment tuning of its annealed Gaussian variance rather than the method's intrinsic capability.

E.2 Optical Flow Method

We compare four optical flow methods on the same three DAVIS sequences and strides used in the deformable motion ablation. All other pipeline components (SAM3 segmentation, MonST3R geometry, TPS deformation model) are kept identical; only the correspondence source differs. Flow is computed directly between keyframes. Note that RAFT is warm-started with ego-motion flow computed from MonST3R camera poses and depth; the other methods receive no pose initialization.

RAFT (Teed & Deng, 2020) iteratively refines optical flow via a recurrent update operator on a 4D correlation volume built from all-pairs feature matching (20 iterations, `raft-things` weights).

GMFlow (Xu et al., 2022) replaces iterative refinement with a single global matching step in feature space, enabling faster inference through a transformer-based architecture.

UniMatch (Xu et al., 2023) extends GMFlow into a unified model for flow, stereo, and depth.

CoTracker (Karaev et al., 2024) performs multi-frame point tracking via a transformer operating on dense grid queries, natively handling long-range correspondences.

Table 11 shows that CoTracker achieves the best overall PSNR and SSIM. RAFT and GMFlow perform comparably. UniMatch performs significantly worse, particularly at large strides (s=25) where centroid errors on blackswan reach 112.4 px versus 13.6–14.1 px for the other methods. These results motivate our default choice of CoTracker for DAVIS sequences; swapping the correspondence source changes PSNR by at most 0.7 dB, which quantifies the pipeline's sensitivity to this upstream module.

Table 11: Ablation: optical flow method on three DAVIS sequences (blackswan, camel, breakdance), averaged over the development strides $s \in \{2, 10, 25\}$. Runtime is the total pipeline time per stride.

Method	CE (px)↓	IoU↑	PSNR (dB)↑	SSIM↑	s/run↓
CoTracker (Karaev et al., 2024)	**16.2**	**0.292**	**18.06**	**0.465**	158
RAFT (Teed & Deng, 2020)	16.5	0.292	17.96	0.461	133
GMFlow (Xu et al., 2022)	16.1	0.290	17.93	0.460	**123**
UniMatch (Xu et al., 2023)	36.8	0.247	17.28	0.440	46

E.3 Physical Parameter Source (Stiffness-Adaptive TPS)

We study whether VLM-predicted continuous physical priors improve deformable interpolation, and how robust the pipeline is to noisy material estimates. The stiffness-adaptive variant uses the VLM-predicted Young's modulus E in three ways: (i) it maps E log-linearly to the TPS regularization parameters – the smoothing weight $\lambda_{\text{tps}} \in [10^{-4}, 0.05]$, the maximum number of control points $N_c \in [30, 150]$, and the MAD residual-filter multiplier $\in [1.5, 4.0]$; (ii) it applies *stiffness-dependent temporal easing*, mapping the linear blend fraction α through a symmetric Beta CDF $\alpha_{\text{eff}} = F_{\text{Beta}(a,a)}(\alpha)$ with $a = 1 + \text{clip}(\log_{10}(E/10^3) \cdot 0.3,\ 0,\ 3)$, so stiffer materials receive an S-shaped ease-in/ease-out profile; and (iii) *stiffness-dependent residual attenuation*: the non-rigid TPS displacement is scaled by $\gamma = 1/(1 + (E/E_{\text{ref}})^{0.5})$ with $E_{\text{ref}} = 10^8$, attenuating spurious non-rigid deformation for stiff segments while preserving the residual for flexible ones.

We compare three conditions on blackswan, camel and breakdance at strides 2, 10 and 25:

1. **No priors** (headline configuration) – fixed defaults: $\lambda_{\text{tps}} = 10^{-4}$, $N_c = 100$, MAD multiplier 3.0, no attenuation, linear temporal blend.
2. **VLM priors** – SAM3 assigns $E = 10^6$ Pa to all deformable segments in these sequences ($\lambda_{\text{tps}} = 0.001$, $N_c = 105$, MAD multiplier 3.06, $\gamma = 0.91$, eased blend $a = 1.9$).
3. **Noisy VLM** – each segment's E is randomly perturbed by a factor up to $100\times$ ($E' = E \cdot 10^U, U \sim \mathcal{U}(-2, 2)$) to simulate inaccurate VLM predictions.

VLM priors change PSNR by $+0.01$ dB averaged over all scenes and strides relative to the fixed defaults, with the largest gain on breakdance ($+0.04$ dB); this is why the headline configuration keeps the fixed regularizer. When the VLM-estimated E is perturbed by up to $100\times$, the noisy condition drops -0.04 dB compared to accurate VLM priors: wildly incorrect stiffness values shift the residual attenuation (γ ranges from 0.03 to 0.99 under $100\times$ noise) and the outlier filter threshold. The effect is most pronounced on camel (-0.04 dB) and breakdance (-0.06 dB), where non-rigid deformation is largest. In all conditions the discrete phase dispatch is unchanged; the continuous parameters are secondary to it.

E.4 Fallback Frequency and the Diagnostic Gates

How often do the fallbacks fire? We instrumented every fallback site with a reason code and measured, on a 10-sequence DAVIS subset spanning animals, people, vehicles, and water (blackswan, breakdance, camel, car-roundabout, dog, elephant, horsejump-high, kite-surf, tennis, train), the fraction of dynamic segment-intervals that revert to the per-pixel linear fallback, both by count and weighted by the number of dynamic-region pixels the segment covers in that interval. Table 6 reports the deformable branch, where almost all fallbacks occur; solid segments fall back on 3.4% of segment-intervals (0.3% of solid area) at s=5 and never at s=25, and liquid segments on 8.8% (<0.01% of area) at s=5 and never at s=25. Two observations. First, counting segment-intervals overstates the effect: a 200-pixel sliver counts as much as the main object, whereas by area the deformable fallback covers 0.4% of the dynamic region at s=5 and 8.1% at s=25; TPS carries >99% of deformable pixels at s=5 and ∼92% at s=25. Second, the dominant category is a segment with no rigid pre-fit or fewer than 50 correspondences surviving robust filtering – a segment where *no* parametric model could be fitted, TPS included – which is a limit of the correspondence front-end on small or poorly tracked segments, not of TPS. At s=25 most of the fallback *area* (6.4 of 8.1 points) is genuine rejection of an oscillating TPS solve on large segments. Including the articulated breakdance sequence raises the counts to 44.1%/1.8% (s=5) and 46.8%/8.0% (s=25) by count/area.

Is the fallback the source of the robustness? We added a switch that disables the *diagnostic* gates – the sites where a model was fitted and then rejected – and forces the fitted model to be used regardless

Table 12: **Gates-off ablation**: PSNR change (dB) when the diagnostic fallback gates are disabled and every fitted model is used regardless, on the four sequences where a gate fires.

	breakdance	elephant	dog	camel	mean
s=5	−0.08	0.00	0.00	0.00	−0.01
s=25	+0.01	0.00	−0.01	0.00	

(the structural fallbacks, where no model exists because too few correspondences survived, cannot be switched off). On the four sequences where a diagnostic gate fires (Table 12; sequences where none fires return zero by construction and are excluded), the mean change is -0.01 dB and the worst case -0.08 dB: switched off entirely, the results are unchanged within run-to-run noise. The gate is a safety net; it does not carry the reported numbers.

E.5 Where PhysGraph Stops Working

PhysGraph has well-defined failure regimes (see also Figure 5):

Articulated objects (*e.g.* a cabinet with a sliding drawer; a person reaching for a cup; the breakdance dancer) are returned as a single SAM 3 segment. The relative motion of the parts then appears as a non-rigid residual that the TPS branch absorbs, producing visually plausible but physically incorrect output. A part-aware segmentation head, or a learned articulation prior, would address this.

Plastic, fracture, and hyperelastic regimes are out of scope for the elastic branch. TPS is a linear-elastic radial-basis interpolant; it cannot represent plastic deformation, fracture, or strongly anisotropic stiffness. Constitutive models (*e.g.* material point methods) are the right tool there; PhysGraph's contribution is orthogonal and could front-end such solvers as a phase router.

Per-scene global 4D reconstruction is what optimization-based 3DGS variants (MoSca, Shape of Motion, Dynamic 3DGS) target, and which are the appropriate tool when (a) the full video is observed and (b) per-scene compute is acceptable. PhysGraph trades off scene-specific photometric quality for sparse-input robustness, zero failures, and less compute.

Continuous physical parameters matter much less than the discrete phase. The physical-parameter ablation (Section E.3) shows that swapping the fixed regularizer for a VLM-predicted, stiffness-adaptive one changes PSNR by 0.01 dB, and that a $100\times$-noisy stiffness costs <0.04 dB. The phase dispatch carries the contribution; the continuous values fine-tune.