GeoLens: Where to Prune Transfers Across 3D Foundation Models, How Much Does Not

Anonymous authors
Paper under double-blind review

Abstract

Pruning is a standard way to make large pretrained models cheaper to run, and 3D foundation models need it, above all in robotics and driving. They have become the default front-end for geometry from images, returning depth, point maps and camera pose in one feedforward pass, and on the multiview, streaming and recurrent models the bypassable blocks take a quarter to half of the latency. Yet in a model that emits dense geometry a badly chosen block warps the scene, and to our knowledge whether training-free block pruning is safe there has not been evaluated. **GeoLens** is the first cross-architecture evaluation of it: eight architectures across four inference paradigms and 26 indoor RGB-D sequences, **208 cells** under one depth acceptance test, plus **304 cells** on ETH3D, TartanAir and ScanNet, every number regenerated from released per-cell files. It yields a recipe and two limits. *Where* to prune transfers in domain: one fixed set of five blocks per architecture, taken from unlabeled scoring of its other sequences, passes **94%** of cells and matches per-sequence re-scoring cell for cell. An oracle that spent a labeled pass on every block, measuring its damage on the benchmark itself, gains at most about two points on that set (95% interval): one unlabeled choice per architecture suffices, with no retraining. A prior fit on other architectures picks sets that pass 87% of cells in its seven-model pool and $17/26$ outside it. *How much* is the first limit: *capacity*, the largest bypass ratio sustaining $\geq$80% acceptance, spans $3.2\times$ across the roster ($2.0\times$ under a wider block count) and shifts with the evaluation distribution; out of domain no $K{=}5$ set we tried clears the bar on three of eight architectures on ScanNet. It does not follow from the per-block ranking, because single-block safety does not compose into set safety, and no pre-registered structural scalar of six tracks it under every block count. The second limit is the one that matters in deployment: the acceptance test licensing the recipe does not certify the task. **On nine cells the acceptance test accepts, trajectory error grows by up to** $2.54\times$, across five of the seven architectures we checked. On the multiview, streaming and recurrent models, bypass saves 5.8–11.5% of wall-clock at $K{=}5$ and up to 22.5% at capacity. We release the benchmark, the per-cell results and the pre-registrations.

1 Introduction

3D foundation models (3D-FMs) have become the default front-end for geometry from images. One feedforward pass now returns depth, point maps and camera pose, replacing a pipeline of matching, triangulation and bundle adjustment (Wang et al., 2024; Leroy et al., 2024; Wang et al., 2025a; 2026). That pass is not cheap: most of these models expose 24 bypassable block instances, and on five of the eight measured here the bypassable blocks take a quarter to half of the latency. The cost matters most where geometry is consumed online, on the on-board compute of a robot or a car. There the world moves while the model runs, so latency itself costs accuracy (Li et al., 2020), and 3D-FMs already serve as the geometry front-end of dense SLAM (Murai et al., 2025; Maggio et al., 2025) and of reconstruction along driving sequences (Fei et al., 2024; Deng et al., 2026). A cheaper pass buys frame rate, or compute for planning and control, and training-free pruning reaches it without

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.

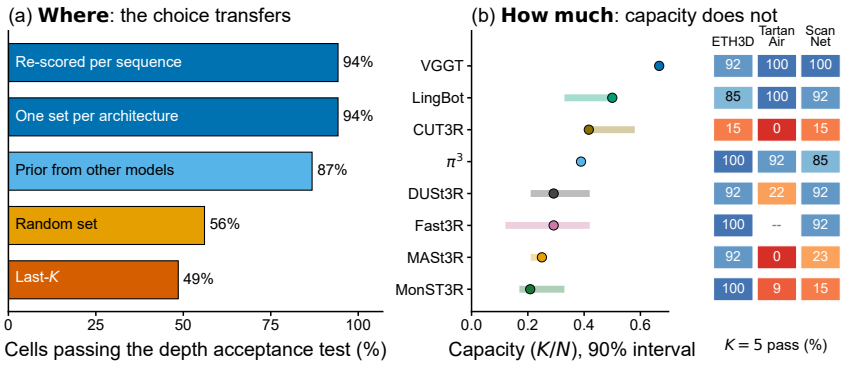

Figure 1: **Where to prune transfers across 3D foundation models. How much does not. (a)** $K{=}5$ acceptance over the 208 benchmark cells, by what the block set is allowed to know about the deployment (208 cells for the first two bars and for last-K, 182 for the prior, 1040 set×cell evaluations for random). **(b)** *Capacity*, the largest bypass ratio sustaining $\geq$80% acceptance, with its 90% bootstrap interval; beside it, the share of cells the $K{=}5$ set passes on ETH3D, TartanAir (Fast3R excluded; its pair-model misses are mis-ranking, §4.3) and ScanNet.

retraining a foundation model. A practitioner therefore needs to know which blocks can be skipped, and whether the pose these systems consume survives the cut.

For language models the recipe is training-free: score each block's importance with a single forward pass, then identity-bypass the safest K (Men et al., 2025; Gromov et al., 2025). To our knowledge no one has tried it on a 3D-FM (§2). Two differences make that transfer a real question: these models contain no language model, so the late-layer redundancy the recipe leans on is not guaranteed, and they emit dense metric geometry, so a badly chosen block warps depth where a language model would shift a distribution over words.

Table 1: **Four working defaults of LLM block pruning, tested on 3D-FMs.** Two fail, one is unsettled, one needs far less than it assumes.

Working default of LLM block pruning	On 3D foundation models
Drop the last layers	Fails: last-K passes 49% of cells, the scored set 94% (§4.2)
Removing fewer blocks is safer	Fails: acceptance is not monotone in K; 52 prunes fail while a superset of them passes (§5.2)
Calibrate on the target	Needs less: one set per architecture, chosen blind to the sequence, matches per-sequence re-scoring cell for cell; a prior from other models, 87% in-pool (§4)
Read removability off redundancy	Unsettled: no structural scalar of six is significant under every block count; the one that correlates is confounded (§5.3)

What we find. **GeoLens** treats block bypass as an evaluation problem (Fig. 2). The selector is unchanged by design; what is new is the evaluation, a controlled cross-architecture test of the LLM block-pruning recipe on metric geometry. It returns a recipe with two limits (Fig. 1) and, on the way, tests four working defaults of that literature (Table 1): two fail, one is unsettled, one needs far less than it assumes. Concretely, we contribute:

- ***Where* to prune transfers, in domain.** One fixed $K{=}5$ set per architecture, blind to the sequence, passes 94% of cells and matches per-sequence re-scoring cell for cell, and the unlabeled scoring that picks it is not separated by an exact test from an oracle that spent a labeled pass on every block. A prior fit on *other* architectures picks sets passing 87% of its pool's cells and 17/26 outside it; it ranks per-block output shift at Spearman $\rho{=}{+}0.53$ but depth damage only at $+0.09$ (§4).
- ***How much* does not.** Capacity spans 3.2× and moves with the evaluation distribution: out of domain no $K{=}5$ set we tried clears the bar on three of eight architectures on ScanNet. Nor is it read off structure: no pre-registered scalar of six correlates under every block count. Single-block safety does not compose, replicating for metric geometry what Sieberling et al. (2025) report for language models (§5).

- **The acceptance test does not certify the task.** Bypass saves 5.8–11.5% of wall-clock at $K{=}5$ outside the pair models, but depth acceptance fails to protect the trajectory on five of the seven architectures we checked: nine depth-accepted cells lose up to $2.54\times$ (§6).
- **The instrument.** Eight architectures across four inference paradigms (Table 2), 208 cells under one depth acceptance test and 304 more on three further datasets, fourteen pre-registration documents, and every number regenerating from released per-cell files that ship with the record of the claims we retracted (Appendix K). To our knowledge it is the first cross-architecture, cross-dataset pruning benchmark for 3D-FMs.

2 Related Work

3D foundation models. Our roster spans four inference paradigms (§3): pair-based encoder-decoders (Wang et al., 2024; Leroy et al., 2024; Zhang et al., 2025), bidirectional multiview transformers (Wang et al., 2025a; Yang et al., 2025a; Wang et al., 2026), a causal streaming model (Chen et al., 2026a) and a recurrent stateful decoder (Wang et al., 2025b). Existing 3D-FM benchmarks evaluate full models across tasks (Cong et al., 2025); none analyses which blocks can be bypassed at inference, or how prunability compares across the family.

Training-free layer / depth pruning. The recipe scores blocks by input-output similarity in one pass (Men et al., 2025; Gromov et al., 2025) or by the measured effect of a removal (Song et al., 2024; Zhang et al., 2024), and drops the least important. Later variants add replacements, compensation or recovery training and reach vision-language and action models (Shopkhoev et al., 2025; Chen et al., 2026b; Ma et al., 2025; Yuan et al., 2025; Yang et al., 2025b), all on language-model backbones that typically need target calibration. GeoLens keeps ShortGPT's score (Men et al., 2025) unchanged, so no effect comes from a scoring choice of ours (§3). Three findings there shaped this study: EvoPress (Sieberling et al., 2025) reports that per-layer errors do not add and that pruning further can recover performance, Hinostroza et al. (2026) that a cosine score is a weak proxy for a layer's measured removal damage, and Lu et al. (2026) that dropping a language model's last quarter, with a small fine-tune, already competes with scored selection. The first two recur here (§5.2, Appendix B); the third does not, last-K passing 49% against the selector's 94% (§4.2), so a weak per-block instrument can still be a strong set-level one. A further question, whether how much can go is readable from architecture, goes untested there (§5.3; unsettled). Table 1 tests its four working defaults.

Efficiency work on 3D-FMs. Existing acceleration of 3D-FMs targets the token, attention, precision and cache axes, mostly on VGGT (Shen et al., 2025; Shu et al., 2025; Li et al., 2026; Sun et al., 2025; Feng et al., 2026; Liu et al., 2026); Spark3R (Tang et al., 2026) and Lite3R (Zhang et al., 2026) carry the token and attention axes across architectures. None removes transformer blocks, and none measures whether a block selector transfers across architectures. Those axes are orthogonal to depth; we measure how bypass stacks with token merging (§6). Closest to us, AVGGT (Sun et al., 2025) restructures VGGT's attention training-free and RegimeVGGT (You et al., 2026) reports U-shaped layer redundancy in VGGT on the token axis. Both are single-family methods; neither is a benchmark. Appendix N covers these lines, and training-time depth adaptation, in full.

3 Setup

Models and bypass. We benchmark eight 3D-FMs from official checkpoints, with their paradigms and block counts in Table 2. What we bypass is the aggregator's global blocks in VGGT-1B and LingBot-Map-Long, and the decoder in the other six. π^3 alternates frame and global decoder blocks and CUT3R-512 applies its decoder recurrently once per frame. "Layer" and "block" both denote one transformer block (attention, cross-attention where the architecture has it, and MLP), always bypassed whole. Writing block ℓ as $B_\ell(\mathbf{x}) = \mathbf{x} + \mathbf{u}_\ell(\mathbf{x})$, with $\mathbf{u}_\ell$ its whole residual update, the *identity-bypass hook* replaces B_ℓ by the identity (Fig. 2); f is the unpruned model and f_S the model with every block of a set S bypassed at once. We report the *bypass ratio* K/N where N is the number of bypassable block instances; the notation table that opens the appendix lists every symbol.

Datasets and depth metric. 7-Scenes and TUM-RGBD give the 208-cell benchmark; ETH3D, TartanAir and ScanNet give the 304 cross-dataset cells of §4.3; CO3D windows are used only for

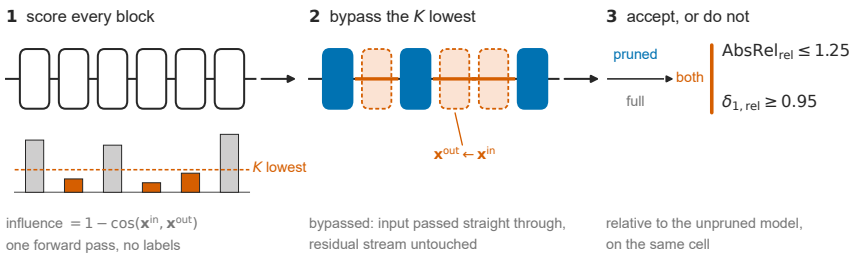

Figure 2: **The evaluation.** Scoring is label-free and costs one forward pass; the K least influential blocks are replaced by the identity, leaving the residual stream untouched; acceptance is joint, relative to the unpruned model on the same cell. Schematic: six blocks, $K{=}3$, illustrative scores.

the cross-architecture prior and its per-block sweep; and a pre-registered out-of-domain study on KITTI is reported in Appendix E. The benchmark evaluates dense depth on all seven 7-Scenes scenes (Shotton et al., 2013) and TUM-RGBD (Sturm et al., 2012) (fr1/desk, fr1/xyz, fr2/xyz), for 26 valid sequences; the planned heads/seq-03 is absent from the release, so heads contributes two. Each sequence contributes 30 windows of $V{=}4$ frames (frame stride 4, window step 8). Predicted target-view depth is compared to sensor depth after median-scale alignment, so metric scale is not tested, and every metric is reported relative to the unpruned model on the same cell.

Acceptance test (PASS). A *cell* is one (architecture, sequence) pair. Writing d for predicted and d^* for sensor depth, with $\langle\cdot\rangle$ the mean over one window's valid pixels:

$$\hat{s} = \operatorname{median}\big(d^*/d\big), \qquad \text{AbsRel} = \big\langle |\hat{s}d - d^*|/d^* \big\rangle, \tag{1}$$

$$\delta_1 = \big\langle \mathbf{1}\big[\max(\hat{s}d/d^*,\, d^*/\hat{s}d) < 1.25\big] \big\rangle, \quad \text{AbsRel}_{\text{rel}} = \text{AbsRel}_{\text{pruned}}/\text{AbsRel}_{\text{full}},$$

$$\delta_{1,\text{rel}} = \delta_{1,\text{pruned}}/\delta_{1,\text{full}}, \qquad \text{PASS} \iff \text{AbsRel}_{\text{rel}} \le 1.25 \ \wedge\ \delta_{1,\text{rel}} \ge 0.95.$$

The two bounds allow at most a 25% depth-error increase and a 5% δ_1 drop; both are swept in §4.1, and Appendix A gives the validity mask and the released function every cell was measured with.

Selectors. Our primary deployable selector is **block-influence**, ShortGPT's score (Men et al., 2025) with nothing changed, so that a difference between architectures is attributable to the architecture and to no scoring choice of ours. Two properties make it the one to deploy: it costs one forward pass where any damage-measuring alternative costs N, and it is calibration-free. A labeled greedy search finds better sets (§7); this paper measures how the unchanged score transfers. On the evaluation sequence's own 30 windows W, with $\mathbf{x}^{\text{in}}_{\ell,t}$ and $\mathbf{x}^{\text{out}}_{\ell,t}$ block ℓ's input and output at token t, and $\langle\cdot\rangle_{t\in w}$ the mean over every token block ℓ processes in window w, the score and the bypassed set are

$$\text{BI}_\ell = 1 - \frac{1}{|W|}\sum_{w\in W} \Big\langle \cos\big(\mathbf{x}^{\text{in}}_{\ell,t}, \mathbf{x}^{\text{out}}_{\ell,t}\big)\Big\rangle_{t\in w}, \qquad S_K = \operatorname*{arg\,min}_{|S|=K} \sum_{\ell\in S} \text{BI}_\ell, \tag{2}$$

that is, the K least influential blocks. Since $\mathbf{x}^{\text{out}} = \mathbf{x}^{\text{in}} + \mathbf{u}_\ell$, the score reads the angular size of the update ($\approx \frac{1}{2}\|\mathbf{u}_\ell\|^2/\|\mathbf{x}^{\text{in}}\|^2$ when it is small and orthogonal to the stream; Appendix A; cf. Gromov et al., 2025), so a low score means the block already acts nearly as the identity that bypass substitutes. Calibration-free means no depth labels and no held-out calibration set (not camera calibration); the score does observe the unlabeled test frames, at the cost of one scoring pass (≈6 s on VGGT). We also evaluate the **FZD prior** (Free-Zone Discovery), a cross-architecture predictor of per-block bypass damage, fit to output shift on CO3D (its output against the full model's, no ground truth). A model's *free zone* is its bypass-safe region. We fit the prior on *other* architectures and apply it to a held-out one with no target calibration (§4.2). The two selectors span three *deployment modes*, all measured: per-sequence re-scoring, one leave-one-sequence-out set per architecture, and transfer to an architecture the prior never saw. An **upper-bound ceiling** takes the better of the two per cell; choosing it needs ground truth, so it is not deployable. Two baselines: **Last-K**, the standard depth-prune heuristic, and **Random**, which draws 15 seeded block sets shared across sequences, so its sampling unit is the set (Appendix B). No 3D-FM is retrained; everything is forward-pass only on a single GPU (FZD adds one ridge fit). "The selector" below means block-influence. Unless a K is named, results are at $K{=}5$, fixed with the two cutoffs in the first pre-registration (2026-05-19) before the benchmark was run; it is also the largest grid point every architecture passes at $\ge 80\%$ in domain (Fig. 4).

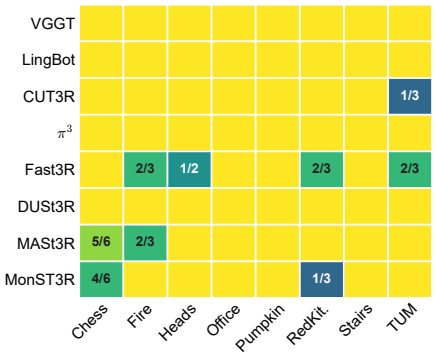

Figure 3: **Training-free block-influence pruning at $K{=}5$ preserves depth across architectures and scenes.** Each tile is one architecture×scene: the fraction of its sequences whose block-influence $K{=}5$ pruning passes the acceptance test, written in wherever it is below 1 (unannotated tiles: every sequence passes); colour runs from dark (0) to yellow (1); RedKit. is Red Kitchen and the TUM column pools the three TUM trajectories; $196/208{=}94\%$ overall.

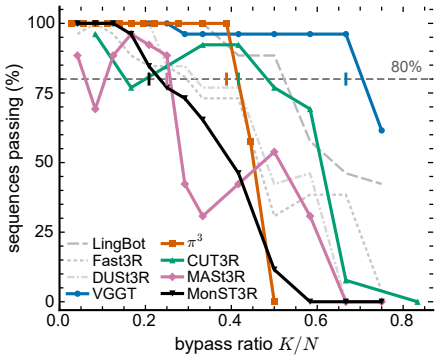

Figure 4: **Capacity spans $3.2\times$ on the full benchmark, and the curves are not monotone.** Acceptance pass rate over all 26 sequences vs. bypass ratio K/N under block-influence; *capacity* is the largest ratio with $\geq 80\%$ pass (Table 2), marked by a tick on the 80% line. The $K{=}2$ dips of MASt3R and CUT3R are nested-set effects (§5.2). Dimmed curves: LingBot (dashed), Fast3R (dotted), DUSt3R (dash-dot).

Table 2: **Every architecture keeps at least 22 of 26 sequences at $K{=}5$, yet their capacity spans $3.2\times$.** N: bypassable block instances (pair-Siamese: 12 decoder blocks × 2 branches). Full: the unpruned model's absolute AbsRel. Cap.: capacity, the largest ratio K/N with $\geq 80\%$ of the 26 sequences passing acceptance under block-influence (Fig. 4). $K{=}5$ AbsRel/δ_1/Pass: depth relative to the full model and acceptance pass rate, mean over the 26 valid sequences. Spd.: $K{=}5$ wall-clock saving under real removal, timing the selector's own set (Appendix I; ≈ 0 = within noise).

					$K{=}5$ block-influence			
Model	Paradigm	N	Full	Cap.	AbsRel	δ_1	Pass	Spd.
VGGT	multiview	24	0.086	**0.67**	0.97	1.00	100%	6.5%
LingBot	streaming	24	0.095	0.50	0.93	1.01	100%	5.8%
CUT3R	recurrent	12	0.078	0.42	1.05	1.00	92%	11.5%
π^3	multiview	36	0.063	0.39	1.01	1.00	100%	7.6%
Fast3R	multiview	24	0.108	0.29	1.04	1.00	85%	6.5%
DUSt3R	pair	24	0.074	0.29	1.05	1.00	100%	≈0
MASt3R	pair	24	0.088	0.25	1.11	0.99	92%	≈0
MonST3R	pair	24	0.079	0.21	1.12	0.99	85%	≈0

4 Where to Prune: Lawful In Domain, Mostly Portable

Pruning asks where and how much, and the first is answered in three widening scopes: across the sequences of one architecture (§4.1), across architectures (§4.2) and across evaluation distributions (§4.3).

4.1 One Fixed Set Passes 94% of 208 Cells, Blind to the Sequence

Table 2 and Fig. 3 give the headline, and Fig. 5 shows what a pass and a failure look like. At $K{=}5$ the calibration-free block-influence selector passes the acceptance test on **94%** of the 208 architecture×sequence cells ($196/208$), and every architecture on at least 22 of its 26 sequences. The *fixed* set, one set per architecture (for each sequence the modal set of that architecture's other 25 sequences), never sees the sequence it prunes and reproduces $196/208$ with no cell changing verdict, the two sets differing on 53 cells. On a $3{\times}3$ grid around Eq. 1's cutoffs, $\geq 6/8$ architectures still pass $\geq 2/3$ of their cells at every point. Misses and the grid are in Appendices B and E.

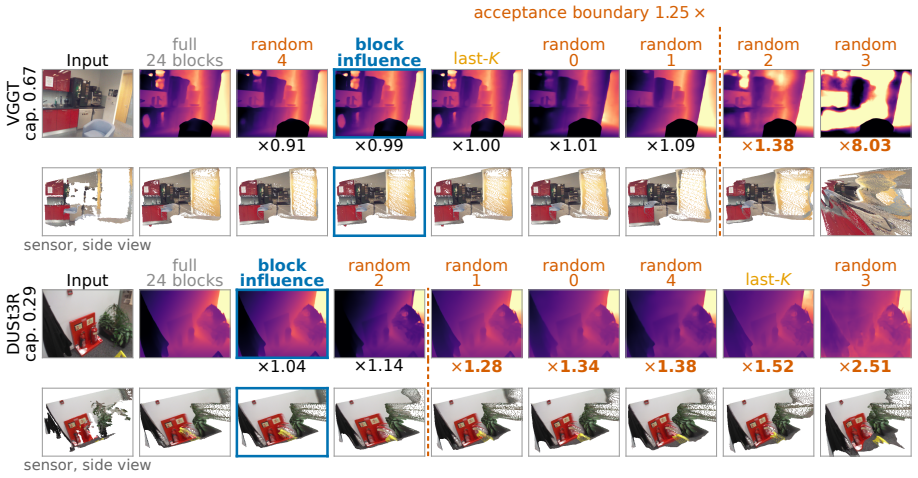

Figure 5: **A passing five of 24 blocks leaves the scene intact; a bad five can shatter it.** Seven $K{=}5$ sets on two cells, ordered by cell-level relative AbsRel, boundary at $1.25\times$; each image is the window nearest block-influence's cell mean, on cells fixed by rule before rendering; below each row the same maps are back-projected and viewed from one raised side, sensor first, one scale per row. Acceptance is metric: DUSt3R's $\times 1.04$ and $\times 1.28$ look alike, while VGGT's random 3 shatters the scene (Appendix B).

Table 3: **No deployable arm passes more cells than one unlabeled scoring pass.** Joint pass rate (Eq. 1) at $K{=}5$ over the valid cells, with Wilson 95% intervals (indicative: cells within a scene are correlated). †Needs the FZD pool or a per-block sweep, so covers the pool's seven models (182 cells). ‡Random: 15 sets reused across cells, so the bracket is the range of per-set pass rates. §Five of each cell's eight lowest-influence blocks, six draws per cell, pre-registered (Appendix C). More in Appendix B.

Selector ($K{=}5$)	PASS	95% CI	Depl.?
Block-influence (ours)	**94**% (196/208)	[90, 97]	yes
FZD prior†	87% (158/182)	[81, 91]	yes
SLEB-style iterative	83% (173/208)	[77, 88]	yes
Position+magnitude†	77% (140/182)	[70, 82]	yes
Random (15 sets)‡	56% (583/1040)	[4, 100]‡	yes
Last-K	49% (101/208)	[42, 55]	yes
Pool-random, 5 of the 8 safest§	82% (1018/1248)	[79, 84]	yes
Single-block oracle, CO3D damage†	75% (136/182)	[68, 80]	no (N passes)
Single-block oracle, benchmark damage	91% (190/208)	[87, 94]	no (N passes, GT depth)
Ceiling (best of two)†	96% (175/182)	[92, 98]	no

4.2 One Scoring Pass Matches an Oracle That Measures Every Block

Against the other selectors. Table 3 compares every arm at $K{=}5$: SLEB-style iterative re-scoring (Song et al., 2024) reaches 83% but loses CUT3R, and the naive baselines trail far behind (random 56%, last-K 49%).

Against an oracle that measures damage. A single-block oracle bypasses each block in turn, measures its damage D_ℓ, and keeps one set per architecture, $S^{\text{orc}} = \arg\min_{|S|=K} \sum_{\ell \in S} D_\ell$, at N forward passes to the score's one. With D_ℓ measured on the benchmark itself, as relative AbsRel against sensor depth, it passes 190/208 against block-influence's 196/208; an exact McNemar test does not separate the two ($p{=}0.33$), and the paired difference of the two pass rates over the same cells, $\text{pass}_{\text{BI}} - \text{pass}_{\text{orc}} = +2.9$ points, carries a scene-cluster bootstrap 95% interval of $[-2.1, +7.8]$, which bounds any advantage of the oracle at about two points. With D_ℓ measured instead as output shift on out-of-domain CO3D windows it passes only 75% of the cells it covers, a gap concentrated on CUT3R and owed to the instrument (Appendix B).

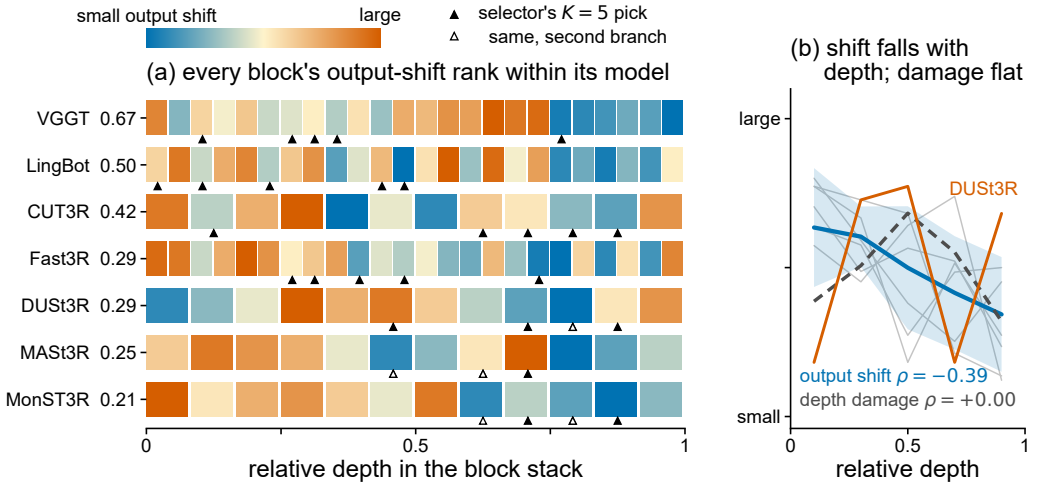

Figure 6: **The free zone, drawn by output shift. (a)** Every block of the seven architectures in the prior's pool, by relative depth, coloured by its rank in its own model's single-block output shift on 30 CO3D windows; labels give capacity, triangles the modal block-influence $K{=}5$ set. **(b)** The same ranks binned by depth: deeper blocks shift the output less (blue, pooled Spearman -0.39), DUSt3R excepted; the benchmark's depth damage, the dashed grey line, does not fall with depth ($+0.00$). The picks use a row's five bluest blocks on 17 of 35 slots and pass more cells than those five, 93% against 75% on the 182 cells both cover (π^3, swept later, is not drawn; Appendix D).

Against its own safe region. The sharper control keeps the score's region and randomises the choice inside it: five blocks drawn from each cell's eight lowest-influence blocks pass 82%, indeterminate against the bands locked before the run. Paired by cell the selector wins 39 disagreements to 2 (exact McNemar $p < 0.001$), 23 of them on CUT3R. The draws match it on π^3 and VGGT (100%) and collapse to 26% on CUT3R, a spread that tracks how much of each model the eight-block pool covers (exploratory; Appendix C).

Across architectures. By output shift on CO3D, deeper blocks are individually safer, though not along one curve and not by the benchmark's depth damage (Fig. 6). The FZD prior fits a ridge predictor on eight residual-stream signals per block and its position in the stack, from one forward pass over 30 CO3D windows. With no target calibration it ranks a held-out architecture's per-block output shift at mean leave-one-model-out $\rho{=}+\mathbf{0.53}$, but its depth damage only at $+0.09$ (exploratory; Appendix D). Its five-block sets carry across: they pass 87% of the pool's cells. On π^3, held out of the pool by pre-registration, the ranking transfers ($\rho{=}+0.69$; $+0.45$ on depth) but the set passes only $17/26$. Inside the pool, position and magnitude alone retain 96% of the prior's leave-one-out ρ ($+0.51$ of $+0.53$) yet deploy at only 77%, collapsing on DUSt3R (Appendix B).

4.3 Under Shift, Failures Are Mis-ranking or a Lost Operating Point

The ordering survives shift. Recomputed on CO3D and TUM-RGBD, each model's output-shift ordering correlates at a median $+0.62$ across distributions, $+0.75$ within the five feedforward models; the two state-carrying ones reorder most (Appendix E). On ETH3D block-influence keeps $\geq 80\%$ of cells on 7 of 8 architectures, all but CUT3R. A registered control implicated ETH3D's sparse ground truth, so we ran two complete-coverage benchmarks, TartanAir and ScanNet, where the fixed set is within two cells of per-scene re-scoring on 6 of 7 and 6 of 8 architectures (Appendix F).

What fails is of two kinds. On TartanAir's pair family the *ranking* fails: one random draw passes where every informed arm misses. On ScanNet's MASt3R, MonST3R and CUT3R, and on CUT3R over TartanAir, the *operating point* fails: no $K{=}5$ arm reaches the bar, though the full model is feasible. Block choice holds in domain; its budget does not port, which is the second question.

5 How Much to Prune Does Not Transfer: Sets Do Not Add Up

The second question has the opposite answer. With $\mathrm{pass}_a(K)$ the fraction of architecture a's 26 sequences whose K-set passes Eq. 1, and D_ℓ, D_S the damage of bypassing block ℓ alone and set S,

$$\mathrm{Cap}(a) = \max\{K/N \;:\; \mathrm{pass}_a(K) \geq 0.8\}, \qquad \Delta(S) = D_S - \sum_{\ell \in S} D_\ell. \tag{3}$$

Capacity varies with model and distribution (§5.1), does not follow from the ranking as Δ is neither zero nor of one sign (§5.2), and no structural scalar tracks it under every block count (§5.3).

5.1 Capacity Spans 3.2× and Moves With the Evaluation Distribution

Sweeping K gives a capacity curve per architecture (Fig. 4, Table 2). VGGT tolerates removing 16 of 24 blocks, MonST3R only 5: a $3.2\times$ spread (90% interval $[2.7\times, 5.3\times]$; $2.0\times$ if the never-bypassed frame-attention blocks count in N), whose ordering largely survives the threshold grid, the pass cutoff and a sequence bootstrap (Appendix E). It crosses paradigm lines and does not track accuracy: π^3, the most accurate unpruned model, keeps every sequence through $K{=}14$ and none at $K{=}18$.

Capacity moves with the distribution. Re-read on a TUM-weighted subset, capacity shifts in both directions, and CUT3R's fall from 0.42 to 0.08 is the one move large enough to stand on six sequences (Appendix E). Out of domain no $K{=}5$ set we tried clears the bar on three architectures on ScanNet or on CUT3R over TartanAir (§4.3). A model can hold one benchmark and lose another: MonST3R keeps $13/13$ ETH3D scenes and $2/13$ ScanNet rooms, yet all 13 rooms at $K{\leq}3$ (Appendix F).

5.2 Set Safety Is Not the Sum of Single-Block Safety

Acceptance is not monotone in K. MASt3R passes 88% of sequences at $K{=}1$, 69% at $K{=}2$ and 96% at $K{=}4$, on nested sets. Across the 208 cells, 52 prunes fail while a superset of them passes, counted as pairs of a cell c and adjacent swept budgets $K_i < K_{i+1}$ with $\mathrm{PASS}(c, S_{K_i}) = 0$ and $\mathrm{PASS}(c, S_{K_{i+1}}) = 1$; re-measured under pre-registration, most of these repairs are separated improvements, far more than a global null predicts (Appendix G; Fast3R's are exploratory, its forward pass being non-deterministic). Removing *more* blocks can repair a failing prune.

Additivity is indeterminate pooled and mis-signed per model. The registered pooled test is indeterminate: summed single-block damage predicts joint damage only weakly ($\rho{=}{+}0.35$ over 496 sets, on a grid without depth ground truth) and its sign changes with the architecture (Appendix G). Table 10 makes it concrete on one scene (exploratory): co-bypassing each model's individually-safest blocks, the sum predicts no damage on the three pair models and π^3 while the measured effect is *damage*, each interval excluding zero; LingBot's predicted gain does not appear.

Where the sum holds. VGGT's joint damage stays below zero across the grid and CUT3R's compounds with K (Fig. 10), both close to the sum at $K{=}8$ (Table 10); the departures are the four above and, at deployment, the non-monotone dips. Joint damage does not rank capacity either, and an earlier contrary claim is retracted (Appendix K). The benchmark-measured oracle (§4.2) fits this failure at deployment: its individually-safest blocks do not improve on a set chosen without measuring damage.

5.3 Five of Six Structural Scalars Fail; the Sixth Is Confounded

Following a pre-registration that fixed the candidates before computing them, we tested six structural predictors of capacity: width, removable-block count, residual magnitude, parameter count, effective dimensionality and inter-block CKA (Fig. 11a). Five fail the corrected bound (residual magnitude is nominal only, $p{=}0.047$). The sixth, parameter count, correlates ($\rho{=}{+}0.87$, exact $p{=}0.007$) but is confounded with the roster's size families, so it licenses no structural proxy. The verdicts also depend on the block count: counting the never-bypassed blocks lifts inter-block CKA to $+0.95$ and drops parameter count to $+0.42$, so no scalar is significant under every N. At $n{=}8$, where only $|\rho| \gtrsim 0.72$ reaches $p < 0.05$, the nulls exclude only large effects (Appendix H). The unpruned model's own AbsRel does not predict capacity either ($\rho{=}{+}0.02$).

The matched pair. VGGT and Fast3R share exposed depth, width and paradigm, yet differ $2.3\times$ in capacity (Fig. 11b; $1.1\times$ with VGGT's never-bypassed blocks counted), so the driver lies outside those three axes. An iterative-refinement reading (Greff et al., 2017; Jastrzębski et al., 2018; Veit et al., 2016) is consistent with every measurement here but untested causally (Appendix J).

6 What $K{=}5$ Buys and Whether It Keeps the Task

6.1 $K{=}5$ Saves 5.8–11.5% Outside the Pair Models

Wall-clock. Table 2 reports the saving at $K{=}5$ under real block removal on one RTX 5090: 5.8–11.5% on the five multiview, streaming and recurrent models, whose bypassable blocks take 28–55% of the forward. The three pair models, whose timing noise exceeds the effect, gain nothing measurable. At each architecture's own capacity the saving grows to 22.5% on π^3 and 20.6% on VGGT, with $26/26$ and $25/26$ passing (Appendix I). Bypass stacks with FastVGGT's token merging (Shen et al., 2025) as registered at $V{=}16$ and 64; at $V{=}4$, where merging alone is slower than the full model, the stack trails bypass alone, failing a registered clause (Appendix I).

6.2 Depth Acceptance Fails the Task on Five of Seven Architectures

Let $\mathrm{ATE}(f)$ be the RMSE of a model's camera centres after Umeyama Sim3 alignment to ground truth, averaged over the non-overlapping chunks of every fourth frame that cover a full TUM sequence, and $\mathrm{ATE}_{\text{rel}} = \mathrm{ATE}(f_S)/\mathrm{ATE}(f)$; a cell fails the task when it passes Eq. 1 yet $\mathrm{ATE}_{\text{rel}} > 1.25$. Block-influence $K{=}5$ keeps $\mathrm{ATE}_{\text{rel}} \leq 1.07$ on VGGT and π^3, where a random $K{=}5$ costs 2.1–$4.7\times$. The recurrent CUT3R loses accuracy on all three sequences, by $2.54\times$ on the one the acceptance test accepts (Appendix M). Fast3R and the three pair models emit no pose, so we recover camera centres from their point maps and score them against ground truth under three pre-registered guards against the convention failure this project once retracted; all pass (Appendix L). **Eight of their twelve cells are accepted by the acceptance test and exceed $1.25\times$ on trajectory**, up to $2.085\times$ and on every one of the four; the single cell the acceptance test rejects costs $0.993\times$.

Practical recommendations. (i) Score $\approx$30 unlabeled windows from the deployment domain with block-influence and bypass the $K{=}5$ least influential: no labels, no retraining, one scoring pass. (ii) Re-measure K in that domain (§5.1). (iii) Validate on the downstream task as well as on depth.

7 Limitations

The selector is not optimal: a pre-registered labeled greedy search beats it slightly (-0.091 mean relative AbsRel), mostly on MASt3R (Appendix E). The 208-cell benchmark is indoor depth; three further benchmarks test it (Appendices E–F). Scale-aligned metrics do not test metric scale, and no trajectory was measured on LingBot. MapAnything (Keetha et al., 2026), Depth Anything 3 (Lin et al., 2025) and MUSt3R (Cabon et al., 2025) are untested. The mechanism analysis is correlational over $n{=}8$ checkpoints with collinear axes, so no mechanism is established (cf. Shi et al., 2026).

8 Conclusion

Training-free block pruning transfers to 3D foundation models as a recipe with two limits. *Where* to prune is lawful in domain: one fixed five-block set per architecture, chosen blind to the sequence, holds on 94% of 208 cells and is not separated from an oracle that measured every block on the benchmark. *How much* does not transfer: capacity spans $3.2\times$, moves with the evaluation distribution and does not follow from the per-block ranking, because single-block safety does not compose. And depth acceptance failed the task on five of the seven architectures we checked. A label-free budget predictor and a task-certifying acceptance test remain open, with the released cells as their test bed; the matched pair VGGT and Fast3R, $2.3\times$ apart in capacity, is where the budget search should start.

AI USE STATEMENT

In this work we used generative AI tools for the following tasks with required disclosure: to propose and refine hypotheses, to design and give feedback on the research methodology and experiments, to implement methods, to clean and reformat datasets, to help develop the conceptual framework, and to interpret results. We have not used generative AI tools to generate synthetic datasets (every dataset here is a pre-existing public release) or to assist with translation; formulating or proving mathematical claims and qualitative or thematic data analysis are not applicable to this work, which states no new mathematical results and analyses no qualitative data. Additionally, we used generative AI tools for tasks with recommended disclosure: to draft parts of the paper and edit it for readability, to suggest its structure, to identify and summarise related literature, to create and edit software code and figures, and to format references.

We have reviewed all AI-assisted work. AI-drafted text was revised by the authors and checked against the released results; AI-written code was run by the authors and its outputs were checked against the released per-cell files; the paper's numbers were audited against those files by script; hypotheses, thresholds and controls proposed with AI assistance were fixed in the pre-registration documents before the corresponding data existed; literature surfaced by AI tools was cited only after the authors had checked the source and its bibliographic data. The authors set the research direction, decided which studies to run and when to stop, and made all positioning, retraction and scoping decisions. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work measures inference-time pruning of publicly released 3D reconstruction models on publicly released datasets. It involves no human subjects, collects no personal data and releases no new imagery. 7-Scenes, TUM-RGBD, ETH3D, TartanAir, KITTI and CO3D are used under their published licences and terms; ScanNet is used under its terms of use, and we release only its per-cell metrics, never its frames. The released benchmark, per-cell results and pre-registration documents are anonymised for review. Block bypass lowers the inference cost of existing models and adds no capability beyond theirs, so we see no misuse risk specific to it. Pre-registered hypotheses and thresholds, a record of the claims we retracted or demoted (Appendix K) and the AI-use disclosure above are part of the submission.

REPRODUCIBILITY STATEMENT

The protocol this paper measures — the identity-bypass hook, the selector, the windowing and the acceptance test — is defined in §3, with the threshold and distribution-shift robustness checks in Appendix E and the per-benchmark protocols in Appendices F and M. Claims we retracted or demoted along the way are collected in Appendix K. The anonymous supplementary material contains every per-cell result, the analysis scripts that regenerate every reported figure and table, the deployment harnesses of the fixed-set, random-set, held-out-prior and timing studies, the download scripts for the two complete-coverage benchmarks, and the pre-registration documents that fixed each study's hypotheses, thresholds and kill criteria before the measurements they govern; its `README` describes the layout, the file conventions and how each number is reproduced. Each pre-registration records its own lock date, self-declared and corroborated by file modification times, not independently timestamped. All models are official public checkpoints and all datasets are public releases; ScanNet is used under its own terms of use, and we release its per-cell metrics, never its frames. The release also pins the environment the measurements ran in — package versions, the CUDA and driver versions, a SHA-256 for every checkpoint file and the upstream commit of every model repository — read off the run container and shipped raw, with the limits of a post-hoc reading stated alongside them. Every measurement in this paper ran on a single RTX 5090. The first run we timed end to end, the 234-cell oracle deployment of Appendix B (the benchmark-measured oracle on 208 cells, π^3's CO3D-measured set on 26), took 1.95 GPU-hours at one selector arm per cell; the release records 34,300 arm evaluations of that shape, each 30 windows of $V{=}4$ frames, so reproducing them costs roughly 60–290 GPU-hours, the spread depending on how much of each cell's model load is shared by the arms measured on it. Later runs were timed end to end as well: the pool-random control (208 cells, seven arms each) took 3.2 GPU-hours, the two trajectory studies 1.1 and 1.0 (Appendices L

and M) and the token-merging timing study 2.1 (Appendix I). Exploratory runs that did not reach the release are not counted.

REFERENCES

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations (ICLR)*, 2023.

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. MUSt3R: Multi-view network for stereo 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2503.01661.

Lin-Zhuo Chen, Jian Gao, Shangzhan Zhang, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, and Yinghao Xu. LingBot-Map: Geometric context transformer for streaming 3D reconstruction. *arXiv preprint arXiv:2604.14141*, 2026a.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining redundant layers to compress large language models. *arXiv preprint arXiv:2403.19135*, 2024. Proposes LLM-Streamline.

Xinrui Chen, Hongxing Zhang, Fanyi Zeng, Yongxian Wei, Yizhi Wang, Xitong Ling, Guanghao Li, and Chun Yuan. Prune&Comp: Free lunch for layer-pruned LLMs via iterative pruning with magnitude compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026b. arXiv:2507.18212.

Wenyan Cong, Yiqing Liang, Yancheng Zhang, Ziyi Yang, Yan Wang, Boris Ivanovic, Marco Pavone, Chen Chen, Zhangyang Wang, and Zhiwen Fan. E3D-Bench: A benchmark for end-to-end 3D geometric foundation models. *arXiv preprint arXiv:2506.01933*, 2025.

Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. VGGT-Long: Chunk it, loop it, align it – pushing VGGT's limits on kilometer-scale long RGB sequences. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3D awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21795–21806, 2024.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations (ICLR)*, 2020.

Xin Fei, Wenzhao Zheng, Yueqi Duan, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Jiwen Lu. Driv3R: Learning dense 4D reconstruction for autonomous driving. *arXiv preprint arXiv:2412.06777*, 2024.

Weilun Feng, Haotong Qin, Mingqiang Wu, Chuanguang Yang, Yuqi Li, Xiangqi Li, et al. Quantized visual geometry grounded transformer. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.21302.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012.

Klaus Greff, Rupesh K. Srivastava, and Jürgen Schmidhuber. Highway and residual networks learn unrolled iterative estimation. In *International Conference on Learning Representations (ICLR)*, 2017.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations (ICLR)*, 2025.

Cristian Hinostroza, Rodrigo Toro Icarte, Christ Devia, Andres Carvallo De Ferari, Eugenio Herrera-Berg, Denis Parra, and Jorge F. Silva. Rethinking layer relevance in large language models beyond cosine similarity. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2605.14075.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. DynaBERT: Dynamic BERT with adaptive width and depth. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Stanisław Jastrzębski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. In *International Conference on Learning Representations (ICLR)*, 2018.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, et al. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*, 2026. arXiv:2509.13414.

Steven Kolawole, Lucio M. Dery, Jean-Francois Kagy, Virginia Smith, Graham Neubig, and Ameet Talwalkar. Everybody prune now: Structured pruning of LLMs with only forward passes. *arXiv preprint arXiv:2402.05406*, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. STream3R: Scalable sequential 3D reconstruction with causal transformer. *arXiv preprint arXiv:2508.10893*, 2025.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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

Haotang Li, Zhenyu Qi, Shaohan Henry Wang, Kebin Peng, Zi Wang, Qing Guo, Sen He, and Huanrui Yang. PaceVGGT: Pre-alternating-attention token pruning for visual geometry transformers. *arXiv preprint arXiv:2605.08371*, 2026.

Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards streaming perception. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Ji Liu, Dehua Tang, Yuanxian Huang, Li Zhang, Xiaocheng Zeng, Dong Li, et al. UPDP: A unified progressive depth pruner for CNN and vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. arXiv:2401.06426.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 6035–6044, 2020.

Xuanyi Liu, Chunan Yu, Deyi Ji, Qi Zhu, Lingyun Sun, Xuanfu Li, Jin Ma, Tianrun Chen, and Lanyun Zhu. StreamCacheVGGT: Streaming visual geometry transformers with robust scoring and hybrid cache compression. *arXiv preprint arXiv:2604.15237*, 2026.

Yao Lu, Hao Cheng, Yujie Fang, Zeyu Wang, Jiaheng Wei, Dongwei Xu, Qi Xuan, and Zhaowei Zhu. Reassessing layer pruning in LLMs: New insights and methods. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2411.15558.

Ji Ma, Wei Suo, Peng Wang, and Yanning Zhang. Short-LVLM: Compressing and accelerating large vision-language models by pruning redundant layers. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2025. arXiv:2507.23362.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics (ACL)*, 2025. arXiv:2403.03853.

Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: Generalizing lottery ticket initializations across datasets and optimizers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? Uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations (ICLR)*, 2021.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10901–10911, 2021.

Weining Ren, Xiao Tan, and Kai Han. Speed3R: Sparse feed-forward 3D reconstruction models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Findings*, 2026. arXiv:2603.08055.

Thomas Schöps, Johannes L Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. FastVGGT: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

Boyu Shi, Chang Liu, ChuanBao Gao, Xu Yang, and Xin Geng. Understanding performance collapse in layer-pruned large language models via decision representation transitions. *arXiv preprint arXiv:2605.07271*, 2026.

Dmitriy Shopkhoev, Ammar Ali, Magauiya Zhussip, Valentin Malykh, Stamatios Lefkimmiatis, Nikos Komodakis, and Sergey Zagoruyko. ReplaceMe: Network simplification via depth pruning and transformer block linearization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.02819.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Zhijian Shu, Cheng Lin, Tao Xie, Wei Yin, Ben Li, Zhiyuan Pu, et al. LiteVGGT: Boosting vanilla VGGT via geometry-aware cached token merging. *arXiv preprint arXiv:2512.04939*, 2025.

Oliver Sieberling, Denis Kuznedelev, Eldar Kurtic, and Dan Alistarh. EvoPress: Accurate dynamic model compression via evolutionary search. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. arXiv:2410.14649.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. SLEB: Streamlining LLMs through redundancy verification and elimination of transformer blocks. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

Xianbing Sun, Zhikai Zhu, Zhengyu Lou, Bo Yang, Jinyang Tang, Liqing Zhang, He Wang, and Jianfu Zhang. AVGGT: Rethinking global attention for accelerating VGGT. *arXiv preprint arXiv:2512.02541*, 2025.

Zecheng Tang, Jiaye Fu, Qiankun Gao, Haijie Li, Yanmin Wu, Jiaqi Zhang, Siwei Ma, and Jian Zhang. Spark3R: Asymmetric token reduction makes fast feed-forward 3D reconstruction. *arXiv preprint arXiv:2605.06270*, 2026.

Andreas Veit, Michael J. Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a. arXiv:2503.11651.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b. arXiv:2501.12387.

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

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. SkipNet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 409–424, 2018.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2507.13347.

Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S. Davis, Kristen Grauman, and Rogerio Feris. BlockDrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8817–8826, 2018.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2246–2251, 2020.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. BERxiT: Early exiting for BERT with better fine-tuning and extension to regression. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 91–104, 2021.

Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000 images in one forward pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a. arXiv:2501.13928.

Yantai Yang, Yuhao Wang, Zichen Wen, Zhongwei Luo, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. EfficientVLA: Training-free acceleration and compression for vision-language-action models. *arXiv preprint arXiv:2506.10100*, 2025b.

Jinhao You, Shuo Lyu, Zhuohang Lyu, Tanxuan Li, Zibo Zhao, Jiaxiang Hu, Kai Tang, and Yichen Guo. RegimeVGGT: Layer-wise spatially preserving redundancy removal for visual geometry grounded transformer. *arXiv preprint arXiv:2606.18439*, 2026. Concurrent work (June 2026).

Qianhao Yuan, Qingyu Zhang, Yanjiang Liu, Jiawei Chen, Yaojie Lu, Hongyu Lin, Jia Zheng, Xianpei Han, and Le Sun. ShortV: Efficient multimodal large language models by freezing visual tokens in ineffective layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2504.00502.

Haoyu Zhang, Zeyu Zhang, Zedong Zhou, Yang Zhao, and Hao Tang. Lite3R: A model-agnostic framework for efficient feed-forward 3D reconstruction. *arXiv preprint arXiv:2605.11354*, 2026.

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. arXiv:2410.03825.

Yang Zhang, Yawei Li, Xinpeng Wang, Qianli Shen, Barbara Plank, Bernd Bischl, Mina Rezaei, and Kenji Kawaguchi. FinerCut: Finer-grained interpretable layer pruning for large language models. *arXiv preprint arXiv:2405.18218*, 2024. Presented at the NeurIPS 2024 Workshop on Machine Learning and Compression.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4D visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025. Known as StreamVGGT.

NOTATION

Symbols of the main text, with where each is defined. Symbols that appear in a single appendix are defined where they are used.

Symbol	Meaning	Defined in
a, c	an architecture; a cell, one (architecture, sequence) pair	Eq. 3, §5.2
ℓ, N	a block; the number of bypassable block instances (Table 2)	§3
K, K/N	the number of bypassed blocks; the bypass ratio	§3
K_i, K_{i+1}	adjacent budgets of the swept K grid	§5.2
V	frames per window, or per forward pass in §6.1	§3
B_ℓ, $\mathbf{u}_\ell$	block ℓ, $B_\ell(\mathbf{x}) = \mathbf{x} + \mathbf{u}_\ell(\mathbf{x})$; its residual update	§3
f, f_S	the unpruned model; the model with every block of S bypassed	§3
W, w, t	a sequence's 30 windows; one window; a token	Eq. 2
$\mathbf{x}^{\text{in}}_{\ell,t}$, $\mathbf{x}^{\text{out}}_{\ell,t}$	block ℓ's input and output at token t	Eq. 2
BI_ℓ	the block-influence score	Eq. 2
S, S_K, S^{orc}	a set of blocks; the K blocks of lowest BI_ℓ; the oracle's set	Eq. 2, §4.2
D_ℓ, D_S	the damage of bypassing block ℓ alone; of bypassing S at once	§4.2, Eq. 3
d, d^*, $\hat{s}$	predicted depth; sensor depth; the median scale aligning them	Eq. 1
$\langle\cdot\rangle$, $\langle\cdot\rangle_{t\in w}$	the mean over one window's valid pixels; over the tokens block ℓ processes in window w	Eq. 1, Eq. 2
$\mathbf{1}[\cdot]$	the indicator, 1 when its argument holds and 0 otherwise	Eq. 1
AbsRel, δ_1	mean absolute relative error; fraction of pixels within $1.25\times$; each per window, then averaged over the cell's windows	Eq. 1
$\text{AbsRel}_{\text{rel}}$, $\delta_{1,\text{rel}}$	the pruned value over the unpruned value on the same cell	Eq. 1
PASS, $\text{PASS}(c, S)$	the acceptance test; its verdict (1 pass, 0 fail) for set S on cell c	Eq. 1, §5.2
$\text{pass}_a(K)$	the fraction of architecture a's 26 sequences that pass at K	Eq. 3
pass_{BI}, pass_{orc}	the pass rates of block-influence and of the oracle over the same cells	§4.2
$\text{Cap}(a)$	capacity, the largest K/N with $\text{pass}_a(K) \geq 0.8$	Eq. 3
$\Delta(S)$	non-additivity, $D_S - \sum_{\ell\in S} D_\ell$	Eq. 3
$\text{ATE}(f)$, ATE_{rel}	the Sim3-aligned camera-centre RMSE; $\text{ATE}(f_S)/\text{ATE}(f)$	§6.2
ρ, p	a Spearman rank correlation; a p-value	throughout
n	a sample size: the architectures a correlation runs over (§5.3), or the blocks, sets, cells or images a statistic runs over where stated	throughout

A HOW A DEPTH MAP BECOMES A PASS

Eq. 1 is counted per window, over the pixels a sequence's sensor depth actually covers. A pixel is valid when the sensor value is neither the dataset's zero nor its saturation code and the prediction is finite and positive; a window with fewer than 500 valid pixels returns no metric. The median ratio $\hat{s}$ makes both metrics invariant to any constant rescaling of the sensor depth, which is why 7-Scenes (millimetres) and TUM (a $5000\times$ fixed point) need no per-dataset branch, and why metric scale itself is untested. Everything above is `depth_metrics()` in `research/scripts/server/_phaseF1_fixedset.py`, the same function every reported cell was measured with, including the qualitative renders.

The block-influence score. For any nonzero $\mathbf{v}_1, \mathbf{v}_2$, $1 - \cos\angle(\mathbf{v}_1, \mathbf{v}_2) = \frac{1}{2}\left\|\mathbf{v}_1/\|\mathbf{v}_1\| - \mathbf{v}_2/\|\mathbf{v}_2\|\right\|^2$, so Eq. 2 is half the mean squared chord between a block's normalised input and output tokens. When the update is orthogonal to the stream, $\cos\angle(\mathbf{x}, \mathbf{x}+\mathbf{u}) = (1 + \|\mathbf{u}\|^2/\|\mathbf{x}\|^2)^{-1/2}$, hence

$$1 - \cos\angle(\mathbf{x}, \mathbf{x}+\mathbf{u}) = \frac{1}{2}\,\frac{\|\mathbf{u}\|^2}{\|\mathbf{x}\|^2} + O\Big(\frac{\|\mathbf{u}\|^4}{\|\mathbf{x}\|^4}\Big),$$

the approximation quoted in §3. The released per-cell files store the score as the harness computed it: one minus the token-mean cosine summed over a block's calls and divided by the number of windows. That is Eq. 2 wherever a block runs once per window. CUT3R's decoder runs once per frame, $V{=}4$ times per window, so its stored field is $1 - V\bar{c}_\ell = V\,\mathrm{BI}_\ell - (V-1)$, with $\bar{c}_\ell$ the mean cosine, and it is negative; the map is increasing, so the ranking, every selected set and every reported number are those of Eq. 2.

Where the two constants come from. Neither is fitted. The 1.25 inside δ_1 is that metric's standard threshold; the 1.25 and 0.95 of Eq. 1 are constants of the protocol rather than quantities estimated from it, and the downstream study of Appendix M reused them unchanged, its pre-registration recording the bound as the benchmark's own constant. What the paper reports about them is a sweep: §4.1 moves both over a $3{\times}3$ grid and Appendix E carries the capacity ordering across all nine settings.

B SELECTOR PASS RATES

Fig. 7 repeats the construction of Fig. 5 on a second VGGT cell, from TUM.

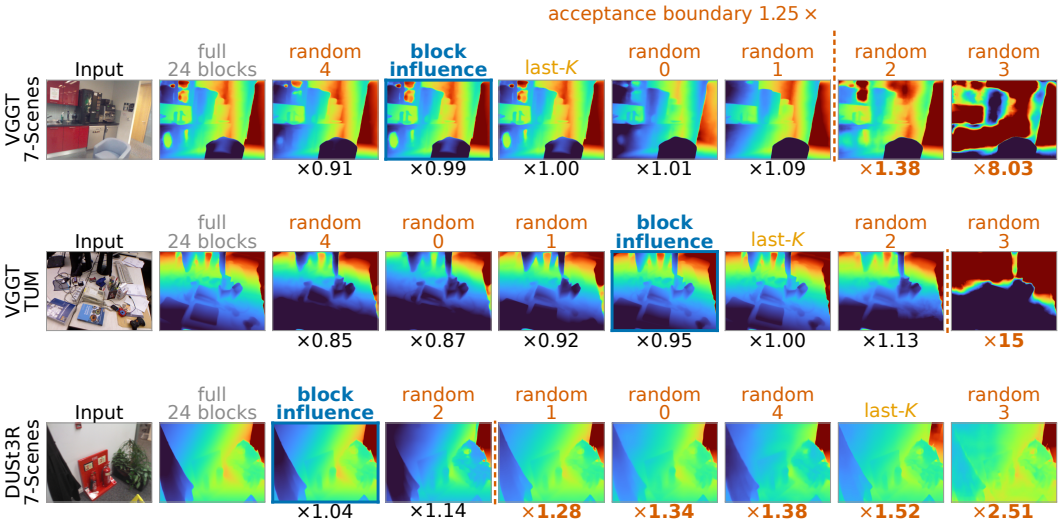

Figure 7: **Figure 5 with a second VGGT cell, on the other dataset.** Same construction: seven $K{=}5$ sets, ordered by cell-level relative AbsRel, boundary drawn per row. Cells were fixed by rule before rendering, the median cell per dataset by block-influence relative AbsRel (DUSt3R's is its 7-Scenes median; VGGT's 7-Scenes cell ranks 13 of 23), and each image is the window whose block-influence ratio is closest to its cell mean. Over VGGT's 26 cells, $86/130$ (66%) cell×seed random draws pass. The TUM row is VGGT's median TUM cell and shows the same pattern as the 7-Scenes one, block-influence at $\times 0.95$ among six of the seven choices that pass. Two displayed windows disagree with their cell's verdict, because acceptance is a mean over ≈30 windows and a single window is not: TUM random-4 is a $\times 1.27$ window of a $\times 0.85$ cell, DUSt3R random-2 a $\times 1.37$ window of a $\times 1.14$ cell. The $\times 15$, $\times 8.03$ and $\times 2.51$ panels are one and the same set, $\{7, 10, 12, 15, 18\}$, which fails on all 26 VGGT cells and all 26 DUSt3R cells: a single choice of five blocks that no sequence and neither architecture survives.

The 12 misses at $K{=}5$. Fast3R on *fire*, *heads*, *redkitchen* and TUM; MonST3R on *redkitchen* $\times 2$ and *chess* $\times 2$; CUT3R on two of the three TUM trajectories (fr1/desk, fr2/xyz); MASt3R on *fire* and *chess*. π^3 passes all 26.

The random baseline, resampled. The 56% random row draws 5 block sets per block count (N=12/24/36) from a seeded generator (`default_rng(42)`), fixed across all 26 sequences and shared by depth-matched models, so it is 15 distinct sets in all, five each at depth 12/24/36, and its sampling unit is the set and not the cell. Its 1040 evaluations are therefore not 1040 independent draws, and Table 3's interval entry for it is the observed *range* of per-set pass rates (mean 54%, s.d. 29 points), not a Wilson interval: a randomly chosen K=5 set is anywhere between 4% and 100% safe. Five is thin against $\binom{24}{5}$=42504, which is why the downstream study redraws 20. We therefore redrew independently: 10 fresh sets per architecture (`default_rng(2026)`, seeded apart from the benchmark's 42; 80 sets in all, model-specific rather than depth-shared) evaluated on an 8-cell-per-architecture subgrid. They pass 67% (429/640), with per-set s.d. 36 points and a set-clustered 95% CI of $[59, 75]$. The original 15 sets restricted to those same 64 cells pass 59%, consistent within set-level error. Block-influence passes 63/64 (98%) there, 23 points above the resample band's upper edge.

The two further baselines, per architecture. SLEB-style iterative re-scoring passes 26/26 on VGGT, LingBot, π^3 and MASt3R, 23/26 on DUSt3R, 21/26 on Fast3R and MonST3R, and 4/26 on CUT3R (173/208). The position+magnitude prior passes 26, 25, 18, 23, 8, 19 and 21 of 26 on VGGT, LingBot, CUT3R, Fast3R, DUSt3R, MASt3R and MonST3R (140/182; no π^3 row: π^3 post-dates the FZD pool, and its released per-cell files mark the FZD-prior and CO3D-oracle arms `not_run`).

Does the score predict single-block damage? Cosine importance correlates weakly with removal damage in LLMs (Hinostroza et al., 2026). Set-level acceptance is what the benchmark measures; whether ShortGPT's score also ranks *individual* blocks by their bypass damage is a separate question, and here too the answer is mostly no. Per architecture, the Spearman correlation between the block-influence score (mean over the model's sequences) and the leave-one-out single-block relative AbsRel is $+0.56$ on VGGT (p=0.005) and $+0.58$ on π^3 (p<0.001), $+0.38$ on Fast3R (p=0.07), and within ±0.1 on LingBot, CUT3R, DUSt3R and MonST3R, with MASt3R at -0.37 (p=0.08). The score's 94% therefore does not rest on per-block damage prediction; it rests on the sets it picks composing well (§5.2), the same reason neither single-block oracle improves on it.

How much the per-sequence K=5 set varies. Block-influence is scored on each evaluation sequence's own frames, so the K=5 set can differ across sequences. Over the 26 sequences it takes 1–5 distinct values per model (VGGT 5, DUSt3R 5, π^3 4, MonST3R 4, MASt3R 3, LingBot 2, CUT3R 2, Fast3R 1); the modal set covers 58–100% of sequences and every pairwise Jaccard is ≥ 0.43, with all sets drawn from a pool of 5–8 blocks per model.

The fixed-set result. Whether that variability matters is a measurement, and we ran it. For every (architecture, sequence) cell we take the modal K=5 block-influence set of that architecture's *other* 25 sequences, a leave-one-sequence-out set that by construction never sees the sequence it is asked to prune, and evaluate it under the same acceptance test. On 155/208 cells that set is already the one the benchmark used, so the released value is the fixed-set value; the remaining 53 were run fresh with a harness derived from the benchmark script by patching only the config under evaluation (its baseline reproduces the benchmark's bit-for-bit). Result: 196/208 (94.2%), and the agreement is cell-for-cell: *zero* of the 53 cells where the sets differ change verdict (the other 155 are identical by construction). On those 53 cells the fixed set gives the lower relative AbsRel on 32 and the higher on 21, so the per-sequence re-scoring is not even directionally better. The per-architecture sets are in Table 4. The same holds one level up: the modal set of the sequences from the *other scenes* is identical to the leave-one-sequence-out set on all 208 cells, so the fixed set is chosen without seeing the scene, not only the sequence, and a scene-level cluster bootstrap over the eight scene groups puts the 94% at $[91, 98]$, next to the sequence-level Wilson $[90, 97]$. The practical reading is that a practitioner needs unlabeled frames from the *architecture*, not from the sequence, and that the 87% of the transferred prior (§4.2) prices the strictly harder case where even the architecture is unseen, inside the pool; on the one architecture outside it the prior's set passed 17/26 (Appendix D). Per architecture (VGGT, LingBot, CUT3R, Fast3R, DUSt3R, MASt3R, MonST3R) the prior passes 26, 26, 18, 23, 23, 19 and 23 of 26 and the CO3D-measured single-block oracle 26, 24, 1, 21, 20, 19 and 25, so the oracle's deficit is CUT3R's; it beats block-influence on MonST3R (25 vs. 22). The benchmark-measured oracle's paired difference against block-influence,

$+2.9$ points over the 208 cells, has bootstrap 95% intervals of $[-1.9, +7.7]$ with cells independent, $[-2.1, +7.8]$ with the eight scene groups as clusters and $[+0.0, +6.2]$ with architectures as clusters (`research/scripts/_oracle_paired_ci.py`).

Table 4: The $K{=}5$ sets. The modal block-influence set over the 26 sequences and how many sequences pick exactly it (the leave-one-sequence-out set of the text and the leave-one-scene-out set coincide on every cell); the FZD prior's set, its five highest-predicted-safety blocks, one per architecture; the CO3D-measured single-block oracle's set, each model's five blocks of lowest single-block bypass damage on the same 30 CO3D windows; and the mean per-cell Jaccard overlap between FZD and oracle (F∩O), FZD and block-influence (F∩B), and oracle and block-influence (O∩B). Pair models index their 24 branch instances. ${}^{*}\pi^{3}$ post-dates the prior's pool: its FZD set is the held-out prediction of Appendix D (17/26) and its oracle set, from the sweep run for that test, passes 26/26 when deployed (`phaseF9_co3doracle_pi3_*`). Script: `research/scripts/_audit_review_followups.py`.

model	modal block-influence set	covers	FZD set	oracle set	F∩O / F∩B / O∩B
VGGT	$\{2, 6, 7, 8, 18\}$	16/26	$\{3, 18, 19, 21, 22\}$	$\{18, 19, 20, 22, 23\}$	0.43 / 0.14 / 0.13
LingBot	$\{0, 2, 5, 10, 11\}$	16/26	$\{2, 17, 18, 19, 22\}$	$\{8, 11, 18, 20, 22\}$	0.25 / 0.11 / 0.16
CUT3R	$\{1, 7, 8, 9, 10\}$	23/26	$\{6, 7, 8, 9, 10\}$	$\{1, 4, 6, 9, 10\}$	0.43 / 0.67 / 0.43
${}^{*}\pi^{3}$	$\{5, 6, 12, 14, 16\}$	17/26	$\{12, 14, 17, 18, 34\}$	$\{12, 14, 15, 16, 20\}$	0.25 / 0.27 / 0.41
Fast3R	$\{6, 7, 9, 11, 17\}$	26/26	$\{9, 15, 17, 20, 21\}$	$\{9, 17, 18, 20, 22\}$	0.43 / 0.25 / 0.25
DUSt3R	$\{5, 8, 9, 10, 21\}$	15/26	$\{7, 8, 9, 10, 11\}$	$\{0, 1, 7, 8, 9\}$	0.43 / 0.51 / 0.31
MASt3R	$\{5, 7, 8, 17, 19\}$	23/26	$\{6, 7, 8, 9, 10\}$	$\{5, 6, 9, 10, 11\}$	0.43 / 0.25 / 0.11
MonST3R	$\{8, 9, 10, 19, 21\}$	19/26	$\{4, 6, 7, 9, 10\}$	$\{7, 8, 9, 10, 11\}$	0.43 / 0.23 / 0.41

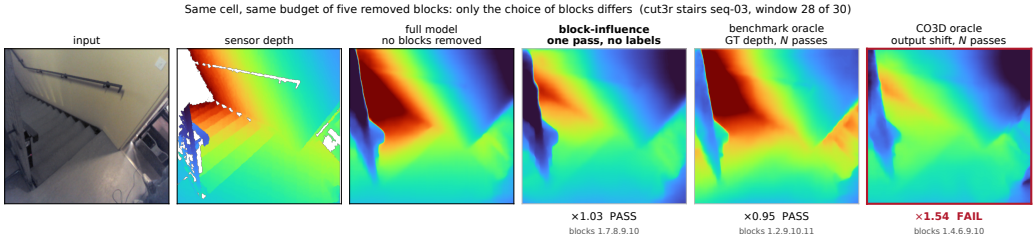

Figure 8: **The oracle gap is the instrument's, not the block choice's.** One cell, one budget of five removed blocks; only *which* five varies. CUT3R on 7-Scenes stairs/seq-03, that architecture's median cell by block-influence relative AbsRel, at the window closest to the cell mean, both fixed by rule before rendering as in Fig. 5. Labels are the cell-level relative AbsRel against the $1.25\times$ acceptance bound; white pixels in the sensor panel are invalid depth. Block-influence ($\times 1.03$, one unlabeled pass) and the oracle whose damage is measured on the benchmark ($\times 0.95$, N passes with ground-truth depth) are both accepted and neither is distinguishable from the full model by eye. The oracle whose damage is measured as an output shift on 30 CO3D windows ($\times 1.54$) flattens the stairwell. CUT3R is where that arm is worst, 1/26 cells against the benchmark-measured oracle's 24/26, and is also the architecture whose per-block ordering transfers worst between the two distributions ($\rho{=}-0.35$, Appendix E); it is drawn for that reason; the difference is atypical. Depth arrays are stored, so the panel re-plots without a GPU (`research/scripts/server/_qual_render_oracle.py`, `figures_src/gen_fig_oracle_qual.py`).

Fig. 9 gives the AbsRel side of Table 3 per architecture for four of its arms.

Iterative re-scoring and the two-feature prior, in detail. The SLEB-style variant re-scores the remaining blocks after each bypass (five rounds, same signal and windows as block-influence). It repairs MASt3R's toxic pair ($24 \to 26$ sequences) but collapses on the recurrent CUT3R ($24 \to 4$): after the first bypass the recurrent state is already corrupted, the influence signal measured on that state reads early blocks as safe (e.g. picking $\{1, 2, 4, 7, 9\}$ where one-shot correctly picks $\{1, 7, 8, 9, 10\}$), and AbsRel inflates 1.3–2.0×. Iterative re-scoring redistributes composition risk across architectures; it does not remove it. The fixed position+magnitude set (the ∼96%-of-FZD two-feature configuration, refit with the original pipeline and deployed per model) holds on most models but passes 8/26 on DUSt3R, whose bilateral free-zone pattern the cross-architecture pool lacks: the rank correlation the

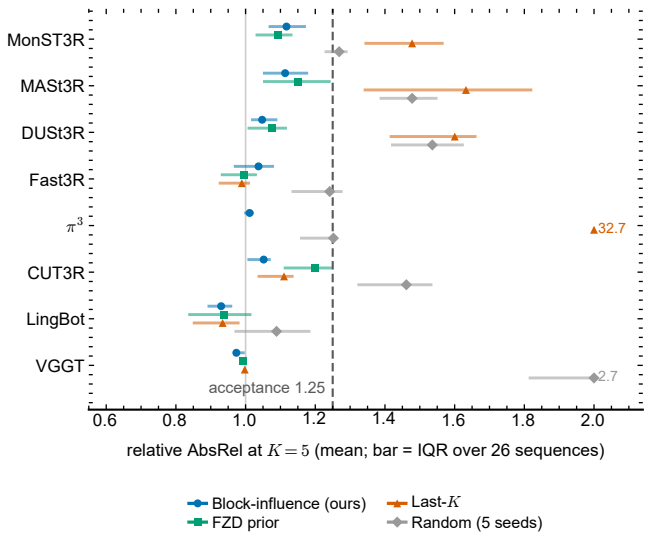

Figure 9: **The calibration-free selectors stay within the AbsRel side of the acceptance test where baselines break it.** Mean AbsRel relative to the full model at $K{=}5$ over the 26 benchmark sequences, all eight architectures; the joint pass rate is in Table 3. Block-influence stays at or below the $1.25\times$ line on all eight (0.93–1.12), the FZD prior on the seven it covers (0.94–1.20). Baselines exceed it on most; bars above 2.0 are clipped and labeled.

two features retain ($+0.51$ LOO) does not survive deployment on the outlier, which is what the seven learned FZD signals buy.

C POOL-RANDOM: IS IT THE SCORE, OR THE REGION?

The deployed selector passes 94% and a random five of N passes 56%, but that comparison cannot say whether the score's *ordering* carries the result or whether identifying a small safe region is enough. Registered before the run (`prereg_G5_poolrandom_control.md`, locked 2026-09-15): per cell, take the $P{=}8$ lowest block-influence blocks as a pool and draw six five-subsets from it, seeded, alongside the selector's own five in the same run so every cell is self-paired. The bands were locked with it: pooled pass rate $\geq 89\%$ would mean the ordering is not what carries the result, $\leq 80\%$ that it is, and anything between is reported as indeterminate. 208 cells, 1248 draws, no failed runs.

Registered verdict: indeterminate. The pooled rate is 81.6% (1018/1248), inside the band. The paired test is not ambiguous: the selector passes 197/208 cells (94.7%) against the draws' majority 160/208, the two disagree on 41, 39 of them in the selector's favour, exact McNemar $p < 0.001$.

Table 5: G5 (pre-registered), per architecture. *pool-random*: the pooled pass rate of the 6×26 draws from that architecture's eight lowest-influence blocks; *block-influence*: the selector's own five, measured in the same runs; *pool, by cell*: cells where a majority of the six draws passes. Capacity and N are Table 2's.

model	N	capacity	pool-random	block-influence	pool, by cell
π^3	36	0.39	**100.0%**	26/26	26/26
VGGT	24	0.67	**100.0%**	26/26	26/26
LingBot	24	0.50	96.8%	26/26	25/26
Fast3R	24	0.29	92.9%	23/26	24/26
MonST3R	24	0.21	87.2%	22/26	22/26
DUSt3R	24	0.29	84.6%	26/26	21/26
MASt3R	24	0.25	64.7%	24/26	15/26
CUT3R	12	0.42	**26.3%**	24/26	1/26

The pooled number is an average over a $3.8\times$ spread (Table 5). On π^3 and VGGT the score's ordering is worth nothing: any five of the eight safest do exactly what it does. On CUT3R it is worth almost

everything: the selector passes $24/26$ where a majority of random draws from its own safe pool passes $1/26$.

What orders that spread is how much of the model the pool covers. Spearman against capacity is $+0.47$ (exact permutation p=0.24). Against the pool's share of the model, P/N, it is -0.71 (p=0.07), which with P fixed is N read backwards; counting VGGT's and LingBot's frame blocks in N it is -0.88 (p=0.006), and counting pair branches once -0.89 (p=0.011). These were computed after the run and are exploratory; at n=8 only $|\rho| \gtrsim 0.72$ reaches $p < 0.05$ (`research/scripts/_g4r4_instrument_checks.py`). Our own stated expectation before the run — that a narrow safe region would make the ordering matter more — is not supported: MonST3R has the roster's smallest capacity and sits at 87%, while π^3 sits below CUT3R in capacity and at 100%. CUT3R is also the one architecture whose pool covers two thirds of the model (8 of 12 blocks against 8 of 24 or 36), so the fixed P does not pose an equally hard question to every architecture; that is a limitation of the design, registered as a reporting rule and left unrepaired.

A replication, incidentally. The block-influence arm here re-measures the benchmark's headline selector a month after the released run, on the same cells with the same code. Seven of the eight architectures reproduce cell for cell ($26/26$, $26/26$, $24/26$, $26/26$, $26/26$, $24/26$, $22/26$); the eighth is Fast3R at $23/26$ against $22/26$, and Fast3R is the architecture whose forward pass Appendix E reports as non-deterministic.

D Cross-architecture transfer of the FZD prior

The prior. For a held-out architecture a, let $\boldsymbol{\phi}_{a',\ell}$ be block ℓ's eight signals, z-scored within architecture a', together with its relative position $\ell/(N_{a'}-1)$, and $y_{a',\ell}$ its within-model rank of output-shift damage on the pool's 30 CO3D windows. Over the seven-model pool $\mathcal{A}$ the prior fits

$$(\hat{b}, \hat{\mathbf{w}}) = \arg\min_{b,\mathbf{w}} \sum_{a' \in \mathcal{A}\setminus\{a\}} \sum_{\ell} \left(y_{a',\ell} - b - \mathbf{w}^\top \boldsymbol{\phi}_{a',\ell}\right)^2 + \alpha\|\mathbf{w}\|^2, \qquad \alpha = 1,$$

predicts $\hat{y}_{a,\ell} = \hat{b} + \hat{\mathbf{w}}^\top \boldsymbol{\phi}_{a,\ell}$, and deploys the K blocks of lowest $\hat{y}_{a,\ell}$. Table 6 reports, per held-out a, the Spearman correlation between $\hat{y}_{a,\cdot}$ and the measured damage; the eight signals are listed under the size-matched control below.

Reading Figure 6. Its colours are this appendix's single-block bypass sweep, each block's output shift against the full model on the pool's 30 CO3D windows, ranked within its model; its triangles are the modal block-influence K=5 set of Table 4. The three pair-Siamese models measure 12 decoder positions where the benchmark bypasses 24 block instances, so their marks are drawn at position (index mod 12) with the fill showing which branch. Panel (b)'s pooled Spearman -0.39 runs over the 120 block positions of the seven-model pool, pair models once per decoder position; the 192 of Appendix H counts Table 2's instances. The trend is the output-shift instrument's: ranked by the benchmark's depth damage, the same 120 positions give $+0.00$. DUSt3R, which crosses the trend, is also the fold the prior finds hardest ($+0.252$, Table 6). π^3 is not drawn because its sweep was run later, for the held-out test below. The five bluest blocks of each row are exactly the CO3D-measured oracle's set, and the triangles use them on 17 of 35 slots (11.5 expected by chance), so the figure's closing contrast is Table 3's, restricted to the 182 cells all three arms cover: block-influence passes $170/182$ (93%), 19 points above that oracle's $136/182$ (75%) and level with the benchmark-measured oracle's $164/182$ (90%). The first gap is the instrument's (§4.2). That the benchmark-measured oracle, safest block by safest block, does no better than a set chosen without measuring damage fits the additivity failure of §5.2 at deployment, which Table 10 shows on the same depth instrument.

Held out of the pool entirely: π^3. π^3 post-dates the pool. The test was pre-registered on 2026-09-09 before any of its signals existed (`research/prereg/prereg_FZD_pi3_heldout.md`; thresholds fixed, analysis script written first and extended after the run for reporting, the lock self-declared, as for every pre-registration here): fit the prior once on all seven pool models, measure π^3's eight signals and single-block bypass damage on the pool's own 30 CO3D windows (`phase31_pi3_perlayer_*`), predict its 36 decoder blocks, deploy the five predicted-safest on all 26 cells (`phaseF7_fzdset_pi3_*`). **H1, the ranking transfers** ($\rho \geq +0.40$): supported,

Table 6: Leave-one-model-out Spearman correlation between the predicted per-block ranking of each held-out architecture and its measured single-block bypass damage, with no target calibration. *Output shift* is the instrument the prior is fit to, the bypassed model's output against the full model on the pool's 30 CO3D windows; DUSt3R is its outlier (a bilateral free-zone pattern absent from the training pool), and a quadratic-position-augmented variant raises its mean to $+0.567$ (we report the simpler model). **Depth damage* is the benchmark's instrument, single-block leave-one-out relative AbsRel against sensor depth, pair models' two branches averaged per decoder position; this column was computed after the fact and is exploratory (`research/scripts/_g4r4_instrument_checks.py`). The prior transfers on the instrument it is fit to and barely on depth. The last row is the pre-registered held-out test of the paragraph below: π^3 never enters the pool and the prior is fit on all seven.

			LOO ρ against	
Model	N	Paradigm	output shift	depth damage*
Fast3R	24	multiview	$+0.630$	$+0.373$
MonST3R	24	pair	$+0.615$	-0.189
LingBot	24	streaming	$+0.560$	-0.005
CUT3R	12	recurrent	$+0.552$	-0.217
MASt3R	24	pair	$+0.538$	-0.021
VGGT	24	multiview	$+0.536$	$+0.513$
DUSt3R	24	pair	$+0.252$	$+0.210$
Mean	—	—	$\mathbf{+0.53}$	$+0.09$
π^3, held out of the pool	36	multiview	$+0.690$	$+0.454$

$\rho{=}+0.690$ ($p{=}3\times10^{-6}$) against the same-instrument damage, above every pool fold in Table 6, and $+0.454$ ($p{=}0.005$) against the benchmark's leave-one-out AbsRel, below only VGGT's fold. **The registered blindness prediction fails.** The two-feature position+magnitude prior reaches $+0.361$ and -0.159 on the same two targets, not within 0.1 of the full prior as the 96% figure predicted; on π^3's alternating decoder the other seven signals carry half of the transfer. **H2, the set deploys** ($\geq$ $21/26$): falsified, $17/26$, median relative AbsRel 1.20, the three TUM cells at 1.7–$2.1\times$, δ_1 never the binding metric. The release's controls bracket it: block-influence $26/26$ (median 1.01), its fixed set $9/9$ where run, random $K{=}5$ sets 3, 8, 17, 26 and 26 of 26 (mean 16), last-K $0/26$; the prior's set sits at the random mean. The loss is one block. The prior ranks block 34 of 36 (indices from 0; the penultimate) safest (predicted damage rank 0.21 against 0.39–0.42 for $\{12, 14, 17, 18\}$), extrapolating the pool's late-block safety; on π^3 bypassing that block shifts the output by 1.16 times the unpruned output's standard deviation on the pool's instrument, ninth most damaging of 36, and it is fourth most damaging under the benchmark's, where the other four blocks read 0.99–1.00. Exploratory, run after the verdict and outside the pre-registration: the same set with block 34 replaced by the next-ranked block 16 passes $26/26$ at a median relative AbsRel of 0.99 (`phaseF7b_fzdminus34_pi3_*`). A second exploratory check locates the miss: refit without the position feature, the prior still ranks π^3 at $\rho{=}+0.67$ and moves block 34 from first to nineteenth of 36 (its five predicted-safest become $\{4, 6, 12, 14, 15\}$, not deployed), so the miss is the position feature extrapolating the pool's late-block safety, not the eight signals. The pool licenses no late-block rule in its place: VGGT's last block is its safest and its penultimate block its fifth-safest of 24, and the penultimate blocks of LingBot and Fast3R are their fourth-safest. The abstract, §1 and §4.2 therefore scope the prior's claim to the seven-model pool and state the held-out result; the demotion is recorded in Appendix K. Script: `research/scripts/_fzd_pi3_heldout.py`; the π^3 signal and bypass adapter is `research/scripts/server/_phase31_pi3.py`.

Size-matched control: what the pool buys. The main text once read the gap between the leave-one-model-out $+0.53$ and a within-model cross-validation as showing that the cross-architecture pool is *necessary*. Those two fits differ in training size as well as in pool composition, so we ran the control that separates them, with the same recipe as Table 6 (eight signals from one forward pass over 30 CO3D windows per model, namely update magnitude $\|\mathbf{u}_\ell\|/\|\mathbf{x}_\ell\|$, parallelism $\cos(\mathbf{u}_\ell, \mathbf{x}_\ell)$ for block input $\mathbf{x}_\ell$ and residual update $\mathbf{u}_\ell$, norm growth, effective rank, top-singular share and shape diagnostics, z-scored within model, plus position, within-model damage rank as the training target, ridge $\alpha{=}1$, Spearman against the held-out model's raw damage; the script reproduces every stored per-fold ρ to three decimals). Per held-out model, training on (A) the full pool of the other six, (B) a

random subset of that pool the size of half the held-out model's blocks, and (C) half of the held-out model's own blocks, each of B and C averaged over 300 draws:

held out	n	A full pool	B matched pool	B′ matched, same half	C within-model
CUT3R	12	$+0.55$	$+0.13$	$+0.11$	-0.10
DUSt3R	12	$+0.25$	$+0.09$	$+0.10$	$+0.51$
Fast3R	24	$+0.63$	$+0.22$	$+0.20$	$+0.34$
LingBot	24	$+0.56$	$+0.07$	$+0.07$	$+0.00$
MASt3R	12	$+0.54$	$+0.11$	$+0.11$	$+0.24$
MonST3R	12	$+0.62$	$+0.06$	$+0.05$	$+0.18$
VGGT	24	$+0.54$	$+0.12$	$+0.10$	$+0.33$
mean		$+0.53$	$+0.11$	$+0.11$	$+0.21$

At matched size the cross-model pool is no better than the model's own blocks; it is worse on five of seven, and B′, which tests on the same held-out half as C, makes that a like-for-like contrast. The pool's advantage is that it is 8–18× larger, and other architectures are the only source of that many damage-labeled blocks that costs the target no labels. Script: `research/scripts/_audit_fzd_size_matched.py`.

E Robustness details: thresholds, distribution shift, and out-of-domain transfer

Determinism. Re-running 104 (cell, K) block-influence measurements for Appendix G reproduced the released window-mean AbsRel to 10^{-5} on VGGT, LingBot, CUT3R, DUSt3R, MASt3R and MonST3R (π^3 has no reversal to re-run). Fast3R did not: its *unpruned* baseline moved by up to 0.0036 AbsRel between runs, the same block set by 0.001–0.0035, and at 3 of its 22 re-run endpoints two near-tied blocks swapped in the influence ranking, once moving AbsRel by 0.038 ($0.195 \to 0.157$ at $K{=}10$ on heads seq-02). Fast3R's selected sets and verdicts are therefore reproducible only up to this run-to-run jitter of its forward pass; the other six re-run architectures' are exact. Every number in this paper is from the released files, which record one realisation.

Distribution shift. Beyond the threshold-grid robustness of §4.1, we tested whether the per-block safety ordering itself survives a change of *data distribution*. The measured object is the *per-block bypass-damage vector*, which records for each block the damage of bypassing it alone against the full model. No selector is involved. We compute it per architecture on CO3D (object-centric) (Reizenstein et al., 2021), TUM-RGBD, and 7-Scenes (indoor) with one byte-identical sweep in which only the data loader changes, then correlate it across distributions (Table 7). The sweep covers the seven original models, as π^3 has no cross-dataset per-block sweep.

The ordering is mostly preserved under distribution shift, at a median full-vector correlation of $+\mathbf{0.62}$, with 71% of comparisons at or above $+0.40$. The boundary is recurrence and streaming. Both state-carrying models reorder ($2/2$: streaming-KV LingBot, recurrent CUT3R), while every feedforward and bidirectional model preserves the ordering in the mean ($5/5$), with means from $+0.40$ to $+0.87$ and a median of $+0.75$ over their 15 comparisons; the $+0.40$ is DUSt3R's, which, like LingBot, reorders against object-centric CO3D.

Out-of-domain pruning transfer (outdoor). In a pre-registered out-of-domain study on the four ETH3D scenes (Schöps et al., 2017), with $K{=}5$ re-scored on target frames over the seven original models, $4/7$ pass every scene. VGGT misses only far-range *terrace* ($1.33\times$), MonST3R misses *meadow* at the δ_1 margin, and the recurrent CUT3R fails all four. An apparent failure on KITTI (Geiger et al., 2012) is most consistent with a sparse-LiDAR ground-truth measurement artifact, under pre-registered controls and a self-consistency analysis; the full controls and the per-scene outcomes follow below.

How much of capacity is the candidate pool? K/N divides by the blocks the sweep could bypass. VGGT and LingBot expose their aggregators' 24 global blocks (the hook is the aggregator's global block list) and never the 24 frame-attention blocks beside them; π^3 exposes both its frame and global

Table 7: Cross-distribution stability of the per-block output-shift ranking (descriptive; the sweep's registered hypothesis was killed and is in the released kill list): full-vector Spearman correlation of the per-block bypass-damage vector between dataset pairs (means computed from unrounded correlations). Median over the 21 architecture×dataset-pair comparisons is $+0.62$ ($71\% \geq +0.40$). The two non-transferring models are the non-feedforward, state-carrying ones (streaming LingBot, recurrent CUT3R); the five feedforward/bidirectional models preserve the ordering in the mean. LingBot and DUSt3R are indoor-stable (TUM-7S $+0.76$ / $+0.91$) and reorder mainly against object-centric CO3D.

arch	CO3D-TUM	CO3D-7S	TUM-7S	mean
VGGT	$+0.85$	$+0.80$	$+0.94$	$\mathbf{+0.87}$
Fast3R	$+0.75$	$+0.77$	$+0.92$	$+0.81$
MonST3R	$+0.57$	$+0.87$	$+0.62$	$+0.69$
MASt3R	$+0.55$	$+0.55$	$+0.74$	$+0.62$
DUSt3R	$+0.15$	$+0.15$	$+0.91$	$+0.40$
LingBot	-0.08	-0.08	$+0.76$	$+0.20$
CUT3R	$+0.41$	-0.35	$+0.15$	$+0.07$

decoder blocks; the pair models expose 24 decoder instances. Counting the frame-attention blocks in N (they were never bypassed, so every capacity K is unchanged) moves the ratios, the ordering and the structural correlations of §5.3:

pool	VGGT	LingBot	spread	top	params ρ	CKA ρ
Table 2 (N=24)	0.67	0.50	3.2×	VGGT	$+0.87$ (p=0.007)	$+0.61$ (n.s.)
VGGT N=48	0.33	0.50	2.4×	LingBot	$+0.73$ (0.047)	$+0.61$ (n.s.)
VGGT and LingBot N=48	0.33	0.25	2.0×	CUT3R	$+0.42$ (n.s.)	$+0.95$ (0.001)

The other six capacities do not move (p: exact permutation, n=8; both aggregators, VGGT's and LingBot's, alternate 24 frame-attention and 24 global blocks, and both are hooked at the global list). What survives every pool is that capacity varies at least twofold and that no structural scalar is significant under all three; what does not survive is any single correlate, parameter count under the first pool and inter-block CKA (centred kernel alignment) under the third, so the verdicts of §5.3 hold for Table 2's pool and the pre-registered redundancy test is not decisive. The VGGT-vs-Fast3R contrast, 2.3× at 24 exposed blocks each, is 1.1× once VGGT's frame blocks count. Script: `research/scripts/_g4_artifact_checks.py`.

How much of capacity is threshold and sampling noise? Capacity is a threshold ($\geq 80\%$ acceptance) on a threshold (the joint criterion) over 26 binary outcomes, so we quantify both sources of arbitrariness from the released per-cell results at no additional compute. *Sampling.* Resampling the 26 sequences with replacement (2000 resamples) and recomputing every capacity per resample gives 90% percentile intervals VGGT 0.67 $[0.67, 0.67]$, LingBot 0.50 $[0.33, 0.50]$, CUT3R 0.42 $[0.42, 0.58]$, π^3 0.39 $[0.39, 0.39]$, DUSt3R 0.29 $[0.21, 0.42]$, Fast3R 0.29 $[0.12, 0.42]$, MASt3R 0.25 $[0.21, 0.25]$, MonST3R 0.21 $[0.17, 0.33]$. Twenty of the 28 pairwise orderings hold in $\geq 90\%$ of resamples. The eight that do not all join overlapping intervals: five inside the four-model low-capacity cluster (0.21–0.29), π^3 against DUSt3R and Fast3R, and LingBot against CUT3R (`research/scripts/analyze_capacity_bootstrap.py`). The headline spread is itself the bootstrap median: $3.2\times$, 90% interval $[2.7\times, 5.3\times]$. *Thresholds.* Recomputing capacity over the 3×3 acceptance grid of §4.1 crossed with pass cutoffs $\{70, 80, 90\}\%$ (Table 8 lists the eleven points on the default's two axes; `research/scripts/analyze_capacity_sensitivity.py` prints all 27) leaves the ordering in place: Spearman against the default ordering has median ρ=$+0.90$ (min $+0.55$, $16/27$ grid points at $\geq +0.80$), VGGT is strictly the highest-capacity architecture at 25 of 27 grid points (tied at one, lower at one), and the cross-architecture spread, max over min non-zero capacity, is 1.9–8.0× (median 4.0×; the zero capacities, MASt3R at the 1.10× points, sit out of the minimum), so the reported 3.2× sits at the conservative end. What is *not* robust is any individual value: at the strictest grid point MASt3R and CUT3R fall to 0.00 and 0.08. What survives the grid is the claim we make, that capacity varies several-fold across architectures and cannot be read off structure. A specific capacity number is a property of a stated acceptance test, not of the architecture alone.

Table 8: Capacity under alternative acceptance thresholds (sensitivity analysis, not pre-registered), recomputed from the released per-cell results. Rows 1–9: the 3×3 threshold grid at the default $\geq 80\%$ pass cutoff (* marks the paper's setting). Rows 10–11: the default thresholds at pass cutoffs 70% and 90%. ρ is the Spearman correlation of that row's architecture ordering with the default ordering.

AbsRel / δ_1	VGGT	LingBot	CUT3R	π^3	DUSt3R	Fast3R	MASt3R	MonST3R	ρ
1.10 / 0.90	0.67	0.42	0.42	0.39	0.17	0.29	0.00	0.17	+0.94
1.10 / 0.95	0.67	0.42	0.42	0.39	0.17	0.21	0.00	0.17	+0.94
1.10 / 0.98	0.67	0.42	0.08	0.39	0.17	0.12	0.00	0.17	+0.62
1.25 / 0.90	0.67	0.50	0.50	0.39	0.29	0.42	0.25	0.29	+0.90
1.25 / 0.95*	0.67	0.50	0.42	0.39	0.29	0.29	0.25	0.21	+1.00
1.25 / 0.98	0.67	0.50	0.42	0.39	0.29	0.17	0.17	0.17	+0.96
1.50 / 0.90	0.67	0.75	0.58	0.39	0.58	0.42	0.50	0.42	+0.64
1.50 / 0.95	0.67	0.50	0.58	0.39	0.42	0.42	0.50	0.29	+0.70
1.50 / 0.98	0.67	0.50	0.42	0.39	0.29	0.25	0.17	0.17	+0.99
pass $\geq 70\%$	0.67	0.50	0.50	0.39	0.42	0.42	0.25	0.29	+0.90
pass $\geq 90\%$	0.67	0.33	0.42	0.39	0.21	0.12	0.21	0.17	+0.81

The $K=5$ verdict on the threshold grid. On the 3×3 grid of Eq. 1's cutoffs, at least 6 of the 8 architectures still pass two thirds of their 26 cells at $K=5$ (§4.1). Block-influence passes per grid point (`research/scripts/analyze_k5_threshold_grid.py`):

AbsRel / δ_1	VGGT	LingBot	CUT3R	π^3	DUSt3R	Fast3R	MASt3R	MonST3R	$\geq 2/3$
1.10 / 0.90	26	26	21	26	20	21	11	12	6/8
1.10 / 0.95	26	26	21	26	20	21	11	12	6/8
1.10 / 0.98	26	26	19	26	20	20	11	12	6/8
1.25 / 0.90	26	26	24	26	26	23	24	24	8/8
1.25 / 0.95*	26	26	24	26	26	22	24	22	8/8
1.25 / 0.98	26	26	21	26	25	20	19	18	8/8
1.50 / 0.90	26	26	26	26	26	26	26	26	8/8
1.50 / 0.95	26	26	26	26	26	25	25	24	8/8
1.50 / 0.98	26	26	23	26	25	22	19	20	8/8

Only MASt3R and MonST3R fall below two thirds, and only at the strictest AbsRel cutoff; * marks Eq. 1, whose row sums to 196.

The TUM-weighted subset of §5.1. Chess seq-01–03 with the three TUM sequences, the six sequences of the original high-K evaluation. Under block-influence alone their capacities are VGGT 0.75, LingBot 0.33, CUT3R 0.08, π^3 0.44, DUSt3R 0.42, Fast3R 0.33, MASt3R 0.25 and MonST3R 0.21 (`figures_src/gen_fig0_teaser.py` prints them). On six sequences one flip moves a pass rate by 17 points. The retracted capacity numbers of Table 14 (row 1) came from the same six sequences with two selectors pooled; these do not.

Out-of-domain pruning transfer (outdoor), full detail. We pre-registered an out-of-domain study and ran it on KITTI (Geiger et al., 2012) (driving) and the four ETH3D (Schöps et al., 2017) scenes (courtyard/facade/meadow/terrace; near and far range, structured and open). Selection is re-scored on each OOD sequence's own frames, the same protocol as in-domain. Under the joint acceptance test, $4/7$ architectures (DUSt3R, Fast3R, LingBot, MASt3R) pass every scene; VGGT passes $3/4$, missing only far-range *terrace* (AbsRel $1.33\times$); MonST3R passes $3/4$, missing *meadow* at the δ_1 margin (0.94 vs. the 0.95 bound); CUT3R fails all four (AbsRel 1.4–$2.6\times$, δ_1 preserved). That is the same state-carrying model whose capacity collapses on TUM and which fails the cross-distribution ranking test above. We did not take the apparent KITTI *driving* failure at face value. Six alternative explanations were pre-registered and tested one at a time, and all six are refuted. It is not camera motion or baseline: the failure persists at minimal frame stride. It is not outdoor content: Fast3R and MASt3R fail KITTI yet pass every ETH3D scene. It is not aspect-ratio squashing: a square center-crop does not recover it. It is not depth range: far-range *facade* (6–28 m) and *meadow* (4–30 m) pass. It is not structure density: open, sparse *meadow* passes. And it is not the selector: on KITTI, Fast3R's per-block safety ranking correlates at $\rho=1.00$ with its in-domain ranking and selects an *identical* top-5 set, and still fails. The same blocks are chosen; only the verdict changes.

The one explanation we could not refute is measurement. A self-consistency analysis (pruned vs. full output, no ground truth) shows the pruned depth moving *less* on KITTI than on the ETH3D scenes the same model passes (7.7% vs. 15.1% for Fast3R; 37% vs. 47% for MASt3R): the output barely changes, and the relative-AbsRel test against sparse LiDAR turns that into a failure. CUT3R is the case the battery does not rescue. At a fraction-matched prune ratio ($K{=}3$ of its 12 blocks) it passes $22/26$ of its in-domain cells and $0/4$ on ETH3D, where all four scenes land at relative AbsRel 1.56–1.83 against the 1.25 bound, so its out-of-domain collapse is real and not a property of the metric.

These out-of-domain rates rest on few cells and on heavily overlapping windows, and we read them as directional. Each architecture is evaluated on all four ETH3D scenes and on three KITTI drives. Within a scene the harness slides a $V{=}4$ window one frame at a time (stride 1, step 1, capped at 30), so a scene of n images yields $\min(30, n{-}3)$ windows and consecutive windows share three of their four frames: 30 from courtyard's 38 images, 20 from terrace's 23, 12 from meadow's 15. The in-domain benchmark instead samples video at stride 4 and step 8, where consecutive windows share two frames of four, so an ETH3D cell carries less independent evidence than its window count suggests. The six refutations are not rate estimates and do not carry this caveat.

A second benchmark: all 13 ETH3D scenes. The out-of-domain evidence above is a spot check on four scenes. To put the selector claims on a second benchmark, we pre-registered and ran the full protocol on all 13 ETH3D scenes and all 8 architectures: 104 cells, $K{=}5$, selection re-scored per scene. Windows here are *disjoint* ($V{=}4$, stride 1, step 4), unlike the step-1 sliding windows used for the spot check above, so a scene of n images contributes $\lceil (n{-}3)/4 \rceil$ independent windows, 107 in total per architecture. The pre-registration fixed the hypotheses, the thresholds, the controls and the kill criteria before the run, and entered version control before any of the data existed.

Table 9: **The second benchmark, 104 cells.** Block-influence $K{=}5$ keeps $\geq 80\%$ of cells on 7 of 8 architectures, against a pre-registered threshold of ≥ 6 of 8. The random column is the pre-registered per-cell *mean* over its five seeds. Pooling individual draws as in Table 3 instead gives 0–91%, with VGGT at 38%: its mean is dragged past the bound by single draws of up to 22.7×. The italicised entry cuts against us: last-K beats block-influence on VGGT, reversing the in-domain order (49% against 94%). Random is the other caveat. It passes more draws here than in domain on Fast3R and the three pair models (66–91% vs. 37–69%), whose full-model ETH3D baselines sit at absolute AbsRel 0.24–0.27 against 0.02–0.05 for the other four, so the relative acceptance bound is looser for them and the test discriminates weakly. The four strong-baseline models move the opposite way (43–78% in domain to 0–49%).

	cell pass rate over 13 scenes		
Model	block-influence	random (5-seed mean)	last-K
π^3	**13/13**	0%	0%
Fast3R	**13/13**	92%	92%
MonST3R	**13/13**	92%	77%
VGGT	12/13	8%	*100%*
DUSt3R	12/13	77%	77%
MASt3R	12/13	62%	69%
LingBot	11/13	38%	77%
CUT3R	2/13	0%	8%

H1 confirmed. 7 of 8 architectures keep $\geq 80\%$ of cells (Table 9), over all 13 scenes and over the 11 with ≥ 5 windows. CUT3R at $2/13$ is the sole failure, the architecture that also reorders under distribution shift and collapses out of domain.

H2, the fixed set, keeps every per-architecture total on 7 of 8. One set per architecture, the modal $K{=}5$ set over its 13 cells with an outcome-blind lexicographic tie-break, applied to every scene with no re-scoring, matches per-scene re-scoring cell for cell (0.0 pp) on VGGT, LingBot, π^3, Fast3R, DUSt3R and MonST3R. On MASt3R the total matches ($12/13$ both ways) through two offsetting flips: courtyard passes only under re-scoring (δ_1 rel. 0.968 vs 0.935) and pipes only under the fixed set (1.204×, δ_1 0.814 vs 0.940×, 1.030). On CUT3R the fixed set does *better* ($3/13$ against $2/13$), a 7.7 pp gap, outside the pre-registered ±5 pp band. That band was mis-specified: one cell of thirteen is 7.7 pp, so it admits only exact matches. We report the miss and keep the band.

H3 splits between the two registered views. Block-influence beats both baselines on 7 of 8 architectures over all 13 scenes but on 5 of 8 over the 11-scene subset, against a threshold of ≥ 6. Both views were registered in advance so the split could not be settled by choosing one.

Controls. Every cell clears the full-model feasibility gate, so no architecture is excluded for being unable to do this domain. The windowing change is not what moved the results: of the 28 cells that exist under the step-1 setting, 2 reverse their verdict under step 4, 7.1% against a 20% bound, both on the thinnest scene and both fail→pass. The sparsity control, however, **fails**. Its logic is the one that diagnosed KITTI: a cell whose output barely moves yet whose ground-truth verdict says fail is being judged by the ground truth rather than by the pruning. On 9 of the 16 failing cells the GT-free self-consistency sits below that architecture's passing median, a KITTI-pattern fraction of 0.56 against a pre-registered bound of 0.25. Seven of the nine are CUT3R, but the control fires without it (2 of 5, 0.40). ETH3D ground truth covers 18.5% of pixels (6.1–35.8% by scene) against 84.4% on 7-Scenes and 76.3% on TUM, with KITTI at 4.0%. We therefore read this benchmark as a second one, not a stronger one, and the registration commits us to a complete-coverage arm before the sparsity question can be called settled. That arm ran; Appendix F reports it.

Greedy selector ceiling. The document that pre-registered the out-of-domain study also fixed a labeled ceiling for the selector before the run: greedy forward selection on the first third of each cell's windows, minimising mean AbsRel directly, scored on the last third (block-influence is computed on the same first third) on nine cells ({VGGT, MASt3R, CUT3R} × {chess seq-01, office seq-01, TUM fr1/desk}) at K=5. Both selectors pass all nine on those held-out windows, so block-influence is safe at this budget without labels. Greedy still buys a margin: the mean held-out relative AbsRel gap is −0.091, concentrated on MASt3R, where block-influence scores ×1.00/1.22/0.95 against greedy's ×0.84/0.97/0.87 and the office cell sits just inside the 1.25 bound. On VGGT the gap is small. Greedy is itself a heuristic, so the true optimum is at least this far above block-influence. Both outcomes were pre-committed.

F TWO COMPLETE-COVERAGE BENCHMARKS: RESULTS AND CAUSE ANALYSIS

Both arms were registered two-sidedly (`prereg_E2_tartanair_complete_coverage.md`, `prereg_E3_scannet_third_benchmark.md`) after ETH3D's sparsity control fired and before any frame of either dataset met any model. Protocol, acceptance test, feasibility gate and selector arms are byte-inherited from the ETH3D harness; the registered deltas are the loaders, TartanAir's dual-view sky handling (masks at 1000 m and an 80 m sensitivity view, which agree on every hypothesis verdict; two single cells flip), and the per-benchmark modal-set arm. E2: 6 TartanAir environments × first two released trajectories, 100-frame heads, 96 cells. E3: the first 13 distinct rooms of the official ScanNet v2 val split (_00 scans; several roster models trained on the train split, none on val — near-distribution but unseen), 104 cells.

Results. Cells passing the acceptance test per architecture (block-influence / indoor modal set / random mean / last-K), non-excluded cells only:

	TartanAir (of 12)				ScanNet (of 13)			
model	b-infl	fixed	rand	last-K	b-infl	fixed	rand	last-K
VGGT	12	12	5	12	13	13	4	13
LingBot	11	10	5	8	12	11	8	11
π^3	11	11	0	0	11	11	0	0
Fast3R	excluded: 4 infeasible cells, bound 3				12	12	8	13
DUSt3R	2	6	0	0	12	13	3	0
MASt3R	0	0	0	0	3	4	0	0
MonST3R	1	1	1	0	2	2	2	0
CUT3R	0	1	0	0	2	8	0	2

Non-excluded denominators: TartanAir 12, except LingBot, CUT3R and MonST3R (11) and DUSt3R (9); ScanNet 13 everywhere. The random column is the registered random-*mean*; the five per-seed arms are in the release, and one of them — seed 1, $[3, 9, 10, 11, 23]$, one set shared across the 24-block

models by the harness's fixed seed — passes DUSt3R $8/9$, MASt3R $10/12$ and MonST3R $10/11$ on TartanAir, identically in both registered views, with median relative AbsRel $1.08\times$ on MASt3R against 1.52/1.93/$2.50\times$ for block-influence, the indoor set and last-K. The other four draws fail broadly on the same models (0–2 cells each), so the passing draw is an existence result about the ranking, not a deployable baseline. No seed reaches the bar on ScanNet's failing trio (best $6/13$) or on CUT3R anywhere (0 across all five). Feasibility excluded 10 TartanAir cells and no ScanNet cell: the forest cell seasonsforest/P002 on 5 of 8 architectures; three more on Fast3R (gascola/P003, neighborhood/P001, seasonsforest/P001), whose four exceed the registered per-architecture bound of three, so it sits out the TartanAir hypothesis counts; and two more on DUSt3R (neighborhood/P001, seasonsforest/P001). Failures spread across all sequences on both benchmarks; no single scene drives a verdict. Full per-cell records: `phaseE2_*`/`phaseE3_*` in the release.

Registered verdicts. Against the locked thresholds (pass rate $\geq 80\%$ on ≥ 6 of 8 architectures; fixed set within 2 cells of re-scoring; KITTI-pattern fraction $\kappa \leq 0.25$, defined below): **H1 FAILED** on both benchmarks (E2: 3 of 7; E3: 5 of 8), **H3 FAILED** on both (3 of 7; 4 of 8), **H2 held** on 6 of 7 and 6 of 8, counting the cells whose verdict differs between the two arms in either direction (violations: DUSt3R 4 cells and CUT3R 6, both with the fixed set the *better* arm; LingBot 3 discordant cells on ScanNet, a net difference of one), and **H4 CONFIRMED** on both. The registered failure branches are the scoping the main text adopts. One caveat on wording: H3's registered baseline is the random *mean*, whose per-seed arms are reported above — on the render-side pair family one seed passes, so the TartanAir failures there are selector failures, not budget failures. The caveat this appendix used to carry, that capacity needs a K-sweep and only $K{=}5$ was run out of domain, is discharged below.

The sparsity question, settled as registered. The KITTI-pattern fraction κ (named f in the pre-registration) — GT-failing cells whose GT-free self-consistency sits below their architecture's passing median — was the reason these arms exist. Its null expectation fixes the reading: under an informative ground truth a GT-failing cell has moved more than a passing one, so κ tends to 0; under an uninformative one the failing cells are distributed like the passing ones, so κ tends to 0.5; the registered bound of 0.25 sits between.

benchmark	GT coverage	κ	bound
ETH3D	18.5%	0.56	0.25
TartanAir	100% (rendered)	$\mathbf{0.07}$	0.25
ScanNet	near-complete RGB-D	$\mathbf{0.19}$	0.25

On both complete-coverage benchmarks the accept/reject verdict tracks real output change; the verdict-flip pattern follows ground-truth coverage, not the models. The ETH3D study's (E1) $\kappa{=}0.56$ is consistent with ETH3D's 18.5% coverage as the cause — the diagnosis Appendix E gives KITTI (4.0%), now measured where coverage cannot be the cause, though the dense legs also change domain and exclusion set, so this is consistency short of isolation. The registered reattribution branch (output pathology) is not taken. One denominator note: on TartanAir 23 of the 41 failing cells belong to architectures with no passing cells, hence no passing median, and can never be pattern cells; restricted to eligible cells $\kappa = 3/18 = 0.17$, same verdict. E3 has no such dilution.

Failure texture. The two benchmarks fail differently. On ScanNet the failing architectures degrade uniformly and moderately: median relative AbsRel over failing cells 1.5–$1.8\times$ across MASt3R, MonST3R and CUT3R with relative δ_1 still ≈ 0.99 — scene structure survives, precision erodes past the bound on nearly every room. On TartanAir both metrics degrade (median over failing cells 1.4–$1.7\times$ on MASt3R, MonST3R and CUT3R and $1.2\times$ on DUSt3R, tails to $3.6\times$; relative δ_1 down to 0.47). In neither case is the output destroyed; the redundancy margin is simply gone.

Why capacity collapses there: an exploratory reading. The registered facts are the tables above; the following interpretation is exploratory and marked as such. (i) The two benchmarks fail differently in kind. On ScanNet's trio and on CUT3R everywhere, no released arm reaches the bar while the full model clears feasibility — there the redundancy margin itself is thin at $K{=}5$, the block-level form of a known compression result: compression damage concentrates on atypical, underrepresented inputs (Hooker et al., 2019), and layer-removal tolerance in LLMs is likewise task-dependent (Gromov et al., 2025; Men et al., 2025). On the render-side pair family, redundancy demonstrably survives (the

seed-1 set above) and what breaks is the *ranking*: every informed signal, scored on off-distribution activations, picks harmful blocks that a blind draw avoids. (ii) The MASt3R/MonST3R dissociation (near-perfect on ETH3D, 2–3$/13$ on ScanNet val) moves with capture style, not with indoor/outdoor: ETH3D is sharp wide-baseline DSLR stills, ScanNet is handheld video whose stride-1 windows have centimetre baselines. In that low-parallax regime a pair matcher's depth leans on refinement rather than triangulation, which is consistent with the δ_1-intact texture: layout survives, the eroded part is exactly what the pruned blocks refined. *This reading was registered and tested after the fact; the test is below and it passed, so (ii) alone among these five is no longer exploratory.* (iii) Re-scoring inside a collapsed domain is worse than importing the indoor set (CUT3R $2/13$ vs. $8/13$; DUSt3R on TartanAir $2/9$ vs. $6/9$): block-influence scored on off-distribution activations mis-ranks, the same corrupted-diagnostics mechanism as SLEB re-scoring on CUT3R in domain. Selector diagnostics require a healthy model. (iv) VGGT's last-K parity on all three further benchmarks (exact ties or better) says its late blocks are free out of domain; a practitioner needs no selector for that architecture there. (v) A relative-denominator artifact is excluded by absolute numbers: the same pruning costs MonST3R -0.002 absolute AbsRel on ETH3D and $+0.016$ on ScanNet (MASt3R -0.002 vs $+0.019$), and π^3, the sharpest baseline in the roster, passes $11/13$ — the dissociation lives in the models, not in the relative metric.

THE PARALLAX TEST AND THE OUT-OF-DOMAIN K-SWEEP

Both were registered (`prereg_E4E5_parallax_and_ksweep.md`) before any frame ran at any new stride or K, with thresholds hardcoded into the evaluation script. All cells are the same 13 ScanNet rooms, $V{=}4$, 25 disjoint windows, same acceptance test.

Does parallax explain the ScanNet failures? Widening the within-window stride at a fixed window count also widens the span of the room covered, so a naive stride comparison confounds baseline width with coverage. Three configurations separate them: **A** is the benchmark's setting (stride 1, step 4), **B** widens both (stride 8, step 32), and **C** widens coverage alone (stride 1, step 32). Block-influence cells passing, of 13:

model	A	B	C	B−A	B−C
MASt3R	3	**7**	3	+4	+4
MonST3R	2	**8**	3	+6	+5
CUT3R	2	**7**	5	+5	+2
VGGT (control)	13	13	13	0	0

The registered bars were a 4-cell gain at B on ≥ 2 of the three (met by all three) and a 3-cell margin over C on ≥ 2 (met by MASt3R and MonST3R), with VGGT permitted to lose at most 3 cells and losing none. The test is about the cause of the failures, not a repair: passing the 80% bar takes 11 of the 13 rooms, and at B the three reach 7, 8 and 7, so a wider baseline explains the deficit without removing it. CUT3R is the mixed case and is reported as such: coverage alone carries it from 2 to 5, the wider baseline adds 2 more. The imported indoor set moves the same way (MASt3R $4{\to}7$, MonST3R $2{\to}6$), so the effect is the domain's, shared by both selectors.

Is there a working K out of domain? Block-influence cells passing, of 13, at configuration A:

model	$K{=}2$	$K{=}3$	$K{=}5$	$K{=}7$	$K{=}10$
MASt3R	5	5	3	2	2
MonST3R	**13**	**13**	2	2	0
CUT3R	4	**7**	2	0	0

MonST3R passes every room at $K{\leq}3$ having passed $2/13$ at $K{=}5$: out of domain its capacity is not absent, the budget was too large. The result does not generalise, which is the other half of it — MASt3R peaks at 38% and CUT3R at 54%, so no swept budget restores them — and CUT3R reproduces the in-domain repair phenomenon out of domain, four rooms at $K{=}2$ against seven at $K{=}3$.

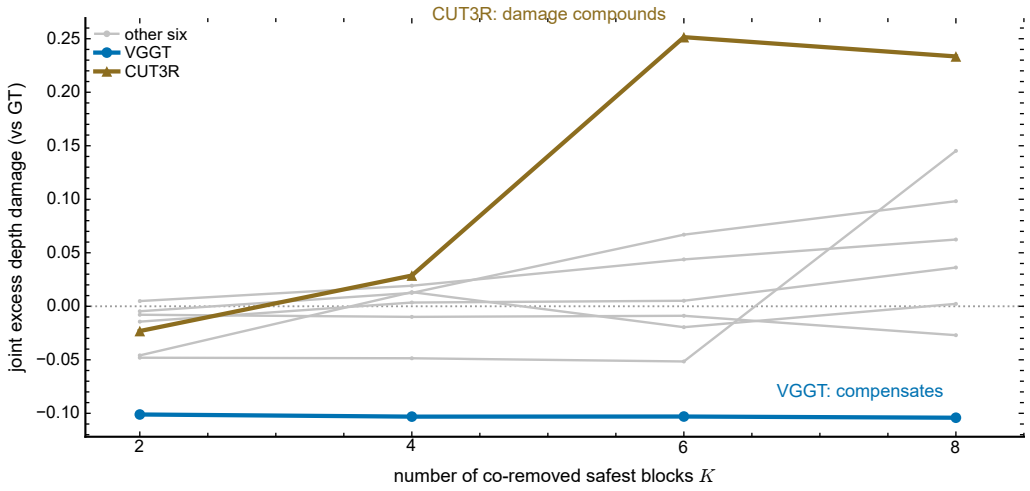

Figure 10: **Single-block safety does not add up.** Exploratory, one scene: for each architecture we bypass its K *individually*-safest blocks (by leave-one-out bypass damage) *together* and measure the joint excess depth damage: AbsRel against ground-truth depth, relative to the full model, mean over 60 windows on 7-Scenes *chess* seq-01–03, the same metric family as the benchmark. VGGT's joint damage stays below zero as K grows while CUT3R's climbs to $+0.25$ by $K{=}6$ (§5.2).

G Composition details

The $K{=}2$ dips, per set. The two non-monotonic dips of §5.2 are error cancellation seen at the acceptance threshold. On MASt3R at $K{=}2$, bypassing $\{5, 7\}$ alone inflates AbsRel to $1.43\times$, while adding two more blocks brings it back to 0.91–$1.14\times$. On CUT3R at $K{=}2$, $\{7, 9\}$ inflates AbsRel to $1.64\times$ and its superset $\{7, 8, 9, 10\}$ passes at 1.05–$1.23\times$ on $4/26$ sequences. In both the failing set is a subset of the passing one, so no block in the failing pair is individually responsible.

The 52 reversals, measured. Sweeping every adjacent step of the K-ladder ($K{=}1$–$8, 10, 12, 14, 16, 18$) over all 208 cells for a fail→pass reversal of the block-influence arm finds 52: MASt3R 25, Fast3R 11, DUSt3R 6, CUT3R 5, LingBot 3, VGGT 1, MonST3R 1. The released sweep files above $K{=}7$ store window means only, so we re-ran both endpoints of every reversal (45 cells, 104 K-points, block-influence arm only) with the sweep scripts patched to write per-window values, under a pre-registration fixed before the run (`research/prereg/prereg_CI_rerun_repairs.md`). Validity gate: the re-run window means must equal the released ones to 10^{-3}; 97 of 104 endpoints match to 10^{-5}, and all seven mismatches are Fast3R (Appendix E), whose reversals are therefore reported separately. Per reversal, on the same 30 windows: the paired difference of relative AbsRel between the failing and the passing budget is *improvement-separated* if its moving-block bootstrap 95% interval (block length 3, since adjacent windows overlap by half; 4000 resamples) excludes zero, and the reversal is *pass-separated* if the passing budget's interval lies entirely below 1.25.

	reversals	improvement sep.	BH $q{=}0.05$	pass sep.	both
six deterministic architectures	41	33	32	15	**14**
Fast3R (exploratory)	8	5	4	1	1

Under a global null at $\alpha{=}0.05$ one expects 2.1 of 41 to be separated on both sides. An i.i.d. window bootstrap would have called 38 improvements separated instead of 33. The fourteen: MASt3R at $K{=}10{\to}12$ on pumpkin seq-01/02 ($2.75 \to 1.05\times$, $2.27 \to 1.03\times$) and fr1/xyz ($1.18 \to 0.85$), at $K{=}8{\to}10$ on redkitchen seq-03 ($1.41 \to 0.70$) and fr2/xyz, at $K{=}12{\to}14$ on redkitchen seq-03, at $K{=}2{\to}3$ on chess seq-01, heads seq-02, redkitchen seq-01 and fr2/xyz, and at $K{=}3{\to}4$ on fr1/xyz and redkitchen seq-02; CUT3R at $K{=}3{\to}4$ on fr1/xyz ($1.40 \to 1.07$); VGGT at $K{=}16{\to}18$ on fr2/xyz ($1.50 \to 0.94$). The largest reversal, MASt3R office seq-02 at $K{=}10{\to}12$ ($3.83 \to 1.19\times$), is improvement-separated but not pass-separated, as the pre-registration anticipated for a passing

value 0.06 under the bound. Three Fast3R reversals are excluded because the re-run selected a different block set at one endpoint (Appendix E). The count reported before this run, three cells, came from half-widths borrowed from $K \leq 7$ and was, as the appendix then said, a lower bound (`research/scripts/_audit_repairs_ci_rerun.py`).

Two damage measures appear in §5.2. The global $\rho = +0.35$ ($n = 496$) compares, for every (architecture, sequence, arm) $K = 5$ set of the CO3D/TUM K-set grid, the sum of single-block damages with the measured joint damage. Damage there is *Procrustes pose drift*: the rigid Procrustes alignment of the pruned model's predicted point map to the full model's on the same window, summarised as the mean rotation shift in degrees, with no ground truth. For pair models the point map is the second view in the first view's frame.

Window overlap and the intervals. The composition windows are $V = 4$ frames at stride 4 with step 8, so adjacent windows share half their frames and the i.i.d. window bootstrap behind Table 10 is anti-conservative. Recomputed from the released per-window values with a moving-block bootstrap (block length 3, 2000 resamples), the $K = 8$ excess intervals are VGGT $[-0.15, -0.06]$, LingBot $[-0.10, +0.08]$, CUT3R $[+0.16, +0.31]$, π^3 $[+0.09, +0.20]$, Fast3R $[-0.09, +0.04]$, DUSt3R $[+0.005, +0.08]$, MASt3R $[+0.03, +0.17]$ and MonST3R $[+0.03, +0.09]$: 1.2–1.55× as wide as the table's (`research/scripts/analyze_composition_blockboot.py`), with the same four sign-flips and VGGT's below-zero excess clear of zero (DUSt3R by the least). The benchmark windows (§3) overlap the same way, which is why the reversal re-run above uses a moving-block bootstrap as well.

Do the two damage instruments agree? Nine TUM cells (VGGT, LingBot and CUT3R on fr1/desk, fr1/xyz and fr2/xyz) appear in both the K-set grid, which records Procrustes pose drift, and the benchmark, which records depth AbsRel against sensor depth. For the 37 $K = 5$ sets whose block lists are identical in both files (the informed arms; the random draws differ), pose drift and relative depth AbsRel correlate at Spearman $+0.19$ (cell-level bootstrap 95% interval $[-0.40, +0.60]$), with within-cell pairwise concordance $28/57 = 49\%$. On these cells the two instruments order the sets differently, so the pooled $+0.35$ of §5.2 is a statement about pose drift and Table 10's sign flips a statement about depth; neither converts into the other. Script: `research/scripts/_g4_artifact_checks.py`.

The pooled correlation depends on which sets are asked. Each of the 31 (architecture, sequence) cells carries sixteen $K = 5$ sets: six informed arms (block-influence, FZD, a ground-truth-best set, last-K, and two position-prior variants) and ten seeded random draws. Intervals are cell-level block bootstraps (2000 resamples), since the sets inside a cell share windows and blocks. Before 2026-09-07 this paper reported $+0.29$ ($n = 403$), the value on the arm set of the original run (three informed arms, ten draws); the released files had since gained three informed arms and, for the three phase-36 models, lost draws 5–9, which `research/scripts/_restore_phase36_seeds.py` puts back. The number moved with the arm set, not with any re-measurement, and every variant is in the release's audit script:

arm set	n	ρ	95% CI
all sixteen arms (reported)	496	$+0.35$	$[+0.19, +0.47]$
six informed arms only	186	$+0.56$	$[+0.34, +0.68]$
ten random draws only	310	$+0.19$	$[+0.00, +0.38]$
original run's arm set (three informed, ten draws)	403	$+0.29$	$[+0.13, +0.43]$

Per architecture on all arms: VGGT $+0.08$, LingBot $+0.15$, CUT3R $+0.43$, DUSt3R -0.27, Fast3R $+0.32$, MASt3R $+0.24$, MonST3R -0.31 (DUSt3R and MonST3R negative at $p < 0.05$). On the informed arms alone the negative architectures are VGGT (-0.50) and LingBot (-0.38), both at $p < 0.05$, while DUSt3R ($+0.50$), MASt3R ($+0.74$) and MonST3R ($+0.47$) are positive at $p < 0.05$; on the random draws alone only DUSt3R (-0.35) is negative at $p < 0.05$. LingBot changes sign between the informed and random families (-0.38 vs. $+0.12$); VGGT does not (-0.50 vs. -0.27). Script: `research/scripts/_audit_additivity_ci.py`. The pre-registered support bound for additivity was $\rho \geq 0.40$; the pooled statistic on all arms does not reach it but its interval includes it, so the registered test is indeterminate rather than failed; the informed arms alone exceed it; and neither reading rescues a predictor whose sign depends on the architecture and on the sets it is asked about. The claim of §5.2 therefore rests on that sign inconsistency and on Table 10's

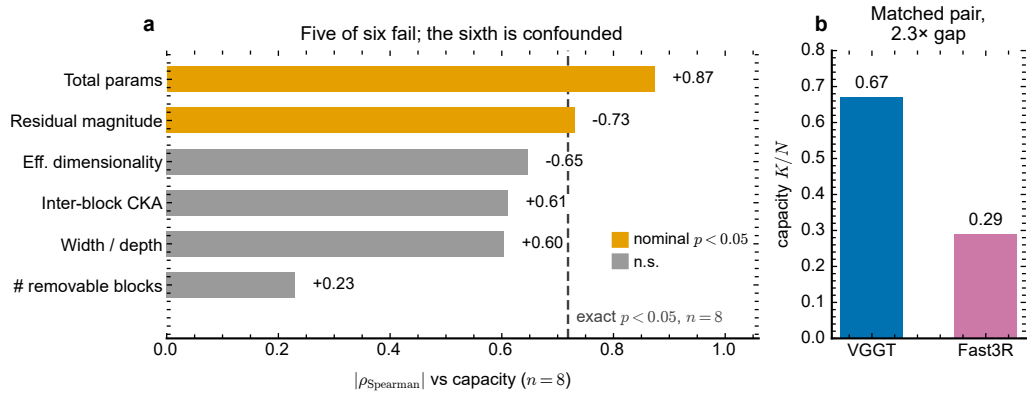

Figure 11: **No single structural factor explains capacity.** (a) |Spearman ρ| between each candidate predictor and the full-benchmark capacity (n=8). All six predictors were pre-registered and are tested against the corrected full-benchmark capacities (Appendix K); five fail. The orange bars are nominally significant at n=8 (exact permutation: total parameters $+0.87$, p=0.007; residual magnitude -0.73, p=0.047; the dashed line is the exact two-sided p=0.05 threshold for these ranks, $|\rho|$=0.72); the parameter correlation clears the six-test Bonferroni bound but is confounded with the roster's size families and untested out of sample (see text). (b) The cleanest controlled comparison: VGGT and Fast3R are matched on exposed depth (24 bypassable blocks), width (1024), and paradigm, yet differ 2.3× in capacity (1.1× at total depth, Appendix E).

four sign-flips with intervals excluding zero, not on the pooled value. Table 10 and Fig. 10 instead use depth AbsRel against ground truth on the K=8 individually-safest sets (60 windows, 7-Scenes *chess*).

Table 10: Co-bypassing each model's K=8 individually-safest blocks (exploratory, one scene): additive-predicted vs. actual joint excess depth damage (AbsRel against ground-truth depth, relative to the full model; 60 windows on 7-Scenes *chess*; bootstrap 95% CIs over windows). The last column compares actual to the additive prediction: sign-flip = predicted improvement, measured damage (all four CIs exclude zero); LingBot's predicted -0.33 improvement does not appear; CUT3R is approximately additive at K=8 while its damage climbs with K to $+0.25$ by K=6 (Fig. 10). Rows ordered by full-benchmark capacity; the *actual* column does not significantly rank it (ρ=-0.29, n.s., n=8; the earlier ρ=-0.81 against the stress-protocol capacity is retracted, Appendix K).

Model	Cap.	Add.-pred	Actual [95% CI]	vs. additive
VGGT	0.67	-0.13	-0.10 $[-0.14, -0.07]$	≈additive
LingBot	0.50	-0.33	$+0.00$ $[-0.05, +0.07]$	off by $+0.33$
CUT3R	0.42	$+0.29$	$+0.23$ $[+0.18, +0.29]$	≈additive; grows in K
π^3	0.39	-0.10	$+0.15$ $[+0.10, +0.19]$	sign-flip
Fast3R	0.29	-0.02	-0.03 $[-0.07, +0.02]$	≈additive
DUSt3R	0.29	-0.00	$+0.04$ $[+0.01, +0.07]$	sign-flip
MASt3R	0.25	-0.08	$+0.10$ $[+0.05, +0.15]$	sign-flip
MonST3R	0.21	-0.03	$+0.06$ $[+0.04, +0.08]$	sign-flip

H MECHANISM ANALYSIS DETAILS

Predictors (correlational; n=8). Against the full-benchmark capacities (Fig. 11a), the Spearman correlation of capacity with: model width (ρ=$+0.60$), number of individually-removable blocks ($+0.23$), per-block residual magnitude (-0.73, exact permutation p=0.047), total parameters ($+0.87$, p=0.007), effective dimensionality (-0.65), and inter-block CKA redundancy ($+0.61$). The pre-registered analysis was run against an earlier capacity protocol, under which width was the only nominally significant correlate ($+0.80$, confounded with depth: all 1024-wide models are 24-deep); against the corrected capacities width is not significant. The parameter-count correlate reaches exact permutation p=0.007 at n=8, below the 6-test Bonferroni bound (α_{corr}=0.0083). One rule covers all six: the predictors were registered, the capacity vector was corrected and the roster extended with π^3

after registration, and every correlation here is the registered predictor on that corrected outcome, the five nulls included. The correlate is confounded with the roster's size–family structure, since the three ≈1B-parameter models, VGGT, LingBot and π^3, are three of its four highest capacities, and it is untested out of sample. Residual magnitude (-0.73, exact p=0.047) is nominal only and fails the corrected bound. The p-values are exact two-sided permutation values over the 8! orderings; the t-approximation gives 0.005 and 0.040. We claim neither as the cause. Consistent with §5.2, $153/192$ block instances (Table 2's N summed, pair branches counted separately, π^3 included) are individually removable at relative AbsRel ≤ 1.05, yet joint damage stays below zero for VGGT and compounds for CUT3R: whatever drives capacity lives in training, data or design choices invisible to these predictors.

Three pre-registered hypotheses: two self-falsified, one pool-dependent. Three mechanistic hypotheses from the pre-registration each failed their own test; (i) and (iii) stay dead against the corrected capacities, and (ii) depends on the candidate pool (Appendix E). *(i) Attention bandwidth:* reducing VGGT's input from V=4 views (V^2=16 cross-view interactions) to V=2 (V^2=4) leaves its capacity unchanged. *(ii) Inter-block redundancy:* under Table 2's pool adjacent-block CKA does not significantly track capacity (ρ=+0.61, n.s.) and CUT3R has the *highest* CKA yet mid-table capacity; with the never-bypassed frame-attention blocks counted in N it correlates at +0.95 (Appendix E). *(iii) Effective dimensionality:* the ratio of representation effective rank to width does not track capacity (ρ=−0.65, n.s.); Fast3R has the lowest ratio yet mid-table capacity (0.29).

I WALL-CLOCK TIMING DETAILS

Why merging alone loses at V=4. Re-matching tokens in every global block costs ≈1.3 ms per block, more than the attention it saves at 5.5k tokens. We did not profile this, so it explains the measured slowdown without measuring it.

What the protocol costs. Scoring one sequence is 30 hooked full-model forwards, about 6 s on VGGT at 185 ms per V=4 window. Validating one K-set costs 30 pruned forwards. A full capacity sweep runs 13 K-points × 26 sequences × 30 windows, roughly 10k forwards, and takes about 30 min per 24-block architecture on one RTX 5090.

Table 11: Real wall-clock saving $1 - t_{\text{pruned}}/t_{\text{full}}$ (a timing measurement, not pre-registered) (medians of 30 timed forwards after 5 warm-ups; full-model medians in parentheses), V=4, TUM fr1_desk, FZD K-sets; the late-K sets timed in the same runs are within 0.5 pp, so the saving depends on K only. Saving is linear in K. Timed at each architecture's own capacity rather than at K=5, it is 22.5% on π^3 (K=14, 26/26 sequences still passing), 20.6% on VGGT (K=16, 25/26), 14.0% on LingBot (K=12, 23/26) and 11.5% on CUT3R, whose capacity is the K=5 point (Table 2's block-influence set; the FZD set here gives 11.6%); per-block saving stays flat across each sweep, so the linearity holds over three times the range it was fitted on. CUT3R gains ∼2× per K because its recurrent decoder has 12 blocks, so each bypassed block is 1/12 of the decoder (1/24 for VGGT/LingBot).

Model	K=3	K=5	K=7
VGGT (24L, 185 ms)	+3.7%	+6.4%	+9.0%
LingBot (24L, 185 ms)	+3.5%	+6.0%	+8.5%
CUT3R (12L rec., 259 ms)	+6.5%	+11.6%	**+16.3%**
Mean	+4.6%	+8.0%	+11.3%

Re-timing with the selector's own sets; π^3. Table 11 times the FZD sets. A later independent session (same GPU model and protocol; fresh full-model medians 179/176/259 ms) times the per-sequence block-influence sets of the timed sequence together with the FZD and late-K sets in the same runs: block-influence saves 3.8/6.5/9.1% (VGGT), 3.5/5.8/8.4% (LingBot) and 7.1/11.5/17.0% (CUT3R) at K=3/5/7, the three set families are within 0.6 pp of one another at every (model, K), and the two sessions agree within 0.8 pp cell for cell. The saving is a function of K alone, so Table 2 reports these selector-set numbers. The same protocol on π^3 (36 blocks, V=4, full-model median 106 ms, real block removal, timing its modal block-influence set) gives 4.7/7.6/11.1% at K=3/5/7; with t_K the median forward time after removing K blocks, the saving $s(K) = 1 - t_K/t_0$ gives a

per-block saving $\beta = s(K)/K$ that is flat across each sweep, and the bypassable stack's share of the forward is $N\beta$: $\sim$55% for π^3, $\sim$31% for VGGT and Fast3R and $\sim$28% for LingBot and CUT3R, which is why π^3 saves more per bypassed block despite its deeper stack.

Pair models: no reliable K=5 gain. The same real-removal protocol on TUM freiburg1_desk (5 warm-up + 30 timed pairs, median): Fast3R $86.8\pm0.9 \rightarrow 81.2\pm1.3$ ms ($+6.5\%$, outside noise); MASt3R $90.2\pm8.9 \rightarrow 86.6\pm10.6$ ms ($+4.0\%$); MonST3R $86.8\pm21.5 \rightarrow 90.7\pm14.3$ ms (-4.5%); DUSt3R $76.4\pm8.8 \rightarrow 86.8\pm12.4$ ms (-13.6%). For the three DUSt3R-family models the per-pair standard deviation (9–22 ms) exceeds the effect: their inference path (CPU-side pairing, heads) dominates and the decoder blocks are a minority of the latency, so we report their K=5 speedup as indistinguishable from zero and omit the noise-dominated medians.

Stacking with FastVGGT token merging (pre-registered). FastVGGT (Shen et al., 2025) merges tokens inside every global-attention block of VGGT (ToMe-style bipartite matching at its default merge ratio 0.9 from the first global block, first frame protected); bypass removes whole global blocks, so a bypassed block never merges. We run its public code with our VGGT-1B weights and preprocessing (its hard-coded 37×28 patch grid set to 37×37 for our 518×518 inputs) under four conditions, full, bypass (K=5, the per-sequence block-influence set), merging and both, with four predictions fixed before the run (supplementary ledger; 2.1 GPU-hours spent, twice the pre-registered estimate, on the repeated timing pass described below). Write s_B, s_C for the single-axis wall-clock savings and q_B, q_C for the single-axis median relative AbsRel. **P1**: the stacked saving is within 3 pp of the multiplicative combination $1-(1-s_B)(1-s_C)$ at every V. **P2**: bypass is the faster axis at V=4 and merging at V=64, with the stacked model fastest at every V. **P3**: the stacked model passes at least as many sequences as the weaker axis minus two, and its median relative AbsRel stays within $1.05\,q_Bq_C$. **P4**: the K=5 set re-scored under merging overlaps the no-merging set (Jaccard ≥ 0.6) on at least $20/26$ sequences. Three kill criteria were set alongside them, none of which fired. *Quality* (26 sequences, acceptance test): every condition passes $26/26$; median AbsRel relative to the full model is q_B=0.986 (bypass), q_C=0.934 (merging), and 0.928 (both), under the pre-registered bound $1.05\,q_Bq_C$=0.967 (P3). Merging alone *improving* AbsRel ($25/26$ cells, δ_1 unchanged) was not predicted and is reported as exploratory; we have not tested whether it is token-averaging smoothing rewarded by scale-aligned AbsRel, and with $q_C < 1$ the P3 bound tests retention of that gain rather than bounded damage. Re-scoring the K=5 set *under* merging (re-scored on the first 15 of the 30 windows) returns the no-merging set on $15/26$ sequences and swaps one block on $11/26$, always within the seven-block pool $\{2, 3, 6, 7, 8, 18, 20\}$ (Jaccard ≥ 0.6 on $26/26$, median 1.00, min 0.67; P4); the re-scored sets are no better (median AbsRel 0.936 vs. 0.928). Re-scoring the same 15 windows with merging *off* separates the two possible causes: of the 11 swaps, 7 return to the no-merging set (so merging caused them), 2 reproduce the merged set (the window count caused them) and 2 are mixed; the other 15 sequences never swapped. The re-scoring is therefore driven by merging rather than by the shorter scoring pass, and it costs nothing either way. *Wall-clock* (Table 12; single RTX 5090, 5 warm-up + 30 timed, real module removal): bypass grows slowly with V ($+7.9\% \rightarrow +15.7\%$, as the global blocks' share of the stack rises with quadratic attention) while merging goes from a net cost at V=4 to $+38.1\%$ at V=64. The stacked saving falls 1.4 pp (V=16) and 2.0 pp (V=64) short of the multiplicative prediction $1-(1-s_B)(1-s_C)$, inside P1's $\pm$3 pp band, and lands 3.1 pp above it at V=4, 0.1 pp outside the band in the favourable direction. Both deviations have one cause: merging's cost or saving is paid per global block and bypassed blocks carry neither, so the post-hoc per-block model $t_D = t_B + \frac{N-K}{N}(t_C - t_A)$ reproduces all three stacked savings within 0.3 pp (Table 12, per-block column; not pre-registered). P2's crossover (bypass the faster axis at V=4, merging at V=64) is confirmed. Its second clause, that the stacked model is the fastest condition at every V, fails at V=4, where merging alone is slower than the full model and drags the stacked model (-9.2%) below bypass alone ($+7.9\%$). No kill criterion fired. Bypass saves $\approx5\%$ of weight memory (five 12.6M-parameter blocks) and no activation memory (full-model peak $8.3/9.7/11.3$ GB at V=4/16/64); the token axis is the memory lever at large V. Three implementation notes. (i) The fork calls `cuda.synchronize()+empty_cache()` and clears its RoPE caches at every aggregator layer (memory hygiene for its 1000-frame regime), a per-layer allocator/sync overhead unrelated to either method and invoked identically by all four conditions. As shipped it raised the V=4 baseline to 758 ms and timing noise 7$\times$ (std 54–290 ms), swamping both effects. That first grid, run before the decision to disable it, fails P1 at V=4 (gap 7.3 pp; 2.4 pp at V=16, inside the band) and is kept as a diagnostic. V=4: 758 ms, bypass $+11.5\%$, merging -10.4%, both -4.9%; V=16: 2220 ms, $+0.2\%$, -21.8%, -24.0%. Timing runs disable it;

quality runs are unaffected. (ii) The pre-registered control that the corrected baseline match stock VGGT within 5% was *not* met: 146 ms vs. 185 ms (Table 11). The timed fork model omits VGGT's point head and runs its camera and depth heads in bf16, whereas the stock model of Table 11 runs camera, depth and point heads in fp32; the five bypassed blocks save the same absolute time in both (11.6 vs. 11.9 ms), so the gap is confined to the heads, and Table 12 percentages are relative to the fork's smaller forward (hence bypass +7.9% here vs. +6.4% in Table 11). Re-running the grid with the point head enabled confirms this: the baseline moves to $158/757/5581$ ms at $V{=}4/16/64$ and bypass to $+7.1/{+}9.8/{+}15.1\%$ (from $+7.9/{+}10.8/{+}15.7\%$), merging to $-18.7/{+}10.4/{+}37.2\%$ and the stack to $-8.0/{+}18.1/{+}44.5\%$. The regime picture is the same, shifted by the head cost, with every standard deviation below 4 ms. (iii) Exploratory latency decomposition (not pre-registered; 3 warm-up + 10 timed, same inputs): the aggregator, the DINOv2-L patch encoder plus 24 frame-attention and 24 global blocks, of which only the global blocks are bypassed, is $87/92/96\%$ of the fork's forward at $V{=}4/16/64$ respectively, and the heads $13/9/4\%$. Removing five global blocks removes $9.6/11.8/16.4\%$ of the aggregator, so the bypassable global blocks are $\approx 46/57/79\%$ of it, inferred from $5/24$ blocks and not timed separately. That predicts $8.4/10.9/15.7\%$ end-to-end against the measured $7.9/10.8/15.7\%$. Timing bypass *before* the full model gives $8.2/11.0/15.6\%$, so there is no ordering or thermal drift. This is why $5/24$ global blocks buy $\approx 8\%$ at $V{=}4$: the encoder and frame-attention blocks, never bypassed, are roughly half the aggregator, and the heads 13%.

Table 12: Block bypass ($K{=}5$) stacked with FastVGGT token merging on VGGT-1B, V frames per forward. Columns: the fork's full-forward latency t_A; wall-clock saving $1 - t/t_A$ of bypass (t_B), merging (t_C), and both (t_D) (+ = faster, − = slower; medians of 30); the pre-registered multiplicative prediction from the single-axis savings s_B, s_C; the post-hoc per-block prediction $t_D = t_B + \frac{19}{24}(t_C - t_A)$ ([‡]not pre-registered: merging's per-block cost or saving is paid only by the 19 of 24 global blocks that remain); and the aggregator's share of the forward ([†]exploratory decomposition, note iii). Merging alone is slower than the full model at $V{=}4$ and gives the larger saving at $V{=}64$; the stacked saving is within 2 pp of the multiplicative prediction once both single-axis savings are positive and within 0.3 pp of the per-block prediction at every V.

V	full (ms)	bypass $K{=}5$	merging	both	$1{-}(1{-}s_B)(1{-}s_C)$	per-block[‡]	aggregator[†]
4	146	+7.9%	−22.0%	−9.2%	−12.3%	−9.5%	87%
16	708	+10.8%	+11.2%	+19.5%	+20.8%	+19.7%	92%
64	5325	+15.7%	+38.1%	+45.8%	+47.8%	+45.9%	96%

J THE ITERATIVE-REFINEMENT INTERPRETATION, IN FULL

One reading is consistent with all measurements; it is not causally tested. On the iterative-refinement view of residual networks, successive blocks incrementally refine a shared residual stream rather than computing fresh representations (Greff et al., 2017; Jastrzębski et al., 2018; Veit et al., 2016). A block then contributes little once the geometric estimate has largely converged, leaving a near-identity map that is cheap to bypass. This matches what transfers in our data: the bypass-safe blocks are predominantly those of *small relative update magnitude*, the magnitude-and-position axes that account for about 96% of the transferable FZD prior (§4.2), which is to say low-amplitude refinements rather than representation-forming blocks. The same lens reframes capacity as *utilisation*: a feedforward model that front-loads representation formation leaves many late refinement blocks (high capacity), whereas the recurrent CUT3R reuses one block stack $V{=}4$ times per forward, raising per-block utilisation. It is the only model whose joint damage compounds with K (§5.2) and the one that collapses hardest under state-stressing TUM trajectories (§5.1). A causal test would need controlled retraining, which we did not run; §7 states the correlational scope.

K HONESTY APPENDIX: CLAIMS WE RETRACTED OR DEMOTED

Tables 13 and 14 list every claim we retracted or demoted during this study, with what killed it and what replaced it.

Table 13: Retractions and demotions, part 1: claims killed by their own pre-registered tests. Rows 1–3 are the three pre-registered mechanistic hypotheses, two self-falsified by their designated tests and the redundancy verdict pool-dependent (Table 14); row 4 is the strongest correlate, demoted for confounding and then killed by the capacity correction (Table 14, row 1); row 5 is an out-of-domain verdict revised by its pre-registered controls.

Claim (pre-registered where noted)	What killed it	What replaced it
Attention bandwidth explains capacity: VGGT's V^2=16 cross-view interactions (H1, pre-reg.)	V=4 → V=2 ablation: capacity unchanged	view count ruled out
Inter-block redundancy (adjacent-block CKA) predicts capacity (H2, pre-reg.)	ρ=−0.09 under the pre-registered protocol; +0.61 n.s. against the corrected capacities (+0.67 at n=7), not predictive under either; +0.95 with the frame-attention blocks counted in N	not decisive, pool-dependent (Table 14)
Effective dimensionality (rank/width) predicts capacity (H3, pre-reg.)	ρ=−0.40 n.s. pre-registered; −0.65 n.s. corrected (−0.68 at n=7)	rank rejected
Width drives capacity (ρ=+0.80, p=0.03)	width perfectly collinear with depth in the 7-model roster; fails 6-way multiple-comparison correction; and ρ falls to +0.60 (n.s.; +0.66 at n=7) against the corrected capacities	demoted, then killed outright by the capacity correction (Table 14, row 1); no causal claim
Initial read: block-influence K=5 fails out-of-domain on KITTI driving	pre-registered controls: persists at minimal stride and under square center-crop; far-range/open ETH3D scenes pass; pruned-vs-full self-drift is *smaller* on KITTI than on passing ETH3D scenes (7.7% vs 15.1% Fast3R; 37% vs 47% MASt3R)	re-read as sparse-LiDAR ground-truth artifact; OOD verdict downgraded to directional (Appendix E)

Table 14: Retractions and demotions, part 2: claims revised by recomputation or re-audit. Rows 1–3 follow from recomputing capacity on the full benchmark under a single selector protocol; row 4 from a per-scene re-audit of the ETH3D transfer claim; row 5 replaces an asserted composability claim by the measured token-axis composition (Appendix I); row 6 demotes a reversal count to the reversals the release's intervals can adjudicate; row 7 restates the additivity correlation on the arm set the release actually carries; row 8 records the six-scalar relabel; row 9 demotes the redundancy verdict to pool-dependent; row 10 scopes the prior's "no target calibration" to its pool after the held-out test; row 11 scopes the one-pass-versus-oracle claim to the CO3D-measured oracle; row 12 corrects what the oracle baseline knows; row 13 names the instrument behind the cross-architecture ranking; row 14 corrects the pool-random spread's correlates; row 15 corrects the free-zone figure's reading.

Claim (pre-registered where noted)	What killed it	What replaced it
Capacity spans 8× (0.67–0.08)	recomputation on all 26 benchmark sequences under the deployed selector (the original numbers pooled two selectors on a TUM-heavy 6-sequence subset): CUT3R 0.08 → 0.42, MonST3R 0.33 → 0.21	capacity spans 3.2× on the benchmark and shifts with evaluation distribution within a model (§5.1)
Joint excess damage at K=8 ranks capacity (ρ=−0.81, exploratory)	against the corrected full-benchmark capacities: ρ=−0.29, n.s., n=8 (−0.50 at n=7); the original held only against the stress-protocol capacity definition	retracted; replaced by the deployment-level non-monotonicity evidence (§5.2)
Conservative guidance $K \leq 2$	full-benchmark K-sweep: MASt3R passes 69% and CUT3R 77% at K=2 (nested-set dips), both ≥ 85% at K=5	guidance inverted: K=5 is the uniformly safe grid point; small K is not automatically conservative (§5.2)

Claim (pre-registered where noted)	What killed it	What replaced it
ETH3D transfer: "$6/7$ architectures keep $\geq 80\%$ acceptance"	per-scene re-audit under the acceptance test on all four scenes: $4/7$ pass every scene; VGGT and MonST3R each miss one; the original count used three scenes and AbsRel only	restated per-scene (Appendix E); later superseded by the 13-scene benchmark, on which §4.3 now reads "7 of 8 architectures on ETH3D"
Block bypass "composes with the token, precision, and cache axes" (asserted)	stacked with FastVGGT merging (pre-registered P1–P4): quality composes ($26/26$) and the $K{=}5$ set needs no re-scoring (identical on $15/26$, one-block swap on $11/26$; median Jaccard 1.00); wall-clock multiplies within 2 pp at $V{=}16/64$ and misses the 3 pp band by 0.1 pp at $V{=}4$ (favourable direction); the union-best clause fails at $V{=}4$, where merging alone is 22% slower; the first timing pass was invalid (the fork's per-layer `empty_cache` inflated the baseline 3–5×) and was re-run with it disabled; the added 5% baseline control was not met (146 vs 185 ms, head configuration)	narrowed to the measured token axis, regime-qualified (§6, Appendix I); precision/cache composition marked untested
"52 failing prunes are repaired by co-bypassing more blocks, in 7 of 8 architectures" (§5.2)	audit against the release's own bootstrap intervals: 9 of 52 fail on δ_1 alone; 30 of the other 43 fail by less than one half-width of the bound; the release carries no intervals above $K{=}7$, so only the three MASt3R $K{=}10{\to}12$ recoveries (2.3–3.8× $\to\leq$ 1.19×) are adjudicable	demoted to the two $K{=}2$ dips and three cells, then re-measured under a pre-registration: 33 of 41 reversals on the deterministic architectures are separated improvements and 14 are separated on both sides; 52 stays a raw count (Appendix G)
"Summed single-block damage predicts joint damage at only $\rho{=}{+}0.29$" (abstract, §1, §5.2)	the $+0.29$ ($n{=}403$) reproduces only on the original run's arm set; the release carries sixteen arms per cell, on which the pooled value is $+0.35$ $[+0.19, +0.47]$, $+0.56$ on informed arms alone, $+0.19$ on random draws alone, with the negative architectures depending on the arm set (DUSt3R and MonST3R over all arms, VGGT and LingBot over the informed arms); a "cross-architecture predictor $+0.04$" clause had no locatable source	restated as arm-set-dependent with no shared sign; the $+0.04$ clause removed; claim carried by the sign inconsistency and Table 10 (Appendix G)
"None of six pre-registered structural scalars predicts capacity" (abstract, §1, §5.3)	total parameters, one of the six, correlates at $+0.87$ with the corrected capacities (exact permutation $p{=}0.007$, under the Bonferroni bound) and had been set aside as "doubly post-hoc" while the five nulls on the same corrected vector kept their registered status; one rule now covers all six	restated as five of six failing, the sixth confounded with the roster's size families and untested out of sample (§5.3, Appendix H)
"Inter-block redundancy self-falsified; CUT3R has the highest CKA yet mid-table capacity" (§5.3, Appendix H)	the verdict depends on the capacity normalisation: with the never-bypassed frame-attention blocks of VGGT and LingBot counted in N, inter-block CKA correlates at $+0.95$ (exact $p{=}0.001$) and parameter count falls to $+0.42$; no scalar is significant under all three pools	demoted to pool-dependent (Table 13, row 2); §5.3 claims no structural proxy under any pool and states the pool it uses (Appendix E)

Claim (pre-registered where noted)	What killed it	What replaced it
"A prior fit on other architectures needs no target calibration" (abstract, §1, §4.2)	the pre-registered held-out test on π^3 (2026-09-09): the ranking transfers (ρ=+0.69) but the prior's K=5 set passes 17/26 against the registered 21, at the random mean, one late block extrapolated as safe; the registered blindness prediction also fails there (+0.36, not within 0.1 of the full prior)	scoped to the seven-model pool; the held-out numbers stated in the abstract, §1, §4.2 and Appendix D
"A one-pass score beats the measured-damage oracle" (abstract, §4.2 heading and body, Fig. 6 caption)	the pre-registered deployment of a benchmark-instrument oracle (`prereg_oracle_benchmark_loo.md`, locked 2026-09-10, run 2026-09-14, 0 failed cells): scoring each block by leave-one-out relative AbsRel against sensor depth instead of by output shift on CO3D lifts the oracle from 75% to 190/208 (91%), against block-influence's 196/208 (94%); the two disagree on 26 cells, 16 to 10, which an exact McNemar test does not separate (p=0.33). Registered H1 indeterminate (89–94% band), H2 supported (block-influence at least matches the oracle on 7 of 8 architectures, bound 6)	scoped to the CO3D-measured oracle throughout; both oracle rows now in Table 3, the instrument explanation in §4.2, the picture in Fig. 8
"A label-free score beats the *ground-truth* single-block oracle" (abstract, §1, §4.2)	the oracle arm is built from each model's single-block bypass sweep on the same 30 CO3D windows, whose damage is the output shift against the full model, so it uses no depth labels either; the asymmetry we claimed does not exist	restated as a one-pass score beating an oracle that measured every block, N forward passes against one; pass rates unchanged (93% vs. 75% on the 182 covered cells)
"The safety ranking behind such sets is shared across architectures" (ρ=+0.53; abstract, §1, §4, conclusion)	re-audit (2026-09-16): the +0.53 is measured on the instrument the prior is fit to, output shift on CO3D; against the benchmark's single-block depth damage the same folds give +0.09 (−0.22 to +0.51), and the selector's sets are not built from that ranking	named as an output-shift ranking with the depth value beside it (§4.2, Table 6); the cross-architecture claim rests on the prior's deployed sets, 87% of pool cells and 17/26 outside
"A spread no registered structural quantity orders" (pool-random control, §4.2; Appendix C gave +0.52 against capacity and −0.29 against the pool's share)	recomputation: +0.47 and −0.71, the share being N read backwards; under the two alternative block counts it reaches −0.88 and −0.89 ($p\leq$0.011)	the spread tracks how much of each model the eight-block pool covers (exploratory)
Fig. 6: "the marks mostly do not sit in the blue", deeper blocks "individually safer", and the selector's lead over the CO3D-measured oracle "because" single-block damage does not add	the triangles use a row's five bluest blocks on 17 of 35 slots (11.5 by chance); the −0.39 trend is the output-shift instrument's (+0.00 on depth damage); §4.2 attributes that lead to the instrument	caption and §4.2 name the instrument and give both counts; only the tie with the benchmark-measured oracle is read as non-additivity (Appendix D)

L PAIR-MODEL TRAJECTORY ACCURACY

Appendix M checks the task on three of the four architectures with native pose heads; LingBot's was not measured. The three pair models and Fast3R emit point maps and no pose, so this recovers camera centres from the point maps themselves: for a window with view 0 as reference, the same view's points are read once in view 0's frame and once in their own, and a Procrustes fit between the two places that camera in frame 0. The sequence loader, the ground-truth parser, the windowing and `umeyama_rmse` are the same code as Appendix M, with chunks of 4 frames

at stride 4 where that appendix uses 8; otherwise only the pose extractor differs. Registered as `prereg_D2_pair_trajectory.md`, locked 2026-09-15 before any cell ran.

Guards. This project has retracted a pose-convention result whose reported error was the norm of the ground truth, identical across two architectures. Recovering pose indirectly is more exposed to that failure, not less, so three guards were locked with the rule that any failure voids the study and forbids reporting a single trajectory number. Feeding ground truth in as the prediction gives 1.7×10^{-10} (bound 10^{-6}); the closest pair of architectures on any sequence differ by 2.4×10^{-4} m, so no two agree to machine precision; and permuting the predicted frame order makes every cell worse. All three pass. The frame convention is scanned on the unpruned model only and then frozen; both conventions occur across the roster.

Table 15: D2. Absolute trajectory error (ATE-RMSE) in metres, windowed and Sim3-aligned, for the unpruned model and for the deployed $K{=}5$ set; *depth* is the deployed set's verdict under Eq. 1, read off the released per-cell files; on DUSt3R fr1/xyz and fr2/xyz and the three MonST3R cells the fixed set differs from the per-sequence set, and its verdict comes from the fixed-set run. Bold marks the registered failure condition: depth accepts, trajectory exceeds $1.25\times$.

model	sequence	full	$K{=}5$	ratio	depth
MASt3R	fr1/desk	0.0177	0.0369	$\mathbf{2.085}\times$	PASS
Fast3R	fr1/xyz	0.0139	0.0257	$\mathbf{1.845}\times$	PASS
DUSt3R	fr1/desk	0.0205	0.0368	$\mathbf{1.796}\times$	PASS
MASt3R	fr1/xyz	0.0172	0.0302	$\mathbf{1.761}\times$	PASS
Fast3R	fr1/desk	0.0171	0.0297	$\mathbf{1.730}\times$	PASS
DUSt3R	fr1/xyz	0.0191	0.0279	$\mathbf{1.461}\times$	PASS
MonST3R	fr2/xyz	0.0037	0.0050	$\mathbf{1.348}\times$	PASS
DUSt3R	fr2/xyz	0.0043	0.0055	$\mathbf{1.265}\times$	PASS
MonST3R	fr1/xyz	0.0121	0.0149	$1.234\times$	PASS
MonST3R	fr1/desk	0.0222	0.0221	$0.997\times$	PASS
MASt3R	fr2/xyz	0.0059	0.0057	$0.965\times$	PASS
Fast3R	fr2/xyz	0.0057	0.0056	$0.993\times$	FAIL

Registered verdict: H2. Eight cells are depth-accepted and over the bound (Table 15), all four architectures contribute at least one, and the worst is $2.085\times$ — the trajectory error doubles on a prune the acceptance test passes. The one depth-rejected cell sits at $0.993\times$. It was rejected at 1.252 and passes at 1.230 when re-measured in the pool-random run, Fast3R's forward pass being non-deterministic (Appendix E).

Scope. $n{=}12$: four architectures, three TUM sequences, one budget ($K{=}5$). Centres are recovered, since these models emit none, and the convention is chosen on the unpruned arm; both are reported above. This is the same n per architecture as Appendix M's existing check, and it is small.

M Downstream trajectory evaluation

Depth accuracy is a proxy. The deployment question is whether a *task* breaks. We evaluate windowed Umeyama-Sim3 ATE-RMSE over full TUM sequences on the three of the four models with native pose heads, VGGT, π^3 and CUT3R (LingBot's was not measured). The pair-Siamese models recover pose only through global alignment and are not included. Chunks are 8 consecutive frames at stride 4, fed jointly; predicted camera centres are Sim3-aligned to ground truth per chunk, so scale is fitted; we report the mean over a sequence's chunks relative to the unpruned model. Conditions are the full model, block-influence $K{=}5$, and 20 random $K{=}5$ sets. The three VGGT rows come from the earlier study that motivated this one (the released `phaseHardA` files). The π^3 and CUT3R arms were pre-registered before any of their trajectory data existed, with the bounds, controls and kill criteria fixed in advance and both outcomes of each test written down first; that session also re-ran VGGT on fr1/desk and reproduced the earlier numbers (full-model ATE 17.8 mm in both, block-influence $0.96\times$ in both).

Two results, one in each direction (Table 16). On the feedforward models the saving is task-neutral: block-influence costs VGGT and π^3 at most $1.07\times$, at most 0.15 mm and 0.5 mm, while random sets at the same budget cost 2.1–$4.7\times$ (Table 16). π^3's unpruned trajectory is in fact the most accurate

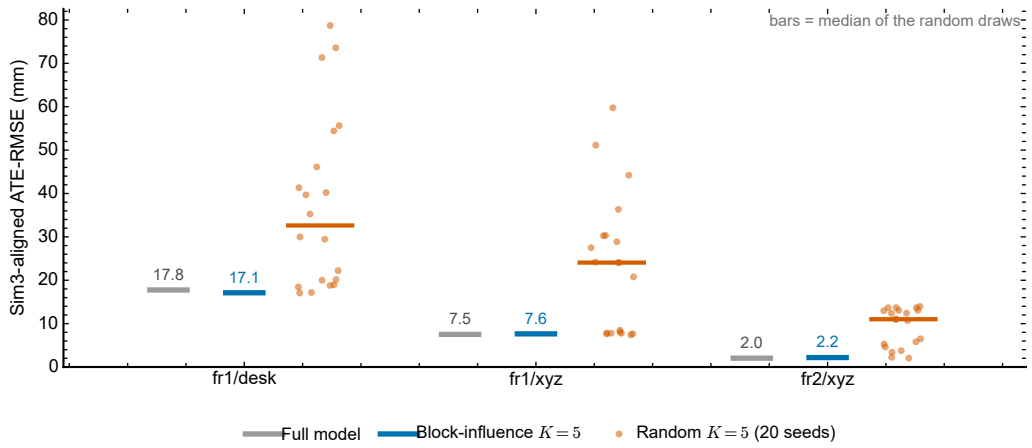

Figure 12: **Pruning preserves downstream windowed trajectory accuracy (VGGT).** Umeyama-Sim3-aligned ATE-RMSE (mm) on three TUM sequences: full model, block-influence $K{=}5$, and every one of the 20 random $K{=}5$ draws (bar = their median). Block-influence is within $1.07\times$ of the full model; the random *mean* is 2.1–$4.5\times$ worse, and the draws span an order of magnitude, the luckiest of them reaching the informed set.

Table 16: **Windowed Sim3 trajectory error, relative to the unpruned model.** Random is the mean over 20 sets (medians 1.8–$5.4\times$; Fig. 12's bars are medians). The depth column is the benchmark's own $K{=}5$ verdict on the same cell; the trajectory harness re-scores block-influence on its own chunks, which swaps one block on π^3 fr1/desk, CUT3R fr2/xyz and VGGT fr1/xyz, and only the first swapped set has a measured depth verdict (pass). The two feedforward models keep trajectory within $1.07\times$; the recurrent CUT3R loses it on all three sequences, including the italicised cell, which the acceptance test *accepts*.

model	sequence	depth $K{=}5$	full (mm)	block-influence	random
VGGT	fr1/desk	pass	17.8	$\mathbf{0.96}\times$	$2.11\times$
VGGT	fr1/xyz	pass	7.5	$\mathbf{1.02}\times$	$3.08\times$
VGGT	fr2/xyz	pass	2.0	$\mathbf{1.07}\times$	$4.54\times$
π^3	fr1/desk	pass	7.4	$\mathbf{1.07}\times$	$3.30\times$
π^3	fr1/xyz	pass	5.0	$\mathbf{1.04}\times$	$3.42\times$
π^3	fr2/xyz	pass	1.7	$\mathbf{1.05}\times$	$4.69\times$
CUT3R	fr1/desk	fail	25.7	$1.52\times$	$2.16\times$
CUT3R	fr1/xyz	*pass*	15.3	$\mathit{2.54}\times$	$2.76\times$
CUT3R	fr2/xyz	fail	5.5	$1.81\times$	$2.09\times$

of the three. On VGGT relative-pose rotation error separates the same way, at $\times 0.88$–1.02 against $\times 1.7$–4.9 over 5 seeds.

The recurrent CUT3R does not follow, and this is the pre-registered test failing against us. We predicted that its depth-accepted cell (fr1/xyz) would keep its trajectory and its clearly depth-rejected cell (fr1/desk) would not. The rejected cell did break ($1.52\times$), but so did the accepted one, by $2.54\times$: from 15.3 to 38.8 mm, against a chunk whose own ground-truth extent is 78 mm. All three CUT3R cells break while their depth verdicts read fail/pass/fail, so across these three cells the depth verdict carries no discriminative signal for trajectory and the accepted cell is a false negative. That was an $n{=}3$ observation on one architecture when it was written, and Appendix L has since enlarged it: the three pair models and Fast3R fail the same way, on eight further depth-accepted cells. **Depth acceptance does not imply task safety, and not only on state-carrying models.** One post-hoc observation, offered as post-hoc: CUT3R reads depth off its self-view point map and pose off a separate head on the same recurrent state, so bypass can damage one while sparing the other. We did not predict this and do not test it.

Controls. Four, all pre-registered. *Pose convention* was fixed per model on the *full* model only, by adopting the candidate that minimises its ATE; the bias therefore runs against our own hypotheses, since shrinking the denominator makes the pruned conditions harder to pass. π^3 selected cam-to-world by a $4.6\times$ margin and CUT3R translation-first cam-to-world from eight candidates, the same choice on every sequence. *Reproduction*: the harness re-ran the stored VGGT cell and returned 0.963 against 0.963 and 2.108 against 2.11. *Selector consistency*: the sets recomputed inside this harness match the benchmark's stored $K{=}5$ sets at Jaccard 0.67–1.00, where an exact match is not expected because the two score block-influence on different window lengths. *Feasibility*: full-model ATE over ground-truth chunk extent is 0.06–0.35 for every cell against a pre-registered 0.5 bound, so no cell is one where the model simply cannot track. No kill criterion fired. The study cost 1.0 GPU-hour.

N Extended related work

3D foundation models. DUSt3R (Wang et al., 2024) reconstructs 3D from unposed image pairs; MASt3R (Leroy et al., 2024) adds matching; MonST3R (Zhang et al., 2025) handles dynamics. These are pair-based encoder-decoders. VGGT (Wang et al., 2025a) and Fast3R (Yang et al., 2025a) use bidirectional multi-view transformers; LingBot-Map (Chen et al., 2026a) adds causal streaming KV-cache attention; CUT3R (Wang et al., 2025b) is a recurrent stateful decoder; π^3 (Wang et al., 2026) removes the reference frame with a permutation-equivariant decoder. Later models postdate the benchmark: any-prior MapAnything (Keetha et al., 2026), Depth Anything 3 (Lin et al., 2025), causal-streaming STream3R (Lan et al., 2025) and StreamVGGT (Zhuo et al., 2025), and memory-augmented MUSt3R (Cabon et al., 2025). The protocol runs unchanged on them (untested). Our eight models were selected for paradigm coverage (offline bidirectional, recurrent-state, causal-streaming, pairwise) and public checkpoints. None of these works analyses which blocks can be bypassed at inference, or how prunability compares across the family.

Training-free layer / depth pruning. ShortGPT (Men et al., 2025) and the layer-similarity pruning of Gromov et al. (2025) rank blocks by a single-pass importance signal (input-output cosine / block-influence) and drop the least important. SLEB (Song et al., 2024) verifies and eliminates redundant transformer blocks, FinerCut (Zhang et al., 2024) prunes attention and feed-forward layers separately, by the change each removal makes to the model's output, and Bonsai (Kolawole et al., 2024) prunes structures by forward-pass-only perturbation; gradient-based variants (Ma et al., 2023) add Taylor importance and a recovery step. Newer variants add linear replacements (Shopkhoev et al., 2025), magnitude compensation (Chen et al., 2026b), or replacement training (Chen et al., 2024); extensions cover vision-language and action models (Ma et al., 2025; Yuan et al., 2025; Yang et al., 2025b). All are designed within language-tower families, typically need a target-model calibration set, and score quality by perplexity or VQA accuracy. We port the forward-pass-only idea to 3D-FMs, benchmark it cross-architecture, evaluate it with dense geometry and a downstream task, and add a calibration-free cross-architecture prior. GeoLens keeps ShortGPT's score unchanged and tests four working defaults the recipe leaves implicit in models that emit metric geometry; Table 1 states them and how each came out.

Three findings in that literature we re-measure, and we state them here in full because our results are their counterparts. EvoPress (Sieberling et al., 2025) starts from the observation that per-layer compression errors are not independent, and reports that pruning a model *further* can recover performance. Both hold here: co-bypassing each model's individually-safest blocks flips the sign of the predicted damage on four architectures (§5.2), and acceptance is not monotone in K (§5.2). Hinostroza et al. (2026) show that cosine similarity is a poor proxy for the accuracy a layer's removal actually costs, and propose measured damage in its place. The per-block half of that holds here too — score-to-damage correlations fall within ± 0.1 on four of the eight architectures (Appendix B) — while the set-level half does not, since the sets that same score picks pass 94% and the benchmark-measured oracle does not separate from them (§4.1, Appendix K). Lu et al. (2026) benchmark layer pruning in language models and find a trivial choice hard to beat, dropping the final quarter and fine-tuning a small part of what remains; that one fails here, where last-K passes 49% of cells against the selector's 94% and is no better than the random baseline's 56% (§4.2). We claim none of the three as new. What the benchmark adds is which of them survive a change of model class. Unlike unified pruning methods spanning CNNs and ViTs (Liu et al., 2024), we benchmark whether block importance itself transfers across 3D-FM architectures.

Efficiency work on 3D-FMs. Existing acceleration of 3D-FMs spans token merging (Shen et al., 2025; Shu et al., 2025), learned token pruning (Li et al., 2026), attention restructuring and sparsification (Sun et al., 2025; Ren et al., 2026), quantization (Feng et al., 2026), and streaming KV-cache compression (Liu et al., 2026); most of it is built for VGGT. Spark3R (Tang et al., 2026) reduces query and key–value tokens asymmetrically, training-free, on VGGT, π^3 and Depth Anything 3 among others; Lite3R (Zhang et al., 2026) replaces dense attention with sparse linear attention under quantization-aware training on VGGT and Depth Anything 3. None removes transformer blocks, and none measures whether a block selector transfers across architectures. We benchmark the orthogonal depth axis on eight models spanning four paradigms, and measure how block bypass stacks with FastVGGT token merging on VGGT: quality and the block set survive merging at FastVGGT's default setting, and the wall-clock savings compound at $V \geq 16$ (19.5%/45.8% stacked), so the stack pays only where merging itself pays (§6). The precision and cache axes are untested. Closest to us, AVGGT (Sun et al., 2025) analyses VGGT's layer roles training-free, independently corroborating that specific VGGT blocks contribute nothing to correspondence. But it restructures attention without removing blocks, covers two same-family models, and offers a method, not a benchmark: no selector comparison, no composition or capacity analysis, no cross-architecture transfer. Concurrently (June 2026), RegimeVGGT (You et al., 2026) reports U-shaped layer redundancy in VGGT with depth-dependent token compression. Our single-block bypass sweep, an output-shift instrument on 30 CO3D windows (Appendix D), agrees with that placement on VGGT: its most damaging blocks are 15–17 (6.45, 0.85 and 0.81 output s.d.) and its last six blocks lie within 0.03 of zero. The shape is not general: over the seven-model pool deeper blocks shift the output less (pooled Spearman -0.39, Fig. 6; $+0.00$ on depth damage), DUSt3R's ends are both safer than its middle as VGGT's are, and π^3's final blocks are among its most damaging. RegimeVGGT is therefore adjacent to our result but on the token axis, single-architecture, and without a selector benchmark.

Training-time and adaptive depth. LayerDrop (Fan et al., 2020), LayerSkip (Elhoushi et al., 2024), DynaBERT (Hou et al., 2020), early-exit (Xin et al., 2020; Liu et al., 2020; Zhou et al., 2020; Xin et al., 2021; Schuster et al., 2022), and learned gates (Wang et al., 2018; Wu et al., 2018) all require training, whereas GeoLens needs only forward passes. Token-axis efficient ViTs (Rao et al., 2021; Bolya et al., 2023), quantization, and distillation are orthogonal axes in principle. We measure composition with token merging only (§6), and quantization and distillation are untested.

Cross-model representational structure. CKA (Kornblith et al., 2019), SVCCA (Raghu et al., 2017), width/depth analyses (Nguyen et al., 2021), stitching (Lenc & Vedaldi, 2015; Bansal et al., 2021), and pruning-mask transfer (Morcos et al., 2019) ask whether distinct networks share structure. Our cross-architecture safety prior sits in this lineage, and our mechanism analysis tests CKA among its predictors and finds its verdict pool-dependent (Appendix E). Probing work (Belinkov, 2022; El Banani et al., 2024) documents a gap between what a representation encodes and what the model uses, which is why GeoLens predicts bypass safety and does not localise geometry to a "writer" block.