DO CROSS-MODAL UPDATES COMMUTE? PATH DEPENDENCE IN HETEROGENEOUS MULTIMODAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Heterogeneous multimodal learning increasingly relies on aligning signals from sensors with substantially different statistical and semantic properties. Existing work has primarily focused on what should be aligned and how strongly alignment should be imposed, whereas the effect of **update order** is largely obscured by the standard joint-training formulation. This paper studies whether different compositions of the same directional updates lead to the same learned solution. To isolate this question, we introduce a controlled identification protocol that fixes direction-indexed data and augmentation exposure while varying composition order. Across seven heterogeneous sensor conditions and three backbone families, reordered updates produce broad but strongly condition-dependent endpoint differences, establishing path dependence across the evaluated training settings. Complementary diagnostics examine local update compositions and endpoint sensitivity across optimizer configurations. Same-state probes detect local differences, while optimizer comparisons reveal substantial variation in endpoint sensitivity. Building on this characterization, we investigate whether path sensitivity can be controlled. Interventions at the optimizer-state, path-composition, and local-operator levels identify effective regimes, and compute-neutral antithetic block pairing reduces aggregate sensitivity by 15.5% with a small average utility change. Finally, native sequential multimodal training reproduces order-dependent endpoints across 21 fresh seeds. Fixed-scale and randomized balanced schedules yield distinct sensitivity distributions, connecting the phenomenon to schedule design in an existing alternating multimodal algorithm. Together, these results establish update order as a consequential optimization variable in sequential multimodal learning and demonstrate that schedule design can reduce its endpoint effect in the evaluated regimes while retaining matched directional exposure and the original update budget.

1 INTRODUCTION

Cross-modal alignment is a basic mechanism for learning from heterogeneous observations that describe related content (Andrew et al., 2013). Contrastive models such as CLIP and ALIGN established scalable image–text representation learning (Radford et al., 2021; Jia et al., 2021a; Zhai et al., 2023), while later systems extended alignment across broader modality sets and unified encoders (Girdhar et al., 2023; Zhu et al., 2024; Han et al., 2024; Zhou et al., 2025). Recent work continues to revisit the geometry of the shared space itself—what should be aligned, what should remain modality-specific, and whether a single shared geometry is sufficient (Cicchetti et al., 2025; Dufumier et al., 2025; Grassucci et al., 2026; Qian et al., 2026).

Alignment at the representation level, however, does not make the underlying optimization symmetric. Modalities can learn at different rates or become imbalanced under a shared objective (Wang et al., 2020; Fan et al., 2023; Yang et al., 2024); more recent analyses identify gradient conflict between modality encoders and fusion modules (Wei et al., 2025b), while also showing that perfectly balanced learning need not be optimal (Wei et al., 2025a). These findings motivate methods that change how much each modality learns or how cross-modal influence is routed (Jiang et al., 2025). A different variable becomes relevant whenever these interactions are realized sequentially: the order in which matched directional responses are composed over training.

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.

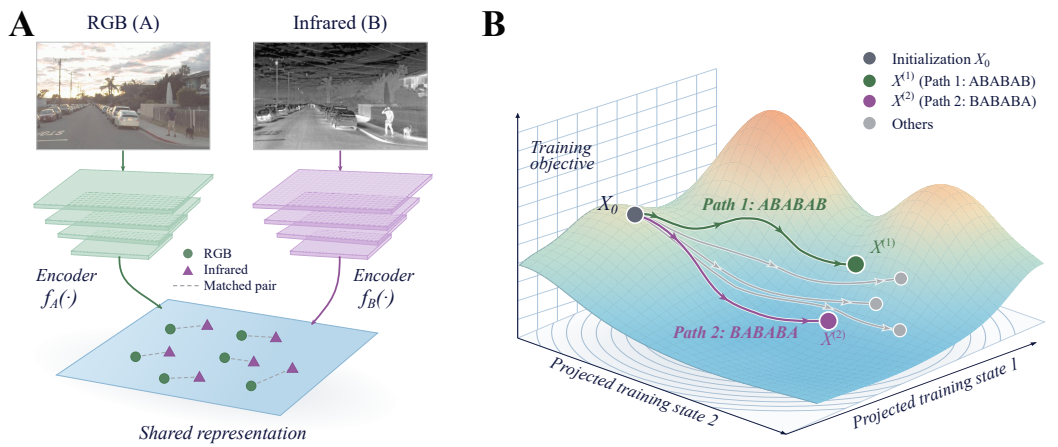


Figure 1: **From heterogeneous alignment to path-dependent learning dynamics.** (A) Paired observations from two modalities are mapped into a shared representation space; alignment can move a matched pair toward a compatible, not necessarily unique, location between their modality-specific representations. (B) We then move from representation geometry to trajectories of the training state X. Different compositions of the same directional updates can follow different routes, and our experiments show that these routes can reach different endpoints at the prescribed training budget. ABABAB and BABABA illustrate one representative pair, with gray curves showing other legal orders. The schematic landscape illustrates the resulting training trajectories.

Order is not entirely absent from learning: stochastic optimization can depend on data permutation (Shamir, 2016; De Sa, 2020), and trajectory-specific data influence can depend on when data are encountered (Wang et al., 2025a). In multimodal learning, MLA makes order explicit through alternating unimodal adaptation (Zhang et al., 2024). Yet most cross-modal alignment is still implemented as a joint update, where both modality gradients are evaluated from the same pre-update state and applied together. Such a step has no literal hidden A→B execution order, but its monolithic form also obscures whether the underlying directional responses would commute if they were realized sequentially. Our question therefore has two direct motivations: understanding methods in which order is explicit, and exposing optimization behavior that joint alignment cannot reveal. We ask: if the same directional updates are composed in different legal orders, do they lead to the same learned solution?

In this work, we turn that question into a controlled identification problem. We expose two directional responses to the same alignment objective and compare legal paths with matched direction-indexed data and augmentation content. The protocol also matches initialization, optimizer, update budget, modality exposure, and schedule statistics. This yields HeteroPathBench, a systematic test of endpoint sensitivity across sensor conditions and encoder architectures. The construction makes directional composition explicit and measurable, complementing standard joint training (Figure 1).

The empirical picture then follows from breadth to diagnostics and control. Across 420 matched pairs, path dependence appears across all seven sensor conditions and three backbone families, with a mean absolute R@5 gap of **2.48 pp** but strong regime variation. Same-state probes characterize local compositions, while optimizer comparisons reveal substantial variation in endpoint PS. Interventions at the optimizer-state, path-composition, and operator levels identify controllable regimes; compute-neutral ABP reduces aggregate PS by **15.5%** with a small average utility change. Native MLA also reproduces the order effect across fresh seeds. Its antithetic scheduling results reveal variation across schedules and seeds, connecting the controlled study to an existing sequential algorithm and a practical setting for studying order.

We summarize our contributions as follows:

1. We identify **directional update order** as a distinct degree of freedom once heterogeneous multimodal interactions are realized through ordered directional updates, and formulate **path independence** as the corresponding learning-dynamics question: whether matched directional invocations yield the same learned solution under different legal compositions.

2. We introduce an **operator-indexed identification protocol** and HeteroPathBench, which match initialization, data and augmentation exposure, and schedule statistics across legal orders, making endpoint sensitivity a controlled empirical object (Section 2).
3. We show that **path sensitivity is broad but strongly regime-dependent**, and characterize its local and endpoint manifestations: same-state diagnostic probes compare directional compositions, while optimizer comparisons reveal substantial variation in final task sensitivity across configurations (Sections 3–4).
4. We identify a practical control regime in which **compute-neutral ABP reduces aggregate path sensitivity by 15.5%**, with a small mean utility change. We also extend the order comparison to native sequential multimodal training: MLA exhibits distinct AV/VA endpoints across 21 fresh seeds. Comparing fixed-scale and randomized antithetic schedules further characterizes the relationship between schedule design and endpoint sensitivity within an existing alternating multimodal algorithm (Sections 5–6).

2 A Methodology for Testing Path Independence

Testing path independence requires a consistent comparison of directional update schedules. In stochastic training, reordering updates can change which minibatches and augmentations are assigned to each direction. We therefore index these inputs by logical invocation, matching each directional call to the same data and augmentation realization across paths. Together with matched initialization, update budget, and schedule statistics, this construction makes directional composition a controlled experimental variable. Endpoint comparisons then measure its effect on the learned representation.

2.1 Path independence as endpoint invariance

Let paired observations x^A and x^B be encoded by $f_A(\cdot;\theta_A)$ and $f_B(\cdot;\theta_B)$ under a shared alignment objective L_{align}. A conventional joint step evaluates both encoder gradients from the same pre-update state and applies them together, so we do not assign it an internal A-before-B order. To expose an orderable object, we instead define two directional responses to the same objective: mode A admits the alignment gradient only to encoder B, while mode B reverses the route. For stochastic realization D, the corresponding update operator is

$$T_{r,D} : X \mapsto X', \qquad r \in \{A, B\}, \tag{1}$$

where X contains model parameters together with optimizer and model state.

Let $\mathcal{U}$ denote a fixed collection of directional invocations and Π a legal class of their orderings. Starting from X_0, $F(\pi; X_0, \mathcal{U})$ is the endpoint produced by composition $\pi \in \Pi$. For an endpoint observable ϕ, path independence over Π requires

$$\phi(F(\pi; X_0, \mathcal{U})) = \phi(F(\pi'; X_0, \mathcal{U})), \qquad \forall\, \pi, \pi' \in \Pi. \tag{2}$$

Viewing training as a composition of state-changing operators follows the classical splitting perspective (Strang, 1968; McLachlan & Quispel, 2002), used here only as a diagnostic lens.

2.2 Isolating order from stochastic exposure

The identification problem is illustrated in Figure 2. Under a global data stream (left), A_1 receives M_1 in $A_1B_1A_2B_2$ but M_2 in $B_1A_1B_2A_2$, coupling order with data exposure. We instead bind data and augmentation draws to logical invocation (right): A_1 always receives D_1^A, B_1 receives D_1^B, and so on. This preserves invocation-specific presentation content across different compositions (Bengio et al., 2009).

Formally, direction A receives $D_1^A, D_2^A, \ldots$ and direction B receives $D_1^B, D_2^B, \ldots$; the k-th invocation of each direction always receives its corresponding realization. Hence the two paths contain the same indexed invocations $\{(A, D_1^A), (B, D_1^B), (A, D_2^A), (B, D_2^B)\}$ and differ only in composition.

Strict comparisons match initialization, architecture, optimizer/LR, batch size, update budget, A/B counts, and schedule statistics to control known training variance (Sutskever et al., 2013; Henderson et al., 2018; Bouthillier et al., 2021; Agarwal et al., 2021). Data and augmentation draws are invocation-indexed, while each update responds to its preceding state. The resulting endpoint differences measure sensitivity to legal composition; implementation details are in Appendix H.1.

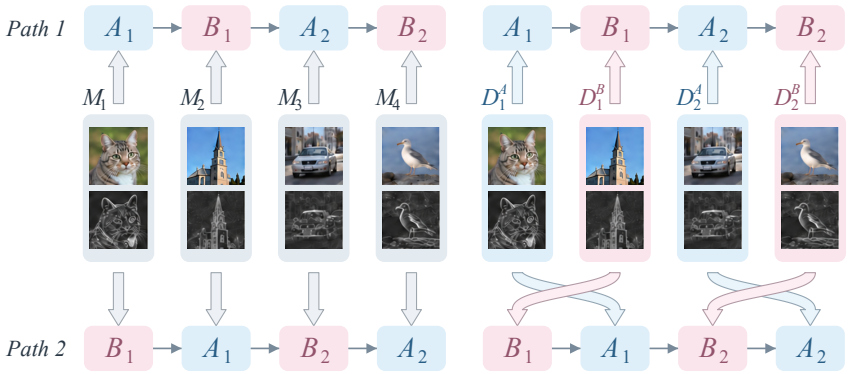


Figure 2: **Matching data and augmentation content across reordered paths. Left:** a global stream makes A_1 receive M_1 in Path 1 but M_2 in Path 2, coupling order with data exposure. **Right:** operator-indexed coupling keeps A_1 tied to D_1^A, B_1 to D_1^B, and so on, preserving each invocation's inputs across the two paths.

2.3 Experimental instantiation and measurement

We instantiate this protocol as *HeteroPathBench* across seven heterogeneous sensor conditions and three backbone families. The conditions span SAR–RGB (SEN12-clear and SEN12-cloudy) (Schmitt et al., 2019), visible–IR (LLVIP) (Jia et al., 2021b), RGB–sonar (Eilat and Sunboat), RGB–depth (NYU) (Silberman et al., 2012), and RGB–event (DSEC) (Gehrig et al., 2021), while the backbone set includes ResNet18, ConvNeXt-Tiny, and ViT-B/16. The main protocol uses 528 directional updates with balanced A/B exposure, allowing the same identification test to be repeated across substantially different sensor relationships and encoder architectures.

For evaluation, we use bidirectional cross-modal Recall@5 (R@5). Recall@5 is the fraction of queries whose true paired sample from the other modality appears among the five most similar candidates. The reported R@5 averages the two retrieval directions,

$$\mathrm{R@5} = \frac{1}{2}\left(\mathrm{R@5}_{A\to B} + \mathrm{R@5}_{B\to A}\right). \tag{3}$$

Higher R@5 therefore means that the learned representation more reliably retrieves the correct cross-modal pair.

Given two strictly matched paths π_a and π_b, we define *path sensitivity* (PS) as the absolute difference between their final R@5 values,

$$\mathrm{PS}(\pi_a, \pi_b) = \left|\mathrm{R@5}(F(\pi_a; X_0, \mathcal{U})) - \mathrm{R@5}(F(\pi_b; X_0, \mathcal{U}))\right|. \tag{4}$$

A larger PS indicates stronger sensitivity of the learned cross-modal retrieval behavior to composition order. Because PS is unsigned, it measures the magnitude of path dependence rather than favoring either ordering.

A single matched pair may still provide an incomplete view of the admissible paths. We therefore complement the pre-fixed primary comparison with complete legal schedule classes under the same matching constraints, so that endpoint dispersion can be assessed without selecting favorable or extreme schedules after observing the results. Full condition definitions, schedule construction, and implementation details are provided in Appendix A.

3 Path Dependence Is Broad but Condition-Dependent

In this section, we study the long-horizon consequence of changing composition order under the matched protocol of Section 2. Across 420 strict matched path pairs, the final bidirectional R@5 differs by **2.48 pp on average** (95% CI [2.01, 2.98]), with a median gap of 1.20 pp. The effect is not confined to a small tail: 96.2% of pairs exceed the 0.01 pp numerical tolerance, 54.8% differ by more than 1 pp, 38.3% by more than 2 pp, and 13.6% by more than 5 pp.

Table 1: **Path sensitivity across sensor conditions and backbones.** Mean absolute final bidirectional R@5 differences (pp) for strict matched paths.

Condition	ResNet18	ConvNeXt-Tiny	ViT-B/16	Mean
DSEC RGB–Event	6.15	6.56	6.67	**6.46**
NYU RGB–Depth	5.89	3.32	5.57	**4.92**
SEN12-clear SAR–RGB	1.57	1.95	2.53	**2.02**
LLVIP Visible–IR	1.79	2.54	1.50	**1.95**
SEN12-cloudy SAR–RGB	0.95	0.68	0.82	**0.82**
Eilat RGB–Sonar	0.61	0.81	1.01	**0.81**
Sunboat RGB–Sonar	0.39	0.23	0.47	**0.36**
Mean	**2.48**	**2.30**	**2.65**	**2.48**

3.1 Broad coverage under a fixed set of strict path comparisons

We first keep the path comparisons fixed and vary the learning regime. The same pre-specified strict path set is evaluated across all seven heterogeneous sensor conditions and three backbone families, giving 420 matched comparisons in total. This design asks whether order sensitivity survives changes in modality relationship and encoder architecture without simultaneously changing which path comparison defines the effect.

Table 1 shows that the effect survives throughout this broad sweep, although its magnitude is far from uniform. DSEC RGB–Event is consistently the most sensitive condition, with mean gaps around 6–7 pp across all three backbones. NYU RGB–Depth is also strongly path-dependent, while Sunboat RGB–sonar remains below 0.5 pp throughout. Removing DSEC and NYU still leaves a mean PS of **1.19 pp** over the other five conditions. Across conditions, ResNet18 (He et al., 2016), ConvNeXt-Tiny (Liu et al., 2022), and ViT-B/16 (Dosovitskiy et al., 2021) yield mean PS of **2.48, 2.30, and 2.65 pp**. A common-batch control preserves this architecture breadth (Appendix C), making the simplest batch-size explanation insufficient.

The broad sweep therefore gives a clear first result: with the strict path set held fixed, path dependence survives across every evaluated sensor condition and backbone family, while its functional magnitude is strongly regime-dependent.

3.2 Dense legal-path enumeration reveals within-regime dispersion

The breadth study above deliberately keeps the strict path set fixed. We next change the emphasis: rather than asking whether the same comparisons generalize across many regimes, we ask how widely endpoints can separate when the admissible path space itself is expanded. We enumerate **120 complete legal schedule classes** over five representative heterogeneous conditions under the same matching constraints. Each class is summarized by the difference between its highest- and lowest-R@5 endpoints.

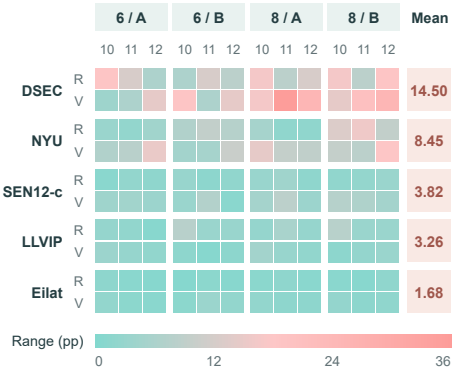


Figure 3: **Legal-schedule endpoint dispersion.** Cells show max–min R@5 (pp); right: condition means. Columns: blocks/final route and seeds; R/V: ResNet18/ViT-B/16.

Figure 3 gives this denser view of the path space. Across all classes, the mean endpoint range is **6.34 pp**, with a median of **3.73 pp** and a maximum of **35.42 pp**. The heatmap also reveals a pronounced condition structure. DSEC has the largest mean range at **14.50 pp**, followed by NYU at **8.45 pp**; SEN12-clear, LLVIP, and Eilat yield mean ranges of **3.82, 3.26, and 1.68 pp**, respectively.

Crucially, the larger ranges in the sensitive conditions recur across multiple legal schedule families and initialization seeds rather than appearing in a single isolated cell. The many-path experiment therefore strengthens the broad result from Section 3.1 in a different direction: path dependence is not only observable under a fixed set of comparisons across many regimes; substantial endpoint dispersion remains when the legal path space within a regime is opened up much more fully.

3.3 Calibrating path sensitivity against training variability

Example ordering and stochastic trajectories are established sources of optimization variability, from curriculum and presentation-order effects to recent evidence that data influence depends on when examples are encountered (Bengio et al., 2009; Wang et al., 2025a). They provide a natural scale for calibrating path sensitivity.

All **63/63** exact repeats reproduce the same R@5, parameters, optimizer state, and buffers. On a matched calibration subset, changing only composition order gives a **2.73 pp** difference, compared with **2.50 pp** when only the operator-indexed stochastic stream changes. These matched comparisons place order-induced variability on a scale comparable to stochastic training variation, establishing directional composition as a practically relevant source of endpoint differences.

The effect also extends to a different schedule geometry: highly alternating ABABAB and BABABA paths differ by **4.01 pp on average**. Alignment-strength and composition-depth sweeps likewise retain nonzero sensitivity across all tested settings; full trends are reported in Appendix C. Together, exact repeatability, stochastic calibration, and alternate schedule geometry establish composition order as a reproducible source of long-horizon variability, motivating the trajectory-level analysis in Section 4.

4 From Local Order Sensitivity to Endpoint Path Dependence

Section 3 shows that changing update order can produce very different endpoint effects across learning regimes: DSEC exhibits multi-point path sensitivity, whereas Sunboat is much less sensitive under the same data-matched protocol. We next examine this variation through two complementary diagnostics: local compositions evaluated from a common checkpoint, and endpoint comparisons under different optimizer configurations. Together, these tests characterize order sensitivity at the update and task levels, providing a basis for examining interventions throughout training.

4.1 Local directional composition at a common state

Using a stateless FP32 SGD probe at a common checkpoint and minibatch, we compare AB and BA, recomputing the second update after the first (Appendices D and H.1):

$$\Delta_{AB}(X, D) = d(T_{B,D}(T_{A,D}(X)), T_{A,D}(T_{B,D}(X))) \,, \tag{5}$$

where $d(\cdot, \cdot)$ measures the relative parameter discrepancy between the two resulting states.

All **252 same-state probes** produce distinct states, including all ResNet18 and ViT-B/16 cases with deterministic forwards. Under this common FP32 SGD diagnostic, the median relative discrepancy decreases from approximately 1.74×10^{-7} near initialization to 9.52×10^{-9} by step 396. Local composition therefore remains order-sensitive across the sampled training checkpoints.

The local discrepancy and endpoint PS characterize order sensitivity at two scales. The probe resolves parameter separation after two directional updates, while endpoint PS measures its task-level counterpart after complete training schedules. The observed local gaps coexist with substantial endpoint differences in sensitive conditions. Together, these measurements motivate a trajectory-level analysis that considers both directional composition and the surrounding optimization dynamics.

4.2 Optimizer dependence of endpoint sensitivity

A complementary comparison measures endpoint sensitivity under different optimizer configurations. Our main experiments use AdamW, which inherits Adam's adaptive first- and second-moment estimation while decoupling weight decay (Kingma & Ba, 2015; Loshchilov & Hutter, 2019). As a contrasting optimization regime, we repeat the matched-path comparison with momentum-free stochastic gradient descent (SGD) (Bottou et al., 2018).

Mean PS is **2.96 pp under AdamW and 0.38 pp under momentum-free SGD** at matched numerical learning rates, demonstrating substantial optimizer dependence of endpoint path sensitivity.

The accompanying controls test the persistence of the effect under changes to optimizer and model state. AdamW with designated state resets retains a mean PS of **3.09 pp**, while freezing BatchNorm

running statistics likewise preserves path-dependent endpoints. Thus, neither retaining optimizer state across blocks nor updating BatchNorm running statistics is required for the endpoint differences observed in these controls.

Together, the diagnostics establish a consistent empirical picture: directional compositions yield distinct local states, and endpoint sensitivity varies with the optimization configuration. These update-level and task-level measurements identify several practical intervention points. The following section evaluates controls acting on optimizer state, schedule composition, and local updates. Diagnostic details are provided in Appendices D and H.1.

5 Controlling Path Sensitivity

Section 4 characterizes local composition and optimizer-dependent endpoint sensitivity. These observations identify several intervention points along the training trajectory. We study controls acting on adaptive optimizer state, path composition, and local directional updates. Each experiment measures the change in PS together with task utility, allowing the robustness–utility balance to be assessed across the same matched training comparisons.

5.1 Multiple levels offer levers for control

Adaptive-state control. Separating route-specific AdamW statistics changes path sensitivity: the strongest variant reduces PS by about **9.9%**. Its nearly **10 pp** loss in mean R@5, however, shows that optimizer-state isolation alone does not provide a useful robustness–utility balance.

Path-level control. Symmetric and antithetic path interventions produce substantially larger gains in favorable regimes, including **38–55%** PS reductions with neutral or positive utility in several condition–architecture settings. Other variants improve mean R@5 while increasing PS, revealing that task utility and path robustness can move independently.

Operator-level control. Local averaging, projection, replay, and cross-evaluated updates provide a third lever, with **30–45%** reductions in selected regimes and modest utility changes. These results identify useful local-update interventions, although their endpoint benefits vary across the tested training regimes.

Across three qualitatively different intervention levels, the recurring reductions establish a consistent result: *path sensitivity is controllable, but controllability is heterogeneous*. Detailed controller definitions and development results are in Appendix E.

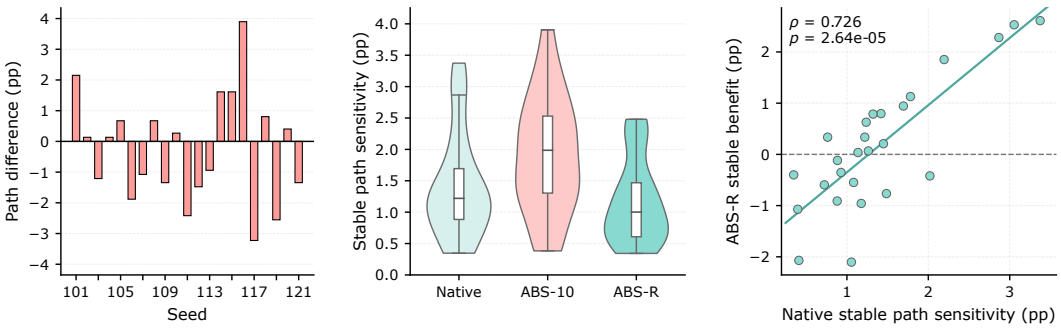
5.2 Control across the evaluated regimes

Across seven conditions, three backbones, and four strict schedule families, we evaluate six complementary controllers spanning scheduling, averaging, replay, and projection (Izmailov et al., 2018; Yu et al., 2020). Table 2 summarizes paired results; full breakdowns are in Appendix F.

Table 2: Path-control results against each method's matched baseline. PSR is relative PS reduction; ΔU and ΔW are changes in paired mean and minimum R@5.

Method	Compute	Pairs	PS↓	PSR↑	ΔU	ΔW
Baseline	1×	336	2.488	–	0	0
ABP	**1×**	**336**	**2.102**	**+15.5%**	**+0.076**	**+0.268**
replay2x	2×	336	2.292	+7.9%	-0.278	-0.180
PHR	2×	84	2.017	+18.7%	-0.664	-0.432
FPC	2×	336	3.259	-31.0%	+0.332	-0.054
SEA	2×	84	3.158	-27.3%	+0.455	+0.117
FBM	2×	84	2.827	-13.9%	+0.206	+0.033

The clearest compute-neutral result is **antithetic block pairing (ABP)**, which maps balanced four-block groups to ABBA or BAAB while retaining directional exposure. Across 336 matched pairs, PS falls from **2.488 to 2.102 pp** (**15.5%**), with a mean-utility change of $+0.076$ pp and a paired minimum-R@5 gain of **0.268 pp**. The paired bootstrap supports the PS reduction ($\Delta\text{PS} = -0.386$



(a) Order Effect (b) Path Control (c) Control Benefit

Figure 4: **Path dependence and antithetic scheduling in native MLA training. (a)** Native AV–VA order produces nonzero endpoint differences across all 21 fresh seeds, with either order favored in different runs. **(b)** Stable sensitivity distributions under native, ABS-10, and ABS-R schedules characterize the effect of schedule design; ABS-R has the lowest mean. **(c)** Baseline sensitivity and observed ABS-R benefit across 26 development and fresh seeds ($\rho = 0.726$). Benefit is native minus controlled sensitivity, so positive values indicate improved path robustness.

pp, 95% CI $[-0.738, -0.117]$). The eight-block families stay unchanged; in the active six-block families, PS falls from **2.572 to 1.801 pp** (**29.98%**).

The condition breakdown reveals a clear control regime. On **NYU RGB–Depth, PS decreases by 34.3%** with a **+0.96 pp** mean-utility change and reductions across all three backbones. The predefined active six-block subset also improves paired minimum R@5 by 0.537 pp. Other controllers illustrate the complementary robustness–utility trade-off: PHR reduces PS by **18.7%** but loses **0.664 pp** mean R@5, whereas FPC, SEA, and FBM have positive mean-utility point estimates while increasing aggregate PS.

Thus, schedule design provides a compute-neutral way to reduce average path sensitivity, with a small mean utility change in the ABP comparison. The condition and schedule breakdowns make the effective control regimes explicit. We next examine native sequential multimodal training, connecting this analysis to order effects and antithetic scheduling within an existing multimodal algorithm.

6 Path Dependence in Native Sequential Multimodal Training

The preceding sections expose directional order in a controlled setting. We finally ask whether the same behavior appears when sequential multimodal updates are already part of the training algorithm itself. We study MLA on CREMA-D, whose native procedure alternates audio and visual adaptation through a shared head. This setting lets us test whether both path dependence and the control structure identified above transfer beyond our diagnostic construction. Figure 4 summarizes the resulting phenomenon, intervention, and sensitivity structure.

6.1 Native order produces path-dependent endpoints

We compare the native Audio→Visual (AV) and Visual→Audio (VA) orders while matching initialization, stochastic stream, modality exposure, optimization settings, and training budget. As shown in Figure 4(a), all **21 fresh seeds** yield nonzero AV–VA endpoint differences, with a mean absolute gap of **1.421 pp** (95% CI [1.024, 1.863]). The sign varies across seeds: AV performs better in 11 cases and VA in 10, showing that the effect does not reduce to a fixed preference for one modality order. The difference also persists beyond the final checkpoint: native MLA gives a last-20-epoch sensitivity of **1.395 pp** and a stable sensitivity of **1.408 pp**. Sequential order therefore remains consequential when it is native to the multimodal training procedure rather than introduced only for diagnosis.

6.2 Antithetic scheduling in native training

Section 5 shows that antithetic ordering provides a practical lever for reducing path sensitivity. We instantiate the same principle within MLA as **Antithetic Balanced Scheduling (ABS)**, which

balances the native AV and VA adaptation orders while preserving total modality exposure. Unlike ABP, which operates on directional blocks in HeteroPathBench, ABS acts directly on MLA's native update sequence.

Figure 4(b) shows that antithetic schedule design shapes the sensitivity distribution. ABS-10 gives stable PS of **1.903 pp**, compared with the native **1.408 pp**; ABS-R gives **1.165 pp**, a **17.3%** lower mean, with a +0.326 pp utility point change. Its paired PS change is -0.243 pp (95% CI $[-0.775, +0.258]$), reflecting substantial seed-wise variation across the 21 fresh seeds under the frozen paired comparison. Figure 4(c) characterizes this variation across 26 development and fresh seeds: baseline sensitivity and observed benefit have Spearman correlation $\rho = 0.726$, with 0.706 on the fresh seeds alone. The descriptive tertile breakdown gives native-to-ABS-R PS of 0.709 to 1.519 pp in the low group and 2.308 to 1.038 pp in the high group. Appendix H.3 reports the paired statistics and association definitions.

These results extend the study beyond the controlled schedule space: native sequential multimodal training also produces order-dependent endpoints. The native comparison establishes the practical relevance of directional order, while the antithetic schedules characterize how its endpoint expression varies with schedule design and seed. Together, the results connect HeteroPathBench to an existing alternating multimodal algorithm. Full protocols and seed-level results are in Appendices G–H.

7 Related Work

Multimodal optimization and sequential training. Multimodal optimization is commonly framed as correcting unequal contribution or conflicting gradients. OGM-GE modulates gradients (Peng et al., 2022); sample-level valuation adapts reinforcement (Wei et al., 2024); MMPareto reconciles unimodal and multimodal objectives (Wei & Hu, 2024); and Data Remixing restructures batches (Ma et al., 2025). Recent work extends this to modality imbalance and challenging inputs (Gong et al., 2025; Wang et al., 2025b; Guan et al., 2026; Nguyen et al., 2026; Jiang et al., 2026). Sequential formulations make order explicit: MLA alternates unimodal adaptation (Zhang et al., 2024), while MedCoSS stages modalities for continual self-supervision (Ye et al., 2024). These methods alter gradients, exposure, or schedule design; we match directional data and augmentation exposure across legal update orders.

Training history and ordered composition. Training history can matter under curricula, reshuffling, and trajectory-aware influence (Bengio et al., 2009; Hacohen & Weinshall, 2019; Mishchenko et al., 2020; Wang et al., 2025a). Continual learning makes task sequence explicit (Lopez-Paz & Ranzato, 2017), while forgetting analyses reveal within-task histories (Toneva et al., 2019). Our construction matches invocation-specific content while varying its temporal composition. Splitting methods study non-commuting compositions and order-dependent error (Strang, 1968; McLachlan & Quispel, 2002). We use this only as language for ordered state transitions; our target is empirical endpoint sensitivity in stochastic multimodal optimization.

8 Conclusion

We study whether heterogeneous cross-modal updates yield the same endpoints when directional interactions are realized in different orders. Under matched direction-indexed data and augmentation exposure, path dependence appears broadly across seven sensor conditions and three backbone families, with substantial variation across learning regimes. Complete legal-schedule classes, exact repeats, and stochastic calibration establish its breadth, reproducibility, and practical scale. Same-state probes resolve local composition differences, while optimizer comparisons characterize the dependence of endpoint sensitivity on training dynamics. These measurements also guide practical interventions: compute-neutral ABP reduces mean path sensitivity by **15.5%**, with a small average utility change. Native MLA further reproduces order-dependent endpoints across 21 fresh seeds and reveals schedule-dependent sensitivity under antithetic balancing. Together, these results establish update order as a measurable and actionable optimization variable in sequential multimodal learning, provide a systematic protocol for studying its effects, and demonstrate a concrete schedule intervention that improves path robustness at the original update budget.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1247–1255, 2013.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gael Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, volume 3, pp. 747–769, 2021.

Giordano Cicchetti, Eleonora Grassucci, Luigi Sigillo, and Danilo Comminiello. Gramian multimodal representation learning and alignment. In *International Conference on Learning Representations*, 2025.

Christopher M. De Sa. Random reshuffling is not always better. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Benoit Dufumier, Javiera Castillo Navarro, Devis Tuia, and Jean-Philippe Thiran. What to align in multimodal contrastive learning? In *International Conference on Learning Representations*, 2025.

Yunfeng Fan, Wenchao Xu, Haozhao Wang, Junxiao Wang, and Song Guo. PMR: Prototypical modal rebalance for multimodal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20029–20038, 2023.

Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021. doi: 10.1109/LRA.2021.3068942.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15180–15190, 2023.

Baoquan Gong, Xiyuan Gao, Pengfei Zhu, Qinghua Hu, and Bing Cao. Multimodal negative learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Eleonora Grassucci, Giordano Cicchetti, Emanuele Frasca, Aurelio Uncini, and Danilo Comminiello. Closing the modality gap aligns group-wise semantics. In *International Conference on Learning Representations*, 2026.

Zhi-Hao Guan, Longfei Huang, and Yang Yang. Anchor-guided gradient alignment for incomplete multimodal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37896–37905, 2026.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2535–2544, 2019.

Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. OneLLM: One framework to align all modalities with language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26584–26595, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, 2018.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4904–4916. PMLR, 2021a.

Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, and Wenli Zhou. LLVIP: A visible-infrared paired dataset for low-light vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 3496–3504, 2021b.

Qing-Yuan Jiang, Longfei Huang, and Yang Yang. Rethinking multimodal learning from the perspective of mitigating classification ability disproportion. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Xun Jiang, Yufan Gu, Disen Hu, Yuqing Hou, Yazhou Yao, Fumin Shen, Heng Tao Shen, and Xing Xu. Multimodal learning on low-quality data with conformal predictive self-calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23041–23050, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6467–6476, 2017.

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

Xiaoyu Ma, Hao Chen, and Yongjian Deng. Improving multimodal learning balance and sufficiency through data remixing. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 41827–41836, 2025.

Robert I. McLachlan and G. Reinout W. Quispel. Splitting methods. *Acta Numerica*, 11:341–434, 2002.

Konstantin Mishchenko, Ahmed Khaled, and Peter Richtarik. Random reshuffling: Simple analysis with vast improvements. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Phuong-Anh Nguyen, Tien Anh Pham, Duc-Trong Le, and Cam-Van Thi Nguyen. BALM: A model-agnostic framework for balanced multimodal learning under imbalanced missing rates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30246–30256, 2026.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8238–8247, 2022.

Chengxuan Qian, Shuo Xing, Shawn Li, Yue Zhao, and Zhengzhong Tu. DecAlign: Hierarchical cross-modal alignment for decoupled multimodal representation learning. In *International Conference on Learning Representations*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Michael Schmitt, Lloyd Haydn Hughes, Chunping Qiu, and Xiao Xiang Zhu. SEN12MS: A curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W7: 153–160, 2019.

Ohad Shamir. Without-replacement sampling for stochastic gradient methods. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, pp. 746–760. Springer, 2012.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1139–1147, 2013.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Jiachen T. Wang, Dawn Song, James Y. Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. In *International Conference on Learning Representations*, 2025a.

Jun Wang, Fuyuan Cao, Zhixin Xue, Xingwang Zhao, and Jiye Liang. CMoB: Modality valuation via causal effect for balanced multimodal learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12695–12705, 2020.

Shicai Wei, Chunbo Luo, and Yang Luo. Improving multimodal learning via imbalanced learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2250–2259, 2025a.

Shicai Wei, Chunbo Luo, and Yang Luo. Boosting multimodal learning via disentangled gradient learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22879–22888, 2025b.

Yake Wei and Di Hu. MMPareto: Boosting multimodal learning with innocent unimodal assistance. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 52559–52572, 2024.

Yake Wei, Ruoxuan Feng, Zihe Wang, and Di Hu. Enhancing multimodal cooperation via sample-level modality valuation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27338–27347, 2024.

Yang Yang, Fengqiang Wan, Qing-Yuan Jiang, and Yi Xu. Facilitating multimodal classification via dynamically learning modality gap. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yiwen Ye, Yutong Xie, Jianpeng Zhang, Ziyang Chen, Qi Wu, and Yong Xia. Continual self-supervised learning: Towards universal multi-modal medical data representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11114–11124, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Xiaohui Zhang, Jaehong Yoon, Mohit Bansal, and Huaxiu Yao. Multimodal representation learning by alternating unimodal adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27456–27466, 2024.

Bo Zhou, Liulei Li, Yujia Wang, Huafeng Liu, Yazhou Yao, and Wenguan Wang. UNIALIGN: Scaling multimodal alignment within one unified model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29644–29655, 2025.

Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, Hongfa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Wancai Zhang, Zhifeng Li, Wei Liu, and Li Yuan. LanguageBind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *International Conference on Learning Representations*, 2024.

AI USE STATEMENT

Generative AI tools were used in a limited capacity to aid or polish writing and to support literature retrieval and discovery, including locating and reading relevant related work.

A Protocol and Experimental Details

This appendix specifies the frozen HeteroPathBench protocol used for the phenomenon and mechanism studies in Sections 2–4. The primary comparisons and their matching rules were fixed before the Final-V2 results were inspected. Unless a secondary track explicitly changes one factor, it inherits the data, model, objective, optimization, and evaluation conventions below.

A.1 Frozen scope and pre-control job inventory

The primary study crosses seven sensor conditions, three backbone families, five seeds, and four strict schedule families, giving 420 matched pairs (840 training runs). The complete frozen pre-control package contains 2,928 accepted jobs; the obsolete 210-job ProbeSelect branch is not included in any reported aggregate.

Table 3: Frozen pre-control evidence used by Sections 3–4.

Track	Jobs
Primary strict paths	840
Exact-repeat controls	63
Stochastic-stream controls	63
Highly alternating paths	90
Legal-schedule ensembles	540
Composition-depth sweep	240
Optimizer/state interventions	360
Alignment-strength sweep	360
Common-batch architecture controls	120
Same-state operator probes	252
Total	2,928

A.2 Conditions, manifests, and evaluation partitions

Every condition is represented by a frozen paired manifest with a training partition and one or more held-out partitions. Training reads only the `train` split. The primary endpoint statistic is always taken from the pre-specified evaluation split in Table 4; neither schedule construction nor training queries evaluation performance.

Table 4: Frozen sensor conditions and primary evaluation partitions.

Condition	Modality pair	Primary split
SEN12-clear	SAR–RGB	`ood_test`
SEN12-cloudy	SAR–RGB	`ood_test`
LLVIP	visible–IR	`test`
Eilat	RGB–sonar	`ood_test`
Sunboat	RGB–sonar	`ood_test`
NYU	RGB–depth	`id_test`
DSEC	RGB–event	`id_test`

The manifest builder preserves paired sample identity and, where source sequence/group information is available, uses deterministic group-aware partitions so that related samples remain in the same partition. Compatible frozen manifests are reused rather than re-split. For DSEC, the data pipeline accepts pre-rendered event frames; when raw event streams require rendering, events from the 50-ms interval ending at the RGB timestamp are deterministically converted into a three-channel image containing positive-event count, negative-event count, and per-pixel recency. No raw dataset is modified by manifest construction.

A.3 Input construction and stochastic views

All modalities retain the full field of view and are resized to 224×224; we deliberately avoid a shared random crop because several modality pairs are not pixel registered. Array-valued non-RGB sensors are deterministically converted to three channels and scaled using finite 1st–99th percentiles before image conversion. RGB-like inputs use ImageNet normalization, while other sensor images use mean/std $(0.5, 0.5, 0.5)$.

Each training pair produces two unimodal views per side (a_1, a_2, b_1, b_2) and one cross-modal view (a_c, b_c). The augmentation family is intentionally mild and is identical across matched paths once its operator-indexed token is fixed. RGB-like inputs use brightness and contrast factors sampled from $[0.8, 1.2]$, Gaussian blur with probability 0.25 and radius in $[0.1, 0.7]$, and additive Gaussian noise with standard deviation in $[0, 0.025]$. Other sensor inputs use contrast $[0.78, 1.22]$, brightness $[0.85, 1.15]$, blur radius $[0.1, 0.8]$ with probability 0.25, and noise standard deviation $[0, 0.04]$. Evaluation views are deterministic functions of pair identity, so every matched run is evaluated on identical tensors.

A.4 Encoder, objective, and optimization defaults

Both modalities use separate copies of the same ImageNet-pretrained backbone. The classifier/head is removed and followed by a modality-specific projection MLP, $d \to 256 \to 128$, with GELU between the two linear layers. The backbone-dependent batch sizes were frozen before the Final-V2 results; the separate common-batch control reruns ResNet18 and ConvNeXt-Tiny at batch size 32.

Table 5: Primary backbone and batch configuration.

Backbone	Projection dim.	Batch size
ResNet18	128	128
ConvNeXt-Tiny	128	64
ViT-B/16	128	32

At every optimizer step, both modality-specific InfoNCE terms remain active; the A/B route changes only which side receives the cross-modal alignment gradient defined in Section 2. The training objective is

$$\mathcal{L} = \mathcal{L}^{A}_{\text{uni}} + \mathcal{L}^{B}_{\text{uni}} + \lambda \mathcal{L}^{(r)}_{\text{align}}, \qquad r \in \{A, B\}, \tag{6}$$

with $\lambda = 5$, temperature 0.07, ℓ_2-normalized embeddings, and symmetric InfoNCE for the cross-modal term. The implementation normalizes each InfoNCE loss by $\log B$, where B is the batch size.

The primary optimizer is AdamW with constant learning rate 10^{-4} and weight decay 10^{-4}. Every long-horizon run uses 528 optimizer steps, BF16 autocast, and global gradient-norm clipping at 5.0. No learning-rate scheduler is used in the frozen Final-V2 trainer. The primary initialization/stream seeds are 10–14; within a strict pair, initialization and stream seed are identical.

A.5 Strict schedule construction and complete legal classes

A schedule is a short string over $\{A, B\}$ whose symbols are expanded into equal-length contiguous blocks. For a schedule of length L, each symbol is repeated for $528/L$ optimizer steps. The four pre-fixed primary families are shown in Table 6. Within each pair, the two schedules have the same length, A/B counts, first and final route, number of switches, and transition counts $(N_{AA}, N_{AB}, N_{BA}, N_{BB})$; only the global composition order differs.

Table 6: Pre-fixed strict schedule pairs.

Family	Path 1	Path 2	Start→end	Steps/block	Switches
6/B	`AABBAB`	`ABAABB`	A→B	88	3
6/A	`BBAABA`	`BABBAA`	B→A	88	3
8/B	`AABABBAB`	`ABAABBAB`	A→B	66	5
8/A	`BBABAABA`	`BABBAABA`	B→A	66	5

The primary matrix is therefore 7 conditions $\times 3$ backbones $\times 5$ seeds $\times 4$ families $= 420$ strict matched pairs. To measure dispersion beyond the pre-fixed pair, we also enumerate every schedule satisfying the same signature. Each 6-block class contains four legal schedules and each 8-block class contains nine. On five representative conditions (SEN12-clear, LLVIP, Eilat, NYU, and DSEC), two backbones (ResNet18 and ViT-B/16), and seeds 10–12, the 18 schedules not already present in the primary matrix complete four equivalence classes per condition/backbone/seed cell, yielding 120 complete legal classes. Enumeration uses only the schedule signature and never endpoint performance.

A.6 Operator-indexed coupling, determinism, and resume behavior

For route $r \in \{A, B\}$, the k-th invocation is assigned a deterministic batch from a hash of the stream seed, route, occurrence index, dataset size, and batch size. Each selected example receives an augmentation token derived from the same route/occurrence key; the dataset combines this token with pair identity to seed the six view-specific transforms. Sampling is without replacement within a batch unless the training set is smaller than the requested batch size. Consequently, reordering a legal path does not reassign minibatches or augmentation draws to logical directional invocations.

The initialization seed controls Python, NumPy, CPU PyTorch, and CUDA RNGs. cuDNN benchmarking is disabled, deterministic cuDNN behavior is enabled, and PyTorch deterministic algorithms are requested in warn-only mode. Checkpoints save model and optimizer states, histories, and RNG states for resumption. Final-V2 uses the initialization-seeded global model RNG stream and invocation-indexed data and augmentations. Final-6 additionally indexes model RNG by route and occurrence. Appendix H.1 specifies the study configurations.

A.7 Evaluation and reporting conventions

Evaluation loaders are unshuffled. Each available split is capped at 1,000 paired examples; if a split is larger, a deterministic evenly spaced subset is used. Retrieval uses ℓ_2-normalized embeddings and cosine similarity. The implementation records Recall@1/5/10 and mean rank in both directions, while the paper uses the bidirectional mean R@5 from Eq. (3) as the primary functional observable and PS from Eq. (4) as the primary path-dependence statistic.

For the 420 primary pairs we report the mean and median PS and threshold exceedance rates; the 0.01 pp threshold is used only as a numerical nonzero tolerance. The 95% interval for the headline mean is a cluster bootstrap with cluster = dataset $\times$ backbone $\times$ seed, so all four schedule-family observations from one such cell move together. Exact-repeat controls rerun the same schedule, seed, and stream; stochastic-stream controls keep the trained path and initialization fixed while changing only the operator-indexed stream seed. Complete legal classes are summarized by their max–min endpoint R@5 range.

For paired control methods, $PSR = 1 - PS_{\text{method}}/PS_{\text{baseline}}$ is always reported together with absolute ΔPS and task utility. The discovery-stage success criterion was frozen as $PSR \geq 30\%$ with mean-R@5 loss no worse than 0.5 pp; $2\times$ methods must additionally not be Pareto-dominated by replay2x. Single-path methods such as PBS are evaluated by utility and win rate and are never assigned an artificial zero PS.

B Relation to Monolithic Joint Updates

The main study concerns ordered compositions of directional responses, not an internal ordering ambiguity in an ordinary joint step. Let J denote the conventional unsplit update, which evaluates both encoder gradients from the same pre-update state and applies them together. By contrast, AB and BA are sequential realizations: the second directional response is recomputed after the first has changed the training state. Thus J, AB, and BA are different update objects, and our path-dependence claims compare the latter two (and longer legal compositions) rather than asserting that J secretly chooses one of them.

The same distinction sets the scope of the broader implication discussed in the main text. A monolithic joint step can hide directional structure because it never exposes a composition order; order becomes an explicit optimization variable only after those directional interactions are decomposed, staged, routed, or otherwise realized sequentially. We therefore use J only as a boundary/reference object: no J–AB or J–BA distance enters the primary path-dependence claims. Quantitatively relating sequential realizations to the unsplit joint update is a separate question from whether matched sequential paths are order-invariant.

C Additional Path-Dependence Characterization

C.1 Complete legal schedule classes

Across 120 complete legal equivalence classes, the mean max–min range is 6.34 pp, median 3.73 pp, and maximum 35.42 pp. These classes are exhaustively enumerated from the frozen schedule

signatures described in Appendix A; endpoint values are not used to choose schedules. The condition-wise means in Table 7 show that the dispersion seen in the headline pair is also present when each admissible class is opened more fully.

Table 7: Complete-class endpoint dispersion by condition. Values are mean max–min R@5 ranges (pp) over the complete legal classes used in Figure 3.

Condition	DSEC	NYU	SEN12-clear	LLVIP	Eilat
Mean range	14.50	8.45	3.82	3.26	1.68

The pre-fixed strict pair spans approximately 52% of the complete class range on average, so the broad phenomenon is not tied to selecting a favorable extreme pair after training.

C.2 Alternation, alignment strength, and composition depth

Table 8: Secondary schedule characterization. Values are mean absolute R@5 gaps unless noted.

Setting	Values / levels	Result (pp)
Highly alternating	ABABAB vs. BABABA	4.01
Alignment strength	$\lambda = 1, 2.5, 5, 10$	3.66 / 2.30 / 2.96 / 3.91
Composition depth	6 / 8 / 12 / 16 blocks	3.11 / 3.13 / 3.74 / 4.12
Legal schedule class	complete class range	6.34 (mean)

The alignment-strength sweep has no monotone trend, whereas the depth sweep increases in aggregate from 6 to 16 blocks; neither statement assumes sample-wise monotonicity. The alternating pair changes schedule geometry while retaining equal A/B exposure, showing that the effect is not specific to coarse all-A-then-all-B blocks.

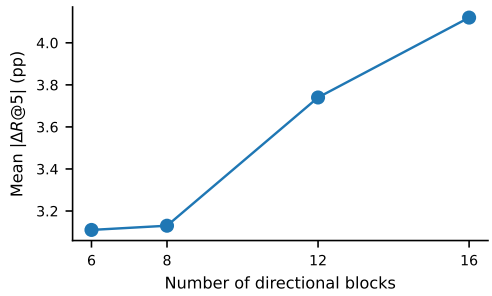


Figure 5: Composition depth increases path sensitivity on average in this sweep; no sample-wise monotonicity is claimed.

C.3 Numerical and stochastic calibration

All 63/63 exact repeats match exactly in R@5, model parameters, optimizer state, and model buffers. On the matched calibration subset, changing path gives 2.73 pp while changing only the stochastic stream gives 2.50 pp; these matched comparisons establish comparable scales for order-induced and stream-induced variability. Common-batch-32 architecture controls yield 2.87/2.26/2.89 pp for ResNet18/ConvNeXt-Tiny/ViT-B/16, compared with 2.48/2.30/2.65 pp in the primary aggregate. Together with Table 8, these controls cover the reliability, strength, composition-depth, alternation, and batch-size checks referenced in Section 3.

D Operator and Optimizer Diagnostics

D.1 Same-state operator probes

The 252 probes span 7 conditions $\times$ 3 backbones $\times$ 3 seeds $\times$ 4 checkpoints at steps 0, 132, 264, and 396. Each probe starts from one saved training state and compares AB with BA while recomputing the second gradient after the first update. The local diagnostic uses stateless FP32 SGD with learning rate 10^{-5}, zero momentum, and zero weight decay; BatchNorm running statistics are frozen during the probe. In the same-batch probe, both routes use the same deterministic minibatch; the paired-batch variant uses the deterministic route-indexed batches assigned to A and B.

For floating tensors in the model state, the reported relative discrepancy is

$$d_{\mathrm{rel}}(X,Y) = \left(\frac{\sum_j \|\theta_j^X - \theta_j^Y\|_2^2}{\sum_j \|\theta_j^X\|_2^2 + 10^{-30}} \right)^{1/2}. \tag{7}$$

Every AB–BA comparison is nonzero. Median relative discrepancy decreases from approximately 1.74×10^{-7} at step 0 to 4.15×10^{-8}, 1.33×10^{-8}, and 9.52×10^{-9} at steps 132, 264, and 396. This quantity measures local two-update separation; Section 4 pairs it with endpoint PS to characterize order sensitivity at both update and task levels.

D.2 Optimizer and model-state controls

The long-horizon controls inherit the same matched schedules and operator-indexed streams as the main protocol and vary only the stated optimization component. Standard AdamW yields about 2.96 pp path sensitivity, while momentum-free SGD yields 0.38 pp. The SGD control uses the track's matched learning rate and weight decay with momentum set to zero. In the AdamW-reset control, model parameters are retained but the optimizer state is cleared at every new schedule-block boundary; its mean PS is 3.09 pp. The BatchNorm control places BatchNorm modules in evaluation mode so running mean/variance no longer update while affine parameters remain trainable; ViT has no corresponding running-state control. These interventions characterize substantial optimizer dependence and the persistence of endpoint path sensitivity across the tested state-handling conventions.

E Systematic Path-Control Study

The discovery study contains 2,016 frozen jobs across three intervention levels. Each round uses fresh development cells and retains the phenomenon protocol's 528 unique directional occurrences, balanced A/B exposure, operator-indexed stochastic content, and matched hardware affinity. The discovery criterion is fixed throughout: a paired method is eligible when PSR is at least 30% and mean-R@5 loss is no worse than 0.5 pp; a $2\times$ method must additionally not be Pareto-dominated by replay2x. Single-path PBS is evaluated only by utility and win rate.

E.1 Adaptive-state isolation

The first round changes how AdamW memory is shared across routes while keeping one gradient evaluation per update. Development uses SEN12-clear, LLVIP, and Eilat; ResNet18 and ViT-B/16; and seeds 21–23, with the remaining conditions held out during this screen. PBS is a purely structural scheduler: within the same schedule-signature equivalence class it lexicographically minimizes maximum prefix A/B imbalance, then cumulative prefix imbalance, without consulting model scores or validation results. Sep-M keeps route-specific first moments with a shared second moment; Sep-V does the converse; Sep-MV separates both moments with route-specific clocks. Sep-MV-Decay additionally decays inactive-route moments on every global step, preserving the global AdamW memory time scale while removing cross-route gradient injection. Sep-MV+PBS combines the separated state with the structural PBS schedule.

Table 9: Adaptive-state study (864 jobs). ΔU is mean-R@5 change relative to baseline.

Method	PS (pp)	PSR (%)	ΔU (pp)
baseline	1.65	0.00	0.00
Sep-M	1.59	3.93	-3.40
Sep-V	1.49	9.92	-9.96
Sep-MV	1.59	3.53	-1.66
Sep-MV-Decay	1.63	1.54	-1.72
PBS	–	–	-0.08
Sep-MV+PBS	–	–	-1.16

PBS beats the matched strict-pair mean in 56.9% of cells. The paired state-isolation variants fall below the frozen eligibility criterion, motivating the subsequent composition-level study.

E.2 Composition and functional symmetrization

The second round acts on pair/block composition. Development uses SEN12-clear, LLVIP, and Eilat; ConvNeXt-Tiny and ViT-B/16; seeds 41–43; and the 6/B and 8/A strict families. ResNet18 is held

out here because common-state and branch symmetrization interact with running-state normalization. ABP maps every four-block 2A/2B group to ABBA or BAAB according only to its first block, preserving operator counts and total calls. SBS replaces an adjacent XY block pair by $X/2, Y, X/2$ when $X \neq Y$, where the halves partition the original X invocations. **Common-State Window (CSW)** evaluates two consecutive gradients at the same pre-update state and then applies them in their original order. SEA runs original and reversed branches from an identical full state and averages model and AdamW states. FBM uses the same two branches but selects the merge weight from $\{0, .25, .5, .75, 1\}$ on a deterministic training-only anchor batch. FPC starts from the SEA merge and applies one small representation-consistency correction per block pair using the training anchor; its correction learning rate is one quarter of the base rate. replay2x replays each of the same 528 unique operator batches twice with half learning rate and square-root Adam betas and serves as the $2\times$ compute/data-presentation control.

Table 10: Composition/functional study (576 jobs).

Method	PS (pp)	PSR (%)	ΔU (pp)
baseline	1.47	0.00	0.00
ABP	1.57	-6.85	+0.03
SBS	1.75	-19.29	-0.56
CSW	2.23	-52.04	-1.95
SEA	2.21	-50.74	+1.87
FBM	2.63	-79.38	+1.62
FPC	2.32	-58.10	+2.02
replay2x	1.52	-3.35	+0.37

Diagnostic cells nevertheless reveal substantial regime-specific control: Eilat+ConvNeXt+FPC reaches 55.3% PSR with +0.23 pp utility; Eilat+ConvNeXt+SEA reaches 38.3% with +0.28 pp; SEN12-clear+ConvNeXt+ABP reaches 43.5% with +0.01 pp; LLVIP+ViT+FBM reaches 53.6% with +2.66 pp. These cells motivate the frozen cross-regime validation in Appendix F; they complement the aggregate discovery results.

E.3 Direct operator-level control

The final discovery round intervenes closest to the local pairwise update and moves to fresh development cells: SEN12-cloudy, Sunboat, and NYU; ConvNeXt-Tiny and ViT-B/16; seeds 61–63; and the 6/B and 8/A strict families. CPJ evaluates both pair gradients at a common pre-pair state, averages them, and uses that canonical direction for the two AdamW transitions. PCJ first applies symmetric PCGrad projection only when the same-state gradients have negative dot product, then averages them. PHR uses the palindromic sequence $X/2, Y/2, Y/2, X/2$ with half learning rate and square-root Adam betas, reusing the exact stochastic operator on each half; the resulting update is a discrete symmetric composition. EMC evaluates both gradients at the common state, makes a virtual clipped-mean midpoint move without touching optimizer state, reevaluates both there, restores the real state, and uses the midpoint mean for the two real transitions. CESG averages the common gradients together with the two cross-evaluated gradients after virtual X and Y displacements. COG uses the same cross-evaluations to estimate $c = (g_Y|X - g_Y) - (g_X|Y - g_X)$ and projects the CESG direction orthogonally to c before the real transitions. replay2x again supplies the $2\times$ control.

Table 11: Direct operator study (576 jobs).

Method	PS (pp)	PSR (%)	ΔU (pp)
baseline	1.96	0.00	0.00
CPJ	2.19	-11.70	-4.14
PCJ	2.50	-27.39	-3.93
PHR	1.96	-0.11	-0.88
EMC	2.46	-25.68	-3.77
CESG	2.33	-18.91	-4.75
COG	2.29	-16.95	-4.02
replay2x	1.92	+2.05	-0.76

Local signals include Sunboat+ConvNeXt+PHR (44.9% PSR, +0.08 pp utility), Sunboat+ConvNeXt+CPJ (42.9%, +0.05 pp), SEN12-cloudy+ConvNeXt+PCJ (34.5%, -0.38 pp), and Sunboat+ViT+PHR (30.8%, -0.26 pp). The three discovery rounds therefore serve as a mechanism-spanning screen, while the final claims rely on the separately frozen full-space study below.

F FULL-SPACE CONFIRMATORY VALIDATION

The final confirmatory study was frozen before observing outcomes. Within this study, all methods use the same linear projection head and route-indexed model RNG; the complete study configuration is given in Appendix H.1. It evaluates PBS, ABP, FPC, SEA, FBM, and PHR together with baseline and replay2x over all 7 conditions $\times$ 3 backbones $\times$ 4 strict schedule families. Core methods use seeds 71–74; supplementary methods use seed 71 but retain the full condition/backbone/family grid. The run completed with 3,528/3,528 valid jobs, zero errors, zero warnings, and passing same-GPU checks for comparison cells and matched pairs.

F.1 JOB ALLOCATION AND AGGREGATE RESULTS

Table 12: Frozen validation job allocation.

Method	Jobs	Role
baseline	672	control
replay2x	672	compute-matched control
PBS	336	core
ABP	672	core
FPC	672	core
SEA	168	supplementary
FBM	168	supplementary
PHR	168	supplementary
Total	3,528	

Across Final V2 (2,928 jobs), the three frozen discovery rounds (864+576+576), and Final-6 (3,528), the pre-MLA study uses 8,472 frozen/confirmatory jobs. The three MLA rounds add 186 non-smoke runs, bringing the updated frozen/confirmatory total to 8,658; smoke/debug runs and superseded development branches remain excluded.

Table 13: Final-6 paired-method summary. Values are computed against each method's matched baseline subset. ΔU and ΔW denote changes in mean and worst-path R@5.

Method	Pairs	PS	PSR	ΔU	ΔW	PS win
baseline	336	2.488	–	0	0	–
ABP	336	2.102	+15.5%	+0.076	+0.268	28.9%*
replay2x	336	2.292	+7.9%	-0.278	-0.180	50.3%
PHR	84	2.017	+18.7%	-0.664	-0.432	50.0%
FPC	336	3.259	-31.0%	+0.332	-0.054	42.0%
SEA	84	3.158	-27.3%	+0.455	+0.117	36.9%
FBM	84	2.827	-13.9%	+0.206	+0.033	42.9%

*ABP is an exact structural no-op on the two eight-block families, so half of the full-matrix comparisons are ties by construction.

PBS is a single-path controller rather than a paired robustness method. Relative to the matched strict-pair mean, its ΔR@5 is +0.115 pp with 53.3% win rate and a cluster-bootstrap 95% CI of approximately $[-0.227, +0.471]$ pp. PBS thus provides a single-path scheduling reference measured by utility and win rate.

F.2 FULL CONDITION AND BACKBONE BREAKDOWNS

Tables 14–15 report every paired controller at the two aggregation levels used in the main paper. Each entry is "PSR / ΔU" in percentage and percentage points; values are always computed against the same-key baseline subset, so supplementary one-seed methods are never compared with a different baseline population.

Table 14: Final-6 by condition: PSR (%) / ΔU (pp).

Condition	ABP	replay2x	PHR	FPC	SEA	FBM
DSEC	+5.6/-0.56	+23.1/+0.07	+32.4/-1.48	-24.5/-1.37	-11.8/-0.52	+14.7/-1.30
Eilat	+8.1/+0.04	-3.4/-0.19	+11.2/-0.13	-24.5/+0.65	-76.5/+0.74	-66.5/+0.86
LLVIP	+10.9/+0.02	+7.0/-0.04	+15.7/-0.61	-62.2/+1.43	-32.3/+1.41	-33.5/+1.61
NYU	+34.3/+0.96	+5.8/-1.96	+4.7/-2.91	-17.8/-1.07	-36.1/-1.06	-24.1/-1.85
SEN12-clear	-2.4/+0.01	-39.8/+0.25	+17.8/+0.98	-74.2/+2.19	-25.7/+1.93	-29.3/+1.59
SEN12-cloudy	+18.0/+0.06	+14.9/-0.06	+37.4/-0.33	-34.1/+0.46	-8.4/+0.74	-17.3/+0.59

Condition	ABP	replay2x	PHR	FPC	SEA	FBM
Sunboat	-2.5/+0.02	-1.9/-0.01	-37.3/-0.17	-18.3/+0.03	-127.1/-0.06	-39.0/-0.06

Table 15: Final-6 by backbone: PSR (%) / ΔU (pp).

Backbone	ABP	replay2x	PHR	FPC	SEA	FBM
ConvNeXt-T	+10.5/-0.05	+16.5/-0.62	+26.4/-0.74	-17.7/-0.07	-32.4/-0.00	-5.4/+0.21
ResNet18	+23.6/+0.14	+20.2/-0.23	+30.9/-0.48	+1.5/-0.62	-14.2/-0.87	-35.2/-0.44
ViT-B/16	+12.9/+0.13	-8.3/+0.01	+7.5/-0.77	-66.7/+1.70	-30.1/+2.24	-9.4/+0.85

These breakdowns expose the regime dependence directly: reductions need not track utility changes, and the same controller can occupy different parts of the robustness–utility plane across conditions and backbones. The four strict schedule families show the same heterogeneity:

Table 16: Final-6 by strict schedule family: PSR (%) / ΔU (pp).

Family	ABP	replay2x	PHR	FPC	SEA	FBM
8/A	0.0/+0.00	+7.6/-0.26	-2.7/-0.92	-55.6/+0.59	-65.4/+0.38	-35.4/+0.21
8/B	0.0/+0.00	+20.8/-0.42	+25.2/-1.05	-14.8/+0.33	+11.5/+0.59	+40.0/+0.32
6/A	+20.8/+0.31	-0.3/-0.18	+25.5/+0.80	-32.3/+0.15	-4.2/+0.71	-15.7/+0.27
6/B	+38.2/-0.01	+2.3/-0.26	+19.2/-1.49	-27.5/+0.26	-72.8/+0.13	-57.2/+0.02

The zero entries for ABP on both eight-block families follow directly from the unchanged schedules under its block mapping.

F.3 ABP: Structural scope and sensitivity dependence

Across all 336 pairs, ABP changes PS from 2.488 to 2.102 pp (15.5% PSR; ΔPS=-0.386 pp). The frozen cluster-bootstrap 95% CI for ΔPS is $[-0.738, -0.117]$ pp. Mean R@5 changes by +0.076 pp (95% CI $[-0.111, +0.291]$ pp), and worst-path R@5 by +0.268 pp. Hence the robustness reduction is supported by the frozen bootstrap, while the utility point estimate is best described as preserved.

Table 17: ABP structural-scope analysis. The active six-block subset is predefined by whether the block permutation changes the schedule.

Scope	Pairs	Base PS	ABP PS	PSR	ΔU
All families	336	2.488	2.102	15.50%	+0.076
Active six-block	168	2.572	1.801	29.98%	+0.151
No-op eight-block	168	2.403	2.403	0.00%	0.000

The active six-block subset also improves worst-path R@5 by +0.537 pp and has a 57.7% PS win rate. NYU provides the clearest cross-backbone signal: PSR is 27.0% on ConvNeXt-Tiny, 34.7% on ResNet18, and 37.8% on ViT-B/16, with corresponding mean-utility changes of +0.35, +0.98, and +1.55 pp. The baseline-PS/ABP-PS-change Spearman correlation is -0.383; the highest baseline-sensitivity tertile has 25.6% PSR. The paired reductions and descriptive sensitivity summaries together characterize the observed condition dependence (Appendix H.3).

F.4 Utility–robustness and compute controls

PHR reduces PS by 18.7% but changes mean R@5 by -0.664 pp and worst-path R@5 by -0.432 pp; in the final $2\times$ comparison it is Pareto-dominated by replay2x. Conversely, FPC, SEA, and FBM have mean-utility point estimates of +0.332, +0.455, and +0.206 pp while their aggregate PS increases. Their frozen cluster confidence intervals for mean utility change cross zero. Together with the condition/backbone tables, this is the full paired breakdown supporting the main-text claim that utility and path robustness are distinct objectives and that control is regime-dependent.

G Published-Algorithm Validation on MLA

This appendix records the frozen MLA development and confirmation evidence used in Section 6. We retain the published MLA model and modality-preprocessing pipeline (Zhang et al., 2024); the intervention changes only the order scheduler and the matched run controls specified here. MLA trains audio–video representations through alternating unimodal adaptation and a shared prediction head. We use CREMA-D with the fixed 6,698/744 train/test split, 100 epochs, batch size 64, and four data workers. Each epoch contains 105 minibatches. Optimization uses SGD with learning rate 10^{-3},

momentum 0.9, and weight decay 10^{-4}; the learning rate is multiplied by 0.1 at epoch 70, and the frozen MLA mixing coefficient is $\alpha_{AV} = 0.55$.

G.1 Matched orders and antithetic schedules

For each seed, all compared modes share the same initialization digest and stochastic-stream digest. Native-AV applies AV on every minibatch and Native-VA applies VA on every minibatch, giving 10,500 native order decisions over 100 epochs. ABS-10 and ABS-R are paired antithetic realizations: each plus/minus pair uses exactly 5,250 AV and 5,250 VA decisions, the two realizations are complements, and total audio/visual route exposure remains equal. ABS-10 uses the fixed-scale antithetic scheduler selected in development; ABS-R uses the frozen randomized antithetic scheduler. The archive records the exact realized schedule by a schedule digest for every run, so confirmation reuses the frozen implementation rather than reconstructing schedules from endpoint results. Round 3 additionally pins all six runs for a seed to the same physical GPU. Four partial smoke metrics in the final archive are excluded from all statistics and job counts.

Final PS is the absolute final-epoch gap; last-20 PS averages the 20 epoch-wise absolute gaps; stable PS averages these two measures. Utility uses the corresponding paired mean accuracies. The exploratory retrospective selector uses the native epoch-21–30 test-accuracy gap, choosing the completed ABS-R trajectory when this gap exceeds the frozen 7 pp threshold (Appendix H.2).

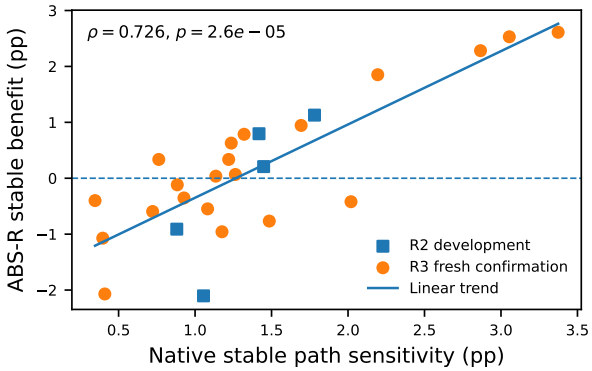


Figure 6: **Baseline sensitivity and observed ABS-R benefit.** Each point represents one CREMA-D seed: 5 development and 21 fresh-confirmation seeds use the same ABS-R schedule. Benefit is native minus ABS-R stable PS, with positive values denoting improved path robustness. The observed Spearman correlation is 0.726; association definitions and paired intervals are given in Appendix H.3.

G.2 Development and fresh-seed confirmation

The MLA evidence was accumulated in three frozen rounds. Round 1 (20 non-smoke runs, seeds 81–85) establishes the native AV/VA order effect and evaluates an earlier deterministic antithetic scheduler. Round 2 (40 runs on the same five development seeds) freezes the scale variants and the exact ABS-R implementation used below; Round 3 is the 126-run fresh-seed confirmation. The development analysis reports native final, last-20, and stable PS of 1.344, 1.289, and 1.317 pp. Among the fixed scheduling variants, ABS-10 satisfies the frozen development eligibility rule and is selected for fresh-seed confirmation. ABS-R is retained as a separately frozen randomized antithetic realization because the development data already show a sensitivity–benefit association.

Table 18: Development scheduling variants. Positive PSR means lower stable path sensitivity.

Variant	Stable PSR	Final ΔU	Last20 ΔU	Stable wins
ABS-1	-20.2%	-0.282	-0.368	2/5
ABS-10	+6.7%	-0.134	+0.166	3/5
ABS-E	+15.9%	-0.202	-0.405	4/5
ABS-R	-13.4%	+0.753	+0.753	3/5

Fresh confirmation uses 21 new seeds (101–121), with 126/126 runs complete. Native final PS is 1.421 pp (95% CI [1.024, 1.863]), last-20 PS is 1.395 pp, and stable PS is 1.408 pp; AV is better on 11 seeds and VA on 10. ABS-10 increases stable PS to 1.903 pp. ABS-R gives 1.165 pp (17.28% PSR),

with +0.326 pp stable utility; its paired PS-change CI is $[-0.775, +0.258]$ pp. The retrospective 7 pp selector gives 1.183 pp and +0.050 pp utility, with a PS-change CI of $[-0.608, +0.066]$ pp. The paired intervals and seed-level values summarize the magnitude and variability of these schedule effects.

G.3 Seed-level ABS-R summary and sensitivity dependence

Table 19 reports every seed contributing to Figure 6. Benefit is native stable PS minus ABS-R stable PS, so positive values denote reduced sensitivity.

Table 19: Seed-level native and ABS-R stable path sensitivity (pp).

Split	Seed	Native	ABS-R	Benefit
Dev	81	1.418	0.622	+0.796
Dev	82	1.055	3.159	-2.103
Dev	83	1.781	0.652	+1.129
Dev	84	0.880	1.791	-0.911
Dev	85	1.448	1.240	+0.208
Fresh	101	2.019	2.440	-0.420
Fresh	102	0.397	1.468	-1.072
Fresh	103	1.176	2.134	-0.958
Fresh	104	0.410	2.480	-2.070
Fresh	105	0.884	1.001	-0.118
Fresh	106	1.237	0.608	+0.628
Fresh	107	0.927	1.280	-0.353
Fresh	108	0.722	1.317	-0.595
Fresh	109	1.220	0.884	+0.336
Fresh	110	0.346	0.746	-0.400
Fresh	111	2.194	0.343	+1.851
Fresh	112	1.321	0.534	+0.786
Fresh	113	1.082	1.630	-0.548
Fresh	114	1.263	1.196	+0.067
Fresh	115	1.485	2.251	-0.766
Fresh	116	3.054	0.524	+2.530
Fresh	117	3.374	0.763	+2.611
Fresh	118	0.763	0.427	+0.336
Fresh	119	2.866	0.585	+2.282
Fresh	120	1.694	0.749	+0.944
Fresh	121	1.136	1.099	+0.037

Across all 26 development and fresh-confirmation seeds, baseline stable PS correlates with observed ABS-R benefit ($\rho = 0.726$; seed-bootstrap 95% CI [0.408, 0.916]); the fresh-seed correlation is 0.706. A descriptive rank-tertile split gives native-to-ABS-R PS of 0.709 to 1.519 pp in the low group (9 seeds), 1.256 to 1.105 pp in the middle group, and 2.308 to 1.038 pp in the high group (8 seeds). These descriptive summaries characterize the observed sensitivity–benefit pattern; Appendix H.3 defines the shared-baseline association and reports the paired treatment-effect intervals.

G.4 Updated job accounting

The three MLA rounds contribute 20, 40, and 126 non-smoke runs, respectively. Together with the pre-MLA total of 8,472 frozen/confirmatory jobs, the updated total is 8,658.

H Implementation and Statistical Details

This appendix specifies the update conventions, native schedules, and statistical definitions used throughout the paper. All results are computed from the frozen study archives.

H.1 Update and randomness conventions

Directional updates. The alignment-gradient factors for encoders (A, B) are $(0, 1)$ under route A and $(1, 0)$ under route B. Both unimodal InfoNCE losses remain active at every step; the route selects the recipient of the cross-modal gradient. Minibatches and augmentation tokens are indexed by stream seed, route, and occurrence in both Final-V2 and Final-6.

Study configurations. Final-V2 uses modality-specific $d \to 256 \to 128$ projection MLPs and an initialization-seeded global model RNG stream, saved and restored with checkpoints. ConvNeXt stochastic-depth masks follow this execution stream. The configured ResNet18 and ViT-B/16 forwards have no active stochastic-depth/dropout contribution; their mean endpoint PS values are 2.48 and

2.65 pp, respectively (Table 1). Final-6 uses linear $d \to 128$ projections and seeds the CPU/CUDA model RNG before each invocation using a hash of the stream seed, route, and occurrence. Every controller is compared with its matched baseline under the same study configuration.

Local diagnostic. The common-checkpoint probe uses FP32 SGD with learning rate 10^{-5}, zero momentum and weight decay, and frozen BatchNorm running statistics. The second gradient is recomputed after the first update. Model RNG proceeds through the two probe branches in execution order. All 252 same-batch comparisons yield nonzero parameter differences, including all 84 ResNet18 and all 84 ViT-B/16 comparisons with deterministic forwards. This local measurement complements the complete-schedule endpoint comparisons under AdamW.

H.2 Native schedules and paired estimates

Schedule construction. Let $g = 0, \ldots, 10499$ index minibatch order decisions. ABS-10 uses bit $\lfloor g/10 \rfloor \bmod 2$, where 0 denotes AV and 1 denotes VA. ABS-R shuffles 5,250 zeros and 5,250 ones once with NumPy `default_rng(20260920)`. Each method pairs this sequence with its bitwise complement. The frozen ABS-R realization is shared across model/stream seeds. Both realizations retain the native algorithm's total audio/visual exposure.

Endpoint and bootstrap definitions. For paired test accuracies a_t, b_t in percentage points, final PS is $|a_{100} - b_{100}|$, last-20 PS is $20^{-1} \sum_{t=81}^{100} |a_t - b_t|$, and stable PS averages these two quantities. Utility uses paired mean accuracies under the corresponding endpoint convention. Intervals use 100,000 paired seed resamples, with RNG seed 20260920 for PS changes and 20260921 for utility changes.

Table 20: Fresh-seed MLA results over 21 seeds. Changes are method minus native, in percentage points; brackets give paired seed-bootstrap 95% percentile intervals.

Method	Stable PS	ΔPS [95% CI]	ΔU [95% CI]
Native	1.408	–	–
ABS-10	1.903	$+0.495$ $[0.022, 0.982]$	$+0.174$ $[-0.215, 0.544]$
ABS-R	1.165	-0.243 $[-0.775, 0.258]$	$+0.326$ $[-0.216, 0.857]$
Selective (retrospective)	1.183	-0.225 $[-0.608, 0.066]$	$+0.050$ $[-0.280, 0.444]$

ABS-R improves stable PS on 11/21 seeds and ABS-10 on 7/21. Mean stable utility is 73.376% for native scheduling and 73.702% for ABS-R. Table 20 presents the mean schedule effects together with their seed-level uncertainty.

Retrospective selection summary. The exploratory selector uses the native epoch-21–30 test-accuracy gap and a 7 pp threshold frozen before confirmation. It selects the completed ABS-R trajectory above the threshold and the completed native trajectory otherwise. This retrospective summary is reported separately from the direct native and universal-schedule comparisons; its computation includes both completed trajectories.

H.3 Sensitivity and benefit summaries

For baseline PS X and controlled PS Y, benefit is $B = X - Y$. The association summaries therefore share the baseline term; in particular,

$$\mathrm{Cov}(X, B) = \mathrm{Var}(X) - \mathrm{Cov}(X, Y). \tag{8}$$

The rank correlations and sensitivity tertiles describe the observed distribution of benefits, while paired mean-change intervals quantify the average schedule effect. MLA has baseline–benefit Spearman correlations of 0.726 across all 26 seeds and 0.706 across the 21 fresh seeds. The early epoch-21–30 test-history gap has an exploratory correlation of 0.300 with later ABS-R stable benefit. For ABP, the baseline-PS/change correlation is -0.383. These association summaries accompany the condition, backbone, and schedule breakdowns.

H.4 ABP MAPPING AND CONTROL POPULATIONS

ABP partitions the block string into consecutive non-overlapping groups of four. A complete group containing two A and two B blocks maps to ABBA if it begins with A, and BAAB otherwise. Other groups and incomplete remainders stay unchanged. Block duration is retained, and per-route occurrence indices are assigned to the resulting sequence, preserving total calls and directional exposure.

Table 21: ABP on the predefined strict pairs. Distance counts differing block positions between the paired paths.

Family	Original pair	Transformed pair	Distance
6/B	AABBAB / ABAABB	ABBAAB / ABAABB	$4 \to 2$
6/A	BBAABA / BABBAA	BAABBA / BABBAA	$4 \to 2$
8/B	AABABBAB / ABAABBAB	unchanged	$2 \to 2$
8/A	BBABAABA / BABBAABA	unchanged	$2 \to 2$

The reported effect is the paired endpoint change under this complete schedule transformation. Its active six-block subset and unchanged eight-block subset follow directly from the predefined mapping. In Final-6, baseline mean R@5 is 23.507% and ABP mean R@5 is 23.582%; $\Delta U = +0.076$ pp with 95% CI $[-0.111, +0.291]$ pp. For each compared path pair, minimum utility satisfies $W = U - \mathrm{PS}/2$ and $\Delta W = \Delta U - \Delta\mathrm{PS}/2$.

PHR, SEA, and FBM use the 84-pair seed-71 subset. On this same subset, replay2x has PS 1.960 pp and mean R@5 23.154%, compared with 2.017 pp and 22.724% for PHR, giving replay2x the stronger aggregate robustness–utility result. Table 2 reports replay2x over all 336 pairs; each method's changes are computed against its own matched baseline population.