When Does Adaptive Test-Time Augmentation Pay Off? A Matched Execution Study in Dense Vision

Anonymous authors
Paper under double-blind review

Abstract

Adaptive test-time augmentation (TTA) can allocate computation usefully and run faster without improving quality–runtime utility. We develop a comparator-aware procedure that measures complete systems and translates utility deficits into improvement requirements. A frozen SegFormer-B2 router on 400 held-out ADE20K images exhibits two effects across RTX 3090, T4 and A100 environments. At 0.8 cross-entropy (CE)/second, A100 batching increases percentage savings over multiscale from 24.7% to 28.3%, but reduces absolute savings from 19.24 to 10.12 ms/image, below the 17.48 ms required by its quality penalty. On T4, routing beats multiscale in utility but loses to native inference, the best static at that preference. The recorded batching reversal on RTX 3090 and A100 holds throughout $[0.75, 0.95]$ CE/second, not only at the historical price. Removing observation/head computation alone leaves substantial deficits; a 2,000-image cross-fitted evaluation retains negative batch-8 mean surplus. Prospectively frozen ACDC calibration changes 30.0% of decisions, but runtime absorbs 85.2% of the CE gain, leaving superiority unresolved. Matched execution identifies the relevant comparators and the improvement needed to outperform them.

1 Introduction

Test-time augmentation (TTA) evaluates transformed inputs and combines their predictions. An adaptive policy reserves additional views for images expected to benefit. Prior work learns which transformations to apply, how to aggregate them, or how many passes to execute (Kim et al., 2020; Shanmugam et al., 2021; Mocerino et al., 2021). However, useful allocation and fewer inference passes do not by themselves establish a favorable quality–runtime trade-off for the executed system.

We fix a quality–runtime exchange rate λ, called the *price*, and define utility as negative cross-entropy (CE) minus priced runtime. A *static* policy applies one action to every image. *Surplus* is the router's utility minus the evaluation-best static utility in a declared menu; positive surplus favors routing. This benchmark differs from a static selected on calibration data for deployment. Image-averaged CE supports additive gain modeling; pooled mIoU can favor a different policy. Relative runtime and the best static both depend on the execution environment.

Figure 1 illustrates the central distinction: avoiding inference passes saves time, but the saving may not compensate for the quality penalty. Removing a measurable observation/head cost need not close that deficit.

Our ADE20K study transfers one fitted SegFormer-B2 router across RTX 3090, T4 and A100 without refitting or changing the price (Xie et al., 2021; Zhou et al., 2017). Routing improves allocation over count-matched random assignment, yet RTX 3090 and A100 favor routing at batch 1 and static inference at batches 8/32 across a shared price range. T4 instead shows why the comparator matters: routing beats multiscale but loses to native inference. The fit–select–refit–evaluate chronology and proxy-oracle comparison are retained in Appendix A.

On ACDC, prospective proxy and complete-cost calibration use the same frozen candidates and preference. Their selections differ on 122/406 final images, but the utility advantage remains unresolved: changing a policy does not establish superiority.

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.

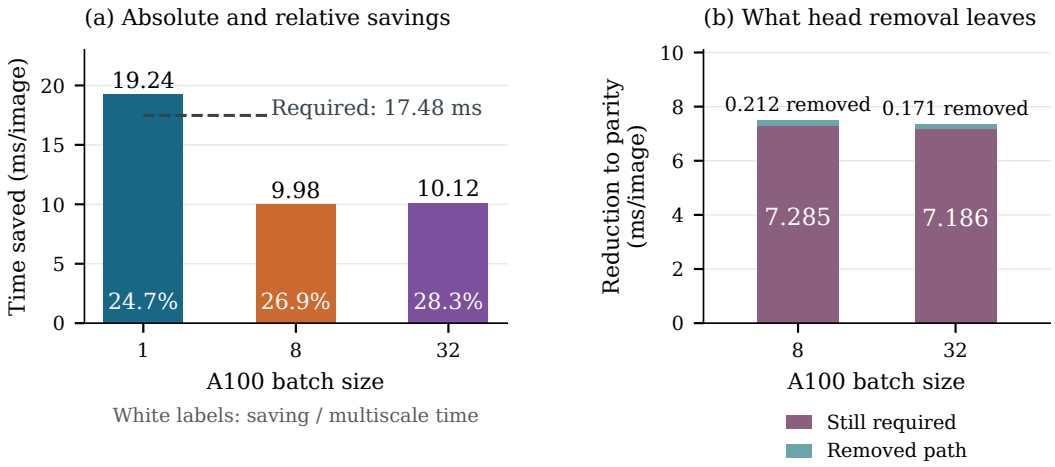

Figure 1: **Time savings must meet the quality-equivalent requirement.** A100, 400 ADE images, $\lambda = 0.8$ CE/second. (a) Percentage savings over multiscale rise with batching, but absolute savings fall below the 17.48 ms/image break-even requirement. (b) Removing observation/head computation leaves 7.285/7.186 ms/image to parity at batches 8/32. These are matched mean comparisons, not confidence bounds or implementation-independent limits.

The contribution combines comparator-aware accounting, matched interventions, and prospective calibration. Together they distinguish useful allocation, executed utility, and the benefit of a measured implementation change. The study does not introduce a new routing architecture or establish an implementation-independent limit on adaptive computation.

Practical evaluation checklist.

1. **Specify.** Declare the quality metric, price, static menu, eligible observations, execution environment, batch size and output contract.
2. **Fit, freeze and calibrate.** Fit candidate policies on development data, freeze them, and select by complete-workload utility on separate calibration data at the declared price.
3. **Measure the comparison.** Freeze the selection; execute it and every eligible static on held-out workloads using the same timing boundary.
4. **Report the decision.** Give quality, runtime, surplus and uncertainty. Translate any deficit into the additional quality gain or time saving needed; a positive mean alone does not establish superiority.

2 Related Work

Learning the value of computation. Metareasoning prices computation through the decisions it changes (Russell & Wefald, 1991; Sezener, 2018); information comparison (Blackwell, 1953) and policy learning (Kitagawa & Tetenov, 2018; Athey & Wager, 2021) formalize its decision value. Cost-sensitive evaluation (Bolukbasi et al., 2017), learned augmentation losses and aggregation (Kim et al., 2020; Shanmugam et al., 2021), embedded-CPU AdapTTA (Mocerino et al., 2021), and calibrated delegation (Pona et al., 2026) are direct precedents. Calibrated scalar uncertainty can predict action value without multivariate regression. We compare an entropy gate under complete execution and study selection from a frozen candidate menu, without claiming the guarantees of a different calibration design.

External augmentation and internal routing. Adaptive TTA selects input variants, including path, stress, and scale/style selection (Yang, 2026a;b; Xie et al., 2024). BranchyNet selects exits, DynamicViT and DToP route tokens, and DynaMask selects mask resolutions (Teerapittayanon et al., 2016; Rao et al., 2021; Tang et al., 2023; Li et al., 2023); Soft MoE and MoNE route rep-

resentations among experts (Puigcerver et al., 2024; Jain et al., 2024). Their sharing, training and dispatch differ from whole-image TTA and require separate execution measurements.

Measuring useful efficiency. Hardware-aware studies distinguish operation counts, runtime and software effects (Williams et al., 2009; Hooker, 2021; Dehghani et al., 2022). Distribution shift and unreliable uncertainty can alter action value (Ben-David et al., 2010; Ovadia et al., 2019). Our study connects plug-in selection (Audibert & Tsybakov, 2007) to matched execution: it separates comparator replacement from pairwise repricing, measures a removable routing path, and quantifies the remaining quality–runtime budget.

3 From action value to executed surplus

We connect executed surplus and favorable prices to adaptive opportunity, fitted recovery, and conditional calibration guarantees.

A common preference and complete workload. Suppressing the environment index, let L_h be mean per-image CE and $T_{h,b}$ complete workload time per image for system h at batch size b. For a price λ in CE/second, a nonempty finite static menu, and a common output contract, define

$$U_h(\lambda, b) = -L_h - \lambda T_{h,b}, \qquad S_{\text{exec}} = U_h - \max_{a \in \mathcal{A}_{\text{static}}} U_a, \quad \lambda \geq 0. \tag{1}$$

Each environment supplies its own measured quality and runtime. Best static maximizes utility at the stated price and batch size. $T_{h,b}$ is normalized workload time, not response latency; internal costs need not be additive.

Reading the units. Equations use seconds/image; tables use ms/image. Surplus is in CE-equivalent units, with $S > 0$ favoring routing. At $\lambda = 0.8$, saving 1 ms/image offsets 0.0008 CE; a table entry of -2.942 in 10^{-3} units means -0.002942, not a percentage change.

The break-even execution condition. For $\lambda > 0$, routing beats a particular static a exactly when

$$\underbrace{T_{a,b} - T_{h,b}}_{\text{measured time saved}} > \underbrace{(L_h - L_a)/\lambda}_{\text{time saving required by quality difference}}. \tag{2}$$

For $L_h < L_a$, the negative right side describes allowable extra time. Sections 5.1–5.4 apply this condition to measured systems.

Proposition 1 (Surplus under a change of cost boundary)**.** *Fix policies, outputs, workload, λ, and batch size b. Let p, e denote proxy and executed costs, with surpluses S_p, S_e against the same menu. Put $s_p(h, a) = U_h^p - U_a^p$ and $d(h, a) = (T_{h,b}^e - T_{h,b}^p) - (T_{a,b}^e - T_{a,b}^p)$. Then*

$$S_e = \min_a \{s_p(h, a) - \lambda d(h, a)\}. \tag{3}$$

If $\max_a |d(h, a)| \leq \epsilon$, then $|S_e - S_p| \leq \lambda\epsilon$, even if the best static changes.

Proof. Each pairwise contrast is $U_h^e - U_a^e = s_p(h, a) - \lambda d(h, a)$. Taking the minimum proves the identity; perturbing every entry by at most $\lambda\epsilon$ changes its minimum by at most the same amount. □

Corrections are *relative to every comparator*; a proxy margin above a valid $\lambda\epsilon$ bound stays positive. We use measured corrections instead. If outputs change, add $-(L_h^e - L_h^p) + (L_a^e - L_a^p)$ inside the minimum (Appendix A.2); cross-platform comparisons therefore use environment-specific diagnostic quality.

Corollary 2 (Fixed-policy price regions)**.** *Fix a nonempty finite static menu and, within each environment e at batch size b, the decisions, outputs, and complete workload times. Then*

$$S_{e,b}(\lambda) = \min_{a \in \mathcal{A}_{\text{static}}} \left\{L_{a,e} - L_{h,e} + \lambda\big(T_{a,e,b} - T_{h,e,b}\big)\right\} \tag{4}$$

is continuous, concave, and piecewise affine. Its favorable-price set $\Lambda_{e,b} = \{\lambda \geq 0 : S_{e,b}(\lambda) > 0\}$ is an interval, possibly empty or unbounded. Across a finite collection $\mathcal{E}$ at the same batch size, a common price gives positive surplus everywhere exactly when

$$\lambda \in \Lambda_b^{\text{common}} := \bigcap_{e \in \mathcal{E}} \Lambda_{e,b}. \tag{5}$$

The derivation is in Appendix A.3. Nonempty regions need not overlap (Section 5.2). These are fixed-policy mean comparisons, not confidence regions or a price-selection rule.

Opportunity and fitted recovery. Under an additive action model, let $V_{a,\lambda} = B_a - \lambda C_a$ be integrable on a finite menu containing native, with $V_{0,\lambda} = 0$. For TTA, $B_a = \ell_0 - \ell_a$ is the per-image CE gain over native and C_a is extra-view proxy time; ℓ_a denotes the per-image action loss. For eligible information $\mathcal{I}$, put $m_a = \mathbb{E}[V_{a,\lambda} \mid \mathcal{I}]$, conditioning all expectations on the fitted training artifact. For an $\mathcal{I}$-measurable h, define $J_{\mathcal{I}} = \mathbb{E}\max_a m_a$, $J_{\mathrm{st}} = \max_a \mathbb{E}m_a$, and $J_h = \mathbb{E}m_{h(\mathcal{I})}$. Then

$$J_h - J_{\mathrm{st}} = \underbrace{J_{\mathcal{I}} - J_{\mathrm{st}}}_{\text{available opportunity}} \underbrace{- \mathbb{E}\Big[\max_a m_a - m_{h(\mathcal{I})}\Big]}_{\text{fitted decision regret}}. \tag{6}$$

Opportunity is positive exactly when no single action is conditionally optimal almost surely; changing labels among ties is insufficient. Positive fitted surplus requires smaller regret than opportunity. Allocation and capacity controls do not identify these unknown terms from a label-oracle gap. This additive model guides candidates, not complete batched execution (Appendix A).

Calibration-to-evaluation transfer. For a nonempty finite menu $\mathcal{H}$ frozen before calibration, let $q(h)$ be complete-cost calibration utility at the fixed price and select $\widehat{h} \in \arg\max_{h\in\mathcal{H}} q(h)$. For evaluation utility U_Q,

$$\max_{h\in\mathcal{H}} |q(h) - U_Q(h)| \le \varepsilon \quad \Longrightarrow \quad \max_{h\in\mathcal{H}} U_Q(h) - U_Q(\widehat{h}) \le 2\varepsilon. \tag{7}$$

If $\mathcal{A}_{\mathrm{static}} \subseteq \mathcal{H}$, the same premise gives

$$U_Q(\widehat{h}) - \max_{a\in\mathcal{A}_{\mathrm{static}}} U_Q(a) \ge q(\widehat{h}) - \max_{a\in\mathcal{A}_{\mathrm{static}}} q(a) - 2\varepsilon. \tag{8}$$

A calibration margin above 2ε suffices; including statics alone only prevents calibration-score regression (Appendix A.5). This selects executable candidates, not joint batch assignments (Appendix A.1). ACDC's five-router menu excludes statics; without an established uniform bound, its held-out contrasts test transfer directly.

4 Experimental protocol

Frozen SegFormer-B2 actions average probabilities from native (one view), native/flip (two), or native/flip/lower/upper (four). Both studies resize inputs to 512/384/640 squares and align outputs to the original grid before averaging. The ADE router selects $\arg\max_a\{\widehat{B}_a(x) - \lambda\widehat{C}_a^{\mathrm{proxy}}(x)\}$ using native hidden-state summaries, PCA64 and entropy; native has zero predicted increments. ADE evaluates a frozen policy on 400 images and separate five-fold fits on all 2,000; ACDC prospectively compares calibration on 406 images. Appendix B gives checkpoints, splits and dispatch; Appendix C reports COCO semantic controls using the Mask2Former-Swin-S model (Liu et al., 2021; Cheng et al., 2022; Caesar et al., 2018).

Selection and execution. The ADE selector takes the first validation-surplus maximum on $0, 0.05, \ldots, 1.50$ CE/second, selecting 0.8 before source refitting. ACDC fixes 0.8 before calibration and final outcomes. This is a common benchmark preference, not an application-derived price. Native inference is shared; selected images form sub-batches for sequential additional views, without masked mixed-action kernels. The 640-pixel pass has a 16-image chunk limit.

Two timing boundaries. *Proxy cost* sums separately timed batch-1 extra views after native inference; native therefore has zero proxy cost, not zero runtime. *Complete runtime* measures the decoded-input-to-output workload, including native inference, transfers, preprocessing, observations, routing, selected views, assembly and synchronization. Decoding, model loading and scoring are excluded. Component profiles diagnose this workload but do not replace complete runtime. The proxy-to-complete contrast changes batching and timing scope, not only router overhead.

Cross-platform comparison. We evaluate the frozen ADE parameters on T4 at batch sizes 1/8 and A100 at 1/8/32, with three matched workload repeats in one session per device. The T4 and A100 allocations agree; one image differs from the original Windows allocation. Quality is recomputed by a separate untimed execution of the same batch groups, not by retaining the discarded timed outputs. These comparisons use environment-specific CE, whereas the original Windows study uses verified cached CE. The platforms also differ in host resources. FP32 tensors are used without AMP; matmul TF32 is disabled and cuDNN TF32 is permitted on T4/A100. Appendix D.5 specifies the environments and output checks. ADE computes hard predictions; ACDC additionally returns probabilities. Contracts match within each comparison, not across the two benchmarks.

Metrics and variation. CE averages valid pixels within images, then images equally. mIoU pools confusion counts before applying the declared class universe: ADE compares active-union and fixed-150 conventions; ACDC uses all 19 classes. Composition sensitivity, workload repeatability and source-refit variation are analyzed separately. ACDC retains its paired recording-block/session bootstrap; the T4/A100 repeat ranges are not population confidence intervals. Appendix B specifies sampling units and the assumptions needed for population interpretation.

5 RESULTS

5.1 RTX 3090 BATCHING REVERSES THE MEAN UTILITY ORDERING

On Windows/RTX 3090 at $\lambda = 0.8$, the router's 0.014222 CE penalty requires **17.778 ms/image saved** against multiscale (Eq. (2)). It saves 21.914 ms at batch 1, but only 14.101/14.131 ms at batches 8/32 (Table 1). Thus mean utility favors routing at batch 1 and multiscale at batches 8/32, although routing remains faster than multiscale. The reversal concerns priced utility, not Pareto dominance; no intermediate batch-size threshold is estimated.

Table 1: **The router remains faster, but loses mean utility at batches 8/32.** Windows/RTX 3090, 400 images, frozen router and verified cached CE. Proxy costs count extra views; complete times include the pipeline. At $\lambda = 0.8$, flip is best under proxy costs; multiscale is best under complete costs at all three batch sizes. Contrasts use unrounded records.

Policy	Mean CE ↓	Extra-view proxy (ms/image)	Complete time (ms/image) ↓		
			Batch 1	Batch 8	Batch 32
Native	0.850571	0	27.541	12.365	11.787
Flip	0.822699	30.922	53.227	24.651	23.533
Multiscale	0.781721	95.939	97.966	50.775	48.845
Frozen router	0.795944	57.459	76.052	36.675	34.715

Router surplus over best static (proxy; complete batches 1/8/32):
$+0.005525;\ +0.003309/-0.002942/-0.002918.$

At batch 8 and unchanged outputs, the router must reduce complete time from 36.675 to below 32.997 ms/image: **3.677 ms/image (10.0%) to parity**. Batch 32 requires 3.647 ms/image (10.5%). Alternatively, batch 8 needs 0.002942 lower CE at unchanged runtime. These targets fix comparator records; any stronger head's extra cost must also be paid.

Allocation is nevertheless useful: CE improves by 0.012870 over expected count-matched random assignment, with composition range $[0.006419, 0.023750]$. Timing-repeat resampling spans zero at batch 1 but remains negative at batches 8/32. Composition and timing analyses are separate, not a joint deployment interval (Appendix D.2).

Comparator-aware accounting. At batch 8, repricing improves the router–flip contrast, but replacing flip with multiscale removes more surplus (Proposition 1). In 10^{-3} CE-equivalent units,

$$\underbrace{5.525}_{\text{proxy surplus}} + \underbrace{11.611}_{\text{pairwise repricing}} - \underbrace{20.078}_{\text{comparator switch}} = \underbrace{-2.942}_{\text{executed surplus}} .$$

This is accounting on unrounded records, not causal isolation of batching, timer scope, or session effects (Appendix E).

5.2 Execution environment changes the break-even comparison

At unchanged $\lambda = 0.8$, Table 2 distinguishes insufficient absolute savings from a change in the best static. Comparisons are matched within each environment, not GPU-only causal interventions.

Table 2: **The best static changes with the execution environment.** At $\lambda = 0.8$, native is best on T4; multiscale on RTX 3090/A100. All rows use 400 images, fixed parameters, and environment-specific quality. Times are complete ms/image; surplus is in 10^{-3} CE-equivalent units. T4/A100: one session, three repeats; RTX 3090: 3/3/10 repeats.

Environment	Batch	Best static	Router time	Static time	Surplus ↑
RTX 3090	1	Multiscale	76.052	97.966	+3.309
	8	Multiscale	36.675	50.775	−2.942
	32	Multiscale	34.715	48.845	−2.918
T4	1	Native	196.192	63.243	−52.776
	8	Native	220.224	75.544	−62.160
A100	1	Multiscale	58.534	77.775	+1.411
	8	Multiscale	27.070	37.051	−5.998
	32	Multiscale	25.652	35.772	−5.886

A100: greater percentage savings, lower utility. From batch 1 to 32, fractional savings rise from 24.74% to 28.29%, while absolute savings fall from 19.241 to 10.120 ms/image, below the approximately 17.48 ms requirement (Figure 1a). All three within-session repeats preserve these utility signs. Removing observation/head computation saves only 0.212/0.171 ms/image at batches 8/32, leaving 7.285/7.186 ms/image to parity (Figure 1b; Appendix D.5).

T4: beating multiscale does not beat the best static. Router-minus-multiscale utility is $+0.053323/+0.061044$ at batches 1/8, but native is best static, giving surplus $-0.052776/-0.062160$. A multiscale-only comparator therefore reverses the conclusion. Every measured T4 method slows per image at batch 8, so the slowdown is not routing-specific (Appendix D.5).

The reversal is not confined to one price. The fixed-record regions (Corollary 2; Figure 2) favor routing at batch 1 and statics at batches 8/32 throughout $[0.75, 0.95]$ CE/second on RTX 3090 and A100. T4/A100 regions are disjoint at batches 1/8. These descriptive comparisons do not reselect the evaluation price, which remains 0.8; Appendix D.5 gives the shared-range check.

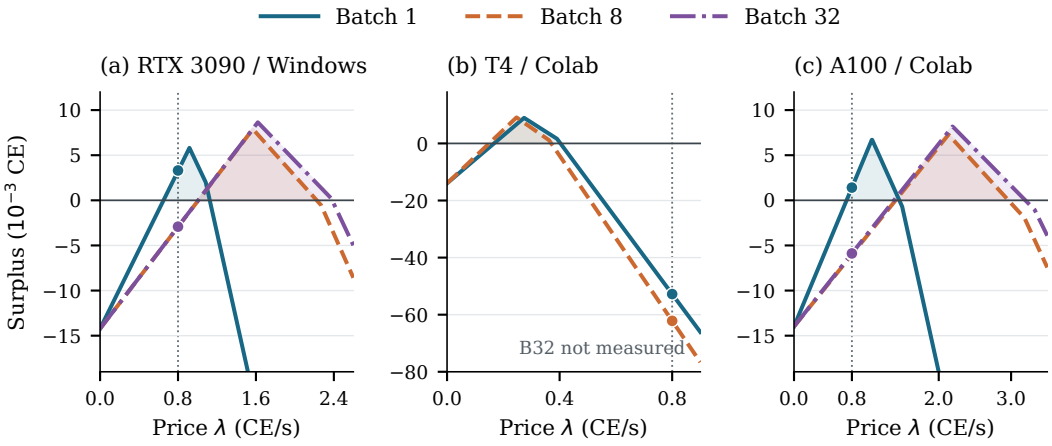

Figure 2: **T4 and A100 have no common favorable price at batches 1/8.** Fixed-policy recorded means; circles mark $\lambda = 0.8$. Shading denotes positive surplus, not uncertainty. Axes differ; T4 batch 32 is unmeasured.

5.3 COHORT AND ROUTING CONTROLS DELIMIT THE EXPLANATION

The cross-fitted RTX 3090 evaluation excludes each of 2,000 ADE images from its generating head's fit ($\alpha = 100$, $\lambda = 0.8$). Routing is faster but mean surplus is negative in all five folds (Table 3). Pooled required/observed savings are 33.246/19.153 ms/image: the original 17.778 ms requirement is workload-specific. Settings are inherited from the inspected study; this is a retrospective check.

Table 3: **The deficit persists across five folds and the tested controls.** Batch 8, $\lambda = 0.8$, three repeats; cached CE and retrospective settings. A: equally weighted excluded-fold images; fold 4 retains decisions but remeasures time. B: separate matched sessions. Changes are first-minus-comparator: positive ΔCE is worse, positive ΔU better (10^{-3} units). Time saved is comparator-minus-first (ms/image).

A. Cross-fitted router versus multiscale

Fold	Router CE	Required saving	Observed saving	ΔU
0	0.714081	23.683	22.170	−1.210
1	0.906493	47.823	18.307	−23.613
2	0.795173	32.388	20.705	−9.346
3	0.870009	44.557	20.441	−19.292
4	0.795944	17.778	14.142	−2.909
Pooled	0.816340	33.246	19.153	−11.274

B. Matched controls on the original 400 images

Comparison	ΔCE	Saved ms/image	ΔU
Supplied actions − live ridge	0	0.512	+0.410
Entropy gate − live ridge	+0.021141	2.073	−19.483

Controller removal leaves a deficit. Removing observation/head work saves 0.512 ms/image but leaves 3.124 ms/image to same-session multiscale parity. Native inference, dispatch, assembly and decisions stay fixed; 64-image hard outputs match.

The entropy gate saves 2.073 ms/image but loses 0.021141 CE ($\Delta U = -0.019483$). It is proxy-calibrated, not a test of complete-cost entropy selection (Appendix F).

5.4 COMPLETE-COST CALIBRATION CHANGES THE OPERATING POINT

Prospective setup. ACDC uses a frozen Cityscapes-trained SegFormer-B2 and all 19 classes (Cordts et al., 2016; Sakaridis et al., 2021): 104 pilot, 1,035 fitting, and 461 calibration images from the 1,600-image training split, with all 406 geographic-validation images held out. Inferred families remain intact within development; Appendix B specifies overlap and chronology.

Both arms share hidden-state summaries and unmasked entropy, with scaling, PCA64 and ridge gain/log-proxy-cost heads ($\alpha = 100$) fitted on the ACDC fitting split. Candidates are

$$h_s(x) = \arg\max_a \{\widehat{B}_a(x) - \lambda_0 s \widehat{C}_a^{\text{proxy}}(x)\}, \qquad s \in \{0.25, 0.5, 1, 2, 4\},$$

with zero native increments and fixed evaluation price $\lambda_0 = 0.8$; s changes the policy, not that preference. Proxy and complete-cost calibration select $s = 1$ and $s = 0.25$, respectively, without refitting. The latter exceeds $s = 0.5$ by only 1.83×10^{-4} calibration utility, so the optimum is not precisely located. Block deletion changes some winners but preserves selector disagreement (Appendix B). Statics are final comparators, not calibration candidates, so Eq. (8) supplies no guarantee against them here. A separate seven-candidate calibration-only boundary check retains $s = 0.25$ without reselection or covering the full augmented menu (Appendix F).

Held-out execution. Five RTX 3090 process sessions run three randomized batch-8 repeats over 14 paired recording blocks. Actual probabilities are scored outside timed microblocks (10^{-8} floor, equal image weights); actions, CE and confusion counts agree across repeats. Table 4 and Figure 3 report frozen comparisons; Appendix B specifies resampling.

Table 4: **Better CE costs more runtime; multiscale retains the highest mean utility.** ACDC, 406 held-out images, five sessions, three repetitions; $U = -L - 0.8T$ uses seconds/image. Rankings are descriptive, not significance claims; contrasts use unrounded records. mIoU is secondary; all 19 classes have nonzero pooled union.

Policy	s	Mean CE	Complete ms/image	mIoU (%)	U
Native	–	1.070347	20.398	49.15	−1.086666
Flip	–	1.007032	35.222	49.70	−1.035210
Multiscale	–	0.923335	69.189	50.16	−0.978686
Proxy-calibrated	1	0.937935	53.432	50.26	−0.980680
Complete-calibrated	0.25	0.927055	65.015	50.08	−0.979068

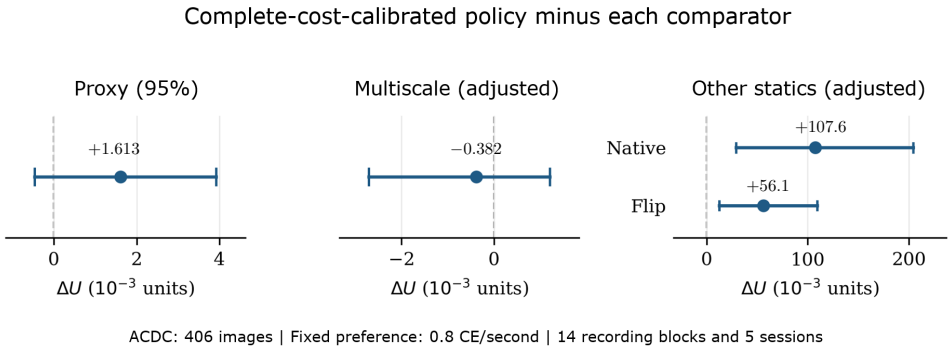

Figure 3: **Superiority to proxy calibration and multiscale remains unresolved.** Positive contrasts favor complete-cost calibration. The proxy interval is paired 95%; static intervals are Bonferroni-adjusted. Units are 10^{-3} CE-equivalent; axes differ. Population interpretation requires the recording-block/session assumptions in Appendix B.

Comparison with proxy calibration. Complete-cost calibration changes 122/406 decisions (30.0%) and raises multiscale use from 61.6% to 89.2%. It reduces CE by 0.010880 while adding 11.584 ms/image; the time charge absorbs **85.2% of the quality benefit**, leaving $\Delta U = +0.001613\,[-0.000457, 0.003917]$. All five session contrasts are positive (0.001530–0.001699), but the paired interval crosses zero. This tests transfer rather than establishing the uniform-error premise of Eq. (7).

Comparison with the best static. The 0.003721 CE penalty against multiscale requires 4.651 ms/image saved; the router saves 4.174 ms/image. At unchanged outputs, a further **0.477 ms/image reduction** reaches mean parity (Eq. (2)). Surplus is $-0.000382\,[-0.002715, 0.001216]$ under the adjusted interval. Contrasts with native and flip are positive, but superiority to multiscale or proxy calibration is not established.

Metric-dependent ordering. Proxy calibration has higher observed mIoU than complete calibration (50.26% versus 50.08%) and lower runtime. It exceeds multiscale by 0.107 mIoU points while saving 15.758 ms/image. These are descriptive point comparisons without an inferential mIoU interval (Appendix B). CE/time favors complete calibration; mIoU and runtime favor proxy calibration, not a metric-independent winner.

5.5 LEARNING AND METRIC CONTROLS

Equation (6) separates opportunity and regret; these controls do not identify either population quantity. Eight CE heads recover 0–7.33% of positive oracle-surplus area without a resolved expressive-head advantage; entropy recovers 5.73% despite worse fixed-price utility (Appendices C.3 and F). Fixing ADE's mIoU denominator changes router-minus-flip from -0.089 to $+0.191$ points; both composition ranges include zero. Six of nine CE- or mIoU-selected policies exceed transported validation cost caps. The appendices separate capacity, refitting, class support, and price-range controls from executed comparisons.

6 IMPLICATIONS AND LIMITATIONS

What an optimized implementation must improve. Hold workload, static menu, output contract, and price fixed. For a revised system j, define signed reductions $\Delta L_j = L_j - L'_j$ and $\Delta T_{j,b} = T_{j,b} - T'_{j,b}$. With $s_a = U_h - U_a$, the relative-loss extension of Proposition 1 yields

$$S'_{\text{exec}} = \min_{a \in \mathcal{A}_{\text{static}}} \left\{ s_a + \Delta L_h - \Delta L_a + \lambda\big(\Delta T_{h,b} - \Delta T_{a,b}\big) \right\}. \tag{9}$$

Output-preserving changes affect relative runtime; batch-aware allocation may also change quality. With unchanged statics, overcoming $D = -S_{\text{exec}} > 0$ requires $\Delta L_h + \lambda \Delta T_{h,b} > D$. Joint allocation and fused multi-action execution remain untested and may change the ordering. The interventions only show that removing the tested observation/head path is insufficient. Residual budgets are not runtime lower bounds, and optimizations benefiting statics require their remeasurement too.

Calibration and scope. The guarantee in Eq. (8) is conditional; ACDC's router-only selector does not establish held-out non-regression. An mIoU objective needs separate calibration and price. These synchronous whole-image TTA studies use frozen backbones. T4/A100 reuse ADE with one session per device and different hosts, precluding GPU-only causal attribution. ACDC recording families overlap across splits; its intervals exclude fitting and selector variation. Other cohorts, sessions, and internal routing architectures need separate measurements.

7 CONCLUSION

Useful allocation, time savings and positive executed surplus are distinct. Matched measurements distinguish absolute-saving shortfalls from comparator replacement and show when removing a controller path is insufficient. Prospective calibration changes decisions without establishing superiority. The resulting comparison procedure and budgets specify what improved systems must achieve relative to their static alternatives.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in the writing of proofs, designing or providing feedback on research methodology or experiments, and implementing methods. We have not used generative AI tools for developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses or interpreting results. Additionally, we used generative AI tools for sourcing and searching for information, identifying relevant literature, creating or editing software code, editing the research paper to improve readability, and formatting references. We have reviewed all AI-assisted work: all AI-generated code snippets and data-processing routines were manually audited and verified; all cited literature was manually cross-checked against original publications to eliminate non-existent citations; and all mathematical formulations, proofs, and empirical conclusions were derived and independently verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

The central study uses public dense-vision benchmarks. Group analyses describe heterogeneity in model behavior; they are not fairness guarantees.

Reproducibility statement

A companion archive contains reconstruction scripts. Public images and weights are external prerequisites for neural inference. The supporting appendices specify estimands, sampling units, devices, and output contracts. The experimental companion and this manuscript-source project are separate artifacts.

References

Susan Athey and Stefan Wager. Policy learning with observational data. *Econometrica*, 89(1): 133–161, 2021. doi: 10.3982/ECTA15732.

Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608–633, 2007. doi: 10.1214/009053606000001217.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24(2):265–272, 1953.

Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 527–536, 2017.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. The efficiency misnomer. In *International Conference on Learning Representations*, 2022.

Sara Hooker. The hardware lottery. *Communications of the ACM*, 64(12):58–65, 2021.

Gagan Jain, Nidhi Hegde, Aditya Kusupati, Arsha Nagrani, Shyamal Buch, Prateek Jain, Anurag Arnab, and Sujoy Paul. Mixture of nested experts: Adaptive processing of visual tokens. *arXiv preprint arXiv:2407.19985*, 2024.

Ildoo Kim, Younghoon Kim, and Sungwoong Kim. Learning loss for test-time augmentation. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2010.11422.

Toru Kitagawa and Aleksey Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018. doi: 10.3982/ECTA13288.

Ruihuang Li, Chenhang He, Shuai Li, Yabin Zhang, and Lei Zhang. DynaMask: Dynamic mask selection for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11279–11288, 2023.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, October 2021.

Luca Mocerino, Roberto G. Rizzo, Valentino Peluso, Andrea Calimera, and Enrico Macii. AdapTTA: Adaptive test-time augmentation for reliable embedded ConvNets. In *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 1–6, 2021. doi: 10.1109/VLSI-SoC53125.2021.9606980.

Alfred Müller. Orderings of risks: A comparative study via stop-loss transforms. *Insurance: Mathematics and Economics*, 17(3):215–222, 1996. doi: 10.1016/0167-6687(96)90002-5.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Edoardo Pona, Milad Kazemi, Mehran Hosseini, Yali Du, David Watson, Osvaldo Simeone, and Nicola Paoletti. Calibrate-then-delegate: Safety monitoring with risk and budget guarantees via model cascades. *arXiv preprint arXiv:2604.14251*, 2026.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *The Twelfth International Conference on Learning Representations*, 2024.

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*, volume 34, 2021.

Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial Intelligence*, 49(1–3):361–395, 1991. doi: 10.1016/0004-3702(91)90015-C.

Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10765–10775, 2021.

Can Eren Sezener. Computing the value of computation for planning. *arXiv preprint arXiv:1811.03035*, 2018.

Moshe Shaked and J. George Shanthikumar. *Stochastic Orders*. Springer, 2007. doi: 10.1007/978-0-387-34675-5.

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Quan Tang, Bowen Zhang, Jiajun Liu, Fagui Liu, and Yifan Liu. Dynamic token pruning in plain vision transformers for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 777–786, 2023.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *International Conference on Pattern Recognition*, 2016.

Samuel Williams, Andrew Waterman, and David Patterson. Roofline: An insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4):65–76, 2009.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, 2021.

Kangxian Xie, Siyu Huang, Sebastian Cajas Ordonez, Hanspeter Pfister, and Donglai Wei. S^3-TTA: Scale-style selection for test-time augmentation in biomedical image segmentation. In *2024 IEEE International Symposium on Biomedical Imaging*, 2024. doi: 10.1109/ISBI56570.2024.10635833.

Youla Yang. Understanding the limits of vision test-time scaling: Path redundancy, instance difficulty, and adaptive compute. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 11450–11456, June 2026a.

Youla Yang. SA-TTS: Stress-aware test-time scaling for vision models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 11480–11487, June 2026b.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

APPENDIX

A PROPOSITIONS, PROOFS AND THEIR MEASUREMENT ROLES

The formal results organize the study around signed action value, a fixed preference and complete execution. The proofs below make the population and recorded-workload objects explicit. Value-of-information, positive-part and selection inequalities are established tools; the paper's contribution is their connection to a matched dense-vision experiment, prospective calibration and measurable runtime targets.

A.1 FROM ORACLE OPPORTUNITY TO A FITTED POLICY

For a finite action set containing the native baseline 0, define integrable

$$V_{a,\lambda} = B_a - \lambda C_a, \qquad V_{0,\lambda} = 0, \qquad m_a = \mathbb{E}[V_{a,\lambda} \mid \mathcal{I}]. \tag{A1}$$

All expectations condition on the fitted training artifact. For a measurable policy h using $\mathcal{I}$, the four relevant values are

$$J_{\mathrm{or}} = \mathbb{E}\max_a V_{a,\lambda}, \quad J_{\mathcal{I}} = \mathbb{E}\max_a m_a, \quad J_{\mathrm{st}} = \max_a \mathbb{E}V_{a,\lambda}, \quad J_h = \mathbb{E}V_{h(\mathcal{I}),\lambda}. \tag{A2}$$

For any best static $a^* \in \arg\max_a \mathbb{E}m_a$,

$$J_{\mathcal{I}} - J_{\mathrm{st}} = \mathbb{E}[\max_a m_a - m_{a^*}].$$

The nonnegative gap is strictly positive exactly when a^* is suboptimal with positive probability; equivalently, no single action is conditionally optimal almost surely. This handles ties: different selected action labels alone do not establish opportunity. The binary closed forms below do not extend unchanged to three actions. The tower property gives $J_h = \mathbb{E}m_{h(\mathcal{I})} \le \mathbb{E}\max_a m_a$. Conditional Jensen gives $\mathbb{E}\max_a m_a \le \mathbb{E}\max_a V_{a,\lambda}$, while ordinary Jensen gives $\max_a \mathbb{E}m_a \le \mathbb{E}\max_a m_a$. Consequently,

$$J_{\mathrm{or}} - J_h = (J_{\mathrm{or}} - J_{\mathcal{I}}) + (J_{\mathcal{I}} - J_h). \tag{A3}$$

The first term is information unavailable to the router; the second is its fitting/approximation shortfall. The empirical oracle and fitted values alone cannot identify the two terms. Better features expand potential information; better fitting seeks to recover more of its value. For nested information sets, conditional Jensen yields $J_{\mathcal{I}_1} \le J_{\mathcal{I}_2}$, but does not order their fitted policies.

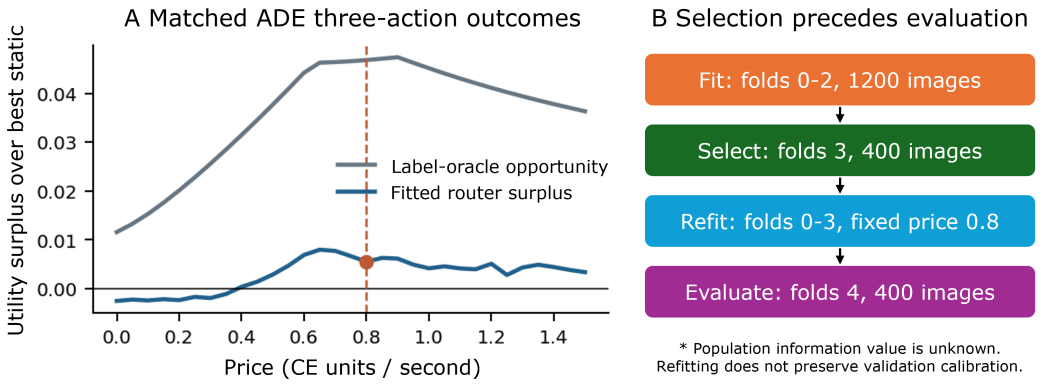

Figure A1: **Proxy opportunity and the frozen-policy chronology.** (a) Label-oracle and fitted-router surplus over the evaluation-best static under extra-view proxy costs. The oracle is not the unknown information-constrained optimum; its gap from the fitted curve does not identify recoverable information value. Decisions vary with price, unlike Figure 2. (b) The historical ADE fit–select–refit–evaluate chronology; the orange point marks the validation-selected price $\lambda = 0.8$.

Signed opportunity and weighted decision loss. In the binary compute/null case, write $V = V_{1,\lambda}$ and $m = \mathbb{E}[V \mid \mathcal{I}]$, with $x_+ = \max(x, 0)$. Selecting compute when $m > 0$ gives $J_{\mathcal{I}} = \mathbb{E}(m_+)$; the best constant decision gives $J_{\mathrm{st}} = (\mathbb{E}V)_+$.

Proposition A1 (Signed opportunity)**.** *For the binary compute/null decision with integrable V,*

$$J_{\mathcal{I}} - J_{\mathrm{st}} = \tfrac{1}{2}\{\mathbb{E}|m| - |\mathbb{E}V|\} = \min\{\mathbb{E}(m_+), \mathbb{E}((-m)_+)\}. \tag{A4}$$

Strictly positive opportunity requires both conditional signs with positive probability, and these two signs suffice.

Proof. Substitute $x_+ = (x + |x|)/2$ and $\mathbb{E}m = \mathbb{E}V$. For $A = \mathbb{E}(m_+)$ and $D = \mathbb{E}((-m)_+)$, the difference is $(A + D - |A - D|)/2 = \min(A, D)$. □

For an $\mathcal{I}$-measurable binary fitted policy h, the same pointwise identity applied to m gives

$$J_h - J_{\mathrm{st}} = \min\{\mathbb{E}(m_+), \mathbb{E}((-m)_+)\} - \mathbb{E}\big[|m|\mathbf{1}\{h \neq \mathbf{1}(m > 0)\}\big]. \tag{A5}$$

Signed mistakes spend the available opportunity. This gives a learning target, while the unknown conditional value m prevents interpreting observed fitted shortfall as a measured information limit. For observed binary decisions $h_i \in \{0, 1\}$, checking $V_i > 0$, $V_i < 0$ and $V_i = 0$ gives

$$\sum_i |V_i|\mathbf{1}\{h_i \neq \mathbf{1}(V_i > 0)\} = \sum_i (V_i)_+ - \sum_i h_i V_i. \tag{A6}$$

Thus mistakes should be weighted by the value they forgo. Weighted logistic in Appendix C.2 is a surrogate for this objective. The binary identity does not make its surrogate optimization exact, or convert an imagewise gain model into an additive model of batched execution.

Imagewise candidates and batch-coupled execution. An imagewise gain/cost rule is a candidate-generation surrogate, not an optimizer of joint batch execution. For a batch $X_{1:b}$, a corresponding joint decision would be

$$\arg\max_{a_{1:b}} \left\{ \sum_{i=1}^{b} \widehat{B}_{a_i}(X_i) - \lambda \widehat{T}(a_{1:b}; X_{1:b}) \right\},$$

where $\widehat{T}$ predicts total batch seconds and depends on the combined action assignment and dispatch. Independent imagewise maximizers need not solve this problem. The study does not fit this joint runtime predictor or execute a joint optimizer. Instead, it freezes imagewise candidates and scores their complete workloads, retaining the measured effects of batch coupling.

A.2 Exact repricing with a changing static comparator

Proposition 1 applies to fixed whole-system records, regardless of dependence among requests. It requires equal outputs between the two cost descriptions for *each* system; different systems may have different outputs. Let $a_p \in \arg\max_a U_a^p$ and define $G = \max_a U_a^e - U_{a_p}^e \geq 0$. Then

$$S_e = S_p - \lambda d(h, a_p) - G. \tag{A7}$$

Indeed, $S_p = U_h^p - U_{a_p}^p$; first reprice this pair and then subtract the gain from replacing a_p by the executed-best static. This identity explains the decomposition in Appendix E. A uniform bound on *relative* corrections gives $\lambda\epsilon$ in the main proposition. If instead each system satisfies $|T_j^e - T_j^p| \leq \epsilon$, only the generic bound $2\lambda\epsilon$ follows. Common cost shifts cancel from all contrasts. For changed outputs, add the relative loss correction $-(L_h^e - L_h^p) + (L_a^e - L_a^p)$ before taking the minimum.

Other workload-level quality objectives. The complete-system identities do not require an additive quality metric. For a fixed higher-is-better workload score M_h, define $U_h^M = M_h - \lambda_M T_{h,b}$. Then

$$S^M = \min_a \{M_h - M_a + \lambda_M (T_{a,b} - T_{h,b})\}.$$

For $\lambda_M > 0$, each pairwise comparison requires $T_{a,b} - T_{h,b} > (M_a - M_h)/\lambda_M$. This applies to pooled mIoU with a fixed class convention. It does not make pooled mIoU imagewise additive, justify the CE-trained gain surrogate for that objective, or transfer the numerical CE/time price to mIoU units. A different deployment objective requires its own selection and evaluation preference.

A.3 A FROZEN POLICY HAS A SINGLE POSITIVE PRICE INTERVAL

We restate the single-environment case of Corollary 2 and prove both its single-environment and cross-environment statements.

Corollary A2 (Frozen-policy price geometry)**.** *Fix a nonempty finite static menu, an execution environment, and a workload. Hold the decisions, outputs, and complete times fixed at batch size b. Then $S(\lambda) = \min_a\{L_a - L_h + \lambda(T_{a,b} - T_{h,b})\}$ is continuous, concave, and piecewise affine. Its strict positive set on $[0, \infty)$ is the intersection of the strict affine inequalities defined by all static comparisons, hence an interval in the relative topology, possibly empty or unbounded.*

Proof of Corollary 2. Fix an environment e and batch size b. Each pairwise contrast

$$\phi_{a,e,b}(\lambda) = L_{a,e} - L_{h,e} + \lambda(T_{a,e,b} - T_{h,e,b})$$

is affine because decisions, outputs, and complete workload times are fixed. Its minimum over the nonempty finite static menu, $S_{e,b}(\lambda) = \min_a \phi_{a,e,b}(\lambda)$, is therefore continuous, concave, and piecewise affine. Moreover,

$$\Lambda_{e,b} = \bigcap_{a \in \mathcal{A}_{\text{static}}} \{\lambda \geq 0 : \phi_{a,e,b}(\lambda) > 0\}.$$

Each set in this finite intersection is convex and relatively open in $[0, \infty)$. The intersection has the same properties and is an interval, possibly empty or unbounded.

For a finite collection $\mathcal{E}$ of environments at the same batch size,

$$\{\lambda \geq 0 : S_{e,b}(\lambda) > 0 \text{ for every } e \in \mathcal{E}\} = \bigcap_{e \in \mathcal{E}} \Lambda_{e,b} = \Lambda_b^{\text{common}}.$$

This establishes Eq. (5); the intersection may be empty even when every individual region is nonempty. □

Figure 2 reprices each environment's recorded decisions, outputs, and times without changing them, so the corollary applies. For the original Windows/RTX 3090 records, the favorable-price intervals are approximately $(0.6490, 1.1261)$, $(1.0086, 2.2251)$, and $(1.0065, 2.3826)$ CE/second at batch sizes 1/8/32. Appendix D.5 reports the other environments' intervals. These describe recorded mean comparisons, not confidence regions or a price-selection rule; the historical price remains 0.8. Their separation does not by itself identify an accelerator-specific mechanism. Curves that reselect actions with price, including Figure A1, represent a different function.

A.4 BINARY OPPORTUNITY, PRICE RANGE AND AREA

For the binary action model, assume $\mathbb{E}|B| < \infty$, $C > 0$ almost surely and $\mathbb{E}C < \infty$. Put $\mu_B = \mathbb{E}[B \mid \mathcal{I}]$, $\mu_C = \mathbb{E}[C \mid \mathcal{I}] > 0$, $R = \mu_B/\mu_C$, $dQ_C/dP = \mu_C/\mathbb{E}C$, and $\bar{R} = \mathbb{E}B/\mathbb{E}C$.

Proposition A3 (Binary opportunity geometry)**.** *For $\Delta(\lambda) = J_{\mathcal{I}}(\lambda) - J_{\text{st}}(\lambda)$,*

$$\frac{\Delta(\lambda)}{\mathbb{E}C} = \mathbb{E}_{Q_C}(R - \lambda)_+ - (\bar{R} - \lambda)_+. \tag{A8}$$

Its positive set is $(\operatorname{ess\,inf}_{Q_C} R, \operatorname{ess\,sup}_{Q_C} R)$, restricted to $[0, \infty)$ for physical prices. It is nondecreasing up to $\bar{R}$ and nonincreasing afterward, with real-line maximum $\frac{\mathbb{E}C}{2}\mathbb{E}_{Q_C}|R - \bar{R}|$; $\max(0, \bar{R})$ is a physical maximizer. Moreover,

$$\int_{-\infty}^{\infty} \Delta(\lambda) d\lambda = \tfrac{\mathbb{E}C}{2} \operatorname{Var}_{Q_C}(R), \tag{A9}$$

$$\int_{0}^{\infty} \Delta(\lambda) d\lambda = \tfrac{\mathbb{E}C}{2}\{\mathbb{E}_{Q_C}(R_+^2) - (\bar{R}_+)^2\}. \tag{A10}$$

The finite-area conditions are $\mathbb{E}_{Q_C}R^2 < \infty$ for the first and only $\mathbb{E}_{Q_C}R_+^2 < \infty$ for the second.

Proof. The tower property gives $\mathbb{E}\mu_C = \mathbb{E}C$, so Q_C is a probability law and $\mathbb{E}_{Q_C}|R| = \mathbb{E}|\mu_B|/\mathbb{E}C < \infty$. Since $m_\lambda = \mu_C(R - \lambda)$, averaging its positive part gives Eq. (A8). Proposition A1 shows strict positivity exactly when $R - \lambda$ has both signs, giving the essential-support interval. For $\lambda \le \bar{R}$, the normalized residual also equals $\mathbb{E}_{Q_C}(\lambda - R)_+$; for $\lambda \ge \bar{R}$ it equals $\mathbb{E}_{Q_C}(R - \lambda)_+$. These expressions prove monotonicity and their value at $\bar{R}$ gives the stated maximum. Tonelli applied to the nonnegative integrands yields

$$\int_{-\infty}^{\infty} \frac{\Delta(\lambda)}{\mathbb{E}C}\, d\lambda = \mathbb{E}_{Q_C}\left[\int_{-\infty}^{\bar{R}} (\lambda - R)_+ d\lambda + \int_{\bar{R}}^{\infty} (R - \lambda)_+ d\lambda\right] = \tfrac{1}{2}\mathbb{E}_{Q_C}(R - \bar{R})^2.$$

For nonnegative prices, integrating each positive part in Eq. (A8) uses $\int_0^\infty (r - \lambda)_+ d\lambda = (r_+)^2/2$ and proves the second identity. These equalities also hold as extended nonnegative equalities if the stated second moments diverge. □

This is a stop-loss representation (Müller, 1996; Shaked & Shanthikumar, 2007). The ratio condition is substantive: $B = 1$ and $C \sim \mathrm{Uniform}(0, 1)$, observed as information, have ordinary finite moments but infinite $\mathbb{E}[\mu_B^2/\mu_C]$. The empirical positive-area ratio instead clips a fitted policy's surplus on a finite grid and divides by oracle area on that grid. Appendix C.6 varies this domain directly; it does not estimate the variance in Eq. (A9). With several actions the Bayes policy can change among several envelopes, so binary connectedness does not follow. The frozen-policy corollary remains valid because its router is held fixed.

A.5 What complete-cost calibration optimizes

Proposition A4 (Complete-cost calibration and evaluation error)**.** *Let $\mathcal{H}$ be a finite frozen menu. Let $q_p(h)$ and $q(h)$ be proxy and complete-cost calibration scores on the same outputs at fixed $\lambda_0 \ge 0$, with $q(h) = q_p(h) - \lambda_0\delta(h)$. For $h_p \in \arg\max_{h\in\mathcal{H}} q_p(h)$ and $\widehat{h} \in \arg\max_{h\in\mathcal{H}} q(h)$, the calibration gain satisfies*

$$0 \le G_{\mathrm{cal}} := q(\widehat{h}) - q(h_p) \le \lambda_0\{\delta(h_p) - \delta(\widehat{h})\}. \tag{A11}$$

If $U_Q(h)$ is value on a declared evaluation workload or distribution and $\max_{h\in\mathcal{H}} |q(h) - U_Q(h)| \le \varepsilon$, then

$$\max_{h\in\mathcal{H}} U_Q(h) - U_Q(\widehat{h}) \le 2\varepsilon. \tag{A12}$$

For any comparator $g \in \mathcal{H}$, $U_Q(\widehat{h}) - U_Q(g) \ge q(\widehat{h}) - q(g) - 2\varepsilon$.

Proof. Selection by q gives $G_{\mathrm{cal}} \ge 0$. Substituting $q = q_p - \lambda_0\delta$ gives $G_{\mathrm{cal}} = q_p(\widehat{h}) - q_p(h_p) + \lambda_0\{\delta(h_p) - \delta(\widehat{h})\}$. The proxy score difference is nonpositive by selection of h_p, proving Eq. (A11). For the evaluation bound, insert $q(h^*)$ and $q(\widehat{h})$ between $U_Q(h^*)$ and $U_Q(\widehat{h})$ for a maximizer h^* of U_Q. The two approximation terms are bounded by ε and the score difference is nonpositive. For the second inequality insert the same two calibration scores in the pairwise value contrast and bound its two error terms below by $-\varepsilon$. □

No independence assumption is required for this deterministic implication; estimating a valid ε for future workloads requires a sampling model. In ACDC, $\mathcal{H}$ consists of the five fixed routing candidates. Complete-cost selection maximizes their observed calibration utility, and the proxy-selected candidate is in that menu. Statics are measured separately on the final cohort; they are not inserted retrospectively into the calibration selector. For a future menu $\mathcal{H}' = \mathcal{H} \cup \mathcal{A}_{\mathrm{static}}$, let $\widehat{h}' \in \arg\max_{h\in\mathcal{H}'} q(h)$. If the uniform-error premise holds over $\mathcal{H}'$, then $U_Q(\widehat{h}') - \max_a U_Q(a) \ge -2\varepsilon$. A calibration margin $q(\widehat{h}') - \max_a q(a) > 2\varepsilon$ suffices for positive evaluation surplus. Merely including statics gives empirical calibration non-regression, not guaranteed test non-regression. The frozen final interval is not a uniform error bound over $\mathcal{H}$. The experiment therefore evaluates the held-out comparison directly. The calibration records in Appendix B instantiate the optimization: $G_{\mathrm{cal}} = 0.004860785 \simeq -0.002170237 + 0.007031021$ (the last digits are rounded). The terms are the proxy-score difference and candidate-relative cost correction. The separate final-cohort difference is 0.001612631. A common cost offset across candidates changes no ordering; the candidate-dependent correction makes calibration consequential.

Linear-price mixtures and measured systems. In an additive action model, an input-independent mixture with probabilities p_a has value $\sum_a p_a \mathbb{E}V_a \leq \max_a \mathbb{E}V_a$. It can still be useful under an expected-cost constraint. Under batch coupling, a linear interpolation of standalone mean times need not equal the time of an executed mixture. The mixtures in Appendix D are measured systems, and their selected images, action shares and subbatches differ from the router's.

B COHORTS, SELECTION AND STATISTICAL INTERPRETATION

Table A1 summarizes the roles and fitting boundaries of the principal experiments. The retrospective all-image and component checks are specified in Appendix F. Appendix D.5 specifies T4/A100 transport of the frozen ADE policy.

Table A1: **Roles of the evidence.** Cross-platform and full-cohort execution extend the original ADE case retrospectively. ACDC freezes selection and analysis before final evaluation; learning and metric controls remain separate.

Question	Fitting / evaluation	What is measured
ADE execution	1,200 fit / 400 select; refit 1,600; final 400	One ridge policy at 0.8; proxy and complete runtime at batch sizes 1/8/32
Cross-platform	Same 400 ADE images and transferred fitted parameters; no refit	T4 batches 1/8 and A100 batches 1/8/32; environment-specific quality and complete times
Full-cohort execution	Five fits: 1,600 fit / 400 excluded-fold evaluation; 2,000 images in total	Retrospective batch-8 execution at 0.8 with fixed $\alpha = 100$; distinct from historical nested-selection metric controls
ACDC calibration	104 pilot / 1,035 fit / 461 calibrate; final 406	Same five candidates, fixed 0.8; proxy versus complete-cost selection; fresh outputs/times
Head capacity	Folds 0–2 fit (1,200) / 3 select (400) / 4 test (400)	Eight CE-head configurations; cached costs, separate from the timed policy
Refit / metric controls	40 source refits; five cross-fitted ADE folds	Common-price variation, class support, and direct validation-mIoU selection

Frozen checkpoints and preprocessing. ADE:
`nvidia/segformer-b2-finetuned-ade-512-512`
Revision: `de01bae28967510f9ddd496c60a969357195400c`.
ACDC: `nvidia/segformer-b2-finetuned-cityscapes-1024-1024`
Revision: `d633b2072669ca68d8f8e309de9b52bfdbf6bf72`.

Both convert uint8 RGB to FP32 in $[0, 1]$, resize directly to 512/384/640 squares with antialiased bilinear interpolation (`align_corners=False`), and normalize by means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$. The ACDC checkpoint name does not denote the executed input size. Flipped logits are restored on the decoder grid before bilinear interpolation to the original image grid and memberwise softmax.

ADE fitting and observation. SegFormer-B2 supplies native 512, native/flip, and native/flip/384/640 probability averages on 2,000 ADE validation images. The timed ridge fits folds 0–2 (1,200), selects price on fold 3 (400), refits on folds 0–3 (1,600), and evaluates fold 4 (400 images). The historical selector maximizes validation surplus over native, flip and multiscale at $0, 0.05, \ldots, 1.50$ CE/second; the first maximum is 0.8, with validation surplus 0.0057678627. Native hidden state and unmasked runtime entropy enter the deployed ridge. The original source-refit protocol and corrected cross-fitted controls are identified separately.

Action-specific dispatch. Transfer decoded uint8 images to GPU, resize/normalize, and run one native batch. For a router, reduce the final hidden map to means/standard deviations, compute decoder entropy, copy the summaries to CPU, apply scaling/PCA and the ridge heads. Let $I_f = \{i : a_i \geq 1\}$ and $I_m = \{i : a_i = 2\}$. Gather native inputs indexed by I_f, flip and forward them; gather originals indexed by I_m, normalize at 384 and 640, then forward these scales in sequence. The 640

pass is chunked at 16 images; there is no padding to the original batch size. ADE uses advanced tensor indexing and ACDC uses GPU `index_select`; both materialize selected tensors. Per-image indices restore native/flip/scale outputs to the original grid before probability averaging and argmax. No masked neural execution or explicit multi-stream overlap is used. Smaller groups and sequential launches can affect utilization, but aggregate times do not identify occupancy or warp divergence.

ACDC prospective sequence. The fixed Cityscapes-trained SegFormer-B2 was used. All 19 Cityscapes classes are retained, with train IDs 0–18 and ignore 255. The 1,600 official training images are allocated by filenames/counts before pixel access: one pilot family closest to 100 images, one calibration family per weather closest to 100, and the rest for fitting, with seeded ties. Actual counts are 104/461/1,035 across 1/4/12 inferred families; timing used the first 40 of the 104 reserved pilot images. Final evaluation uses all 406 official geographic-validation images in 14 filename-derived families. ACDC had not been used in the earlier development of this study; this status was recorded before inference. The published split is geographic; 11 inferred recording families occur on both sides, covering 340 final images. This does not establish unseen-recording transfer or independence of all recording groups.

Native last-stage feature means/standard deviations yield 1,024 channels; scaler, PCA64 and ridge heads ($\alpha = 100$) are fitted only on the 1,035-image ACDC fitting split. Decoder entropy uses no annotation mask. Gain targets are native-minus-action CE, and cost targets are log additive extra-view seconds. Candidate multipliers are 0.25/0.5/1/2/4 at fixed evaluation price 0.8. Both calibration procedures use the same actual calibration losses with their respective cost boundaries. Exact selection ties favor the larger multiplier; action ties favor the lower action index. No final source refit follows calibration.

The ACDC proxy averages three separately synchronized batch-1 measurements per image and extra view after one warmup. It starts with GPU-resident FP32 RGB in $[0, 1]$ and includes member preprocessing, forwarding and original-grid probabilities. Native inference, initial transfer, observations, selection, dispatch, fusion and final argmax are outside this proxy. Complete calibration instead measures the batch-8 pipeline under the output contract below. Their contrast therefore combines a batching change with a measurement-boundary change; it does not isolate the effect of adding router overhead.

Table A2 records the two selectors' calibration scores on the same candidates.

Table A2: The frozen ACDC calibration menu on 461 images at 0.8 CE/second. Times are ms/image; scores use the same actual calibration CE and the named cost boundary. Bold scores mark each selector's maximum. These are calibration records, distinct from the 406-image final results.

s	CE	Proxy time	Complete time	Proxy score	Complete score
0.25	0.849679	76.774	67.197	-0.911098	$\mathbf{-0.903436}$
0.5	0.851541	72.504	65.099	-0.909544	-0.903620
1	0.863433	56.868	56.080	$\mathbf{-0.908928}$	-0.908297
2	0.893015	31.801	40.933	-0.918455	-0.925761
4	0.957317	6.256	25.404	-0.962322	-0.977640

The calibration collection followed its design lock. We used $s = 1$ and 0.25. Local UTC records on September 9, 2026 place calibration lock at 05:27:12, policy selection at 05:42:26, and the final protocol lock at 05:42:38 before evaluation sessions. These are recorded internal freezes, not external preregistration. Across the final evaluation, all 30,450 image executions had exactly matching actions, CE and confusion counts across repeats. The 0.25 versus 0.5 calibration utility margin is 0.000183293; the recorded selection does not precisely identify an optimal scale. The 0.002 design reference was predeclared, not validated as an application margin.

Calibration-composition sensitivity. A retrospective deletion analysis uses the original five candidates, four recording blocks and three repeats. We average loss/time sums across repeats within each block, then divide pooled retained sums by retained image count. Omitting rain/fog/night/snow selects proxy multipliers $0.5/1/1/1$ and complete-cost multipliers $0.25/0.25/0.5/0.25$, respec-

tively. Thus the cost rules select different candidates in every deletion, while the full-cohort winners $1/0.25$ are not invariant. Without night, $s = 0.5$ beats $s = 0.25$ by only 0.000012671 complete utility. Each deletion also removes an entire weather condition: this is composition sensitivity, not a selection probability or confidence interval. All original selections and final contrasts remain frozen; no test outcomes enter this analysis.

ACDC complete-output contract. ACDC uses driver 595.95, Torch 2.1.0/CUDA 11.8/cuDNN 8700, Transformers 4.44.2, sklearn 1.3.2, NumPy 1.26.4 and FP32 with both TF32 flags off. Its contract returns GPU-resident original-grid probabilities and hard predictions for every method. At 1080×1920, one 19-channel FP32 probability tensor is 157.6 MB and the returned int64 label map is 16.6 MB (decimal units). Interpolating and softmaxing each selected view precedes averaging and argmax; output work therefore depends on action choice. It is not a common constant that necessarily cancels. A timed microblock retains at most 40 images before scoring actual outputs outside the timer; larger recording blocks sum these intervals. CE floors probabilities at 10^{-8} and averages valid pixels within images, then images equally. Loading, disk decode, warmup, labels/scoring and serialization are excluded. This probability-output contract differs from the historical hard-label replay.

Three separate variation analyses. Finite-cohort means are arithmetic on the recorded outcomes and workloads. To describe composition sensitivity, ADE uses 1,999 whole-category reweightings with fixed fitted policies; static/oracle comparators are recomputed within a draw when specified. Of the 136 scene categories (not semantic classes) in the 400-image cohort, 92 are singletons, 115 contain at most two images and the largest has 67. Categories can share indoor/outdoor or other conditions. Their bootstrap is a *declared composition-sensitivity analysis*, not evidence of 136 independent population domains. A positive range describes that reweighting law; conditioning on fitted heads alone does not justify population coverage.

Execution-repeat analyses hold image quality and policy fixed. ADE uses all $3^3 = 27$ ordered resamples for three repeats and 19,999 draws for ten repeats, with inverse empirical-CDF quantiles. The source-refit analysis instead changes fitted predictions across 40 source-category resamples. These analyses answer different questions and are not pooled into a joint interval.

ACDC retains the frozen 20,000-draw paired bootstrap of whole recording blocks and whole sessions, seed 20260909. Repetitions are averaged within each block/session/method cell. Draws use the same block weights for CE sums, elapsed seconds and image counts, preserving quality–time pairing and batch structure. Block sizes range from 2 to 78 images; sessions occupy one GPU/time window. The primary interval is the 2.5–97.5 percentile range; three static contrasts use Bonferroni-adjusted percentiles (nominal familywise 95%). This preserves the prospective estimand and all numerical results. Population interpretation requires exchangeable or adequately modeled blocks and sessions, which repetition and filename grouping do not establish. The interval conditions on fitted policies and does not include retraining or selector variation.

Descriptive ACDC mIoU contrast. The proxy-calibrated-minus-multiscale mIoU difference is 0.107 percentage points. This secondary analysis is reported descriptively. An earlier exploratory interval is omitted because its complete resampling specification could not be recovered from the retained documentation. No nominal coverage or population-level mIoU/runtime dominance is claimed. The prespecified CE/time bootstrap, selected policies, and primary interval endpoints are unchanged.

C Comparator, learning, and selection controls

These experiments test the steps connecting signed-value learning to a specified operating point: comparator choice, learner capacity, source refitting, metric definition and price domain. They share frozen backbone outcomes while varying the named choice. Their designs followed inspection of the original cohorts. Unless specified, paired ranges use 1,999 whole-category draws; their composition-sensitivity interpretation and the separate refit analysis are in Appendix B. The 0.005 reporting threshold was fixed before these control runs; it is not a validated equivalence margin. Each subsection retains its fitted-policy and selection boundary. We also used the Mask2Former-Swin-S model for the COCO dataset evaluation.

C.1 TARGET-BEST AND TRANSPORTED STATIC COMPARATORS

Table A3 distinguishes an evaluation-best static from an action selected on the source and transported to evaluation. Nested rows aggregate fold-specific binary actions and selected prices, with fallback folds contributing zero; they do not represent one policy at one price. Legacy ADE B0/B1/B2 scalar scores average entropy/JS over annotation-defined valid pixels, unlike the corrected annotation-free single-B2 router. The latter, selected at $\lambda = 0.8$, has $0.005525\,[-0.007269, 0.019454]$ surplus over target-best static and $0.005525\,[-0.007140, 0.027088]$ over transported static on 400 images. Equal point estimates need not yield equal ranges because the former comparator is reselected.

Table A3: Nested utility surpluses in 10^{-3} CE units, with 9,999-draw category-composition ranges conditional on fitted policies. The target-best comparator removes the positive lower bound of the historical ADE B1 result.

Procedure	Versus target-best static	Versus transported static
ADE B0	$0.802\,[-1.095, 1.999]$	$1.152\,[-0.840, 2.601]$
ADE B1	$4.223\,[-1.651, 5.911]$	$7.312\,[1.772, 10.483]$
ADE B2 nested	$-0.644\,[-5.979, 1.861]$	$2.360\,[-3.680, 6.511]$
COCO nested	$0.262\,[-1.266, 0.932]$	$0.262\,[-0.953, 0.978]$

Exact-count random allocation preserves each router's action counts but randomizes image assignments. Router CE improves over its expectation by $0.012870\,[0.006419, 0.023750]$ on ADE and $0.001443\,[-0.001957, 0.002428]$ on COCO; counts are rematched in every draw. Matching counts does not match seconds: router/random optional costs are $0.057459/0.057375$ s/image on ADE and $0.161400/0.161740$ on COCO. The artifact separates 1,999 allocation draws from 9,999 category-composition draws. COCO's 998-image evaluation has only five dominant-supercategory clusters, limiting bootstrap precision. Semantic confusion-count results do not turn the earlier fixed-partition semantic-relabeling PQ endpoint into adaptive panoptic inference.

C.2 MATCHED SPLITS, TRAINING SIZES, AND COST-SENSITIVE OBJECTIVES

The same images and head family receive category-disjoint and image-random five-fold evaluation with matched train/test sizes. ADE training sizes are 400/800/1,600 (400 test images); COCO uses 999–1,000/1,998–2,001/3,996–4,003 (997–1,004 test images). Smaller sets are nested label-independent subsamples; every image receives one OOF decision per size. Each training subset independently fits embedding scaling, PCA64 and combined-feature scaling. Native embeddings, annotation-free entropy and image size are available initially; native/flip JS is available for the subsequent incremental multiscale decision.

Ridge predicts gain with $\alpha = 100$. Weighted logistic uses the same features, weights $|B - \lambda C|$, sign target $B - \lambda C$, inverse regularization strength $C_{\mathrm{LR}} = 1$, and threshold 0.5. This is a convex cost-sensitive surrogate (Bolukbasi et al., 2017), not exact optimization of empirical policy value. Prices $0.1/0.3/0.5$, primary price 0.3, and head settings preceded fitting. Table A4 compares the resulting full-training policies.

Table A4: Full-training controls at $\lambda = 0.3$, in 10^{-3} CE utility units with pointwise 2.5–97.5% category-composition ranges conditional on fitted policies. The first column is surplus over target-best static; the others are paired policy-value changes. CS denotes weighted logistic.

Setting	Category ridge	Random − category	CS − ridge, category
ADE flip	$-1.045\,[-2.553, 0.106]$	$0.130\,[-0.828, 1.081]$	$-0.190\,[-1.492, 1.120]$
ADE multi	$-1.797\,[-4.453, 0.725]$	$0.911\,[-0.505, 2.640]$	$-1.464\,[-3.566, 0.401]$
COCO flip	$0.024\,[-0.325, 0.482]$	$0.085\,[-0.377, 0.450]$	$-0.057\,[-0.601, 0.592]$
COCO multi	$0.238\,[0.018, 0.469]$	$-0.013\,[-0.222, 0.148]$	$-0.321\,[-0.674, -0.002]$

Image-random splitting does not establish improved policy value at full training size. Quarter-to-full ridge improvement is $0.003117\,[0.001075, 0.005550]$ for ADE image-random flip and $0.000345\,[0.000171, 0.000503]$ for COCO category-disjoint multiscale; the other

six contrasts are unresolved. All 288 settings and 456 paired contrasts are retained. Exponentiated log-cost predictions need not equal arithmetic conditional means. Replacing them by training-mean seconds changes the four category-disjoint primary values, in table order, by $0.000025\,[-0.000095, 0.000137]$, $0.000304\,[-0.000080, 0.000955]$, $0.000032\,[-0.000030, 0.000108]$, and $0.000016\,[0, 0.000054]$. Raw-seconds regression with positive clipping is also retained. These controls do not establish conditional calibration or a Bayes limit.

C.3 Expressive frozen-feature heads

The capacity/utility controls use ADE folds 0–2 for training (1,200 images), fold 3 for selection (400), and fold 4 for testing (400), without post-selection refitting. Train-standardized native 1,024-dimensional embeddings are either retained or reduced by train-only randomized PCA64. Both append annotation-free entropy, log image area and log aspect ratio, followed by train-only combined scaling. Heads predict cumulative native-to-flip and native-to-multiscale gains; native gain/cost is zero. A common $\alpha = 100$ ridge on the three scalar features predicts log optional costs for every head; archived optional times are unchanged.

The four prespecified families are ridge ($\alpha = 100$); ReLU MLP (one 64-unit layer, $L_2 = 0.1$, batch 64, Adam rate 0.001, at most 300 epochs); histogram boosting (100 iterations, depth 3, at most 8 leaves, rate 0.05, minimum leaf 20, $L_2 = 1$); and ExtraTrees (128 trees, depth 8, minimum leaf 20, feature fraction 0.25). MLP early stopping uses a 15% internal training holdout and patience 20, never the external validation/test folds. Targets are train-standardized; no convergence warnings were reported. Two representations, four families and CE/per-image active-union-mIoU gain targets give 16 fitted heads.

The main text reports the eight CE heads' matched positive-area recoveries, using 61 prices in $[0, 1.5]$ and the test-best three-action static envelope. This single-test-fold, joint-action comparison differs from historical binary OOF summaries. At predeclared $\lambda = 0.75$, validation selects full-feature ExtraTrees among the eight CE heads and three statics. Test surplus is $0.003815\,[-0.010008, 0.017396]$, at 0.070078 optional s/image. Its uncertain advantage bounds conclusions to the tested frozen-feature families; new-head overhead is not included in the original router's integrated runtime measurements. Table A5 reports area recovery and paired head differences.

Table A5: Paired head-area sensitivity on the same 400-image test fold and 61-price window $[0, 1.5]$. Areas are percentages of oracle-residual area. Ranges use 1,999 shared category resamples, recomputing static/oracle envelopes; they condition on fitted heads and do not control multiplicity or guarantee population coverage. Differences use unrounded integrals.

Head / features	Positive	Signed	Difference from PCA64 ridge [range]
Ridge / PCA64	5.04	2.31	—
MLP / PCA64	1.68	−14.81	$-3.36\,[-17.92, 0.76]$
Boosting / PCA64	5.19	1.41	$+0.16\,[-14.65, 6.30]$
ExtraTrees / PCA64	7.33	7.29	$+2.29\,[-12.05, 6.51]$
Ridge / full	0.00	−30.87	$-5.04\,[-23.58, -0.03]$
MLP / full	0.31	−22.22	$-4.72\,[-22.74, 0.44]$
Boosting / full	1.51	−4.58	$-3.52\,[-19.02, 2.05]$
ExtraTrees / full	5.57	5.05	$+0.54\,[-10.42, 7.52]$

The unnormalized oracle-residual integral is 0.05460785 over this price window; PCA64 ridge positive/signed residual integrals are 0.00275020/0.00126198. All raw integrals and paired draws accompany the table. Clipping, comparator selection and ratio normalization are nonsmooth, so these ranges remain exploratory sensitivities. The 5.04-to-7.33% point difference does not establish an expressive-head advantage. Predictions on source images from these eight heads are not out-of-fold (OOF).

C.4 CLASS SUPPORT AND DIRECT VALIDATION-MIOU SELECTION

The main metric-support comparison is summarized in Section 5; Table A6 gives its cross-fitted scope. Active-union mIoU averages class IoUs with positive aggregate union; a class absent from ground truth may enter through false positives. Fixed-150 averages all ADE classes, assigning zero to undefined IoUs. Fixed-GT support uses the 135 classes present in the original test ground truth, held fixed through resampling; its results equal fixed-150 times $150/135$, not independent evidence. Full confusion matrices verify identical GT counts across actions and exact compact-statistic reconstruction. Rare classes can disappear in bootstrap draws and shift absolute percentile ranges downward, so paired difference ranges are reported without promising nominal population coverage. Neither common support nor the original point reversal identifies a causal CE-proxy mismatch.

Cross-fitted support scope. The five saved nested outer-fold heads each exclude their evaluated fold from fitting. Re-thresholding their price-independent predictions at 0.8 gives 2,000 OOF decisions; these are not the single-split expressive-head predictions. Intersection/union counts are joined by image ID and pooled before evaluating mIoU. Two 400-image folds reverse point ordering under a fixed denominator, in opposite directions, while pooled support contains all 150 classes (Table A6). The pooled router/flip values are 46.1633/46.1733% under either convention; fold-specific models are not one deployed router or independent confirmation.

Table A6: Fixed-price 0.8 cross-fitted router minus flip. Differences are mIoU percentage points. K_h/K_f are active class counts; pooled metrics aggregate counts rather than averaging fold mIoUs. No uncertainty or prevalence guarantee is inferred from five dependent fitted procedures.

Cohort	GT classes	K_h/K_f	Active difference	Fixed-150 difference
Fold 0	118	136/137	$+0.5692$	$+0.2831$
Fold 1	135	142/141	-0.4548	-0.1677
Fold 2	137	143/143	-0.2497	-0.2381
Fold 3	136	145/147	$+0.3834$	-0.0935
Fold 4	135	144/143	-0.0892	$+0.1915$
Pooled 2,000	150	150/150	-0.0100	-0.0100

In fold 4, bar and lake have router-only predicted support (280 and 23 pixels); barrel has flip-only support (77 pixels), with zero GT/intersection for all three. The exact decomposition $D_{150} = (K_h/150)D_{\mathrm{act}} + ((K_h - K_f)/150)M_{\mathrm{act}}(f)$ uses $M_{\mathrm{act}}(f)$, the active-union mIoU of flip, and gives $-0.085603 + 0.277070 = 0.191467$ points. This support term is not positive IoU from an absent class. Thirty label-independent nested image subsets at sizes 100/200/400/800/1,600/2,000 yield 10/8/5/2/0/0 sign reversals under the fixed OOF predictions. These are subset sensitivities, not population frequencies or replicas of category-disjoint sampling. All 2,000 annotation hashes/counts are retained; the existing 400-image GT counts reproduce exactly. For each of the 16 frozen heads, deterministic selection considers zero plus 121 log-spaced price multipliers in $[10^{-3}, 10^{3}]$, scaled by training gain standard deviation/training mean multiscale optional cost. Together with three statics this gives 1,955 candidates before deduplication. Each chooses predicted gain minus priced predicted cost. Caps are 25/50/75% of training mean multiscale optional cost (24.054/48.109/72.163 ms/image). Among policies within each *actual validation* cost cap, selection separately maximizes CE reduction, active-union dataset mIoU, or fixed-150 dataset mIoU. Native is a feasible fallback; ties follow a fixed order. No test outcome selects a head, price or convention. Transported costs need not match one another or remain below the validation cap. Table A7 shows the frozen selections at their realized costs.

Active-union selection chooses CE-trained PCA64 MLPs at the first two caps and full-feature boosting at the third. Active test differences from CE selection are $+0.1412\,[-0.3995, 0.8753]$, $-0.2521\,[-0.4423, 0.6980]$ and $+0.1983\,[-0.4724, 0.7739]$ percentage points. Fixed-150 selection chooses per-image *active-union*-gain-trained full ExtraTrees, PCA64 ExtraTrees and full boosting. Fixed-150 differences are zero (identical actions), $-0.0559\,[-0.3383, 0.1938]$ and $+0.1611\,[-0.1209, 0.3438]$ points. These comparisons use unequal realized costs and do not estimate a quality advantage at equal test budget. Six of nine policies violate their transported caps; a validation cap alone is not a deployment guarantee. A budgeted system needs a separately validated

Table A7: Frozen deterministic policies selected under common validation cost caps. Test mIoUs are percentages; optional cost is ms/image. Active and Fixed 150 denote the validation metric, distinct from the learned per-image gain target.

Selection	Cap	Test active mIoU	Test fixed-150 mIoU	Test cost
CE	25%	41.4220	39.4889	23.975
CE	50%	41.6273	39.6847	61.091
CE	75%	41.5129	39.5756	74.447
Active	25%	41.5631	39.6235	29.226
Active	50%	41.3752	39.7202	56.592
Active	75%	41.7111	39.7646	71.482
Fixed 150	25%	41.4220	39.4889	23.975
Fixed 150	50%	41.2800	39.6288	54.263
Fixed 150	75%	41.6818	39.7367	74.697

safety buffer or an online budget/fallback rule. Appendix F plots all nine points against realized proxy cost; no complete-execution claim is made for these untimed heads.

Randomized interpolation adds a distinct estimand: mIoU of expected confusion counts $M(\mathbb{E}[C])$ need not equal finite-assignment $\mathbb{E}[M(C)]$. Policies and the input-independent comparator match validation expected cost, then freeze their probabilities. At the 75% cap, the CE-selected policy gives 41.2451% versus 42.1160% active mIoU across 999 assignments, with 145 versus mean 142.002 active classes. Fixing 150 classes reduces the gap from 0.8709 to approximately 0.000104 percentage points. Independent inverse-CDF sampling reproduces the means; confusion expectations agree within 3.73 Monte Carlo standard errors. This surrogate and a preseeded assignment are retained separately, not used to claim expected deployment mIoU superiority.

C.5 Forty source refits: separating policy and objective variation

Forty source-category bootstrap runs independently resample folds 0–3, refit scaling/PCA64 and four gain/log-cost ridge heads, and tune alpha from $\{0.01, 1, 100\}$ with inner-training preprocessing. The historical procedure uses folds 0–2 to fit, selects a price on fold 3, then refits on 0–3. Head fitting and alpha selection do not depend on price, so saved predictions permit exact replay under a common price without new fits. All analyses hold the backbone outcomes and 400 target images fixed. Table A8 separates changes in the fitted policy from changes in the evaluation preference.

Table A8: Three distinct summaries of the same 40 fitted heads. Repricing holds archived actions fixed; re-thresholding recomputes their predicted-net-value argmax at 0.8. Empirical 2.5–97.5 percentiles use linear interpolation and are not population confidence intervals or success probabilities.

Evaluation rule	Median surplus	Empirical range	Count > 0	Count > 0.005
Own selected price	0.001858	$[-0.007966, 0.009055]$	26/40	10/40
Archived actions, repriced 0.8	0.006621	$[-0.001777, 0.013003]$	34/40	27/40
Actions re-thresholded at 0.8	0.005301	$[-0.002254, 0.011212]$	36/40	21/40

Selected prices have median 1, observed range 0.3–1.5, and empirical 2.5–97.5 percentiles 0.54375–1.45125. The first row therefore mixes changing preferences with fitting; it is not reliability under one utility. The common-price row supports more frequent positive point surplus but includes negative refits and does not remove prior cohort inspection. All original selected-action surpluses are reconstructed before re-thresholding.

Five nested rotations use each fold as test, the preceding fold as validation, and the other three for fitting before refit. Their original fold/price/surplus triples are $0/0.65/0.016112$, $1/0.60/-0.001420$, $2/0.90/0.011863$, $3/0.75/0.003688$, $4/0.80/0.005525$. Differing prices and overlapping source data preclude interpreting them as independent deployments. Their common-0.8 actions are used only for the cross-fitted support check above.

C.6 PRICE RANGE, DENSITY, NORMALIZATION, AND OPERATING POINTS

Holding saved OOF predictions fixed, we evaluate annotation-free PCA64 ridge/log-cost controls on $[0, 0.5]$, $[0, 1.5]$ and $[0, 3]$ CE/second with 31/61/151 points. Positive area integrates positive policy-minus-target-best-static surplus; signed area retains harm. Both divide by label-oracle opportunity area against that same static envelope. The 240 archived summaries include mean-cost/raw-cost and explicitly identified legacy-head sensitivities; none selects a price from test outcomes. Table A9 compares the three physical price windows.

Table A9: Binary-action OOF positive-area recovery (%) at 151 prices, with low-range signed recovery. These splits/estimands differ from the joint-action capacity test.

Dataset	Action	$[0, 0.5]$	$[0, 1.5]$	$[0, 3]$	Signed $[0, 0.5]$
ADE	Native to flip	0.33	6.37	3.29	-7.59
ADE	Flip to multiscale	0.00	6.62	3.94	-9.55
COCO	Native to flip	5.36	3.10	2.60	4.94
COCO	Flip to multiscale	7.95	5.48	4.72	7.52

Density changes alter these positive ratios by at most 0.49 percentage points across tested ranges; changing the physical range matters more. The source-predicted break-even scale (mean predicted gain/mean predicted cost) supplies a dimensionless coordinate. Reexpressing the *same physical interval* preserves the ratios; fixing a normalized $[0, 2]$ interval changes physical coverage and is a separate sensitivity, not utility invariance. The artifact retains recoveries at fixed prices $0.1/0.3/0.5$ and validation-selected points, so descriptive areas do not replace operating-point surplus. Full grids, model settings, allocation draws and paired uncertainty records remain in the archived companion.

D EXECUTION MEASUREMENTS AND THEIR COMPARATORS

D.1 ORIGINAL SELECTION AND PROXY TIMING

The original source fit uses folds 0–2. On fold 3, each of the 31 prices $0, 0.05, \ldots, 1.50$ selects actions by predicted net value and is evaluated by mean realized net value minus the best of the three validation statics. The first maximizer is 0.8 (surplus 0.0057678627); source folds 0–3 then fit the final heads, with test fold 4 unused in either selection. This is a historical preference search, not application calibration.

The 256-image initial cache and 1,744-image extension use identical member-timer bodies. Each stored batch-one FP32 time surrounds a call on an already GPU-resident image, with CUDA synchronization before and after: GPU resize, optional flip, normalization, forward, original-grid logit interpolation, inverse flip and softmax. Decode/initial H2D precede the timer; fusion, CE extraction and cache writes follow it. Two model-only warmups (512 and 640) precede collection; members run in fixed native/flip/384/640 order, with one stored time per image/member. Joint proxy costs are 0, t_{flip} and $t_{\text{flip}} + t_{384} + t_{640}$. They omit feature acquisition, CPU routing, fusion and final decision assembly and are not repeated complete-workload measurements. The cache protocol records RTX 3090/FP32/batch one, but does not provide a complete contemporaneous library/driver/TF32 manifest. Differences from later complete runtime are not attributed to an isolated software or hardware factor.

D.2 COMMON-PRICE EXECUTION AND FRESH-OUTPUT VERIFICATION

The primary comparison fixes the original ridge's saved actions, the 400-image cohort, the native/flip/multiscale static menu and price 0.8. It prices complete method time for both router and statics. Best static is flip under optional proxies and multiscale under executed costs, so the surplus change is not just router overhead. Adding the measured B0, B2-384 and two input-independent mixtures does not change the best comparator at this price. Table A10 reports fixed-cohort timing-repeat sensitivity.

Each draw averages matched complete-workload records and reselects best static. With three repeats, interpolation between the 27 resamples gives a positive 2.5% endpoint despite a nega-

Table A10: Fixed-cohort workload-repeat sensitivity at 0.8. Surpluses and endpoints are in 10^{-3} CE. Quantiles use the inverted empirical CDF: all $3^3 = 27$ ordered resamples for three repeats, and 19,999 resamples for ten repeats. Quality and policy are fixed; these are not population confidence intervals.

Batch	Repeats	Mean surplus	2.5–97.5% repeat quantiles	Negative draws
1	3	$+3.309$	$[-0.034, 9.387]$	1/27
8	3	-2.942	$[-3.011, -2.906]$	27/27
32	10	-2.918	$[-2.939, -2.895]$	19,999/19,999

tive resample; the retained inverse-CDF range includes that resample. Draw frequencies are not probabilities of future success. Separately, 1,999 paired category draws give CE improvements over native/flip/multiscale of $0.054627\,[0.040266, 0.075270]$, $0.026755\,[0.015075, 0.045722]$, and $-0.014222\,[-0.021284, -0.009622]$. They resample quality, not a newly rebatched target runtime. The two sampling units are not pooled into a joint deployment interval. For scale, holding the batch-8 multiscale time credit fixed at 0.011280 and shifting only its paired CE range gives $[-0.010003, 0.001659]$. This is a router-versus-multiscale fixed-time sensitivity, not a joint net-value confidence interval or a reselected best-static bound. Tight repeat-only timing ranges therefore do not remove the separate quality-sampling uncertainty.

A subsequent Windows verification session checks the cached-quality approximation with one full replay of each of four methods at batch sizes 1/8/32: 12 workloads and 4,800 image executions, after smoke checks and warmup. All 1,200 live router decisions match, with native/flip/multiscale counts 133/41/226 at every batch size. Maximum absolute setting-mean and per-image CE deviations from the archived quality are 2.25×10^{-6} and 1.57×10^{-4}, respectively. The original inference function bodies, image/annotation/model hashes and frozen numeric policy are verified. A read-only forward hook retains each batch's logits; CE reconstruction preserves decoder-grid flip, original-grid interpolation, per-member softmax, ordered probability averaging and the 10^{-8} clamp. Capture/reconstruction overhead makes these new times unsuitable for replacing the original timing records. This supports close numerical agreement for this session, not bitwise parity, original-timed-logit recovery or uninspected confirmation.

D.3 Synchronous Windows replay

Windows/RTX-3090 replay uses Python 3.9.18, PyTorch 2.1.0/CUDA 11.8 and two CPU threads. Validation with the source fit fixes multiscale probabilities $p_{\mathrm{Q}} = 0.340619$ to match CE and $p_{\mathrm{C}} = 0.225776$ to match optional proxy cost; the other action is flip. Router refitting on folds 0–3 and transfer to fold 4 need not preserve those equalities. Methods execute all 400 test images at batch sizes 1/8/32 with 3/3/10 workload repeats and three warmups. Each mixture randomly orders batches and assigns a prefix of whole batches nearest pN images to multiscale, including the partial final batch. No batch is split by the assignment. At batch 32, the quality mixture selects 128 images in four repeats and 144 in six, averaging 137.6/400 (34.4%). The 16-image partial batch explains the 144-image count. Method blocks are randomly ordered within repeat. B0 and B2-384 run subsequently, one resident model at a time, on the same image orders.

Timing includes transfer, preprocessing, native observations, CPU routing, selected forwards, original-grid assembly and synchronization; it excludes disk decoding, loading and queues. Cached quality is attached after action-parity checks. The main table gives mean time and quality; Table A11 retains the additional batch tails and paired workload contrasts. Mixture CE at batch sizes 1/8/32 is $0.80683/0.81123/0.80899$ for p_{Q} and $0.81183/0.81552/0.81392$ for p_{C}.

The router has lower CE but greater mean time than either measured mixture. Router-minus-p_{Q} CE ranges are $-0.010886\,[-0.012372, -0.008195]$, $-0.015289\,[-0.017882, -0.012237]$, and $-0.013047\,[-0.014670, -0.011359]$ at batch sizes 1/8/32. These describe allocation/repeat variability, not population-quality uncertainty. Derived evaluation-mean static frontiers are descriptive; they are not validation-selected measured systems. Figure A2 retains active-union mIoU, whose denominator sensitivity is reported in the main text.

Table A11: Additional synchronous records: descriptive p95 elapsed milliseconds per *batch*, not per image or request, and router-minus-mixture mean ms/image. Each method has 1,200/150/130 timed batches at batch sizes 1/8/32. The paired mean-time contrasts have 95% ranges from 4,999 whole-workload repeat resamples within one session.

Method	Batch 1	Batch 8	Batch 32
B0 native	16.1	44.2	162.0
B2 at 384	29.4	67.9	245.1
B2 native	33.5	103.8	386.9
B2 flip	67.2	207.5	772.2
B2 multiscale	115.0	424.9	1601.0
B2 router	122.1	366.9	1251.1
Mixture p_{Q}	116.6	411.2	1593.1
Mixture p_{C}	106.1	407.5	1584.1

Batch	Router minus p_{Q}	Router minus p_{C}
1	$6.841\,[2.996, 10.401]$	$13.535\,[9.867, 15.960]$
8	$3.250\,[3.127, 3.453]$	$6.353\,[6.246, 6.497]$
32	$2.414\,[2.095, 2.739]$	$5.371\,[5.096, 5.697]$

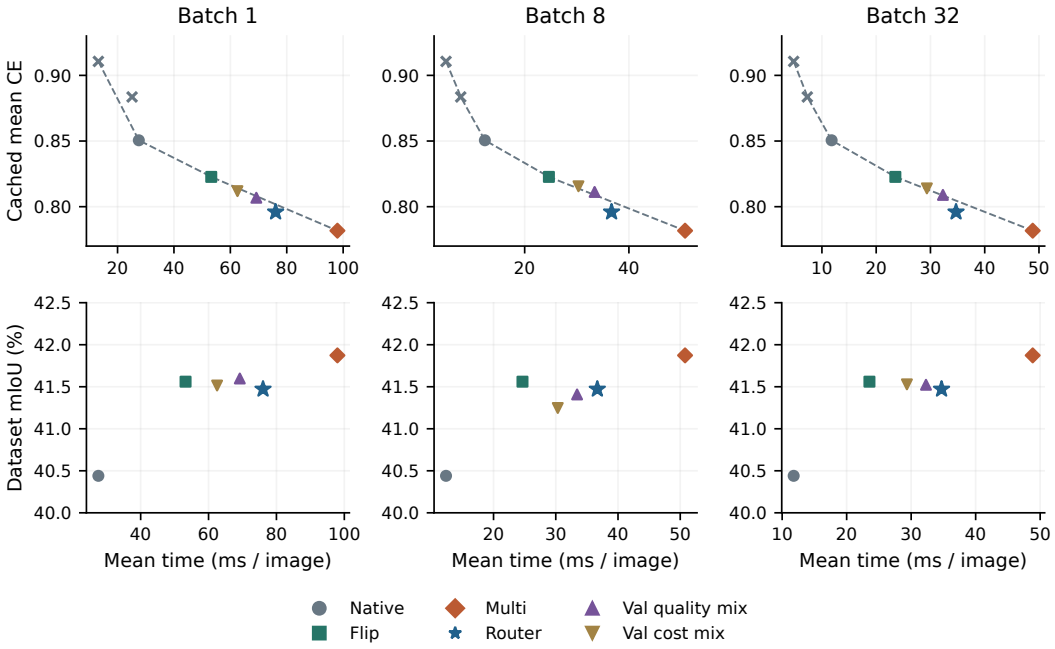

Figure A2: Measured Windows mixtures (triangles) and descriptive interpolated static boundaries (dashed CE curves). Quality uses verified cached actions on 400 ADE images. Active-union mIoU and synchronous means are distinct from fixed-support utility and asynchronous response time.

The separate B0-to-B2 cascade pays a full B2 native pass after B0 when delegated; standalone B2 skips B0 and is a different system comparator. Its validation-selected ridge at price 0.1 has 34.25% calls, CE 0.892835 and conditional target-best surplus $-0.039401\,[-0.109470, 0.031426]$. Integrated mean time is 24.665/15.038/13.903 ms/image at batch sizes 1/8/32. Full head/price and timing records remain in the experimental companion; no main conclusion depends on selecting a favorable cascade setting.

D.4 FRESH WSL2/LINUX REPLAY

On the same physical RTX 3090, four methods receive three repeats of 400 images at batches 1/8/32: 36 workloads, 14,400 image actions and 5,556 timed batches. Python 3.11.8, PyTorch 2.6.0/CUDA 12.4 and Transformers 4.49.0 use FP32, two threads, matmul TF32 off, cuDNN TF32 on, and benchmark/deterministic modes off. The same timing boundary applies. Scaler/PCA/ridge parameters and price 0.8 transfer numerically without fitting; software, JPEG libraries, WSL and policy wrappers still differ. This is a comparison of complete stacks, not an isolated OS effect. Table A12 reports its fresh-output quality and complete runtime.

Table A12: Linux means over three full workloads per setting, with quality recomputed from fresh Linux outputs. mIoUs are percentages. Active-union class counts are 145/143/143/144 in row order; fixed 150 assigns zero to undefined IoUs.

	Mean ms/image			Fresh quality		
Method	Batch 1	Batch 8	Batch 32	CE	Active mIoU	Fixed 150
Native	24.785	13.551	12.995	0.849121	40.4569	39.1084
Flip	45.128	27.413	25.744	0.821538	41.5525	39.6134
Multiscale	88.930	55.199	53.342	0.781561	41.8985	39.9432
Router	67.869	39.651	37.952	0.795538	41.4794	39.8202

A retained strict-parity attempt stops when one image changes native to flip. The amended timing run uses live decisions: 3,600/3,600 match a fresh Linux reference and 3,591 match Windows. All nine mismatches are the same unique image (`ADE_val_00001191`), changing native to flip in each of three repeats at each of three batch sizes. Source JPEG hashes agree, but decoded pixels differ between Pillow 10.2/libjpeg 9 and Pillow 11.1/libjpeg-turbo 3.1; disabling cuDNN TF32 does not remove the decision difference. Fresh action-loss/confusion statistics evaluate the recorded actions without asserting logits parity across all batch compositions. Conditional paired router-minus-flip mIoU ranges are $-0.0731\,[-0.5661, 0.8443]$ active-union and $+0.2068\,[-0.0815, 0.4923]$ fixed-150 points. Both include zero. The archived failed attempt and decoder diagnostics preserve the amendment; they are excluded from the 36-workload timing summary. At the same 0.8 preference, Linux-specific output quality and Linux mean time give surpluses $+0.002871/-0.001538/-0.001665$ at batch sizes 1/8/32, again against multiscale. This is a related complete-stack sensitivity on the same images/GPU with one changed action, not an independent dense-task replication or an isolated OS intervention.

D.5 CROSS-PLATFORM EVALUATION ON T4 AND A100

Experimental design and workload normalization. The original SegFormer-B2 checkpoint and numerical ridge parameters are transferred to T4 and A100 without refitting. All comparisons retain the same 400 ADE images and $\lambda = 0.8$ CE/second. Five systems are evaluated: native, flip, multiscale, live routing and supplied actions. Tesla T4 is evaluated at batch sizes 1/8 and NVIDIA A100-SXM4-40GB at 1/8/32. Each platform contributes one session with three randomized complete-workload repetitions per setting, yielding 30 and 45 method–batch–repeat workloads, respectively. Preliminary functionality checks are excluded. Image order and batch membership are matched across methods within a repeat; method order is randomized. At batch size 32, twelve 32-image groups and one 16-image group cover the cohort. Runtime is total workload seconds divided by 400, not an unweighted mean of batch-normalized times. The 640-pixel pass retains a 16-image chunk limit, so batch size denotes workload grouping rather than every individual forward pass. Table A13 reports all five systems.

Execution environments. Both environments are hosted on Colab. Their manifests record Linux, Python 3.13.15, PyTorch 2.6.0+cu124, CUDA build 12.4, cuDNN 9.1, Transformers 4.49.0, NumPy 2.2.6, Pillow 11.1.0 and driver 580.82.07. Execution uses FP32 without AMP, matmul TF32 disabled and cuDNN TF32 permitted, consistent with the WSL and subsequent Windows output-verification settings; ACDC disables both flags. FP32 tensor storage therefore does not imply that all permitted convolution arithmetic excludes TF32. CPU/BLAS limits are two threads. Host resources

nevertheless differ: A100 exposes 12 Xeon CPUs at 2.20 GHz, versus two at 2.00 GHz for T4. The comparison concerns complete execution environments, not an isolated change in GPU model.

Timing and quality measurement. Timing begins with decoded CPU uint8 images and includes transfer, normalization, native inference, live observations and selection, conditional passes, original-grid assembly, hard argmax and synchronization. Hard maps are computed and discarded. Decoding, loading and scoring are outside the timer. Quality is scored through a separate untimed execution with the same groups and assembly operations. Diagnostic output hashes apply to that execution, not to the discarded timed maps. Utility uses the corresponding device/batch CE; the diagnostic agreement below is not a bitwise comparison with timed outputs.

Table A13: **Complete runtime and fresh quality.** Runtime is mean ms/image over three complete repeats. CE and pooled mIoU agree across each device's batch sizes at the displayed precision. mIoU uses a common 150-class denominator, assigning zero to absent-union classes. A dash denotes an unmeasured setting. The supplied-action system removes the observation/head path using decisions from an untimed prepass. Contrasts are computed from unrounded records.

Method	CE	mIoU (%)	Batch 1	Batch 8	Batch 32
Tesla T4					
Native	0.84913	39.108	63.243	75.544	—
Flip	0.82155	39.614	133.963	151.799	—
Multiscale	0.78157	39.943	280.317	314.000	—
Router	0.79555	39.820	196.192	220.224	—
Supplied actions	0.79555	39.820	195.180	219.452	—
A100-SXM4-40GB					
Native	0.84912	39.109	22.243	9.641	9.230
Flip	0.82154	39.614	40.691	18.316	17.535
Multiscale	0.78158	39.943	77.775	37.051	35.772
Router	0.79556	39.820	58.534	27.070	25.652
Supplied actions	0.79556	39.820	57.881	26.858	25.481

Input, policy and numerical consistency. The recorded integrity validation reports agreement for all 45 A100 completion hashes and 414 integrity conditions, including frozen assets, configuration and seven historical timing functions. These provenance checks are distinct from numerical output agreement. All 400 decoded RGB hashes match between T4, A100 and the prior Linux RTX 3090 replay; all differ from the historical Windows decoder hashes. Decisions match across those Linux stacks: 132 native, 42 flip and 226 multiscale, with only `ADE_val_00001191` changing from native to flip relative to Windows. Within every T4/A100 matched group, prepass, timed and quality-pass decisions agree. Router and supplied-action diagnostic outputs, CE and confusion counts agree exactly. Across devices, hard maps are not bitwise identical despite matched pixels and actions; mean router CE differs by approximately 1.4×10^{-5}. Cross-stack accounting therefore includes the relative-loss correction in Appendix A.2. Active-union mIoU has method-dependent class support; the common denominator in Table A13 avoids that comparison artifact.

A100: absolute savings and residual runtime budgets. Multiscale has the highest static utility in every A100 repeat. The router's approximately 0.013982 CE penalty requires 17.48 ms/image saved by Eq. (2); observed savings are $19.241/9.981/10.120$ ms/image at batch sizes 1/8/32. Relative savings increase from 24.74% to $26.94/28.29\%$, yet the absolute savings become insufficient. Batch 32 improves router time an additional 5.24% over batch 8 without changing the utility ordering. Figure 1 quantifies the residual target: removing observation/head work closes only $2.83/2.32\%$ of the batch-8/32 deficit. Grouping, dispatch, selected inference and assembly remain in this intervention. Strictly positive surplus requires exceeding the stated reduction to parity.

Paired-repeat surplus ranges at batches 1/8/32 are $[1.069, 1.867]$, $[-6.010, -5.975]$ and $[-5.905, -5.858]$ in 10^{-3} CE-equivalent units. These are observed min/max ranges, not confidence intervals.

T4: the relevant comparator is native inference. On T4, native instead minimizes static loss plus priced runtime in every repeat. Router-minus-multiscale surplus is positive, but router-minus-native surplus is $-0.052776/-0.062160$ at batches 1/8. Observation/head removal saves $1.012/0.772$ ms/image, leaving $64.958/76.928$ ms/image to parity. Every T4 method takes longer per image at batch 8, so the effect is not specific to the selection path. The available aggregate measurements do not identify its cause or support generalizing the slowdown to T4 hardware. The intervention quantifies a residual budget without showing which retained component can supply the required reduction.

Fixed-policy price regions. Table A14 intersects the three router-versus-static affine inequalities using unrounded, stack-specific mean records. At both matched batch sizes, 1 and 8, the T4/A100 intervals are disjoint. Thus no single price yields positive mean surplus for this transferred policy on both recorded environments at either of those batch sizes. Endpoints have zero surplus; these open intervals describe repricing the same frozen policy, not newly selected preferences or uncertainty regions. The primary comparison retains 0.8 CE/second. Repeat ranges hold the cohort fixed and describe observed execution variability within each session.

Table A14: **Prices with strictly positive mean router surplus.** Intervals are in CE/second; policies and recorded outputs/times remain fixed. The historical RTX 3090 rows retain their original quality records, whereas T4/A100 use fresh diagnostic quality.

Stack	Batch 1	Batch 8	Batch 32
RTX 3090, Windows	$(0.649, 1.126)$	$(1.009, 2.225)$	$(1.006, 2.383)$
RTX 3090, WSL	$(0.664, 1.143)$	$(0.899, 2.053)$	$(0.908, 2.130)$
T4, Colab	$(0.166, 0.403)$	$(0.149, 0.370)$	—
A100, Colab	$(0.727, 1.456)$	$(1.401, 2.967)$	$(1.382, 3.200)$

A shared interior price range. The interval endpoints in Table A14 imply positive batch-1 and negative batch-8/32 mean surplus for both Windows/RTX 3090 and A100 throughout $[0.75, 0.95]$ CE/second. This range was derived descriptively from the fixed-policy regions; it does not replace the historical price. The accompanying `tools/check_published_summaries.py` verifies the signs from Tables 1 and A13, allowing half a displayed last digit for each CE and runtime entry. All pairwise contrasts at both endpoints are positive at batch 1; the multiscale contrast is negative at both endpoints for batches 8/32. Affineness extends these signs over the entire range. This checks arithmetic and display-rounding robustness, not sampling uncertainty or the raw measurement pipeline.

E Runtime budgets and the cost of changing comparators

Observed means and derived targets. Table A15 applies the break-even execution condition, Eq. (2), to existing measurements, holding outputs and comparator time fixed. These are performance targets, not achieved optimizations or isolated head-overhead measurements.

Table A15: Runtime saving against multiscale required at 0.8 CE/second. ADE rows use historical Windows/RTX 3090 records. All times are ms/image. Additional reduction is the amount needed for mean parity; percentages divide that reduction by complete router time $T_{h,b}$.

Setting	CE penalty	Required saving	Observed saving	Additional reduction
ADE batch 1	0.014222	17.778	21.914	Already above parity
ADE batch 8	0.014222	17.778	14.101	3.677 (10.0%)
ADE batch 32	0.014222	17.778	14.131	3.647 (10.5%)
ACDC batch 8	0.003721	4.651	4.174	0.477 (0.73%)

An exact decomposition of the Windows/RTX 3090 surplus change. The optional proxy chooses flip as best static; executed costs choose multiscale at all three tested batches. Equation (A7)

separates repricing against the original comparator from the gain of the new comparator. Table A16 uses fixed outputs, the same workload and the same 0.8 price.

The change cannot be assigned to isolated router overhead: at batch size 8, repricing the router and flip improves their contrast by 0.011611, but replacing flip by multiscale removes 0.020078. The identity reconstructs the archived means to within 9.1×10^{-17}; it is an algebraic attribution to comparator terms, not a causal component ablation.

Table A16: Windows/RTX 3090 surplus transfer in 10^{-3} CE units. Proxy surplus plus pairwise repricing minus the comparator-switch term equals executed surplus. Positive repricing improves the router–flip comparison; multiscale is then a stronger static. Identities use unrounded records, not sums of rounded display entries.

Batch	Proxy surplus	Pairwise repricing	Comparator switch	Executed surplus
1	5.525	2.971	5.186	3.309
8	5.525	11.611	20.078	−2.942
32	5.525	12.285	20.727	−2.918

Interpreting router–mixture time. The validation-quality mixture fixes $p_{\mathrm{Q}} = 0.340619$ of images to multiscale and the rest to flip, assigned by whole batches. Source refitting and transfer do not preserve its validation-quality match. The ADE router's final action counts are 133 native, 41 flip and 226 multiscale (56.5%). Mixture multiscale shares are 34.0/34.0/34.4% at batch sizes 1/8/32, and its CE is 0.80683/0.81123/0.80899, versus router 0.79594. Router time is greater by 6.841/3.250/2.414 ms/image, while its CE is lower by 0.010886/0.015289/0.013047. Action allocation, selected images, subbatches and the feature/head path all change; the contrast does not isolate any one component.

Appendix F holds selected actions/images and subbatches fixed, compares live and supplied decisions, and verifies the change in uninstrumented complete execution. This intervention isolates the removable observation/head path of the tested implementation, while dispatch remains. The measured mixtures and Linux replay appear in Appendix D.

F Routing, cohort and calibration sensitivity

These analyses address scalar routing, cohort composition, dispatch costs and the calibration-menu boundary. They are retrospective extensions; none changes the original policy locks, prospective selections or primary intervals.

F.1 Scalar routing and measured CPU head costs

Native decoder entropy is computed without labels. We form 21 source-fold (0–2) quantiles, enumerate ordered pairs in both monotone directions, and add three statics (465 candidates). At each of the same 61 prices in $[0, 1.5]$, fold 3 selects a candidate by actual proxy utility; fold 4 supplies the reported curve. Positive/signed oracle-area recovery is 5.73/3.15%, compared with 5.04/2.31% for PCA64 ridge. The paired positive-area difference is $+0.696\,[-19.097, 7.389]$ points under the same 1,999 composition draws. The comparison does not establish a multivariate-head advantage. It complements, rather than duplicates, the historical annotation-free scalar-decile OOF control. At price 0.8 the entropy selector uses $a(x) = 2 - \mathbf{1}\{e(x) > 0.2402484\} - \mathbf{1}\{e(x) > 1.1694953\}$, giving native/flip/multiscale counts 1/261/138, CE 0.817085 and proxy surplus -0.012516. Thus similar integrated recovery can coexist with a worse fixed operating point. A matched batch-8 replay executes entropy, original ridge and multiscale on 400 images over three randomized repeats. Complete times are 34.580/36.653/50.829 ms/image and CE values 0.817085/0.795944/0.781721, respectively. At 0.8, their contrasts against multiscale are $-0.022364/-0.002881/0$; Table 3 reports the paired entropy–ridge contrast. Every timed action matches its frozen record; a 64-image check gives exact hard-output agreement with supplied actions. CE is attached from the verified cache, not newly scored historical logits.

Loading the eight archived CE heads, we separately time source scaling, PCA where applicable, feature assembly, prediction and inverse target scaling and the shared cost head/action argmax on CPU-resident observations (two threads, five warmups, 50 calls per batch size). All predictions agree with the archived values within 1.2×10^{-16}. At batch 8, PCA64 ridge/MLP/boosting/ExtraTrees medians are 0.066/0.067/0.362/2.001 ms/image; full-feature counterparts are 0.055/0.060/0.252/1.999. These exclude GPU feature computation, transfers and complete TTA execution. ExtraTrees' cost is substantial relative to a 3.68 ms budget, but subtracting these standalone measurements from an unrelated complete run would not establish the new system's surplus.

F.2 Metric selection at realized costs

Figure A3 plots pooled mIoU against realized optional cost for all nine frozen policies in Table A7. It involves no test-set reselection or complete head timing. Six policies exceed the transported validation cap: equal validation constraints do not ensure equal test compute. Select dataset mIoU when it is the deployment objective, and check cost compliance separately.

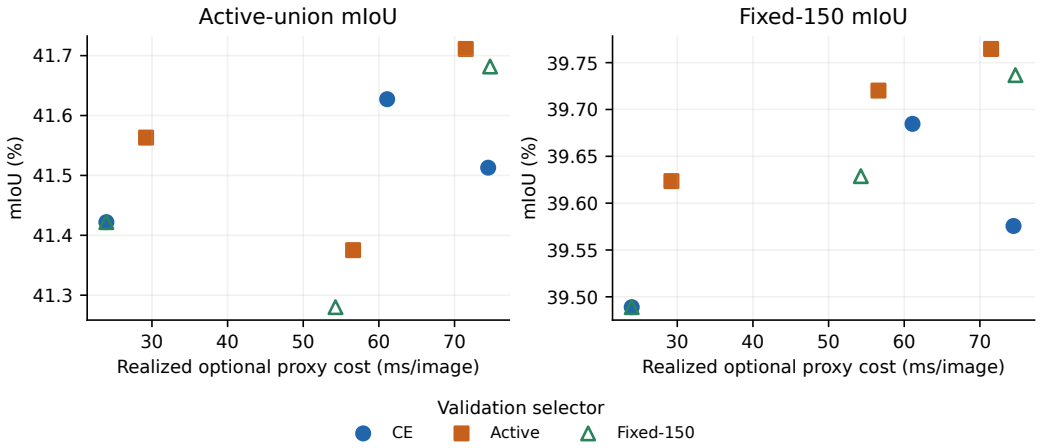

Figure A3: Frozen metric selectors at realized test proxy costs. Each criterion contributes three validation-cap policies; coincident points remain. Costs differ across criteria, so these are not matched-budget comparisons. Lower cost and higher mIoU are preferred.

F.3 Cohort and matched-action execution checks

The batch-8 cohort replay fits scaler/PCA and ridge gain/log-cost heads on 1,600 images and evaluates the excluded 400, rotating over all five ADE folds with $\alpha = 100$ and $\lambda = 0.8$. Fold 4 reproduces all 400 original decisions; the other fits use fixed seeds without price or alpha selection. These settings are inherited from the inspected study. Three randomized full-workload repeats compare live routing with multiscale, attaching historical CE after verifying every live action. The replay measures complete time on all 2,000 images, without claiming fresh probability parity with the cached losses.

Table 3 in the main text reports all five folds and their pooled mean, together with the matched execution controls. The separate historical nested-selection heads used for the support analysis (Appendix C.4) give pooled router/multiscale CE 0.816394/0.789743 and required saving 33.313 ms, with fold values 17.778–48.106 ms. These differ from the fixed-parameter replay; neither makes the original fold's 17.778 ms requirement universal.

Fixed-action intervention and component profiling. On fold 4, supplied frozen actions remove hidden-summary/entropy computation, observation transfers and CPU head evaluation. Native inference, image order, optional-view sub-batches, dispatch and original-grid assembly remain fixed. Three uninstrumented repeats give live/supplied-action/multiscale times of 36.349/35.837/50.491

ms/image. Savings average 0.512 ms (repeat differences 0.357–0.762). All timed actions agree; 64-image hard outputs agree exactly. Same-session parity requires 32.713 ms/image, leaving 3.124 ms after the intervention. This measures a removable path in the tested implementation, not a universal runtime lower bound.

Synchronized stage profiling on the first 64 fold-4 images gives, in ms/image: input preparation 1.281, native forward 9.945, observation/transfer 0.256, CPU head 0.101, flip dispatch/forward 7.401, scale dispatch/forwards 14.993, and output assembly 4.685. Synchronization is included; optional-view stages combine dispatch and forward work, not isolated fragmentation or occupancy. The uninstrumented full-cohort intervention supplies the realized saving.

F.4 ACDC RECORDING COMPOSITION AND CALIBRATION BOUNDARY

The exact existing block timing permits separate summaries for 340 images in 11 overlapping inferred recording families and 66 images in three disjoint families. Complete-minus-proxy utility is $+0.001747$ and $+0.000922$, respectively; complete-minus-multiscale is -0.000444 and -0.000060. The corresponding complete/proxy multiscale shares are 87.35/55.59% and 98.48/92.42%. Complete routing saves 4.915 ms/image against multiscale in the overlap subset and 0.353 ms/image in the disjoint subset. The latter contains 50 night, 13 rain and three snow images, with no fog: recording overlap and condition composition cannot be separated by this contrast. These three families do not supply an independent population validation.

The original multiplier grid was fixed before outcomes; it was not designed to enforce a minimum adaptive fraction. As $s \to 0$, the rule approaches $\arg\max_a \widehat{B}_a(x)$, which need not select multiscale. On the 461 calibration images, $s = 0$ gives 6/5/450 native/flip/multiscale decisions, versus 18/6/437 at $s = 0.25$. Finite-cohort allocations change in steps. All 2,305 cached decisions on the old grid reproduce its live calibration. A separate replay measures $s = 0, 0.0625, 0.125, 0.25$ and three statics on 461 calibration images, retaining the frozen router, contract, batch 8 and 40-image microblocks. Three repeats randomize method order within recording blocks (84 cells; 9,681 image executions). Fresh-output CE, confusions and actions match across repeats; source/lock hashes are unchanged. Table A17 reports the seven calibration candidates.

Table A17: Retrospective ACDC calibration-only boundary check: seven candidates, fresh outputs/times, one session at 0.8. Original multipliers 0.5/1/2/4 are absent; this is not full augmented-menu optimization. mIoU pools 19 classes.

Candidate	CE	ms/image	mIoU (%)	Utility
Native	0.997206	21.176	47.832	-1.014147
Flip	0.926485	35.978	48.613	-0.955268
Multiscale	0.848791	69.986	49.391	-0.904780
Router, $s = 0$	0.849292	68.679	49.361	-0.904235
Router, $s = 0.0625$	0.849413	68.653	49.364	-0.904335
Router, $s = 0.125$	0.849443	68.048	49.364	-0.903882
Router, $s = 0.25$	0.849679	67.634	49.357	-0.903786

Among these seven candidates, $s = 0.25$ has highest observed calibration utility, exceeding $s = 0.125$ by only 0.000096 and static multiscale by 0.000994. The margin over $s = 0.125$ changes sign across repeats ($+0.000002, -0.000274, +0.000560$), so the optimum is not precisely located. Adding these lower multipliers and statics does not improve the calibration mean; this check neither reselects the prospective policy nor establishes held-out superiority.

Measured output-contract work. On the first 40 calibration images (one rain block), three synchronized batch-8 multiscale profiles measure 68.304 ms/image: forwards 56.406, original-grid assembly 6.150, preprocessing/dispatch residual 5.748. Assembly comprises interpolation/softmax 3.773 and fusion/division/argmax 2.377 ms (9.00%). Uninstrumented runtime is 68.013 ms/image, with identical probabilities, labels and actions. Outputs occupy 157.594 MB of FP32 probabilities and 16.589 MB of int64 labels/image. This suffix measurement is not an achieved speedup.