LEARNING TO BE WRONG: TASK-SUFFICIENT INEXACT COMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural efficiency methods usually optimize how much computation to spend. We study how much *local exactness* is worth paying for. A cheaper realization may violate a conventional local contract while preserving downstream utility. We evaluate each candidate with measured runtime against a matched exact reference, a mechanism-specific local-fidelity diagnostic, and signed downstream task regret. Across precision-relaxed, incomplete, and contract-relaxed realizations, nominal work, local fidelity, hardware cost, and task utility frequently disagree. Our main case study groups several target positions in ESM-2 masked-marginal scoring so that one forward evaluates multiple mutations under a changed conditioning context. At $g = 4$, the same frozen random-group procedure yields 3.93–3.99$\times$ measured scoring speedup with mean signed DMS regret 0.0010–0.0012 on TEM-1, influenza HA, and a DHFR assay held out from policy development; at $g = 16$, speedup reaches 12.34–15.59$\times$ with assay-dependent regret. Pairwise interference is predictable in development, while held-out DHFR shows that a fidelity-aware grouping rule can improve exact-score agreement and still worsen task regret. These results make task regret the governing quantity for inexact execution: spend exactness where it changes the task.

1 INTRODUCTION

Most efficient-learning methods ask *how much computation* a model should spend. We ask a complementary question: *how much local correctness is worth paying for?* An intermediate computation can depart from its conventional exact contract while the final task remains essentially unchanged. A matrix product may be incomplete, a selection may stop early, or a scorer may use a different conditioning context. When a cheaper realization preserves the downstream quantity that matters, exact local agreement carries avoidable cost. We call this local departure *wrongness* and treat exactness as a task-dependent computational resource.

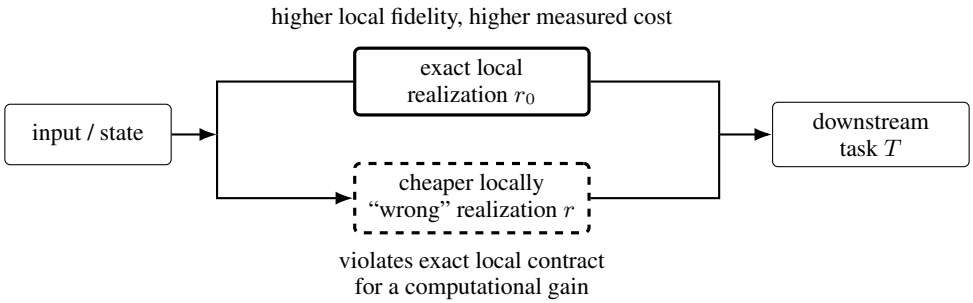
Evaluating this decision requires three separations. Realized runtime can differ from nominal work after data movement, materialization, and kernel effects. Local discrepancies have site-dependent consequences. Individually tolerated changes can interact under composition. Fidelity proxies can also diverge from downstream utility. *Learning to Be Wrong* therefore asks where a cheaper local realization remains sufficient for the final task.

The protein-scoring case in this paper makes the idea concrete. Exact masked-marginal scoring evaluates one target position per context. If several positions are masked together, one ESM-2 forward can score a group of targets, while every target is conditioned on a context that differs from the exact single-mask contract. The grouped realization is locally inexact and computationally cheaper. Its value is determined by the downstream variant-ranking task and the realized hardware cost.

Quality-constrained approximation spans programming systems and neural optimization (Baek & Chilimbi, 2010; Sharif et al., 2019; 2021; Zhao et al., 2023). We study a realization-level empirical object for neural computation:

$$\big(\text{local contract},\ \text{measured cost},\ \text{local fidelity},\ \text{downstream task regret}\big).$$

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.



$$R_T(r) \;=\; U_T(r_0) \;-\; U_T(r), \qquad \min_{r \in \mathcal{R}} C(r) \text{ s.t. } R_T(r) \;\leq\; \epsilon$$

Figure 1: **Learning to Be Wrong as a task-level decision.** A cheaper realization may be locally wrong relative to the exact computational contract yet remain task-sufficient when its downstream regret stays within a task-specific budget. The objective is lower cost, not error; local wrongness is tolerated only when the task does not justify paying for additional exactness.

Heterogeneous forms of local wrongness—matrix error, set overlap, or a changed semantic context—lack a common additive unit. We therefore retain mechanism-specific diagnostics and compare realizations through the downstream task quantity they eventually affect.

Let r_0 be the exact realization and r a cheaper candidate for downstream task T. With utility oriented so that larger is better, we define signed task regret

$$R_T(r) = U_T(r_0) - U_T(r).$$

For a declared task-specific budget ϵ, the operating point is the least costly measured realization satisfying $R_T(r) \leq \epsilon$. This separates local wrongness, downstream regret, and failure. Failure means exceeding the declared task budget.

We test this perspective through an evidence chain designed to expose failure modes as well as useful operating points. Conventional TF32/BF16 controls verify that the measurement stack recovers hardware-supported quality–speed gains. Incomplete realizations—partial GEMM, sampled normalization, and early-stop Top-K—test whether nominal work reduction survives native implementations. Single-site interventions measure how task sensitivity changes with location. Grouped masked-context scoring then provides the main contract-relaxed case, with an explicit semantic change, computational gain, and downstream task metric.

Three findings recur. **First**, local wrongness is a poor proxy for task harm: the same ESM-2 FFN realization has markedly different effects across sites, and single-site tolerance breaks under composition. **Second**, useful relaxation is a measured systems property: several nominal shortcuts disappear against native implementations, whereas grouped protein scoring yields real multi-fold speedups. **Third**, useful wrongness is task-dependent. Across TEM-1, influenza HA, and a frozen held-out DHFR assay, random grouped scoring traces a measured cost–task-regret frontier. At $g = 4$, measured scoring speedup is 3.93–3.99$\times$ while mean signed DMS regret remains 0.0010–0.0012 across the three assays. More aggressive grouping reaches 12–16$\times$ with assay-dependent regret.

These results motivate the learning problem: predict where relaxation remains task-sufficient. We learn pairwise interference on TEM-1 and evaluate structure-aware grouping during development on HA. The predictor contains measurable signal. On frozen held-out DHFR, the same structural bias improves exact-score fidelity while producing worse DMS regret than random grouping. The reversal isolates the harder target for allocation policies: downstream task regret.

Our empirical contributions are:

1. We measure precision-relaxed, incomplete, and contract-relaxed realizations under a common realization-level record and show that nominal work, local fidelity, realized hardware cost, and downstream utility can rank the same candidate differently.

2. We establish strong context dependence in correctness slack: the same ESM-2 FFN relaxation has different task consequences by site, and individually tolerated BERT interventions lose sufficiency under composition.
3. We obtain multi-fold measured cost–regret frontiers from grouped masked-context scoring on three protein assays, together with a frozen held-out reversal where improved local fidelity fails to improve downstream DMS regret.

The experiments cover three realization classes, two neural workloads, three protein assays under one frozen ESM-2 scorer, and one held-out test of the structure-aware grouping rule. Within this scope, the evidence supports a concrete systems principle: local exactness should be purchased according to its measured effect on the downstream task.

2 Related work

Approximate and quality-constrained computing. Controlled approximation under application-level quality constraints is well established. Green exposes programmer-controlled approximation under a quality-of-service target (Baek & Chilimbi, 2010); ApproxHPVM provides an accuracy-aware compiler representation (Sharif et al., 2019); ApproxTuner searches approximation choices under end-to-end quality requirements (Sharif et al., 2021); and ApproxCaliper incorporates application-specific objectives beyond conventional network accuracy (Zhao et al., 2023). Layer-wise approximate hardware assignment is likewise established, including ALWANN (Mrazek et al., 2019). These systems establish both quality-constrained approximation and nonuniform fidelity allocation.

We study the empirical boundary between a *local correctness contract* and the downstream task. Candidate realizations may be numerically approximate, incomplete, or semantically different from exact execution. Their local discrepancies have incompatible units, so we retain a realization-specific diagnostic and compare candidates through measured hardware cost and a common downstream regret quantity. The resulting experiments measure how local contract violation, realized cost, and downstream utility decouple across neural realization classes, including a semantic contract relaxation evaluated on a frozen held-out assay.

Adaptive computation and learned allocation. Adaptive approximation and routing already allocate inexact computation nonuniformly across neural-network layers; AutoApprox, for example, learns per-layer routing over a bank of approximate systolic arrays (Jain et al., 2022). Our development-time grouping predictor tests whether interference created by a contract relaxation has predictable structure. The held-out DHFR result then separates predictable local interference from transferable downstream task-regret improvement. For selection, RTop-K shows how GPU implementation details and early stopping affect row-wise Top-K (Xie et al., 2025); our Top-K experiments use that distinction to separate implementation gains from approximation-attributable gains.

Protein language models and variant scoring. ESM-2 supplies the frozen protein language model in our main case study (Lin et al., 2023), and ProteinGym supplies standardized deep-mutational-scanning benchmarks (Notin et al., 2023). Efficient mutation predictors already exist: Mutate Everything predicts stability effects for all single and double mutations in one forward pass using a trained model (Ouyang-Zhang et al., 2023). Our grouped masked-context scorer instead modifies the conditioning contract of a fixed masked-marginal ESM-2 scorer so that one forward evaluates multiple target positions. Recent work directly benchmarks reduced-precision ESM-2 variant-effect scoring on ProteinGym (Shao, 2026). Together, this prior work places the protein case study as a measured semantic-contract relaxation with a real compute benefit and a downstream scientific task.

3 From local wrongness to task sufficiency

What "wrong" means. Let r_0 denote the conventional exact realization of a local computation and let $r \in \mathcal{R}$ be an alternative realization. We call r locally *wrong* only in an operational sense: it does not reproduce the exact local contract of r_0. The deviation may be numerical, as in lower

precision; procedural, as in incomplete evaluation; or semantic, as in changing the conditioning context of a masked-marginal scorer. Wrongness is therefore mechanism-specific and is not itself something to maximize.

For each realization we record three measured quantities. First, $C(r)$ is execution cost on the target hardware; throughout the main experiments this is synchronized latency, with speedup $S(r) = C(r_0)/C(r)$. Second, $D(r, r_0)$ is a local diagnostic appropriate to the mechanism, such as relative Frobenius error, exact-set overlap, or score agreement under a changed context. Third, $U_T(r)$ is utility on the complete downstream task, oriented so that larger values are better.

Task regret. We define signed task regret relative to exact execution:

$$R_T(r) = U_T(r_0) - U_T(r). \tag{1}$$

The sign is retained. A cheaper realization can score slightly better on a finite benchmark; negative measured regret does not imply that its local error is intrinsically beneficial. It only says that the measured downstream utility did not decrease on that benchmark.

For a task-specific tolerance ϵ, a realization is *task-sufficient* when $R_T(r) \leq \epsilon$. The measured operating decision is

$$r^*(\epsilon) = \arg\min_{r \in \mathcal{R}} C(r) \quad \text{subject to} \quad R_T(r) \leq \epsilon. \tag{2}$$

Equation 2 defines the empirical decision used throughout the audit. Exactness is worth paying for only when relaxing it changes the task beyond the declared budget.

Local wrongness is not a global error budget. Equation 2 deliberately does not sum local diagnostics. Relative matrix error, Top-K overlap, and a changed conditioning context have no common physical unit. Even within one mechanism, site-wise tolerance need not compose. We therefore keep $D(r, r_0)$ as a diagnostic and use $R_T(r)$ as the common downstream currency. This separates three notions: *local wrongness* describes deviation from the exact local contract; *task regret* measures the downstream consequence; and *failure* is only the decision that a realization exceeds a stated budget. The frozen PASS/FAIL gates used during development serve to prevent post-hoc threshold selection; they are not universal standards.

Three realization classes. We study three conventional mechanism classes. *Precision-relaxed* realizations preserve the nominal operation while changing arithmetic fidelity, such as TF32 or BF16; these are positive controls. *Incomplete* realizations execute only part of an exact procedure, such as retained-inner-dimension GEMM, sampled normalization, or early-stop Top-K. *Contract-relaxed* realizations change a local semantic precondition for a computational reason. Grouped masked-context scoring is the main example: each target is no longer conditioned on an otherwise singly masked sequence, but one expensive forward supplies several target scores.

The learning problem. The strongest form of *Learning to Be Wrong* would choose a realization from context rather than use one global setting. Conceptually, a policy π could map an input, site, local state, and regret budget to a realization,

$$r_i = \pi(x, i, z_i, \epsilon).$$

We test four prerequisites for that policy: whether correctness slack varies by site, whether its structure is predictable, whether candidate policies survive measured systems accounting, and whether a development-time fidelity signal transfers to held-out task regret. Section 6 shows that the last step is the difficult one.

Audit rule. Every claimed operating point must therefore make the following tuple auditable:

$$\big(\text{contract}(r),\ C(r) \text{ versus matched exact } C(r_0),\ D(r, r_0),\ R_T(r)\big).$$

A nominal-work reduction without measured savings, a local diagnostic without downstream utility, or a learned proxy without held-out task evidence is insufficient. Rejected realizations remain informative because they identify which link in this chain fails.

Table 1: Realization taxonomy and audit outcome. A realization is retained in the main argument only when its local contract, measured implementation cost, and downstream result can be stated together.

Family	Exact contract → cheaper realization	Audited outcome and role
Precision-relaxed	Same nominal operation; TF32/BF16 arithmetic	Hardware-supported speedups with negligible task change; positive control.
Incomplete	Full GEMM/reduction/selection; partial, sampled, or early-stopped realization	Outcome depends on shape, site, and implementation; several candidates are rejected.
Contract-relaxed	One masked target per context; group several masked targets per forward	Measured scoring-speedup–DMS-regret frontiers on three assays; main case study.

4 Empirical audit protocol

The experiments are organized around four questions that correspond directly to the framework. **Q1: Is the candidate actually cheaper?** We measure synchronized runtime against a matched exact reference rather than infer speed from FLOPs. **Q2: How is it locally wrong?** Each realization keeps its own diagnostic rather than being forced into a common error unit. **Q3: Does the downstream task care?** We evaluate signed task regret under the complete task. **Q4: Does the conclusion survive composition or held-out evaluation?** Development gates, composition tests, and the final frozen protein assay are used to prevent a locally attractive point from being promoted without stronger evidence.

All core systems measurements use two NVIDIA A100-SXM4-80GB GPUs under a fixed PyTorch/CUDA environment. We retain timing samples and compare exact and candidate paths under matched runtime protocols. Where an optimized operator baseline is specifically claimed, such as BERT $\mathrm{F.linear}$, it is identified explicitly. For the protein frontiers, plotted speedup is the lower of the two lane-wise median speedups; every plotted grouped-scoring candidate has candidate p95 below exact p05 on both lanes. The two GPU lanes therefore assess timing reproducibility, not biological uncertainty.

For incomplete operators we separately examine nominal work, kernel latency, and relevant materialization or routing overhead. For the protein case study, the reported speedup is the measured masked-marginal scoring speedup under the stated grouping policy; it is not extrapolated from the number of forward passes. Protein task utility is

$$U_T(r) = \rho_{\mathrm{S}}(s_r, y_{\mathrm{DMS}}),$$

the assay-level Spearman correlation between model mutation scores and ProteinGym `DMS_score`, whose direction is normalized so that larger values indicate higher mutant fitness (Notin et al., 2023). Exact single-mask scoring is the computational reference; it is not biological ground truth.

During development we used frozen task gates to prevent post-hoc threshold selection. The final paper reports continuous cost–task-regret frontiers instead of treating those gates as universal standards. For random grouped scoring, every $g > 1$ uses three deterministic partitions frozen before downstream comparison. Figure 3 reports their mean signed regret and a one-sided bar to the worst observed partition. These bars are retained deterministic-policy ranges, not confidence intervals or biological replicate uncertainty. Full gate definitions, timing procedures, and negative admissions are reported in the appendix.

As a conventional positive control, hardware-supported precision changes validate the measurement stack. On BERT/SST-2, TF32 and full BF16 preserve all 872 predictions while yielding model speedups of 2.64–2.69× and 2.74–2.77×, respectively. On ViT/CIFAR-100, full BF16 changes 12 of 10,000 predictions while yielding 8.13–8.19× speedup. These controls are not contributions; they establish that the audit can recover familiar quality–speed gains when they are genuinely present.

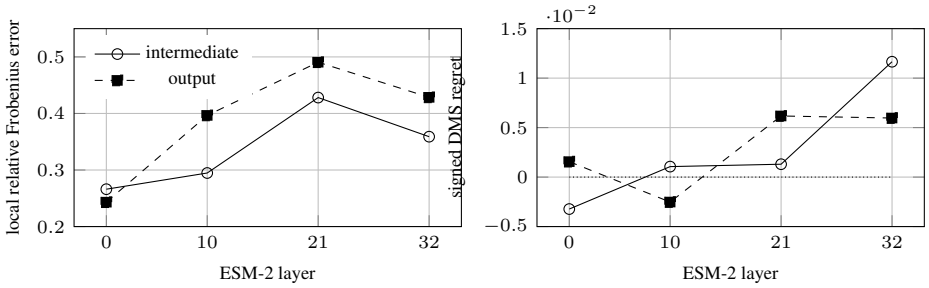


Figure 2: **The same local relaxation has different downstream value at different sites.** R13 applies the same $\rho = 0.75$ partial-FFN realization at four ESM-2 depths. Left: local relative Frobenius error. Right: signed DMS regret relative to exact scoring. Local distortion alone does not determine the task quantity.

5 Empirical results: when local wrongness is useful

5.1 Measured cost can reverse nominal savings

The first requirement is systems reality. On BERT, retained-inner-dimension GEMM exposes clear local cost–distortion tradeoffs in isolated arithmetic, but the tested drop-in partial implementations do not beat optimized native $\mathrm{F.linear}$ at the model-relevant shape. At larger ESM-2 FFN shapes, however, the same family becomes viable: at the primary DMS-like regime ($M = 2304$), partial `addmm` with $\rho = 0.75$ yields 1.11–1.12$\times$ speedup for the FFN intermediate projection and 1.16$\times$ for the output projection across the two A100s. These are operator-level measurements. Together with the rejected LayerNorm and high-fidelity Top-K routes in the appendix, they show that nominal work reduction alone does not establish a useful system point.

5.2 Task sensitivity varies across sites

We next apply the same $\rho = 0.75$ ESM-2 FFN realization at different sites while scoring TEM-1. The exact BF16 scorer has DMS Spearman 0.730850. Across the eight probed sites, local relative Frobenius error ranges from 0.243 to 0.491, while signed DMS regret ranges from -0.00323 to 0.0117. For example, the output projection at layer 10 has local relative error 0.397 and score agreement 0.9981 with exact, yet no positive measured DMS regret; the output projection at layer 21 has higher local error 0.491 and DMS regret 0.00617. Figure 2 shows all eight conditions. The point is not a monotone depth law: it is that the downstream task, not the local diagnostic alone, determines whether a relaxation is acceptable.

Single-site tolerance also fails to compose automatically. A BERT allocation assembled from individually tolerated sites reduces SST-2 validation accuracy by 2.29 percentage points; in the calibration-ranked sparse composition, only the single-site $k = 1$ intervention passes the frozen strict task-quality gate, while $k \geq 2$ does not. Thus, local correctness slack is site-dependent and non-additive in this testbed.

5.3 A semantic contract relaxation yields a real cost–regret frontier

The main case study changes not just arithmetic fidelity but the local scoring contract. Exact masked-marginal scoring follows the zero-shot log-odds construction of Meier et al. (2021): for a single substitution, one target position p is masked at a time. Let $x^{(p)}$ denote the wild-type sequence with position p masked and let a be a candidate amino acid. The exact-reference mutation score is

$$s_p(a) = \log P_\theta(a \mid x^{(p)}) - \log P_\theta(x_p^{\mathrm{WT}} \mid x^{(p)}).$$

A grouped realization chooses a set G, masks every position in G, and evaluates the same log-probability difference for each $p \in G$ under $x^{(G)}$. One ESM-2 forward therefore scores several targets. The candidate is cheaper because it requires fewer forwards, but each target is locally wrong relative to the single-mask conditioning contract. Throughout this section, "exact" denotes exact execution of the single-mask computational reference.

Table 2: Protein assays and the WT-marginal transparency comparator used in this study. The final column gives speedup / signed DMS regret for a one-pass comparator; it is not a proposed realization.

Assay	Length	Variants	Positions	Exact DMS ρ	WT-marginal $\times / R$
TEM-1 Stiffler	286	4,996	263	0.731	62.8 / 0.00891
Influenza HA Doud	565	10,715	564	0.507	375.6 / 0.01784
E. coli DHFR +Lon	159	2,363	157	0.480	25.6 / 0.00082

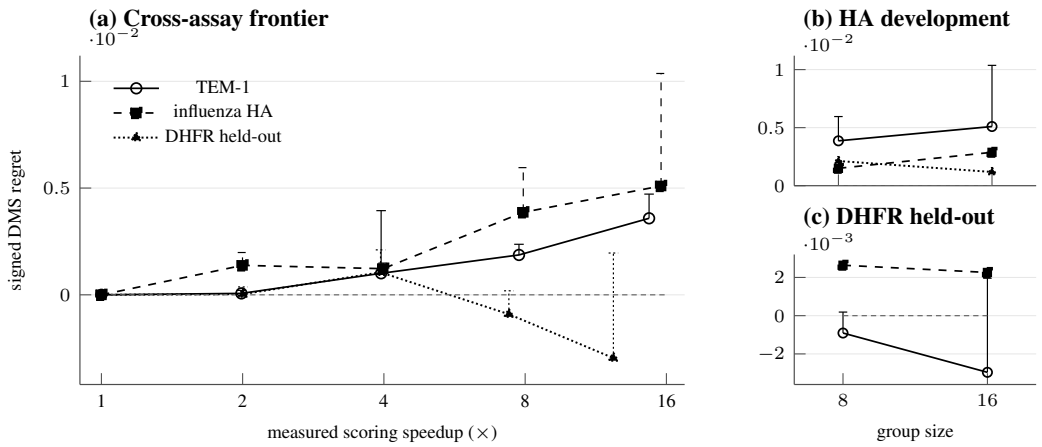


Figure 3: **Measured protein cost–regret frontiers and a held-out structure test. (a)** Exact single-mask scoring ($g = 1$) and random grouped-context realizations at $g \in \{2, 4, 8, 16\}$ across three assays (solid circles: TEM-1; dashed squares: influenza HA; dotted triangles: DHFR). Points average three frozen deterministic partitions; one-sided bars extend to the worst observed partition and are not confidence intervals. Speedup is the lower of two lane-wise median A100 measurements. **(b)–(c)** At matched group size, solid circles denote random grouping, dashed squares denote distance-aware grouping, and the dotted triangles in panel (b) denote the learned HA policy. Structure-aware grouping lowers regret on development HA, but the distance heuristic does not transfer as a task-regret improvement to held-out DHFR. Panels (b) and (c) use independent y-scales.

We evaluate the same frozen random-group procedure on TEM-1, influenza HA, and finally E. coli DHFR, held out from grouping-policy development. Table 2 summarizes the assays and includes WT-marginal scoring as a one-pass transparency comparator. WT-marginal scoring is much faster, with signed DMS regret 0.00891 on TEM-1, 0.01784 on HA, and 0.00082 on DHFR.

Figure 3 is the central empirical result. At $g = 4$, random grouped scoring reaches 3.93–3.99× measured speedup across all three assays, with mean signed DMS regret between 0.0010 and 0.0012. Increasing g moves further along the compute frontier: at $g = 16$, measured speedup is 12.34–15.59×, while mean signed regret ranges from -0.00296 to 0.00510 depending on the assay. Negative finite-benchmark regret records the observed sign of the comparison against the exact computational reference; we attach no biological-benefit interpretation to it. The scientific object is the frontier itself, whose operating point is selected by the downstream regret budget.

6 Learning task-sufficient relaxation

A context-dependent allocation policy must estimate the downstream consequence of relaxing exactness. We test two prerequisites: whether grouping interference is predictable from local context and whether exploiting that signal improves downstream task regret at matched compute.

We use TEM-1 pairwise grouped-mask interference as development supervision. For each probed pair, the target is the 90th percentile of absolute substitution-score drift from exact single-mask scoring over variants at the two positions. A ridge predictor uses sequence-distance, position/residue,

and frozen ESM-2 hidden/logit features. With 2,048 deterministic pair probes, the best tested regularization reaches mean fold Spearman 0.415893 under five position-blocked development folds. The development split therefore contains predictable structure in local interference; held-out task regret is evaluated separately on DHFR.

On the HA development assay, the learned policy and a distance-aware grouping both improve exact-score fidelity relative to random grouping at matched $g = 8$. The learned $g = 16$ policy has signed DMS regret 0.00118 at an adjusted $10.4\times$ speedup. The adjusted runtime includes the test-protein feature forward and measured CPU planning overhead for the one-protein invocation; offline TEM-1 development is excluded from per-protein inference cost. These results show that structured grouping can preserve the exact scorer more closely at matched compute.

DHFR separates fidelity prediction from task utility. At $g = 8$, distance grouping improves local score agreement with exact (0.9962 versus random mean 0.9947) while signed DMS regret rises to 0.00264 versus -0.00091 for random grouping; it also exceeds the worst retained random partition (0.00019). At $g = 16$, distance grouping has regret 0.00226 versus -0.00296 for random grouping and exceeds the random worst case 0.00196. Panels (b)–(c) of Figure 3 show this development/held-out reversal.

The reversal identifies the target for a general allocation policy: downstream task regret under relaxation, together with realized systems cost. The ridge model establishes predictable local interference; DHFR shows that this signal alone fails to determine downstream utility. A stronger policy therefore needs a validated surrogate for task regret and evidence that the selected operating points transfer beyond the development task.

7 Discussion and limitations

Interpretation. *Learning to Be Wrong* treats local exactness as a task-dependent computational resource. A cheaper realization is useful when its measured downstream regret fits the application budget and its implementation delivers a real cost reduction. This decision depends jointly on the local contract, the hardware path, and the final task.

Empirical findings. Three patterns recur across the experiments. Local fidelity is an unreliable proxy for downstream importance: site dependence, composition failure, and the DHFR policy reversal all expose this gap. Correctness relaxation also depends on implementation quality: nominal arithmetic reduction, kernel latency, and end-to-end scoring speed can rank candidates differently. The resulting object is a empirical cost–task-regret frontier from which a task-specific budget selects an operating point.

Scope. The protein evidence covers three assays under one frozen ESM-2 scorer. Three deterministic partitions at each group size quantify sensitivity to the frozen grouping construction. DHFR provides one held-out assay for the structure-aware grouping rule. These experiments establish a cost–regret frontier for grouped masked-context scoring on the tested assays and a concrete failure of local-fidelity transfer on DHFR. Broader claims across ProteinGym, model families, or deployment settings require additional held-out tasks and models.

Open problem. The development-time predictor reveals learnable interference structure, while DHFR shows that fidelity-aware grouping can move local agreement and downstream utility in different directions. A general allocator therefore needs to predict task regret under composition and validate that prediction on tasks and models excluded from surrogate design. The same requirement applies beyond protein scoring: efficient-learning systems need separate measurements of local wrongness, realized cost, and downstream utility whenever those quantities can decouple.

8 Conclusion

In this work we treat exactness as a task-dependent computational resource. We evaluated cheaper local realizations by measured cost, mechanism-specific fidelity, and downstream task regret. Across precision controls, incomplete operators, site-sensitivity experiments, composition tests, and

grouped protein scoring, these quantities repeatedly diverge: nominal savings can disappear in implementation, the same local relaxation can have different task effects by site, and improved fidelity can fail to improve held-out task utility. Grouped masked-context scoring nevertheless produces a substantial measured cost–task-regret frontier across three protein assays. For a declared regret budget, the operating point is the cheapest measured realization whose downstream utility remains within budget. The next step is an allocation policy that predicts this task-regret boundary from context and transfers across tasks and models.

AI USE STATEMENT

Generative AI tools were used substantially for developing and refining the conceptual framework and hypotheses; designing and reviewing experimental methodology; drafting, debugging, and reviewing research code; literature triage; analysis planning; and manuscript drafting and engineering. They were not used to substitute for execution of the reported experiments or for the final verification of experimental outputs and cited sources. The authors reviewed all AI-assisted work, audited the reported quantities against retained artifacts, and take responsibility for the final manuscript, claims, citations, code, and supplementary material.

REPRODUCIBILITY STATEMENT

The supplementary research package records round-level specifications, frozen configurations, environment metadata, aggregate results, raw timing summaries, SHA-256 manifests, and compact original return archives. Main figures read CSV files included with the LaTeX source, and key main-text numerical claims are linked by source comments to retained machine-readable artifacts. The appendix documents the experimental protocol, negative results, implementation controls, statistical treatment, and provenance needed to interpret the main claims. No model checkpoint or dataset cache is redistributed; public model/dataset identifiers and frozen revisions are recorded in the corresponding round artifacts.

REFERENCES

Woongki Baek and Trishul M. Chilimbi. Green: A framework for supporting energy-conscious programming using controlled approximation. In *Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2010. doi: 10.1145/1806596.1806620.

Paras Jain, Safeen Huda, Martin Maas, Joseph E. Gonzalez, Ion Stoica, and Azalia Mirhoseini. Learning to design accurate deep learning accelerators with inaccurate multipliers. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 184–189. IEEE, 2022. doi: 10.23919/DATE54114.2022.9774607.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f51338d736f95dd42427296047067694-Abstract.html.

Vojtech Mrazek, Zdenek Vasicek, Lukas Sekanina, Muhammad Abdullah Hanif, and Muhammad Shafique. Alwann: Automatic layer-wise approximation of deep neural network accelerators without retraining. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, 2019. doi: 10.1109/ICCAD45719.2019.8942068.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko

Franceschi, Yarin Gal, and Debora S. Marks. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379, 2023. doi: 10.52202/075280-2810.

Jeffrey Ouyang-Zhang, Daniel J. Diaz, Adam R. Klivans, and Philipp Krähenbühl. Predicting a protein's stability under a million mutations. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3332.

Qing Shao. Benchmark averages hide the failures that matter: Quantizing esm-2 for protein variant-effect prediction. *bioRxiv*, 2026. doi: 10.64898/2026.08.10.744024. Preprint.

Hashim Sharif, Prakalp Srivastava, Muhammad Huzaifa, Maria Kotsifakou, Keyur Joshi, Yasmin Sarita, Nathan Zhao, Vikram S. Adve, Sasa Misailovic, and Sarita Adve. Approxhpvm: A portable compiler ir for accuracy-aware optimizations. *Proceedings of the ACM on Programming Languages*, 3(OOPSLA):Article 186, 2019. doi: 10.1145/3360612.

Hashim Sharif, Yifan Zhao, Maria Kotsifakou, Akash Kothari, Ben Schreiber, Elizabeth Wang, Yasmin Sarita, Nathan Zhao, Keyur Joshi, Vikram S. Adve, Sasa Misailovic, and Sarita Adve. Approxtuner: A compiler and runtime system for adaptive approximations. In *Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pp. 262–277, 2021. doi: 10.1145/3437801.3446108.

Xi Xie, Yuebo Luo, Hongwu Peng, and Caiwen Ding. Rtop-k: Ultra-fast row-wise top-k selection for neural network acceleration on gpus. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=PHg4rAXFVH.

Yifan Zhao, Hashim Sharif, Peter Pao-Huang, Vatsin Shah, Arun Narenthiran Sivakumar, Mateus Valverde Gasparino, Abdulrahman Mahmoud, Nathan Zhao, Sarita Adve, Girish Chowdhary, Sasa Misailovic, and Vikram Adve. Approxcaliper: A programmable framework for application-aware neural network optimization. In *Proceedings of Machine Learning and Systems*, volume 5, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/89efa87dc8f0a5d18e4ae0a479658f60-Abstract-mlsys2023.html.

A EXPERIMENTAL PROTOCOL AND MEASUREMENT DETAILS

This appendix records the experimental details that are intentionally compressed in the main text. The purpose is not to enlarge the claim set, but to expose the implementation, validation, negative admissions, and development decisions needed to interpret the reported cost–regret frontiers. Unless otherwise stated, all numbers below are copied from the retained machine-readable round artifacts described in Appendix K.

A.1 HARDWARE AND SOFTWARE ENVIRONMENT

All core systems experiments were executed on a shared GPU cluster using two NVIDIA A100-SXM4-80GB GPUs. The validated environment was kept fixed across the later rounds; the experiment packages did not replace the cluster PyTorch/CUDA/Triton stack during a round.

Table 3: Core execution environment used by the retained experiments.

Component	Recorded configuration
GPU	2× NVIDIA A100-SXM4-80GB, compute capability 8.0
Python	3.10.12
PyTorch	`2.3.0a0+40ec155e58.nv24.03`
CUDA	12.4
NCCL	2.20.5+cuda12.4
Triton	2.2.0
Protein model	`facebook/esm2_t33_650M_UR50D`, frozen BF16 inference
BERT control	`textattack/bert-base-uncased-SST-2`, frozen BF16 inference
Vision control	ViT CIFAR-100 checkpoint used in R11; 10,000 test images

A.2 TIMING CONVENTIONS

A speedup is always derived from measured latency, not from a nominal reduction in arithmetic. GPU sections use CUDA-event timing after warmup. Where a candidate and exact implementation are compared directly, the order is alternated across paired measurements to reduce drift. Raw timing samples are retained in the corresponding return archive. Later protein rounds run the two A100s as independent timing lanes and conservatively report the smaller lane-wise median speedup. Candidate p95 and exact p05 are retained as a distribution-separation diagnostic; the two lanes measure systems reproducibility and are not biological replicates.

Cost accounting depends on the realization. For the random grouped-context protein frontiers (R14/R15/R18), the primary reported scoring cost is GPU model-forward time for scoring all mutated positions. For the structure-aware development rounds, additional per-protein feature extraction and measured CPU grouping/planning overhead are reported separately and included in the adjusted speedup. Offline development on TEM-1 is not charged to the per-protein HA inference cost.

A.3 TASK UTILITY, SIGNED REGRET, AND FINITE-BENCHMARK INTERPRETATION

For the ProteinGym case study, the downstream utility is the assay-level Spearman correlation between the model mutation scores and the assay DMS scores,

$$U_T(r) = \rho_{\mathrm{S}}(s_r, y_{\mathrm{DMS}}), \tag{3}$$

with signed regret

$$R_T(r) = U_T(r_0) - U_T(r). \tag{4}$$

The regret is not clipped at zero. A negative value on a finite assay means only that the candidate obtained a slightly larger measured Spearman correlation than the exact reference on that assay; it is not interpreted as evidence that violating the local contract is biologically beneficial. Exact-score fidelity metrics, such as candidate-vs-exact Spearman and top-10% overlap, remain diagnostics rather than definitions of downstream success.

B DEVELOPMENT GATES AND INTERPRETATION BOUNDARIES

Several rounds used frozen binary gates to decide whether a branch of experimentation should continue. These gates were experiment-management devices chosen before inspecting the corresponding result. They are not standards and are not the primary result of the paper; the final presentation uses continuous measurements whenever possible.

Table 4: Representative frozen development gates. The final paper does not universalize these thresholds.

Round	Frozen gate	Role
R02/R03 BERT	accuracy degradation ≤ 0.5 pp and exact-prediction agreement $\geq 99\%$	decide whether single-site tolerance/site heterogeneity warranted a full layer sweep
R13 protein site	score Spearman vs exact ≥ 0.995, top-10% exact overlap ≥ 0.95, DMS-Spearman degradation ≤ 0.01	decide whether an individual ESM-2 FFN site was task-sufficient under the pilot criterion
R14/R15 grouped context	same three task criteria must hold for all three deterministic partitions; systems speedup $\geq 1.5\times$ on both A100s with timing-distribution separation	decide whether a grouped-context point was admitted during development
R16 learning signal	cross-validation Spearman ≥ 0.20 before calling the pair predictor a nontrivial learning signal	prevent a no-signal predictor from being promoted as an adaptive policy
R18 confirmation	validity of exact baseline and a frozen matched-cost comparison; no threshold change after seeing DHFR	final held-out interpretation and stop decision

The distinction between a local fidelity gate and a downstream regret budget is important. For example, random $g = 8$ grouped scoring on HA crossed the earlier exact-fidelity gate but still had a finite measured DMS-regret value. The final paper therefore does not label such a point intrinsically "failed"; it locates it on the empirical cost–regret frontier.

C RETAINED-INNER-DIMENSION GEMM: SINGLE-SITE SENSITIVITY AND COMPOSITION

C.1 BERT SINGLE-SITE MAP

The early BERT experiments use a structured prefix realization of the FFN intermediate projection. If the exact layer is

$$y = xW^\top + b, \tag{5}$$

then a retained-inner-dimension realization uses

$$y_R = x_{:R} W_{:R}^\top + b, \tag{6}$$

with the static weight prefix prepacked and no explicit activation `contiguous()` copy in the later BERT implementation. R02 first probed layers 0, 3, 7, and 11 on the 872-example SST-2 validation set; R03 then mapped all 12 layers. Table 5 records the most aggressive ratio that met the frozen strict single-site gate in R03.

For layers 3, 4, 6, 7, and 8, the strict task gate is first met at $\rho = 0.875$. The associated R01c operator-speed reference is $1.092\times$, below the separate R03 useful-system threshold of $1.10\times$; this systems criterion is distinct from the single-site task gate summarized in Table 5.

The map is heterogeneous: the same retention ratio does not imply the same downstream consequence, and the local relative Frobenius distortion does not order the layers by task importance.

C.2 COMPOSITION STRESS TEST

R04 tested whether the individually tolerated ratios compose. The guided 12-layer vector was assembled directly from the R03 single-site map. It did not satisfy the global criterion: SST-2 valida-

Table 5: R03 BERT single-site sensitivity. The listed ratio is the most aggressive tested ratio satisfying the frozen strict task gate.

Layer	best ρ	acc. degr. (pp)	agreement	local relF
0	0.75	0.000	0.9908	0.276
1	0.75	0.459	0.9931	0.244
2	0.75	0.459	0.9908	0.178
3	0.875	0.229	0.9954	0.129
4	0.875	0.229	0.9954	0.135
5	0.75	0.344	0.9920	0.214
6	0.875	0.459	0.9931	0.141
7	0.875	0.229	0.9908	0.160
8	0.875	0.115	0.9966	0.165
9	0.75	0.459	0.9931	0.275
10	0.50	0.344	0.9966	0.427
11	0.50	0.115	0.9989	0.453

tion accuracy decreased by 2.294 percentage points and exact-prediction agreement fell to 0.9358. Importantly, the measured end-to-end implementation was also slower than the exact model. This negative result prevents a single-site map from being interpreted as an additive global error budget.

Table 6: Representative R04 multi-site compositions. Speedup below one means the tested implementation was slower than the exact model.

Condition	acc. degr. (pp)	agreement	GPU0 speedup	GPU1 speedup
Guided R03 allocation	2.294	0.9358	0.946	0.949
Uniform $\rho = 0.875$	1.261	0.9667	0.944	0.946
Uniform $\rho = 0.75$	3.211	0.9128	0.935	0.939
Uniform $\rho = 0.50$	19.954	0.7179	0.951	0.952

Among 31 same-multiset random permutations used in the R04 comparison, one had a lower prediction-flip rate than the guided allocation, giving the retained finite-set empirical value $(1 + 1)/(31 + 1) = 0.0625$. This is a development comparison over the enumerated/randomized policies, not a population-level significance claim.

C.3 Calibration-ranked sparse composition

R05 used a deterministic balanced 2,048-example SST-2 training subset (1,024 per label) to rank layers at fixed $\rho = 0.5$, then evaluated sparse top-k policies on the frozen validation set. The ranking transferred most strongly for loss regret, but only $k = 1$ met the frozen strict validation criterion; $k \geq 2$ did not. Table 7 shows the full sparse curve.

Table 7: R05 calibration-ranked sparse composition. The model-speed signal is negative for every retained k.

k	guided layers	strict?	acc. degr.	flip rate	GPU0 $\times$	GPU1 $\times$
1	11	yes	0.115	0.00115	0.989	0.992
2	10,11	no	0.573	0.01491	0.988	0.981
3	9–11	no	0.917	0.02752	0.974	0.984
4	8–11	no	1.032	0.03784	0.983	0.977
5	0,8–11	no	1.491	0.04702	0.968	0.971
6	0,1,8–11	no	1.835	0.07110	0.962	0.973

Calibration-to-validation rank correlations were 0.6268 for prediction-flip rate ($p \approx 0.0292$), 0.8392 for loss regret ($p \approx 0.000643$), and 0.410 for accuracy degradation ($p = 0.185$). These values support a site-ranking signal but not successful multi-site composition on this testbed.

D NEGATIVE OPERATOR-FAMILY ADMISSIONS

A central part of the audit is that nominally cheaper computation is not automatically retained. R06–R09 deliberately test attractive shortcuts against stronger implementation baselines.

D.1 BERT IMPLEMENTATION DISCRIMINATOR

R06 compared several partial-GEMM expressions against native `F.linear` for the BERT FFN shape. None produced a median speedup above one. The best tested row was `addmm_2d` at $\rho = 0.5$, with speedups 0.943 and 0.951 on the two A100s. This closes the BERT systems route for the tested realization even though earlier synthetic/strided microbenchmarks showed arithmetic savings.

D.2 SAMPLED LAYERNORM STATISTICS

R07 evaluates sampled LayerNorm-statistics realizations together with an exact Triton control. Native PyTorch LayerNorm remains faster at both tested row counts. At 4,096 rows and hidden size 768, for example, the exact Triton control reaches only 0.441/0.433× the native speed, while sampled variants are approximately 0.34×. This is a systems rejection rather than a claim that sampled normalization lacks a numerical fidelity curve.

Table 8: Representative negative systems controls. "Speedup" is candidate versus the corresponding exact/native reference.

Round	Condition	fidelity	GPU0 ×	GPU1 ×
R06	BERT partial `addmm_2d`, $\rho = 0.50$	not the limiting issue	0.943	0.951
R07	Triton exact LayerNorm control, 4096×768	rel. distortion $\approx 1.5 \times 10^{-5}$	0.441	0.433
R07	sampled stats every 2, 4096×768	distortion 0.0442	0.339	0.346
R09	large Top-K, iter=8, 4096×4096, $k = 128$	hit rate 0.9892	0.985	0.985

D.3 TOP-K IMPLEMENTATION VERSUS APPROXIMATION GAIN

R08 initially showed a large speed gap between an RTop-K implementation and `torch.topk`. R09 therefore separates two questions: (i) the implementation advantage of exact-like RTop-K over `torch.topk`, and (ii) the additional gain attributable to early stopping within the same RTop-K family. Only the latter is counted as an inexact speedup. Table 9 shows that high-fidelity early-stop points survive only at the smallest shape; medium and large shapes lose the required speed margin as fidelity approaches the exact Top-K set.

Table 9: Selected R09 early-stop Top-K points, referenced to exact-like RTop-K rather than `torch.topk`.

Rows	Dim	k	iter	hit rate	GPU0 ×	GPU1 ×
1024	768	32	8	0.9913	1.113	1.107
4096	2048	64	4	0.8565	1.114	1.115
4096	2048	64	8	0.9893	1.035	1.034
4096	4096	128	2	0.5374	1.088	1.088
4096	4096	128	8	0.9892	0.985	0.985

The retained decision was therefore `INEXACT_TOPK_LOW_FIDELITY_ONLY`; no downstream task route was opened.

E PRECISION POSITIVE CONTROLS

R10 and R11 are conventional hardware-supported controls. Their role is to show that the timing and task-evaluation stack can recover familiar quality–speed improvements; they are not claimed as novel approximation mechanisms.

Table 10: Precision controls. Accuracy is the measured task accuracy; speedup is reported for both A100 lanes.

Task	mode	accuracy	agreement	GPU0 ×	GPU1 ×
BERT/SST-2	FP32/TF32	0.924312	1.0000	2.689	2.642
BERT/SST-2	AMP BF16	0.924312	1.0000	1.956	1.960
BERT/SST-2	full BF16	0.924312	1.0000	2.738	2.767
ViT/CIFAR-100	FP32/TF32	0.9148	1.0000	3.976	3.951
ViT/CIFAR-100	AMP BF16	0.9147	0.9989	6.353	6.356
ViT/CIFAR-100	full BF16	0.9146	0.9988	8.185	8.127

The ViT full-BF16 control changes 12 of 10,000 predictions; AMP BF16 changes 11. These controls establish that the audit does not reject approximation by construction.

F ESM-2 FFN SYSTEMS ADMISSION AND SINGLE-SITE TASK REGRET

F.1 OPERATOR-LEVEL SYSTEMS ADMISSION

R12 repeats the retained-inner-dimension realization at ESM-2 650M FFN shapes. Unlike the BERT `F.linear` case, the tested `addmm_2d` implementation becomes operator-positive. The primary DMS-like regime uses $M = 2304$ flattened tokens, corresponding to batch size 8 and the ESM-2 sequence-length regime used by the protein case study.

Table 11: R12 primary ESM-2 FFN operator measurements. These are operator-level measurements, not end-to-end ESM-2 speedups.

Projection	ρ	local relF	GPU0 ×	GPU1 ×	gate
1280→5120 intermediate	0.75	0.4998	1.113	1.116	yes
1280→5120 intermediate	0.50	0.7069	1.410	1.408	yes
5120→1280 output	0.75	0.4989	1.159	1.163	yes
5120→1280 output	0.50	0.7068	1.577	1.576	yes

F.2 PROTEIN SINGLE-SITE LANDSCAPE

R13 inserts only the R12-admitted $\rho = 0.75$ realization at one ESM-2 FFN site at a time while scoring the TEM-1 ProteinGym assay. The model is `facebook/esm2_t33_650M_UR50D`; the assay contains 4,996 substitutions across 263 mutated positions in a 286-residue sequence. The exact BF16 DMS Spearman is 0.730850. Table 12 gives all eight interventions.

Table 12: Complete R13 single-site landscape. The strict column uses the frozen development gate defined in Appendix B.

Layer	proj.	relF	score ρ	DMS ρ	DMS regret	top10	strict
0	int.	0.2661	0.99832	0.73408	< 0	0.964	yes
10	int.	0.2947	0.99732	0.72979	0.00106	0.960	yes
21	int.	0.4281	0.98110	0.72955	0.00130	0.894	no
32	int.	0.3589	0.98528	0.71919	0.01166	0.912	no
0	out.	0.2431	0.99844	0.72929	0.00156	0.964	yes
10	out.	0.3966	0.99814	0.73337	< 0	0.970	yes
21	out.	0.4906	0.99252	0.72468	0.00617	0.920	no
32	out.	0.4285	0.99176	0.72488	0.00597	0.942	no

For R13, uncertainty is computed by 500 paired cluster-bootstrap replicates with mutated position as the resampling unit, because all substitutions at a position share the same masked-model forward. The bootstrap interval is reported but was not itself an automatic pilot gate. The most clearly separated condition is intermediate layer 32, whose bootstrap interval for DMS-Spearman delta is approximately $[-0.01909, -0.00411]$ relative to exact.

G GROUPED MASKED-CONTEXT SCORING

G.1 EXACT AND CONTRACT-RELAXED SCORING RULES

For a target position p, exact masked-marginal scoring creates a wild-type sequence with only p masked, runs ESM-2 once, and scores a substitution a as

$$s_p(a) = \log P_\theta(a \mid x^{(p)}) - \log P_\theta(x_p^{\mathrm{WT}} \mid x^{(p)}). \tag{7}$$

For a group G of target positions, the cheaper realization masks all positions in G simultaneously and reads the logits for each $p \in G$ from a single forward. The score expression is unchanged, but the conditioning context becomes $x^{(G)}$ rather than $x^{(p)}$. This is the local contract change. Group size is therefore both a computational knob and a transparent measure of how many target contexts are coupled.

R14 evaluates $g \in \{1, 2, 4, 8, 16, 32, 64, 263\}$ on TEM-1; R15 focuses on $g \in \{1, 2, 4, 8, 16\}$ on influenza HA; R18 freezes $g \in \{1, 2, 4, 8, 16\}$ for held-out DHFR. For every random $g > 1$ point used in the main comparison, three deterministic partitions are retained. The plotted task-regret point is their mean, while the one-sided error bar reaches the worst retained partition. These bars are not confidence intervals.

G.2 COMPLETE MAIN GROUP-SIZE FRONTIERS

Table 13: Random grouped-context frontiers used by the main cross-protein result. Score ρ and top10 report the worst retained partition; DMS regret is the retained mean used for the main frontier. Speedup is the conservative two-device value.

Assay	g	speedup	DMS ρ	DMS regret	score ρ	top10
TEM-1	1	1.000	0.730850	0	1.000000	1.000
TEM-1	2	1.988	0.730784	0.000066	0.999268	0.976
TEM-1	4	3.946	0.729842	0.001008	0.998663	0.964
TEM-1	8	7.762	0.728983	0.001868	0.996945	0.940
TEM-1	16	14.695	0.727267	0.003584	0.994560	0.928
HA	1	1.000	0.506846	0	1.000000	1.000
HA	2	1.999	0.505467	0.001379	0.998685	0.9739
HA	4	3.986	0.505622	0.001225	0.997362	0.9599
HA	8	7.927	0.502982	0.003865	0.991335	0.9450
HA	16	15.587	0.501748	0.005098	0.984024	0.9132
DHFR	1	1.000	0.479592	0	1.000000	1.000
DHFR	2	2.002	0.479559	0.000033	0.998739	0.9536
DHFR	4	3.935	0.478538	0.001054	0.996841	0.9283
DHFR	8	7.386	0.480500	-0.000908	0.994101	0.9030
DHFR	16	12.335	0.482555	-0.002963	0.989015	0.8819

The exact-score fidelity decreases monotonically enough to expose the semantic effect of grouping, whereas DMS-Spearman regret need not be monotone on a finite assay. That difference is central to the paper: the local diagnostic describes deviation from the reference computation, while task regret describes the selected downstream utility.

G.3 WT-MARGINAL TRANSPARENCY COMPARATOR

A one-pass WT-marginal comparator is retained so that grouped masking is not compared only against a deliberately expensive reference. It runs the unmasked wild-type sequence once and reads logits at all target positions. It is not presented as a new method. Its measured speedup/regret pairs are 62.8×/0.00891 on TEM-1, 375.6×/0.01784 on HA, and 25.6×/0.00082 on DHFR. The DHFR value is a useful reminder that a very cheap baseline can happen to have low measured regret on one assay; the paper does not convert this into a general superiority claim.

H STRUCTURE-AWARE GROUPING AND ANTI-LEAKAGE CONTROLS

H.1 PAIR-INTERFERENCE TARGET AND FEATURES

R16/R17 ask whether the context sensitivity exposed by grouped masking contains predictable structure. The final development round (R17) deterministically samples 2,048 unique TEM-1 position pairs, stratified by sequence-distance quartile (512 pairs per quartile). For each pair, both positions are masked together and the target is the 90th percentile absolute substitution-score drift from the exact single-mask score over substitutions at the two positions.

A single unmasked ESM-2 forward provides test-time position features. Pair features include normalized and log sequence distance, several exponential distance kernels, normalized absolute positions, WT-residue equality, hidden-state cosine and normalized L2 differences at layers 11, 22, and 33, plus unmasked amino-acid logit entropy and confidence-margin statistics. A standardized ridge regressor is fit on TEM-1. The regularization coefficient is selected from a frozen grid by five deterministic position-blocked folds; validation pairs contain at least one held-out-fold position, and training pairs contain no held-out-fold positions. The selected R17 predictor has mean fold Spearman 0.415893.

For HA policy construction, the following information is explicitly forbidden: HA DMS labels, HA exact masked-marginal scores, and HA grouped-mask outcomes. The policy is allowed to use only the wild-type HA sequence, the mutated-position list, and one unmasked ESM-2 feature forward. Thus the downstream HA metrics are evaluation targets, not policy inputs.

H.2 DEVELOPMENT POLICY COMPARISON

Table 14: R17 HA development comparison. Adjusted speed includes the measured test-protein policy overhead used in R17.

Policy	g	DMS regret	score ρ	top10	adjusted $\times$
learned	8	0.002128	0.997508	0.9664	6.017
learned	16	0.001181	0.996090	0.9496	10.404
distance	8	0.001496	0.998321	0.9711	6.582
distance	16	0.002877	0.996312	0.9534	12.344

The learner has a nontrivial development signal, but it does not uniformly dominate the distance heuristic. The retained interpretation is therefore structured interference rather than a generally superior learned router.

H.3 DESCRIPTIVE REGRET BUDGETS

R17 also instantiates Equation 2 on the HA development frontier. Table 15 lists the largest measured speedup among the retained HA candidates for several illustrative values of ϵ. These values were chosen as descriptive slices and are not standards or transferred thresholds.

Table 15: R17 descriptive HA regret-budget curve.

ϵ	selected point	speedup	measured regret
0.0000	exact $g = 1$	1.000$\times$	0.000000
0.0010	exact $g = 1$	1.000$\times$	0.000000
0.0025	learned $g = 16$	10.404$\times$	0.001181
0.0050	distance $g = 16$	12.344$\times$	0.002877
0.0100	random $g = 16$	15.587$\times$	0.005098
0.0200	WT marginal	375.650$\times$	0.017839

H.4 FROZEN HELD-OUT DHFR COMPARISON

R18 freezes the distance rule before evaluation on E. coli DHFR. The distance risk is $\exp(-|i-j|/32)$, and deterministic balanced greedy grouping minimizes within-group risk. No DMS label, exact masked score, or grouped-score outcome from DHFR is used to tune the rule. The result reverses the development intuition: distance grouping improves exact-score fidelity but increases DMS-Spearman regret relative to the random mean at both matched group sizes.

Table 16: R18 held-out matched-cost comparison. Distance grouping improves exact-score fidelity but has higher signed DMS regret than random grouping at both matched group sizes.

g	random regret	distance regret	random score ρ	distance score ρ	distance $\times$
8	-0.000908	0.002639	0.994101	0.996210	6.309
16	-0.002963	0.002261	0.989015	0.993482	10.681

This held-out result is why the main paper treats the learned/distance policies as a boundary result rather than the primary contribution.

I STATISTICAL TREATMENT AND UNCERTAINTY

The experiments contain several different sources of variation, which are intentionally not conflated.

Biological/task sampling. R13 uses a paired cluster bootstrap over mutated positions (500 replicates) because multiple substitutions at the same position share the same masked forward. The bootstrap summarizes uncertainty in the DMS-Spearman difference for that fixed assay/model setup.

Grouping variation. R14, R15, and R18 use three deterministic random partitions for each random group size. The main frontier reports mean signed regret across those retained partitions and a one-sided bar to the worst observed partition. This is a finite policy-range summary, not a confidence interval and not a biological replicate estimate.

Hardware timing variation. The two A100 lanes are independent systems replicates. The paper uses the weaker lane-wise median speedup for protein frontier points and retains p05/p95 timing summaries to check strong separation. These lanes do not create additional biological samples.

Development comparisons. The R04 same-multiset permutation comparison and R05 matched-random comparisons use finite empirical counts to rank development policies. They should not be read as confirmatory hypothesis tests over an underlying population of all possible allocations.

Cross-assay scope. The three ProteinGym assays are selected case studies under a single frozen ESM-2 scorer. They establish repeated empirical frontiers under the tested setup but do not estimate ProteinGym-wide, protein-family-wide, or deployment-wide generalization.

J ROUND-BY-ROUND EVIDENCE MAP

Table 17 records why each round exists. The point of retaining this map is to make negative branches visible rather than presenting only the successful protein case study.

K REPRODUCIBILITY, PROVENANCE, AND ARTIFACT INTEGRITY

The reproducibility package uses a hierarchy of evidence: frozen machine-readable configuration/specification, source code actually executed, raw result artifacts, round-level summaries/TAKEBACK files, and paper-ready derived tables. Narrative prose is downstream of these artifacts; if prose disagrees with a machine-readable artifact, the artifact is treated as the source of truth and the prose must be corrected.

Table 17: Experimental evidence map.

Round	Question	Retained outcome / role
R00	Is the broad idea novel?	No broad novelty claim for approximate or quality-constrained computing; forces narrower empirical contribution.
R01–R01c	Does retained-K GEMM have a real operator frontier?	Yes at some shapes; BERT-like layouts expose copy/stride effects.
R02	Does one realization have different task effects by BERT layer?	Yes; sparse site heterogeneity signal.
R03	Does heterogeneity persist across all 12 layers?	Yes; different layers admit different strict single-site ratios.
R04	Do individually tolerated sites compose?	No; guided multi-site allocation degrades SST-2 and is slower.
R05	Can calibration ranking rescue sparse composition?	Ranking transfers, but only $k = 1$ passes the strict task gate; no model-speed signal.
R06	Can a fair BERT partial implementation beat native `F.linear`?	No tested route does; BERT GEMM systems branch closed.
R07	Does sampled LayerNorm create a useful systems point?	No; even exact Triton control is slower than native.
R08/R09	Is Top-K gain implementation- or approximation-driven?	Exact-like implementation is fast; additional early-stop gain is high-fidelity only at the smallest tested shape.
R10	Can the stack recover a known hardware-supported win?	Yes; BERT TF32/BF16 positive control.
R11	Does the positive control transfer to vision?	Yes; ViT/CIFAR-100 TF32/BF16 control.
R12	Can retained-K become systems-positive at ESM-2 FFN shapes?	Yes at the operator level for tested `addmm`.
R13	Is ESM-2 single-site task tolerance location dependent?	Yes; four of eight probed FFN sites meet the frozen gate.
R14	Does an explicit context-contract relaxation form a useful frontier?	Yes on TEM-1; grouped masking yields measured speed–regret tradeoff.
R15	Is the frontier a single-protein accident?	Same qualitative frontier appears on influenza HA.
R16	Can pair interference be learned across proteins?	Nontrivial signal, but a simple distance heuristic is competitive; initial planner overhead matters.
R17	Does stronger development supervision establish a learned router?	Predictor CV Spearman 0.4159; learned policy does not uniformly dominate distance.
R18	Does the frozen structure rule transfer to an unseen assay?	Frontier persists on DHFR, but distance-aware DMS-regret benefit does not transfer; scientific compute stops.

The compact return archives retain experiment specifications, source hashes, environment metadata, aggregate outputs, timing summaries, and SHA-256 manifests. Model checkpoints, Hugging Face caches, Python environment trees, and large unnecessary tensors are intentionally not redistributed. Public model/dataset identifiers and frozen revisions are retained in the round artifacts instead.

Table 18: Archive-integrity status retained by the final paper package.

Archive range	Integrity status
R01b	Return manifest lists three files (spec/source files) that are absent from the returned ZIP; caveat preserved.
R02	Return manifest lists one tensor artifact absent from the returned ZIP; caveat preserved.
R01c, R03–R18	Manifest-listed artifacts were present and hash-matched in the local audit.
R18	Final held-out round is manifest-complete and hash-matched; exact cross-GPU score Spearman is 1.0.

The early archive-completeness caveats do not support the main protein claims, which are traced to the later R13–R18 artifacts. They are reported rather than silently repaired.

K.1 Frozen revisions and final held-out assay

The final DHFR round freezes ProteinGym repository revision `e936ae69e3c70ff651d47889a389de6f596863b2` and ESM-2 revision `08e4846e537177426273712802403f7ba8261b6c`. The assay contains 2,363 single substitutions on a 159-residue reference sequence, with 157 observed mutated positions in the frozen substitution CSV (positions 2–159; the ProteinGym reference metadata annotates the region as 1–159). The exact BF16 DMS Spearman is 0.479592, and the exact score vectors from the two A100 lanes have Spearman 1.0.

K.2 STOP RULE

R18 is the final planned scientific-compute round. After the frozen DHFR evaluation, no additional operator family, protein, group size, or learned controller is introduced to improve the story. Subsequent work is restricted to recomputing statistics from retained raw data, generating paper figures/tables, verifying citations and numerical traceability, and correcting implementation errors that would invalidate an existing result. This stop rule is part of the audit discipline: negative transfer on DHFR is retained as evidence rather than followed by another tuning round.