WHEN DOES FEW-SHOT TRANSFER LEARNING WORK ACROSS MUTATION ASSAYS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein mutagenesis can generate large numbers of measurements across many proteins, but characterizing any single new protein-assay pair remains expensive. A typical plate has only a few hundred labeled mutants, too few to train an accurate predictor from scratch. Related measurements offer a way to reduce that burden. A model trained on how mutations shift a property in another assay, or in other proteins, may already encode information useful for the new plate. The question is whether that older data can train a predictor for a new assay from only a small target plate, and whether any gain is genuine transfer rather than protein-identity leakage. We study this with three overlap conditions at a fixed 200-label budget. Where source and target assays measure the same proteins, identity itself may carry the signal, so we compare from-scratch training on the plate to sequential transfer. Where the sources share no exact wild-type sequence with the target, a source checkpoint has no shared identity left to exploit; what may still help is a shared initialization that re-specializes from a few target labels. There we apply first-order model-agnostic meta-learning (FOMAML), with a source protein as the task, meta-trained on solubility and thermostability toward a proteolysis-stability (K50) target, beside a random-initialization control. A third condition holds protein identity out of labeled training and spends the whole budget on one protein. Its plate geometry differs, so we report it beside the matched pair rather than on the same axis. When source and target share the same proteins, sequential transfer clearly beats training from scratch on both ranking and calibration. On the identical plate with sequence-disjoint sources, that gain disappears: every arm, including transfer and FOMAML, ranks near chance despite a large source label pool. On a held-out protein with the budget concentrated on that protein alone, transfer again leads from-scratch at every seed and FOMAML clears its random-initialization control, but both remain far below simply fitting the plate. Spreading the same budget across many held-out identities returns every arm to chance. The pattern is that few-shot transfer here is contingent on shared protein identity among the admitted sources (with a remaining confound that relatedness and provenance also change) and on how the labels are spent, not which reuse method is used.

1 INTRODUCTION

A mutagenesis campaign starts in the lab with a wild-type protein and a library of single-point mutants of it. A wild type is one protein's unmutated reference sequence and a mutant is that sequence with substitutions. Each mutant is evaluated for how a physical or chemical property changes relative to the wild type. These properties can be thermostability, solubility, proteolysis resistance, and related readouts. The resulting labels are the experimental record of that campaign. We call that measurement protocol an assay. This is one property, measured under one lab protocol, on one set of mutants. Examples of assays are Tsuboyama $\log_{10} K_{50}$ (Tsuboyama et al., 2023), FireProt ΔT_m (Stourac et al., 2021), or SoluProtMut ΔS (Velecký et al., 2022). Deep mutational scanning and related high-throughput methods make such campaigns feasible (Fowler & Fields, 2014; Tsuboyama et al., 2023), but they remain expensive per protein-assay pair.

A typical physical experiment (called a plate) yields only on the order of a hundred to a few hundred labeled mutants. Protocols are not standardized across labs, so labels from different campaigns do

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.

not sit on a shared scale that one could naively pool and fit (Pucci et al., 2022). That plate is also far too small to train a fully tuned predictor that takes wild type and mutation and returns the property shift from scratch.

What can still help is reuse of related measurements. If two assays measure overlapping mutants, or if other proteins have already been labeled for a related property, a model trained on how mutations shift that property elsewhere may already carry information useful for a new target plate (Dieckhaus et al., 2024). The practical question is whether that existing data can transfer into predicting a new assay from only a few hundred target labels. Additionally how much of any apparent gain is genuine transfer of information, as opposed to protein-identity leakage from shared wild types (Pucci et al., 2022). We study that question by comparing sequential transfer to from-scratch training on the plate, at a fixed label budget that matches what one round of a mutagenesis campaign typically buys. We look at three overlap conditions through that lab lens, each at the same 200-label plate budget (Figure 2; Section 4).

Study 1 asks how much sequential transfer buys when the old measurements and the new plate share the same proteins, so identity itself may carry the signal. Study 2 keeps that identical plate but admits only sources with no shared wild-type sequence, so any remaining gain cannot be memorization of the target proteins. Study 3 holds a protein out of labeled training and spends the whole 200-label budget on it alone. This is the deployment case of predicting the rest of a never-before-labeled protein, and is reported separate from the Study 1–2 axis because its plate geometry differs.

Against those questions we compare three reuse strategies plus a control (Figure 3). From-scratch fine-tuning fits the head on the plate alone. Sequential transfer first trains on all admitted source labels, then specializes on the same plate. Where a protein is the task unit, we also ask whether meta-learning (first-order MAML, FOMAML) can learn an initialization that adapts quickly from a handful of target labels. A random-initialization control applies the same adaptation operator as FOMAML to a head that never saw meta-training, so any gap between them is what meta-training bought.

We score every arm with within-(protein, assay) Spearman ρ as the ranking metric and pooled z-MSE as a calibration check against a named reference (Section 5, Table 1). Means and standard deviations are reported over four seeds $\{42, 43, 45, 46\}$.

Contributions.

- **Same plate, different sources.** Fix the target plate (200 K50 labels on the same 64 proteins, same encoder, same from-scratch baseline) and change only which older measurements are allowed in. With $\Delta\Delta G$ on those same 64 wild types, sequential transfer reaches $\rho = 0.475 \pm 0.010$. With only sequence-disjoint $\Delta S{+}\Delta T_m$ sources (15,986 rows), it falls to 0.029 ± 0.034, and every other arm is near chance too. So more source labels are not enough. Shared wild types matter for the gain we see, but they are not cleanly separated from what else changes in that swap: Study 1's $\Delta\Delta G$ is co-measured with K50 (paired-mutant ρ=0.95), while Study 2 draws solubility and melting temperature from other databases.
- **How the 200 labels are spent matters more than which reuse method is used.** The same label budget ranks mutants well when it is spent on one protein and fails when it is spread thinly. On a never-before-labeled protein with all 200 measurements on that protein alone, transfer reaches $\rho = 0.312$ and from-scratch 0.262, both far above a random-initialization control at -0.015. Spreading those same 200 labels across all 46 held-out proteins ($\sim$4 each) returns every arm to chance.

2 Related work

This paper asks which older mutation measurements help a new plate, and which reuse methods can extract that help at a fixed label budget. The pieces below are the encoder, the public data we compile, and the transfer methods we compare.

Encoders and data benchmarks. Protein language models such as ProtBERT and ESM supply mutant representations (Elnaggar et al., 2022; Rives et al., 2021; Lin et al., 2023; Alley et al., 2019;

Rao et al., 2019) and zero-shot mutant ranking (Meier et al., 2021; Notin et al., 2022), but each new assay is a new label space: a representation alone does not say how to reuse labels from another campaign. We therefore freeze one encoder and treat it as a fixed featurisation shared by every arm, so gaps between methods are not confounded with representation choice, a reader who prefers a different encoder can substitute one without changing the comparisons. On the data side we use a stability-focused compilation (Tsuboyama proteolysis/stability libraries (Tsuboyama et al., 2023), FireProt ΔT_m (Stourac et al., 2021), and SoluProtMut ΔS (Velecký et al., 2022)) rather than a fitness leaderboard. ProteinGym (Notin et al., 2023a) and related suites such as FLIP (Dallago et al., 2021) standardize splits for scoring models, we instead build three source–target overlap policies on one compilation (shared proteins, sequence-disjoint sources, labeled protein holdout) so that identity leakage can be measured rather than assumed away. Overlap between training and test items can inflate apparent generalization (Lewis et al., 2021), once the encoder has been domain-adapted on unlabeled strings, checking labeled splits alone is not enough.

Transfer learning. The practical question is how to move information from admitted source labels into a small target plate. Sequential transfer trains on the source and then specializes on the plate (Yosinski et al., 2014), with no episodic outer loop. From-scratch fine-tuning fits only the plate and is the baseline that transfer must beat. Where a protein is the task unit, first-order MAML (FOMAML) seeks an initialization from which a few gradient steps on a new support reduce query error (Finn et al., 2017; Nichol et al., 2018), we compare it to the same randomly initialized head adapted by the same inner operator, so the gap is what meta-training bought. Closely related protein work fits simple regression models on pretrained embeddings or evolutionary scores (Biswas et al., 2021; Hsu et al., 2022) or conditions on other labeled measurements (Notin et al., 2023b) under low-N budgets, those methods ask how to use a plate well, whereas we ask which source labels are allowed to enter it. Few-shot molecular property prediction studies the same sample-efficiency question on small molecules (Stanley et al., 2021; Altae-Tran et al., 2017), though we do not use those datasets here.

3 Data and representation

Compilation and scaling. We compile four public datasets into one training corpus: (1) Tsuboyama $\log_{10} K_{50}$, (2) pooled folding-stability $\Delta\Delta G$ from Tsuboyama and FireProt, (3) FireProt melting-temperature change ΔT_m, and (4) SoluProtMut solubility change ΔS. After cleaning, the corpus covers $65{,}520$ labeled mutants over 305 protein identities (Table 6). Figure 4 shows how those labels are distributed before any scaling: the sources sit on incompatible units and sign conventions, so we first arrange orientation so that higher values mean more stable throughout.

Scale map. Different sources still sit on incompatible numeric scales, so we build a scale map that brings every admitted source onto a common corresponding scale. That map is a per-campaign z-normalization

$$z = \frac{y - \mu_c}{\sigma_c}, \tag{1}$$

with campaign mean and standard deviation (μ_c, σ_c). The map is reproducible and identical across methods, and it is invertible ($y = z\,\sigma_c + \mu_c$), so predictions can be returned in the original assay units when needed.

Partitions. Each study fixes (μ_c, σ_c) from its own partition: Studies 1 and 2 from the full target-training split, Study 3 from each plate's own support. We train every arm on z-MSE under this map and report that loss as a calibration check, ranking uses within-(protein, assay) Spearman on the raw labels (Section 5), which never depends on (μ_c, σ_c). Appendix A gives the compile detail, identity merge, and splits.

Encoding. A raw mutant string does not tell the encoder where the substitution sits: appending a mutation token at the end leaves the site unmarked, and the tokenizer treats those characters as unknowns. We instead crop a window around each mutated position and mark the site with in-vocabulary `[SEP]` tokens so the model sees the wild-type residue, the mutant residue, and their location in one continuous string. Only the input markup changes, the ProtBERT tokenizer and vocabulary stay fixed. Figure 1 shows this proposed input representation.

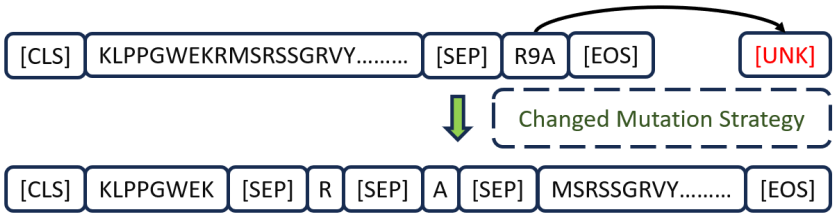

Figure 1: Proposed mutation-site input representation. Top: appending the mutation string leaves the site unmarked and routes indices as unknown tokens. Bottom: in-vocabulary `[SEP]` site marks inside a centered window (Appendix A.2).

4 One controlled axis and one side question

Figure 2 lays out the three studies on one identity axis: Studies 1 and 2 share one k=200 K50 plate and differ only in which source band is admitted, while Study 3 spends the same budget on a labeled holdout and is scored on its own (Appendix A.3). We never average ρ across studies.

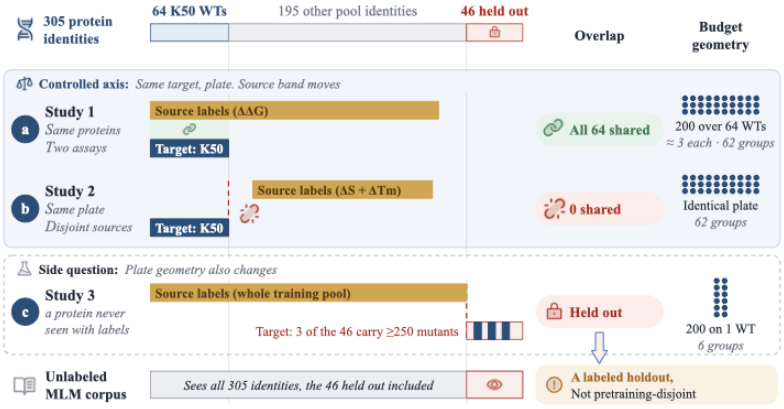

Figure 2: Three studies on one protein-identity axis (305 identities). Study 1 admits $\Delta\Delta G$ sources on the same 64 K50 wild types as the target plate (all 64 shared). Study 2 keeps that identical k=200 plate but admits only ΔS+ΔT_m sources on the 195 other pool identities (0 shared). Study 3 spends the same budget on a labeled holdout (200 on one of three eligible held-out wild types). The unlabeled MLM corpus spans all 305 identities, so Study 3 is a labeled holdout, not pretraining-disjoint.

Study 1: same proteins, two assays. The target plate has 200 new K50 measurements spread over 64 proteins ($\sim$3 each, 62 scored groups), and older $\Delta\Delta G$ measurements of those exact same 64 wild types are admitted as the source. Protein and mutant overlap are left deliberately open. From-scratch uses only the new labels, sequential transfer first trains on the old measurements, then specializes on the same plate. If transfer wins, the gain is useful but ambiguous: the model may have learned general mutation behaviour, or it may simply have learned how these particular proteins tend to behave (identity leakage). Both directions run ($\Delta\Delta G \rightarrow$K50 and K50$\rightarrow \Delta\Delta G$). Only from-scratch and sequential transfer apply: the target assay is already present at training time on other proteins, so there is no held-out protein task for an initialization to adapt to.

Study 2: same plate, disjoint sources. We keep the identical target plate from Study 1 and move the source band onto the 195 other pool identities, admitting only ΔS and ΔT_m (15,986 labeled rows). Every source campaign that shares an exact wild-type sequence with the target is removed (error, not warning). Study 1 has complete protein overlap, Study 2 has none, so the swap removes shared wild types on a fixed plate. Identity is not the only thing that changes: Study 1's $\Delta\Delta G$ source is co-measured with K50 (paired-mutant ρ=0.95), while Study 2's sources differ in physical property and database provenance (Section 7). Sequential transfer on the disjoint sources is allowed as a diagnostic but is not on the default queue, default arms are from-scratch, FOMAML and the random-initialization control on protein-level source tasks (Section 5). That filter blocks supervised leakage of source labels, it does not constrain the encoder, which still sees target-train strings during

unlabeled MLM. On current tables K50 keeps ΔS and ΔT_m, ΔS and ΔT_m each keep only K50, $\Delta\Delta G$ retains no disjoint source and is dropped.

Study 3: a protein never seen with labels. Instead of spreading $\sim$3 shots across 64 proteins, the experimentalist receives a protein never seen with labels, can afford 200 measured mutants, and wants to predict the rest. Because a protein is the task at both ends, sequential transfer, FOMAML and the random-initialization control all apply. Protocol: 15% of identities (46 of 305) are held out of every labeled split, held-out wild types still enter the unlabeled MLM corpus (labeled holdout, not pretraining-disjoint). The study is run twice on the same proteins and budget. The per-protein arm puts all 200 on one held out protein ($\geq$250 distinct mutants required, three of 46 qualify, six scored groups, Appendix A.3). The pooled-plate arm spreads the same 200 across all 46 ($\sim$4 mutants each).

5 METHODS

Frozen encoder and head. We use a frozen protein language model because sequence is the cheapest input a wet lab already has: the alternatives need a solved or predicted structure per wild type (Jumper et al., 2021) or a force field (Schymkowitz et al., 2005). Additionally sequential transfer and meta-learning ask about an initialization or a checkpoint over a fixed featurisation rather than about a particular architecture: any encoder that maps a mutant to a vector can sit underneath. ProtBERT ($\sim$420M) (Elnaggar et al., 2022) provides that map $h \in \mathbb{R}^d$. We freeze it and train only a three-layer MLP

$$f(h) = W_3\,\mathrm{ReLU}(W_2\,\mathrm{ReLU}(W_1 h)) \tag{2}$$

($W_1 \in \mathbb{R}^{256\times d}$, $W_2 \in \mathbb{R}^{128\times 256}$, $W_3 \in \mathbb{R}^{1\times 128}$) that predicts z. The representation therefore does not decide which arm wins: every method sees the same frozen `[CLS]` features, and a richer encoder could lift every arm together without explaining a gap between them (Appendix B.1). Before any labeled run we domain-adapt the encoder with BERT MLM (Devlin et al., 2019; Gururangan et al., 2020) for five epochs on study-train `sep_highlight` strings (43,310 unique mutants, all 305 wild-type sequences). All 46 held-out Study 3 wild types enter that unlabeled corpus via Study 1/2 target-train, so Study 3 is a labeled holdout, not a pretraining-disjoint benchmark.

Objective. All methods minimize z-MSE

$$\mathcal{L}(\theta, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x,z)\in\mathcal{D}} \big(f_\theta(x) - z\big)^2. \tag{3}$$

Figure 3 and Algorithm 1 state the four arms by what, if anything, crosses from a source phase onto the shared k=200 plate.

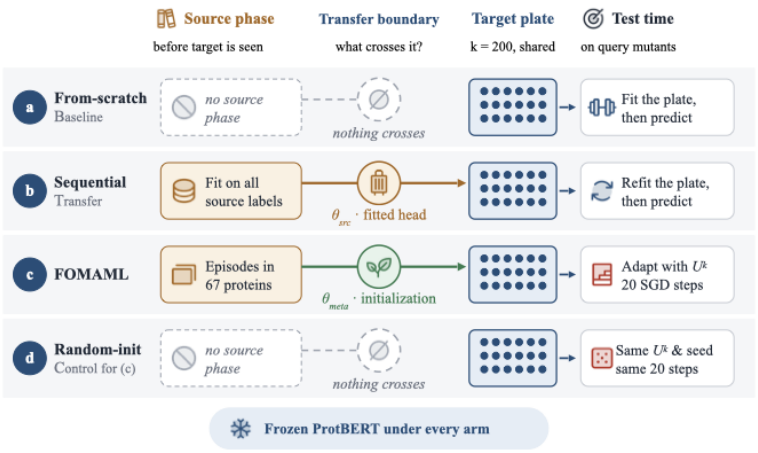

Figure 3: Four arms on a shared k=200 plate, under a frozen ProtBERT. From-scratch has no source phase and fits the plate, then predicts. Sequential transfer fits on all admitted source labels, carries the fitted head θ_{src} across the boundary, and refits the plate. FOMAML meta-trains episodes in 67 source proteins, carries an initialization θ_{meta}, and adapts with $U^{k_{\mathrm{inner}}}$ (20 SGD steps). Random-init applies the same $U^{k_{\mathrm{inner}}}$ and seed with nothing crossing the boundary, so the gap to FOMAML isolates the outer loop. Formal operators: Algorithm 1; full procedures: Appendix B.2.

From-scratch. AdamW (Loshchilov & Hutter, 2019) fits the head on the target plate alone,

$$\theta^\star = \arg\min_\theta \mathcal{L}(\theta, \mathcal{S}_{\mathrm{tgt}}). \tag{4}$$

Nothing crosses from a source phase. A fifth of the plate is held out of gradients and used only to choose the stopping epoch (five-epoch floor, patience five); those rows stay inside the k-label budget. The $\sim$295k-parameter MLP therefore receives gradients from 160 of 200 labels, whereas target-time inner-loop adaptation uses all 200. The nominal budget is equal across arms; the effective fitting budget is not.

Sequential transfer. Train the head on all admitted source-train labels, carry the fitted head θ_{src} across the boundary, then specialize on the same target plate under the same held-out stopping rule,

$$\theta_0 = \arg\min_\theta \mathcal{L}(\theta, \mathcal{D}_{\mathrm{src}}), \qquad \theta^\star = \arg\min_\theta \mathcal{L}(\theta, \mathcal{S}_{\mathrm{tgt}}) \text{ from } \theta_0. \tag{5}$$

With the encoder frozen and dropout off, each source `[CLS]` is a fixed vector, so we embed the source once and fit on the cached matrix, the target plate is always embedded live.

First-order MAML (FOMAML). Where a protein is the task unit (Studies 2–3; Section 4), FOMAML (Finn et al., 2017; Nichol et al., 2018) seeks an initialization θ_{meta} from which a few support steps reduce query error (Figure 3),

$$\min_\theta \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}\, \mathcal{L}\big(U^{k_{\mathrm{inner}}}(\theta, \mathcal{S}_\mathcal{T}), \mathcal{Q}_\mathcal{T}\big). \tag{6}$$

The shared inner operator $U^{k_{\mathrm{inner}}}$ is k_{inner}=20 head-only SGD steps,

$$\theta \leftarrow \theta - \alpha\, \nabla_\theta \mathcal{L}(\theta, \mathcal{B}), \qquad \alpha{=}10^{-3},\ |\mathcal{B}|{=}8. \tag{7}$$

Tasks are source proteins, not assays: assay-level tasks leave too few outer loop tasks to identify, whereas protein-level tasks raise the K50 fold to 67 and make every episode live inside one protein, matching how the target plate is drawn (Appendix B.1). Episodes are $8/8$ support/query (Vinyals et al., 2016), meta-training uses a five epoch floor with patience 5 on source val query MSE. We do not sweep α or k_{inner}.

Random-initialization control. Apply the same $U^{k_{\mathrm{inner}}}$, seed and step count to a head that never saw meta-training. Nothing crosses the boundary, FOMAML and this control differ only in whether the outer loop ran, so that gap (not FOMAML versus from scratch) is what meta-training bought.

Algorithm 1 Four arms on a shared k=200 plate (encoder frozen; head only).

1: **From-scratch:** solve equation 4 on $\mathcal{S}_{\mathrm{tgt}}$ (hold out 20% of k for early stop); score the query.
2: **Transfer:** solve equation 5: fit θ_0 on all admitted $\mathcal{D}_{\mathrm{src}}$, then specialize as in line 1 from θ_0.
3: **FOMAML:** meta-train θ_{meta} by outer Adam on equation 6 with inner operator equation 7; at target, $\theta^\star \leftarrow U^{k_{\mathrm{inner}}}(\theta_{\mathrm{meta}}, \mathcal{S}_{\mathrm{tgt}})$.
4: **Random-init:** $\theta^\star \leftarrow U^{k_{\mathrm{inner}}}(\theta_{\mathrm{rand}}, \mathcal{S}_{\mathrm{tgt}})$ with the same U, seed, and k_{inner} as line 3.

Metrics. We report two quantities (Table 1). The primary ranking metric is within-(protein, assay) Spearman ρ: an unweighted macro-average of Spearman correlation inside each query group with at least five non-constant raw labels ($+1$ perfect order, 0 chance, -1 reversed). The calibration check is pooled z-MSE on campaign z-scores equation 1, the training loss, read against a named reference (train-mean on Studies 1–2, support-mean on Study 3) and never as a substitute for ranking. Variability is the four-seed mean $\pm$ standard deviation over $\{42, 43, 45, 46\}$ (seed 44 is not in the batch). No interval here is a significance claim, we rest every comparison on the per-seed values in Appendix C.

Table 1: What each reported quantity measures. Per-seed values are in Appendix C; algorithms and hyperparameters are in Appendix B.

Quantity	Role	Definition
Within-(protein, assay) Spearman ρ	Ranking (primary)	Unweighted macro-average of Spearman ρ inside each query group with ≥ 5 non-constant raw labels. $+1$ perfect order, 0 chance, -1 reversed.
Pooled z-MSE	Calibration	Mean squared error on campaign z-scores equation 1, the training loss, read against a named reference (train-mean or support-mean), not as a substitute for ranking.
Seeds $\{42, 43, 45, 46\}$	Variability	Four-seed mean $\pm$ std., seed 44 is not in the batch. No hypothesis tests, no interval here is a significance claim.

6 RESULTS

We run every arm over four seeds, $\{42, 43, 45, 46\}$, and report the mean $\pm$ seed standard deviation. That $\pm$ is a spread across seeds, not a confidence interval. Per seed values are in Appendix C. Ranking uses within (protein, assay) Spearman ρ as defined in Section 5, z-MSE is the calibration check against that section's named references.

6.1 STUDY 1: SAME PROTEINS, TWO ASSAYS

When the source and plate share all 64 wild types, sequential transfer pulls far ahead of from-scratch on ranking and also improves calibration (Table 2).

Table 2: Study 1 at k=200. Ranking over 62 groups ($\Delta\Delta G\rightarrow$K50) and 109 (reverse), both arms share the group set. Bold marks the better arm. Per-seed: Appendix C.

Direction/metric	Scratch	Transfer
$\Delta\Delta G\rightarrow$K50 z-MSE	1.104 ± 0.083	$\mathbf{0.787 \pm 0.025}$
$\Delta\Delta G\rightarrow$K50 within-(protein, assay) ρ	-0.008 ± 0.021	$\mathbf{0.475 \pm 0.010}$
K50$\rightarrow$ $\Delta\Delta G$ z-MSE	1.470 ± 0.039	$\mathbf{1.461 \pm 0.036}$
K50$\rightarrow$ $\Delta\Delta G$ within-(protein, assay) ρ	0.003 ± 0.042	$\mathbf{0.316 \pm 0.009}$

On $\Delta\Delta G \rightarrow$K50, ρ rises from chance (-0.008) to 0.475, and z-MSE drops from 1.104 to 0.787, transfer wins at every seed. Identity is open by design: the two assays measure the same proteins, and paired-mutant Spearman between them is 0.95. The reverse direction still ranks well ($\rho = 0.316$ against 0.003 from scratch) even though z-MSE barely moves, because 39 unclipped ± 15 sentinels dominate squared error. In short, under this shared-identity, co-measured source band, transfer rearranges mutants correctly, and calibration follows when the label scale is well behaved.

6.2 STUDY 2: SAME PLATE, DISJOINT SOURCES

On the identical plate with sequence-disjoint sources, every arm collapses toward chance (Table 3).

Table 3: Study 2 K50 at k=200, 62 scored groups. From-scratch matches Study 1 (same plate). Bold marks sequential transfer, the strongest ranking arm. Per-seed: Appendix C.

Target	Scratch	Transfer	FOMAML	Random-init
K50 z-MSE	1.104 ± 0.083	1.145 ± 0.078	1.072 ± 0.017	1.059 ± 0.004
K50 within-(protein, assay) ρ	-0.008 ± 0.021	$\mathbf{0.029 \pm 0.034}$	0.011 ± 0.015	-0.021 ± 0.044

Transfer still leads on ranking ($\rho = 0.029$), with from-scratch at -0.008, FOMAML at 0.011 and random-init at -0.021, but none of these is a usable predictor. Critically, z-MSE does not improve under transfer either: values sit within about 0.05 of one another and of a constant predictor, so there is no calibration gain that would underwrite a transfer claim. Against Study 1's $\rho = 0.475$ under shared wild types, the disjoint $\Delta S + \Delta T_m$ sources (15,986 labeled rows) do not close the gap. Ranking and z-MSE move together here: both say that label volume without that shared-identity source band does not buy transfer (Section 7).

6.3 STUDY 3: A PROTEIN NEVER SEEN WITH LABELS

With all 200 labels on one held-out protein, sequential transfer clearly outperforms from-scratch on both ranking and z-MSE (Table 4).

Table 4: Study 3 at k=200 on the same 46 held out identities. Bold marks the best arm per row. Per-seed: Appendix C.

Plate	Metric	Scratch	Transfer	FOMAML	Random-init
Per-protein	within-(protein, assay) ρ	0.262 ± 0.055	$\mathbf{0.312 \pm 0.041}$	0.043 ± 0.038	-0.015 ± 0.043
Per-protein	z-MSE	1.232 ± 0.254	$\mathbf{1.183 \pm 0.236}$	1.279 ± 0.249	1.301 ± 0.266
Pooled plate	within-(protein, assay) ρ	0.027 ± 0.076	$\mathbf{0.057 \pm 0.025}$	0.000 ± 0.037	-0.002 ± 0.052
Pooled plate	z-MSE	1.778 ± 0.443	1.872 ± 0.463	$\mathbf{1.759 \pm 0.416}$	1.763 ± 0.425

Transfer reaches $\rho = 0.312$ against 0.262 from scratch, and leads at every seed. FOMAML clears its random-initialization control (0.043 against -0.015) but stays far below either plate-fitting arm. Spreading the same budget across all 46 held-out identities returns every method near chance: transfer's $\rho = 0.057$ is the least bad ranking score, while FOMAML is at 0.000 ± 0.037 on the pooled plate and still cannot match simply fitting a concentrated plate.

7 DISCUSSION

The pattern across Tables 2–4 is consistent: sequential transfer is the reuse method that helps at a 200-label budget (Chen et al., 2019; Tian et al., 2020), and meta-learning does not substitute for it on this compilation.

Transfer is contingent on shared identity, not label volume. Studies 1 and 2 hold the plate, budget, encoder and from-scratch baseline fixed and change only the admitted sources. Shared wild types buy a large ranking gain and a clear z-MSE drop. Sequence-disjoint sources leave every arm near chance despite 15,986 labeled rows. Transfer still edges the other arms on that thin plate, but without a reduction in z-MSE there is no calibration signal that would support a strong transfer claim. The controlled axis shows that the gain is contingent on shared protein identity among the admitted sources: remove that overlap, and the same procedure on the same plate falls to chance. We do not claim identity is the causal factor in isolation. Study 1's $\Delta\Delta G$ source is physically related and co-measured with the K50 target (paired-mutant ρ=0.95), whereas Study 2 admits solubility and melting-temperature labels from other databases, so identity overlap, physical relatedness, and provenance change together. Separately, what shared identity buys when it is present is still open: the model may be memorizing specific mutants, or it may be picking up structure shared across those proteins; an exact-match analysis would tell the two apart.

Meta-learning does not deliver a usable predictor. FOMAML was built for this few-shot structure, yet on Study 2 it sits at chance with its control, and on Study 3's concentrated arm it clears that control without approaching the plate-fitting arms (Raghu et al., 2020). On the pooled plate it is again near chance. Assays on incompatible scales, an identity-heavy frozen `[CLS]`, and a thin per-protein support are a poor fit for this operator, we read that as a negative finding about this data and this inner loop, not as a claim that meta-learning cannot work for mutation assays in general.

How the 200 labels are spent shapes every method. Concentrating the budget on one never before labeled protein lets transfer beat from-scratch, spreading it over 46 identities collapses every arm. The same label count is not the same experiment.

Limitations. The mutation-site encoding may also limit every arm (Figure 1): a single substituted site marked with `[SEP]` inside a frozen window is a weak local signal. A richer site encoding, a mutant-minus-wild-type feature, or an unfrozen encoder could change absolute numbers and possibly the gaps. Other limits: compile drops 602 Deng/Dutta rows ($< 1\%$), FOMAML uses a fixed inner operator we do not sweep, from-scratch fits 160 of 200 labels while $U^{k_{\text{inner}}}$ uses all 200, four seeds $\{42, 43, 45, 46\}$, campaign scalers are study-dependent (Appendix B.3).

8 CONCLUSION

Across three overlap conditions at a fixed 200-label budget, sequential transfer is the reuse method that works best. On a shared-protein plate it substantially outperforms training from scratch on both Spearman ranking and z-MSE (Section 5). On the same plate with sequence-disjoint sources it remains the strongest arm, even though every method is near chance once that overlap is removed (with the caveat that relatedness and provenance also change in that swap). On a held-out protein with the budget concentrated on that protein alone, transfer again beats from-scratch on both metrics and at every seed, though with only three eligible plates we treat that as suggestive rather than conclusive. Meta-learning does not offer a comparable practical win on this compilation: FOMAML stays near chance on the thin plate and, where it clears its control, still lags simply fitting the plate. These assays and this frozen representation may not support episodic adaptation at this budget. How the 200 labels are allocated matters as much as which method is used: spreading them over many proteins returns every arm to chance. The mutation-site encoding is a limit as well. Marking a single substituted site with `[SEP]` inside a frozen window may under-specify the local change that the labels describe, better site-aware or difference representations should be tried before concluding how far transfer can go.

AI USE STATEMENT

In this work, we have not used generative AI tools for any of the required-disclosure research tasks (generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, proving claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, translation, cleaning or reformatting datasets, qualitative or thematic analysis, or interpreting results). Those tasks were carried out by the authors. Additionally, however, we used generative AI tools to edit the manuscript for readability (grammar, wording, and polish), and for limited assistance with software-code syntax while implementing author-designed methods. We reviewed all AI-assisted edits and code suggestions, verified them against our own experiments and claims, and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS AND DATA STATEMENT

This study uses public protein-mutation measurements and contains no human-subject or personally identifiable data. Its predictions are research benchmarks, not validated guidance for clinical, environmental, or protein-engineering deployment. Tsuboyama, FireProtDB, and SoluProtMutDB are attributed to their original sources in Section 3. The raw Deng-tagged and Dutta-tagged campaigns are dropped at compile time and are not part of the released table. We redistribute processed splits only, users must obtain raw tables from the original providers and comply with those licenses. This paper grants no additional rights.

REPRODUCIBILITY STATEMENT

Code and data are available at the anonymized repository `https://anonymous.4open.science/r/maml-protein-0733`. Appendix D.1 lists commands and script-to-table mappings. The released default queue contains 30 frozen k=200 cells at two seeds (42, 43), the reported tables use four seeds (42, 43, 45, 46, 60 cells): sequential transfer and from-scratch on the two Study 1 directions, from-scratch, FOMAML and the random-initialization control on Study 2, and those three plus sequential transfer on each of Study 3's two plate geometries. Per-seed values are in Appendix C. Completed run directories store metrics, logs, provenance and fingerprints. The rank and near-duplicate analyses are separate diagnostics, not main-grid cells.

REFERENCES

Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature*

Methods, 16(12):1315–1322, 2019. doi: 10.1038/s41592-019-0598-1. URL https://doi.org/10.1038/s41592-019-0598-1.

Han Altae-Tran, Bharath Ramsundar, Aneesh S. Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *ACS Central Science*, 3(4):283–293, 2017. doi: 10.1021/acscentsci.6b00367. URL https://doi.org/10.1021/acscentsci.6b00367.

Surojit Biswas, Grigory Khimulya, Ethan C. Alley, Kevin M. Esvelt, and George M. Church. Low-N protein engineering with data-efficient deep learning. *Nature Methods*, 18(4):389–396, 2021. doi: 10.1038/s41592-021-01100-y. URL https://doi.org/10.1038/s41592-021-01100-y.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkxLXnAcFQ.

Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. FLIP: Benchmark tasks in fitness landscape inference for proteins. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://openreview.net/forum?id=p2dMLEwL8tF.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Henry Dieckhaus, Michael Brocidiacono, Nicholas Z. Randolph, and Brian Kuhlman. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proceedings of the National Academy of Sciences*, 121(6):e2314853121, 2024. doi: 10.1073/pnas.2314853121. URL https://doi.org/10.1073/pnas.2314853121.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022. doi: 10.1109/TPAMI.2021.3095381.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

Douglas M. Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nature Methods*, 11(8):801–807, 2014. doi: 10.1038/nmeth.3027. URL https://doi.org/10.1038/nmeth.3027.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7):1114–1122, 2022. doi: 10.1038/s41587-021-01146-5. URL https://doi.org/10.1038/s41587-021-01146-5.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1000–1008. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.86. URL https://aclanthology.org/2021.eacl-main.86/.

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. URL https://doi.org/10.1126/science.ade2574.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 29287–29303. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f51338d736f95dd42427296047067694-Paper.pdf.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. URL https://arxiv.org/abs/1803.02999.

Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan N. Gomez, Debora Marks, and Yarin Gal. Tranception: Protein fitness prediction with autoregressive transformers and inference-time retrieval. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16990–17017. PMLR, 2022. URL https://proceedings.mlr.press/v162/notin22a.html.

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 Marks. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/cac723e5ff29f65e3fcbb0739ae91bee-Paper-Datasets_and_Benchmarks.pdf.

Pascal Notin, Ruben Weitzman, Debora Marks, and Yarin Gal. Proteinnpt: Improving protein property prediction and design with non-parametric transformers. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 33529–33563. Curran Associates, Inc., 2023b. doi: 10.52202/075280-1458. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6a4d5d85f7a52f062d23d98d544a5578-Paper-Conference.pdf.

Fabrizio Pucci, Martin Schwersensky, and Marianne Rooman. Artificial intelligence challenges for predicting the impact of mutations on protein stability. *Current Opinion in Structural Biology*, 72:161–168, 2022. doi: 10.1016/j.sbi.2021.11.001. URL https://doi.org/10.1016/j.sbi.2021.11.001.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgMkCEtPB.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems*, volume 32, pp. 9689–9701. Curran Associates, Inc., 2019.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118. URL https://doi.org/10.1073/pnas.2016239118.

Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, and Luis Serrano. The FoldX web server: an online force field. *Nucleic Acids Research*, 33(suppl_2):W382–W388, 2005. doi: 10.1093/nar/gki387. URL https://doi.org/10.1093/nar/gki387.

Leslie N. Smith and Nicholay Topin. Super-convergence: very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006 of *Proceedings of SPIE*, pp. 1100612, 2019. doi: 10.1117/12.2520589. URL https://doi.org/10.1117/12.2520589.

Megan Stanley, John Bronskill, Krzysztof Maziarz, Hubert Misztela, Jessica Lanini, Marwin Segler, Nadine Schneider, and Marc Brockschmidt. FS-Mol: A few-shot learning dataset of molecules. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/8d3bba7425e7c98c50f52ca1b52d3735-Paper-round2.pdf.

Jan Stourac, Juraj Dubrava, Milos Musil, Jana Horackova, Jiri Damborsky, Stanislav Mazurenko, and David Bednar. FireProtDB: database of manually curated protein stability data. *Nucleic Acids Research*, 49(D1):D319–D324, 2021. doi: 10.1093/nar/gkaa981. URL https://doi.org/10.1093/nar/gkaa981.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In *Computer Vision – ECCV 2020*, volume 12359 of *Lecture Notes in Computer Science*, pp. 266–282. Springer, 2020. doi: 10.1007/978-3-030-58568-6_16.

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J. Weinstein, Niall M. Mangan, Sergey Ovchinnikov, and Gabriel J. Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973):434–444, 2023. doi: 10.1038/s41586-023-06328-6. URL https://doi.org/10.1038/s41586-023-06328-6.

Jan Velecký, Marie Hamšíková, Jan Štourač, Miloš Musil, Jiří Damborský, David Bednář, and Stanislav Mazurenko. SoluProtMutDB: A manually curated database of protein solubility changes upon mutations. *Computational and Structural Biotechnology Journal*, 20:6339–6347, 2022. doi: 10.1016/j.csbj.2022.11.009. URL https://doi.org/10.1016/j.csbj.2022.11.009.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3630–3638. Curran Associates, Inc., 2016.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27, pp. 3320–3328. Curran Associates, Inc., 2014.

A DATA, SPLITS AND ENCODING

How four public datasets become the three study trees used in the main text. Section 3 gives the high level picture, this appendix fills in the compile, cleaning, encoding, and split detail.

A.1 DATA COMPILATION AND CLEANING

Compiling. K50 is entirely Tsuboyama $\log_{10} K_{50}$ (Tsuboyama et al., 2023). Pooled $\Delta\Delta G$ joins Tsuboyama folding-stability libraries with FireProt, while ΔT_m is FireProt alone (Stourac et al., 2021) and ΔS is SoluProtMut (Velecký et al., 2022), per-family row and wild type counts are in Table 6. Tsuboyama public files place the variant in the sequence column, we invert that so the sequence field holds the wild type and the mutation string reads as wild type to mutant, matching FireProt. Raw positive K50 labels that are not already log-scaled are stored as $\log_{10}(\text{raw})$. Sequences keep the 20 standard residues, nonstandard characters are stripped and mutation indices remapped.

Sign harmonization. The sources do not share a sign convention, and per-campaign z-scoring removes offset and scale but not a sign flip. As stored, FireProt $\Delta\Delta G$ is positive for destabilizing mutations (Spearman -0.89 against FireProt ΔT_m on matched mutants) while Tsuboyama folding stability is positive for more stable ones (Spearman $+0.95$ against $\log_{10} K_{50}$ on paired mutants). We negate the FireProt $\Delta\Delta G$ rows so that higher means more stable throughout, after which the FireProt $\Delta\Delta G$/ΔT_m agreement is $+0.89$. Identity conflicts (next paragraph) are dropped before orientation tagging.

Cleaning and overlap. Compile drops 602 raw rows whose source tags are `BLAT_ECOLX_Deng_2012` or Dutta Mcl-1: those campaigns stored the same 271-aa sequence under incompatible names. Two smaller $\Delta\Delta G$ filters also run: 12 rows whose sequence column holds a mutation token rather than a wild type sequence, and 12 exact duplicate mutants within a (source, property) pair. After these drops, campaign to sequence and sequence to-identifier are compile time functions over the corpus in Table 6 (305 identities: 46 held out, 259 in the training pool, five of the 259 appear only in source-val). The audit file `identity_audit.json` records the conflict drop and the function checks. Table 5 counts exact wild-type sequence overlap after restoring Tsuboyama wild-type columns. The 64 shared wild types are why Study 1 exists and why $\Delta\Delta G$ cannot be a sequence-disjoint source for K50. K50 is sequence-disjoint from ΔS and ΔT_m (148 proteins, 67 usable meta-tasks on the K50 fold). On those 64 wild types, paired Tsuboyama mutants give Spearman ρ=0.95 and Pearson r=0.73 (per-wild-type median ρ=0.99), Pearson is lower because Tsuboyama ΔG is censored at ± 15 (178 paired sentinels), which enter z-scoring unclipped.

Table 5: Wild-type sequence overlap (shared WT counts).

	K50	$\Delta\Delta G$	ΔS	ΔT_m
K50	—	64	0	0
$\Delta\Delta G$	64	—	7	77
ΔS	0	7	—	10
ΔT_m	0	77	10	—

Table 6: Compiled corpus after cleaning (80/20 split).

Task	Family (rows)	Train / test	WTs
K50	Tsuboyama (21,249)	17k / 4k	64
$\Delta\Delta G$	Tsuboyama (20,686) + FireProt (4,778)	20k / 5k	236
ΔS	SoluProtMut (16,067)	13k / 3k	23
ΔT_m	FireProt (2,740)	2k / 548	135
Total	65,520	—	—

A.2 ENCODING

Section 3 and Figure 1 state the design: mutation-centered windows with in-vocabulary `[SEP]` site marks. Mechanically, let p be the mutation index set. A 512-residue window is cropped about p. At each $i \in p$, processed high to low so later indices stay valid, the string becomes

$$\ldots\ X\ \texttt{[SEP]}\ \mathrm{WT}_i\ \texttt{[SEP]}\ \mathrm{mut}_i\ \texttt{[SEP]}\ Y\ \ldots$$

Residues are space-separated for ProtBERT. The tokenizer budget is 1024 tokens (within ProtBERT's position limit), a second mutation-centered crop enforces that budget if multi-site markup still overflows. MLM corrupts the marked string, not the unmarked wild type. Compile-time wild-type/mutant markup pairs never reach the model.

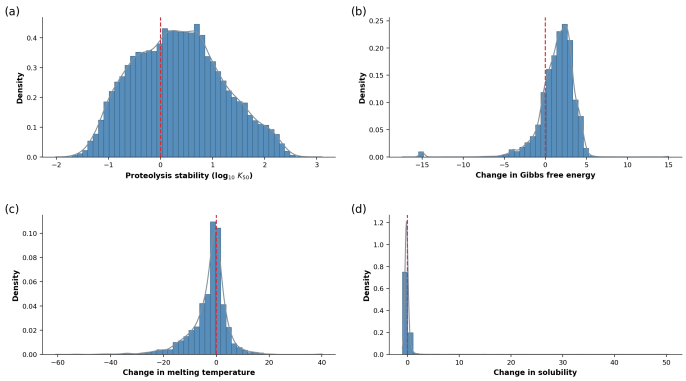

Figure 4: Compiled label distributions prior to z-scoring. Incompatible scales motivate the scale map in Section 3.

A.3 Splits and plates

Identity keys: Tsuboyama rows key on experimental campaign, FireProt and SoluProtMut use wild-type sequence. After cleaning, campaign-to-sequence and sequence-to-identifier are functions. Exact wild-type string overlap between held-out and pool identities is zero.

Study 1. Each assay property receives an $80/20$ row split stratified by campaign (split seed 42). The source property is also split $85/15$ into source-train / source-val. No sequence filter: overlap is intentional. k=200 is a pooled sample from target-train, drawn at run time with the training seed. Directions $\Delta\Delta G \rightarrow$K50 and K50$\rightarrow \Delta\Delta G$ are separate cells (n_{src}=1 each).

Study 2. For each target property, any other property sharing even one exact wild-type sequence with the target table is excluded (error, not warning). Target row-split $80/20$, surviving sources $85/15$. Disjointness is checked between source and (target-train $\cup$ target-test), source-train and target-train, and source-train and target-test. On current tables K50 keeps ΔS and ΔT_m (67 protein tasks), ΔS and ΔT_m each keep only K50 (63 tasks), $\Delta\Delta G$ is dropped. Scaler keys fit on train only. MLM remains target-wild-type-transductive.

Table 7: Study 3 per-protein plates: held-out Tsuboyama libraries with $\geq$250 distinct mutants (enough for k=200 support plus query). Together they supply 602 support rows, 4,108 query rows and six scored (protein, assay) groups. Support and query are mutant-grouped, so a $\Delta\Delta G$/K50 twin cannot straddle the split.

Library	Rows	Distinct mutants	Assays
CBPA2_HUMAN	2,195	1,117	$\Delta\Delta G$, K50
BCHB_CHLTE	1,566	783	$\Delta\Delta G$, K50
ILF3_HUMAN	949	483	$\Delta\Delta G$, K50
Total	4,710	2,383	—

Study 3. Identities are shuffled (seed 42), 15% (46 of 305) are held out of every labeled split. The remaining pool is row split $85/15$ (source-train has 254 identities, five singletons appear only in source val). A protein is eligible for a k=200 plate if it has $\geq$250 distinct mutants, three held-out Tsuboyama libraries meet that bar (Table 7). The pooled-plate fold instead draws one mutant-grouped plate across all 46 held-out identities ($\sim$4 shots each, 49 scored groups). Sequence overlap with the training pool is allowed, only protein identity is forbidden. On the current compile, $\Delta\Delta G$ source-train is an $85/15$ split of the full $\Delta\Delta G$ table.

B Methods, metrics and algorithms

Detail that Section 5 compresses: why the encoder and task unit are fixed, the operators each arm runs, how to read the metrics, and the hyperparameter defaults.

B.1 Encoder choice and task construction

Section 5 freezes one sequence encoder so every arm sees the same features. The practical reason is cacheability: one encoder pass serves every method and seed, and a richer featurisation could lift every arm together without explaining a gap between them.

Protein-level tasks are required for the same reason the main text gives, with the counts made explicit. One task per source assay leaves two tasks on the K50 fold and a single task on each of ΔS and ΔT_m, with a meta-batch of all tasks, a five-epoch floor and patience five halt after roughly ten outer Adam steps. Under protein-level tasks the K50 fold instead carries 67 source proteins with at least 8 labeled rows (ΔS and ΔT_m: 63 each), so every Study 2 fold is multi-task. Assay-level episodes would also split whole proteins across support and query while the deployed plate draws both from the same wild types, protein-level episodes remove that mismatch.

B.2 Algorithms

Companion to Section 5 and Algorithm 1. Defaults that instantiate these steps are in Appendix B.4.

From-scratch. AdamW fits the head on the target plate alone equation 4. A 20% slice of the k labels is held out of gradients and used only to choose the stopping epoch (floor 5, patience 5), the remaining 160 rows receive the weight updates. The encoder stays frozen. Scoring is a live forward pass on the query.

Sequential transfer. Train on all admitted source-train labels, then specialize on the same target plate under the same holdout rule equation 5. Because the encoder is frozen, the source phase embeds once and fits on a cached `[CLS]` matrix, the target plate is always embedded live.

Algorithm 2 Sequential transfer

1: Embed all source-train rows once (frozen encoder), cache `[CLS]` matrix
2: $\theta_0 \leftarrow$ AdamW fit of the head on the cached source matrix equation 5
3: $\theta^\star \leftarrow$ from-scratch specialization of θ_0 on the target plate (same holdout rule)
4: **return** $\theta^\star$

Shared inner operator. FOMAML and the random-initialization control share one SGD operator $U^{k_{\text{inner}}}$ equation 7. Defaults are k_{inner}=20 head-only steps at learning rate α, minibatch size 8, weight decay 0. At meta-train time the operator runs on an 8-row episodic support (each step full-batch), at target time it runs on the k=200 plate.

Algorithm 3 Inner operator $U^{k_{\text{inner}}}$

Require: weights θ, support $\mathcal{S}$, steps k_{inner}=20, α=10^{-3}
1: **for** $t = 1, \ldots, k_{\text{inner}}$ **do**
2: Sample minibatch $\mathcal{B} \subset \mathcal{S}$ (cycle if exhausted)
3: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{B})$ ▷ weight decay 0, head only
4: **end for**
5: **return** θ

First-order MAML (FOMAML). FOMAML seeks an initialization from which $U^{k_{\text{inner}}}$ on a new support reduces query error (Figure 3). Tasks are source proteins with $\geq$8 labeled rows, each episode is an $8/8$ support/query split inside one protein, matching how the target plate is drawn. The outer loop is first-order (`create_graph=False`): query gradients are taken w.r.t. the adapted weights and applied to θ by Adam (Kingma & Ba, 2015),

$$\theta \leftarrow \mathrm{Adam}\left(\theta,\ \frac{1}{n_{\text{ok}}} \sum_{i \in \text{ok}} \nabla_{\theta_i'} \mathcal{L}(\theta_i', \mathcal{Q}_i)\right). \tag{8}$$

Algorithm 4 FOMAML meta-training, one epoch

1: Sample meta-batch of source tasks (default: all n_{src})
2: **for** each task $\mathcal{T}_i$ **do**
3: Draw sequence-disjoint support $\mathcal{S}_i$ and query $\mathcal{Q}_i$ (size $8/8$)
4: $\theta_i' \leftarrow U^{k_{\text{inner}}}(\theta, \mathcal{S}_i)$
5: $g_i \leftarrow \nabla_{\theta_i'} \mathcal{L}(\theta_i', \mathcal{Q}_i)$ ▷ first-order, inner loop not differentiated
6: If g_i is finite, accumulate g_i on gradient buffers of θ and mark task successful
7: **end for**
8: Average accumulated gradients, abort if fewer than half the tasks succeeded, clip to norm 1, Adam step on θ

At target time, apply $U^{k_{\text{inner}}}$ once to the k=200 plate and score the query. Assay-level tasks are not used for claim cells (Appendix B.1).

Random-initialization control. Apply the same $U^{k_{\text{inner}}}$, seed and step count to a head that never saw Algorithm 4. The two arms differ only in whether the outer loop ran.

Episode and plate draws. Meta-episodes use an $8/8$ split inside one source protein. Target k is a pooled sample from target-train (Studies 1–2) or a per-protein / pooled held-out plate (Study 3), drawn with the training seed.

B.3 Metrics and how to read the tables

Primary definitions are in Table 1. Two quantities logged beside them but not used to rank methods are protein-/campaign-macro z-MSE and the paired seed gap,

$$\text{macro} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathcal{L}(\theta, \mathcal{Q}_p), \tag{9}$$

$$\Delta_s = z_{\text{method},s} - z_{\text{from-scratch},s}, \qquad s \in \{42, 43, 45, 46\}. \tag{10}$$

Original-unit MSE $\hat{y} = \hat{z}\,\sigma_c + \mu_c$ is also logged within one physical feature group. Δ_s is not a significance claim.

Reference predictors and campaign scalers. Studies 1–2 report z-MSE against a train-mean predictor that emits μ_c (equivalently $\hat{z}$=0), with (μ_c, σ_c) fixed from the target-training split (z-MSE 1.056 on the current Study 2 K50 build). Study 3 instead extends (μ_c, σ_c) from each plate's own support and compares against that support mean. Val rows with a missing campaign key are discarded, the query never fits a scaler. Exact duplicates were removed at compile, so no replicate noise floor is claimed.

Cautions. Studies are never averaged. From-scratch fits 160 of 200 labels while $U^{k_{\text{inner}}}$ uses all 200. Near-reference z-MSE is also reached by a near-constant predictor. On Study 2, FOMAML is compared to random-init, not to from-scratch. Study 3 plates are mixed-assay, only within-assay ρ is interpretable.

B.4 Hyperparameters

Fixed defaults from the released code (`protein_transfer/pipeline/paths.py`, `helpers.py`, `algorithms.py`, `trainers.py`). None was tuned on a target test split. Protocol procedures are in Appendix B.2.

Shared across labeled arms. The encoder is frozen ProtBERT, only the regression head is trained. Inputs are mutation-centered windows of 512 residues, tokenized to at most 1024 tokens with in-vocabulary `[SEP]` site marks. The head is a three-layer MLP d→256→128→1 with ReLU and dropout 0.1 at initialization, dropout is disabled during comparison runs so `[CLS]` is deterministic. Label budget k=200. Split seed 42 (one split), training seeds $\{42, 43\}$ by default and

$\{42, 43, 45, 46\}$ for the reported tables. Minibatch size 8 is not sharded across GPUs. AdamW early stopping holds out 20% of the plate ($\geq$5 rows) inside the k budget, with a five-epoch floor and patience 5, below the floor the monitor falls back to train loss. ProtBERT was chosen because the site markup needs an in-vocabulary token that can be spliced inline, ESM-2 has no equivalent, so a family swap would change encoding and encoder together. We therefore fix the encoder and report no cross-family control.

Quantity	Default	Notes
Encoder	ProtBERT, frozen	head only
Window / max tokens	512 / 1024	mutation-centered
Head	$d \to 256 \to 128 \to 1$	ReLU, dropout 0.1 at init, off at compare
k	200	plate budget
Batch	8	not multi-GPU
Seeds (default / reported)	$\{42, 43\}$ / $\{42, 43, 45, 46\}$	seed 44 absent from this batch
Split seed	42	one plate draw per training seed
Early-stop holdout / floor / patience	20% ($\geq$5) / 5 / 5	inside k, shared by AdamW and meta
Held-out eligibility (Study 3)	k plus query rows	per-protein plate

Domain-adaptive MLM. BERT MLM (Devlin et al., 2019) runs once before any labeled cell, on study-train `sep_highlight` strings (43,310 unique mutants, all 305 WT sequences, including the 46 Study 3 holdouts),

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \mathcal{M}} \log p_\theta(x_i \mid \tilde{x}). \tag{11}$$

Special tokens (site `[SEP]` and leading `[CLS]`) are never masked.

Quantity	Default	Notes
Epochs / lr / weight decay	5 / 5×10^{-5} / 0.01	AdamW + OneCycle (Smith & Topin, 2019)
Corruption	p=0.15, then 80/10/10	BERT schedule, equation 11
Mask specials	no	`tokenizer.all_special_ids`

From-scratch. Fits only the plate equation 4. Effective gradient budget is 160 of 200 labels, the early-stop slice never enters the loss. Optimizer is AdamW with OneCycle (10% warmup).

Quantity	Default	Notes
Max epochs / min / patience	50 / 5 / 5	monitor = held-out plate MSE
Head lr / weight decay	5×10^{-4} / 0.01	OneCycle, 10% warmup
Scheduler	OneCycleLR (Smith & Topin, 2019)	`pct_start` = 0.1
Freeze encoder	true	comparison default

Sequential transfer. Source phase then plate specialization (Algorithm 2). With the encoder frozen, source `[CLS]` vectors are cached once, target specialization reuses the from-scratch AdamW schedule and holdout rule. A warm-start control shuffles source labels (`--source-labels shuffled`).

Quantity	Default	Notes
Source phase	cached `[CLS]`	frozen encoder $\Rightarrow$ deterministic
Source / target AdamW	same as from-scratch	50/5/5, lr 5×10^{-4}, wd 0.01
Source labels	real	optional shuffle control

FOMAML and random-initialization control. Both arms call the same $U^{k_{\text{inner}}}$ (Algorithm 3) with identical α, step count and seed. FOMAML alone runs the outer loop (Algorithm 4), the control starts from a fresh head. Meta-episodes are 8/8 inside one source protein, the meta-batch is all n_{src} tasks (one outer Adam step per meta-epoch). Inner Jacobian is detached. We do not sweep α or k_{inner}.

Quantity	Default	Notes
Task unit	source protein	$\geq$8 labeled rows, 67 on K50 fold
Episode support / query	8 / 8	inside one protein
Meta epochs / min / patience	100 / 5 / 5	monitor = source-val query MSE
Meta-batch	all n_{src}	one outer Adam step / epoch
Inner steps k_{inner} / lr α / wd	20 / 10^{-3} / 0	SGD, Finn/Nichol
Outer lr β / grad clip	10^{-3} / 1.0	Adam on θ
First-order	yes	`create_graph=False`
Target adaptation	$U^{k_{\mathrm{inner}}}$ on k=200	all 200 labels, no AdamW holdout
Abort rule	$<$ half tasks finite	skip outer step that epoch

At meta-train time, support size equals batch size, so each of the 20 inner steps is a full-batch update on the same 8 examples, at target time the operator takes 20 minibatch steps over 200 rows. That mismatch is intentional and fixed: we report the operator as deployed, not a retuned one.

C FULL RESULTS

Table 8 gives per-seed ranking and Table 9 per-seed calibration for every cell the protocol admits. Means here are the values the main-text tables report. Seed 44 is not in the batch.

Table 8: Per-seed within-(protein, assay) Spearman ρ at k=200. Study 1 admits only from-scratch and transfer. Study 2 from-scratch matches Study 1 $\Delta\Delta G\rightarrow$K50 (same plate).

Study	Fold	Method	42	43	45	46	Mean
Study 1	$\Delta\Delta G\rightarrow$K50	from-scratch	-0.012	-0.028	$+0.022$	-0.013	-0.008 ± 0.021
		transfer	$+0.461$	$+0.478$	$+0.476$	$+0.485$	$+0.475 \pm 0.010$
Study 1	K50$\rightarrow \Delta\Delta G$	from-scratch	$+0.058$	$+0.002$	-0.007	-0.043	$+0.003 \pm 0.042$
		transfer	$+0.308$	$+0.322$	$+0.307$	$+0.325$	$+0.316 \pm 0.009$
Study 2	K50	from-scratch	-0.012	-0.028	$+0.022$	-0.013	-0.008 ± 0.021
		transfer	$+0.019$	$+0.034$	$+0.072$	-0.009	0.029 ± 0.034
		FOMAML	$+0.024$	$+0.021$	$+0.009$	-0.008	0.011 ± 0.015
		random-init	-0.015	$+0.014$	$+0.001$	-0.085	-0.021 ± 0.044
Study 3	per-protein	from-scratch	$+0.316$	$+0.302$	$+0.214$	$+0.216$	$+0.262 \pm 0.055$
		transfer	$+0.354$	$+0.317$	$+0.256$	$+0.323$	$+0.312 \pm 0.041$
		FOMAML	$+0.031$	$+0.022$	$+0.019$	$+0.100$	$+0.043 \pm 0.038$
		random-init	-0.017	$+0.029$	-0.073	-0.000	-0.015 ± 0.043
Study 3	pooled plate	from-scratch	-0.056	$+0.037$	$+0.001$	$+0.124$	$+0.027 \pm 0.076$
		transfer	$+0.029$	$+0.046$	$+0.067$	$+0.086$	$+0.057 \pm 0.025$
		FOMAML	$+0.022$	-0.046	$+0.037$	-0.012	0.000 ± 0.037
		random-init	-0.078	$+0.012$	$+0.038$	$+0.020$	-0.002 ± 0.052

Table 9: Per-seed pooled z-MSE at k=200, same cells and conventions as Table 8. Reference predictors: 1.056 on the K50 folds, 1.447 on K50$\rightarrow \Delta\Delta G$, and a per-seed support mean on Study 3 (1.302 ± 0.269 per-protein, 1.438 ± 0.153 pooled plate on seeds $\{42, 43\}$).

Study	Fold	Method	42	43	45	46	Mean
Study 1	$\Delta\Delta G\rightarrow$K50	from-scratch	1.063	1.058	1.228	1.068	1.104 ± 0.083
		transfer	0.771	0.790	0.821	0.767	0.787 ± 0.025
Study 1	K50$\rightarrow \Delta\Delta G$	from-scratch	1.456	1.446	1.528	1.448	1.470 ± 0.039
		transfer	1.507	1.471	1.434	1.431	1.461 ± 0.036
Study 2	K50	from-scratch	1.063	1.058	1.228	1.068	1.104 ± 0.083
		transfer	1.158	1.116	1.246	1.061	1.145 ± 0.078
		FOMAML	1.069	1.062	1.061	1.097	1.072 ± 0.017
		random-init	1.064	1.057	1.056	1.058	1.059 ± 0.004
Study 3	per-protein	from-scratch	1.099	1.414	1.474	0.942	1.232 ± 0.254
		transfer	1.093	1.312	1.430	0.899	1.183 ± 0.236
		FOMAML	1.191	1.423	1.531	0.972	1.279 ± 0.249
		random-init	1.198	1.499	1.535	0.973	1.301 ± 0.266
Study 3	pooled plate	from-scratch	1.299	1.529	2.266	2.019	1.778 ± 0.443
		transfer	1.287	1.739	2.342	2.120	1.872 ± 0.463
		FOMAML	1.309	1.519	2.208	1.998	1.759 ± 0.416
		random-init	1.293	1.529	2.211	2.020	1.763 ± 0.425

D REPRODUCING THE EXPERIMENTS, AND COMPUTE

D.1 REPRODUCING THE EXPERIMENTS

From the repository root after `pip install -r requirements.txt`, one command rebuilds compile, splits, encoder, labeled grid, aggregates, and diagnostics for the reported seeds:

```
python scripts/run_all.py --seeds 42,43,45,46
```

Study 2's sequential-transfer cell is not on the default queue; add it with `--stages grid --study unseen_protein_fewshot --fold k50 --method transfer --seeds 42,43,45,46`. Method ids: `finetune` (from-scratch), `transfer`, `fomaml`, `random_init`. Each labeled cell stores a run fingerprint and is skipped unless that fingerprint changes.

Command	Paper output
`python scripts/build_dataset.py`	Compiled dataset, overlap ledger, and `identity_audit.json` (Appendix A.1).
`python scripts/preprocess_data.py`	Three study directory trees (Appendix A.3).
`python scripts/run_method.py`	One (method × study × fold × seed × k) cell, `orchestrate_runs.py` loops over it.
`python scripts/pretrain_foundation_model.py`	Domain-adaptive MLM encoder shared by all labeled runs.
`python scripts/orchestrate_runs.py`	Default frozen k=200 queue, pass `--seeds 42,43,45,46` for the reported tables.
`python scripts/evaluate_runs.py`	Per-study aggregates in `logs/eval/`, paired Δ.
`python scripts/generate_report.py`	Checklist at `logs/report/report.md`.
`python scripts/eval_rank_metrics.py`	Spearman correlations.
`python scripts/eval_homology.py`	Near-duplicate check on the held-out versus training pool.

`scripts/run_all.py` is the single entry point and is resumable by stage markers and per-cell fingerprints. Studies 1 and 2 use target-train campaign scalers, Study 3 uses support-only scalers on per-protein plates.

D.2 COMPUTE

Labeled training for this paper ran on four NVIDIA A800 40 GB GPUs, one independent (method, study, fold, seed) job per visible GPU. The AdamW minibatch of 8 is not distributed across GPUs. Cost is dominated by two arms: sequential transfer's source `[CLS]` cache and head fit, and FOMAML's discarded inner-loop clones (k_{inner} steps per episode that are not kept). From-scratch is cheap by comparison. Reported tables use seeds $\{42, 43, 45, 46\}$.