Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Does Residual Non-Parameterizability Predict When Learned Residual Control Helps? A Controlled Negative Benchmark on a Simulated Manipulator

Dikshant Dikshant

dikshant1103@gmail.com

Motilal Nehru National Institute of Technology https://orcid.org/0009-0006-9434-610X

Research Article

Keywords:

Posted Date: August 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10779392/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Does Residual Non-Parameterizability Predict When Learned Residual Control Helps? A Controlled Negative Benchmark on a Simulated Manipulator

Dikshant · dikshant1103@gmail.com

Abstract

Question. For a rigid manipulator the uncertainties most often studied are exactly linear in the parameters of the rigid-body model — a class of classical adaptive control handles with no training data — so it is natural to expect a learned residual to earn its place precisely when the residual leaves that span. We define ρ_{unexp}, the fraction of model-mismatch torque energy that no constant linear-in-parameters fit to the adaptive regressor can explain, and ask whether it predicts when learned residual control helps.

Method. A controlled comparison in which the learned and classical arms share one control-law implementation — they differ by a single boolean, the classical arm is bit-identical to a standard composite adaptive controller, and the learned arm's search space contains the classical controller as an interior point — with one tuning protocol, five seeds, six mismatch families, two architectures and five offline-data budgets. The independent unit is the **task**: 240 tasks each evaluated under five seeds, so the 1,200 task-seed evaluations are repeated measurements and every interval resamples whole tasks.

Result. The learned controllers won **33.8%** of task-level comparisons (95% task-clustered CI **[27.9%, 39.6%]**; Wilcoxon over task medians p = 6.0e-05, sign test p = 5.4e-07) and lost on all five training seeds individually, despite an offline-data advantage of 216,000 samples. The result held at every tested data budget, for both architectures, and on an independently pre-registered ladder of seven residual mechanisms (0 of 21 cells positive; pooled win rate 21.8%). ρ_{unexp} did not predict benefit: its correlation was indistinguishable from zero (Pearson -0.034 per task, p = 0.60) and no fitted model outperformed a constant.

Conclusion. Within this benchmark the tested learned residual controllers were inferior to classical composite adaptive control, and residual non-parameterizability was not a reliable criterion for predicting when learning would help. Scope: one simulated 2-DOF manipulator, one controller formulation, one additive-torque residual interface, six mismatch families, one training protocol, two architectures and the data budgets tested. We make no claim beyond it, and in particular no claim that learned residual control does not work.

1. Introduction

Conclusions about learned control depend on how the comparison is set up. Kunapuli, Welde, Jayaraman and Kumar [kunapuli2025] showed for quadrotor tracking that once the objective, the tuning data and the feedforward information are made symmetric between a reinforcement-learning controller and a geometric controller, the published performance gaps shrink substantially. **That unequal comparisons can inflate or reverse apparent learning advantages has therefore already been reported; we adopt equalization as method and do not claim it as a finding.**

What remains open is prediction. For a rigid manipulator the uncertainties most often studied — link mass, inertia, centre of mass, payload, viscous and Coulomb friction — are exactly linear in the parameters of the rigid-body model, a class Slotine and Li's adaptive controller [slotine1987; slotine1989] handles with a

Lyapunov argument and no training data. It is natural to hypothesise that a learned residual becomes useful precisely when the residual leaves that span. We define ρ_{unexp}, the fraction of model-mismatch torque energy that no constant linear-in-parameters fit to the regressor can explain, build a benchmark in which the learned and classical arms share one control-law implementation and one tuning protocol, and test that hypothesis. **It was not supported.**

Contributions.

1. A controlled benchmark in which the two arms share a **bit-identical** control law and one tuning protocol, with a full leakage audit, and in which the learned arm's search space contains the classical controller as an interior point.
2. A **controlled falsification of a proposed criterion**: residual explainability did not predict learning benefit, with no threshold and no monotone trend.
3. A **bounded negative result** — under that protocol the learned residual did not outperform classical composite adaptive control, across data budgets and two architectures — with statistics computed at the level at which the experiment is actually independent, the task, using rank-based paired endpoints.
4. A demonstration that **ratio-based improvement metrics are mechanically inflated** by reference-controller degradation, with the algebraic identity made explicit and the corresponding rank statistics shown to be null.
5. A **mechanistic analysis** of why several apparently promising readings of residual-learning benefit do not survive controlled testing, and measurements showing that residual explainability itself depends strongly on regressor choice and excitation.

We do not propose a controller.

2. Related work

Adaptive robot control. [slotine1987] gives asymptotic tracking for uncertainty linear in the rigid-body parameters; the composite form [slotine1989] adds a prediction-error term. [guo2022] surveys composite adaptation and learning; we could not obtain its full text (Section 5).

Learned residual and semiparametric dynamics. Rigid-body models augmented with a learned correction date to [nguyentuong2010], developed through incremental and derivative-free semiparametric estimators [camoriano2016; romeres2020; giacomuzzo2024] and physics-structured networks [lutter2019; sutanto2020; gupta2020]. Learned residuals with online adaptation and stability guarantees include [shi2019; oconnell2022] and control-oriented meta-learning [richards2021; shi2021]. **Our learned arm is the frozen-basis, adapted-output-weights recipe from this line, used unmodified.**

Fair and equalized comparison. [kunapuli2025] is the direct ancestor of Section 4.1.1; **that result is theirs and we do not claim it.** [cirrincione2026a] baselines a learned correction against Slotine-Li on a 2-DOF arm and notes qualitatively that gains depend on whether the unmodelled dynamics fall outside the regressor span, without equalizing tuning budgets or defining a metric.

Parametric / non-parametric separation. [smith2019] identifies the non-stationarity that arises when both components adapt simultaneously and proposes a consistency transform; [kon2022] penalises the learned component in the subspace of the physics model by orthogonal projection, obtaining uniquely identifiable coefficients. **The correction used diagnostically in Section 4.5 is theirs; we claim no novelty for it.**

Model mismatch and excitation. Sensitivity of identification to excitation and unmodelled dynamics is long-standing [gautier1992; armstrong1988]; we could not obtain the latter's full text, which bounds what we claim in Section 4.4.

3. Benchmark and protocol

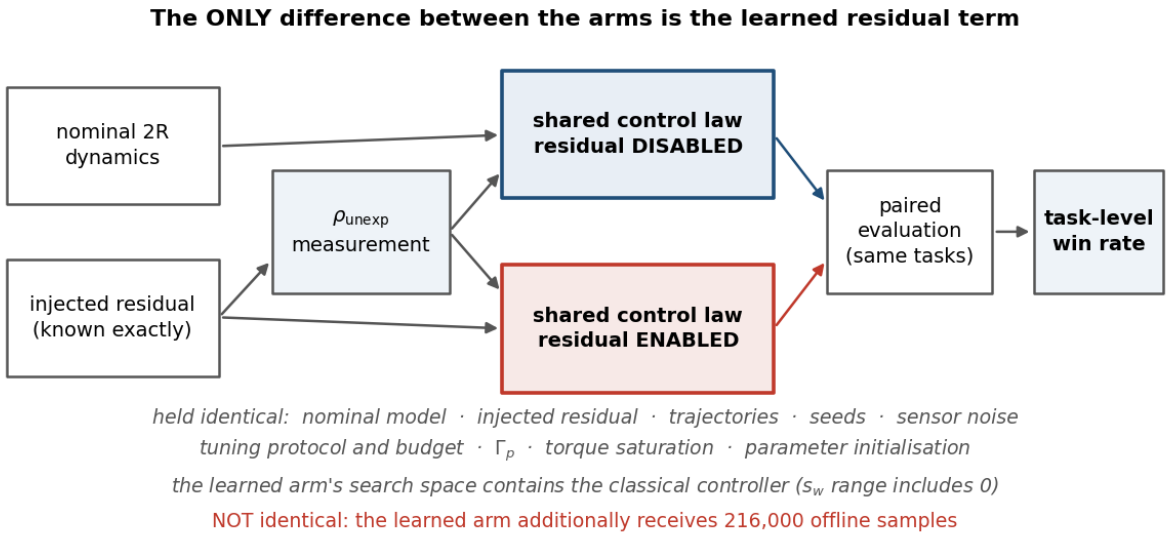


Figure 1: Controlled benchmark protocol. A physical residual is injected into the nominal 2R dynamics with its value known analytically; the same residual is measured by ρ_{unexp} and presented to two control arms that share one implementation. Nominal model, residual, trajectory, seeds, noise, tuning protocol, prediction-error gain, saturation and initialisation are identical; the arms differ only in whether the learned residual term is enabled. Evaluation is paired per task. This diagram describes the evaluation methodology; it is not a network architecture, and it does not establish any performance result.

Plant and mismatch. MuJoCo 3.11.0, planar 2R arm, RK4 at 1 kHz, control at 500 Hz, all engine damping and friction zeroed; every mismatch is injected through `qfrc_applied` by a module carrying its own hidden state, so the injected torque is analytically known. The analytical model, the skew-symmetry identity the adaptive proof relies on, and the engine's spurious-torque floor are validated numerically (supplement). Six mismatch families are used: inertial/payload/friction (exactly linear in the parameters); Stribeck excess; gear ripple with load-coupled friction; LuGre bristle dynamics with backlash and actuator lag; a drifting variant; and a synthetic non-parameterizable field used only to populate the high-ρ end. Each has a test asserting its analytical limit (supplement). **Tasks:** 240 test tasks drawn once over six families, three parameter shells, four severities and four trajectory families, each evaluated at five training seeds — 1,200 paired evaluations, sensor noise 2e-4 rad, six-second episodes.

Residual explainability metric. With $r(t)$ the model-mismatch torque and Y the Slotine-Li regressor along the reference trajectory,

$$\theta^* = arg\,min_\theta \sum_t \|r(t) - Y(t)\theta\|^2$$

$$\rho_{unexp} = \frac{\sum_t \|r(t) - Y(t)\theta^*\|^2}{\sum_t \|r(t)\|^2} \in [0,1]$$

$\rho_{unexp} = 0$ means the residual lies entirely inside the span, 1 entirely orthogonal to it. **This is the unexplained fraction, the complement of an explained-variance convention**, and one constant θ is fitted over the whole trajectory — the relevant question for an adaptive law converging to a single parameter estimate. It is a property of the triple (residual, regressor, trajectory distribution), not of a physical disturbance; the last two dependencies are quantified in Section 4.4. It was verified against constructed cases with known answers, rank-deficient regressors are flagged rather than silently pseudo-inverted, and the linear-in-parameters family measures exactly 0.0000 across all 576 configurations. θ^* is fitted in-sample; an even/odd refit changes it by less than 0.001 in every case tested, so in-sample optimism is negligible at 12,000 equations and nine parameters.

Controllers. Every arm runs one implementation:

$$v = \dot{q}_{des} - \Lambda e, \quad \dot{v} = \ddot{q}_{des} - \Lambda \dot{e}, \quad s = \dot{e} + \Lambda e$$

$$\tau = Y(q,\dot{q},v,\dot{v})\hat{a} - K_D s \; [+ \Phi_\theta(q,\dot{q})\hat{w}]$$

$$\dot{\hat{a}} = -\Gamma Y^T s - \Gamma_p Y_f^T \left(Y_f \hat{a} - \tau_f\right)$$

$$\dot{\hat{w}} = -\Gamma_w \Phi^T s - \sigma(\hat{w} - \hat{w}_0)$$

The bracketed term is present iff `use_residual` is true. $\hat{a}$ starts at the nominal parameters in every arm and is projected onto a physically admissible box; saturation, filtering, initialisation and timesteps are identical. Both arms receive 200 TPE trials on the same 16 tuning tasks with the same objective, with a held-out 16-task validation set measuring the tune-to-validation gap; the learned arm's search space has two extra dimensions that exist only because a residual exists, and the trial budget is equal regardless. Network-training, gain-tuning, tuning-validation and test task sets have pairwise empty intersections, verified programmatically, and the test set contains mismatch and trajectory families that never appear in training. Architecture A is a recurrent (GRU) residual basis, architecture B a parameter-matched feed-forward MLP; both adapt only their output weights online over a frozen basis.

What is held equal. The residual is the only experimental variable:

held equal	how
control law	one implementation; arms differ by a single boolean. Residual off is **bit-identical** to composite adaptive control (max abs torque difference 0.000e+00 over 4,000 steps)
prediction-error adaptation	Γ_p = 0.30, a protocol constant in **both** arms, never tuned per arm

tuning	one optimiser, objective, trial budget and task distribution
search space	S_w range **includes zero**, so the learned arm's space strictly contains the classical controller
evaluation	paired on identical plant realisations; disjoint train/validation/test seed offsets
seeds	five training seeds, all reported
not equal	the learned arm additionally receives 216,000 offline samples. The asymmetry **favours the learned arm** and is stated wherever the result is quoted

Metric and statistical unit. The primary metric is end-effector Cartesian RMSE; learning benefit is the paired log-ratio

$$L = log_{10}\left(\frac{RMSE_{classical}}{RMSE_{learned}}\right)$$

L > 0 means learning helps.

Raw-error ratios are avoided because the denominator approaches zero on the in-span family. Every comparison is paired on the identical mismatch draw, trajectory and noise realisation, and a label-permutation control on the paired pipeline returns significance in 4.5% of permutations against a nominal 5%. **Each task is evaluated under several training seeds, so evaluations are clustered within tasks and are not independent.** Unless stated otherwise all confidence intervals are percentile bootstraps that **resample whole tasks** (10,000 resamples, fixed seed), and any test requiring independent observations is run on task-collapsed data (one median per task). Rank-based paired statistics — win rate, Wilcoxon and sign tests over task medians — are the pre-declared primary endpoints; the mean log-ratio is a secondary descriptive quantity, for the reason established in Section 4.1.1. Corrected values are regenerated by `scripts/fa1_clustered_stats.py`.

Multiplicity. Several metrics and analyses were examined over the same evaluations, and p-values for different metrics on one comparison are **not independent confirmations** — the Wilcoxon and sign tests are alternative summaries of the same paired data as the win rate. We designate the task-level win rate as the single primary endpoint, report the rest as descriptive support, and apply Holm correction only within families of comparisons where several arms are tested against one reference. No p-value here survives a correction for the full set of analyses the project performed.

4. Results

4.1 Primary negative result

Unit of analysis. Each of 240 tasks is evaluated under five training seeds, so the 1,200 evaluations are not independent. Every interval below resamples whole **tasks**; where a test requires independent observations, each task is collapsed to its median over seeds. An earlier version of this section bootstrapped rows, which produced intervals almost exactly half the correct width.

statistic	rows (N = 1200)	task level (N = 240)
learning wins	32.9%	**33.8%**
95% task-clustered CI	[27.6%, 38.3%]	**[27.9%, 39.6%]**
Wilcoxon (task medians)	—	**p = 6.0e-05**
sign test (task medians)	—	**p = 5.4e-07**

mean L	-0.0784, CI [-0.158, -0.001]	-0.0450, CI **[-0.127, +0.035]**
median L	-0.0954	-0.0897

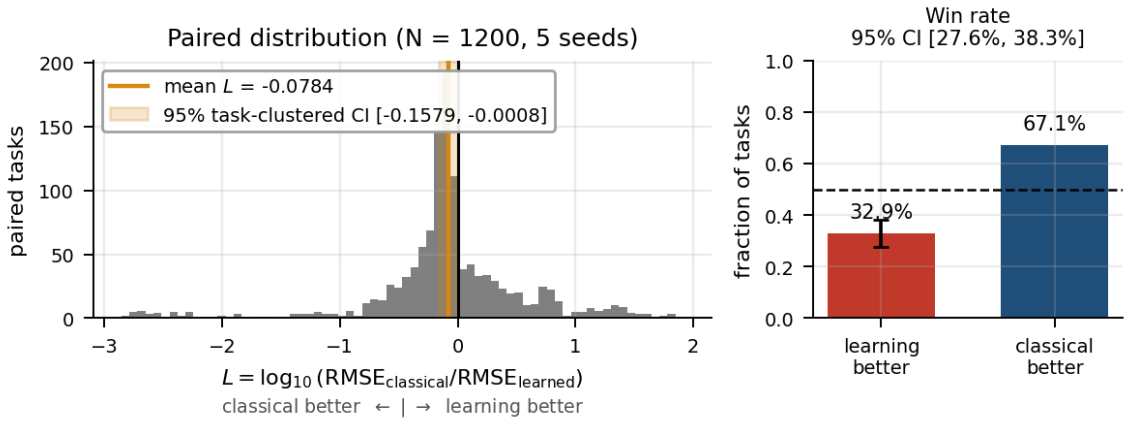


Figure 2: Primary paired result. Distribution of the paired log-ratio L over N = 1200 evaluations (240 tasks x 5 training seeds); right panel gives the win rate with its 95% task-clustered interval. The shaded band is the task-clustered CI on mean L, [-0.1579, -0.0008]; resampling rows instead of tasks would halve it. Learning wins 32.9% of evaluations and 33.8% of tasks. Negative L means the classical arm tracked better. Axes are untruncated and the full distribution including both tails is shown. This establishes the outcome for this benchmark and these conditions only; it does not establish anything about learned control in general.

The rank-based result is robust; the magnitude is not. The win rate excludes 50% by a wide margin under either unit, the Wilcoxon and sign tests over task medians are highly significant, and the direction holds in **all five** seeds individually (per-seed win rates 30.4%-35.8%, mean L -0.194 to -0.024). The mean log-ratio on the independent unit is -0.045 with a task-clustered interval **[-0.127, +0.035] that includes zero** (paired t over 240 task means, p = 0.27), so we report the rank statistics as primary and treat the size of the deficit as undetermined — consistent with Section 4.1.1, which shows independently that log-ratio magnitudes are the wrong instrument here.

Within this benchmark, the tested learned controllers were inferior to the classical composite adaptive controller. The claim is scoped to: a simulated manipulator; the tested controller formulation; the tested additive-torque residual interface; the six mismatch families evaluated; the tested training protocol; the two architectures tested; and the offline-data budgets tested. It is not a claim that learning is inferior to adaptive control, nor that learned residual control does not work, and it is not a claim of equivalence — the task-level win-rate interval lies entirely below 50%, which is evidence of inferiority within this benchmark rather than of a null.

A focused check on the exactly-in-span family, where there is nothing outside the span to learn, gives the same direction: with each arm at its own tuned optimum the classical arm is ahead by 1.18x (95% task-clustered CI [1.07, 1.30]) and the learned arm wins 19.0% of tasks (CI [10.3%, 28.7%]).

4.1.1 The improvement metric is inflated by baseline degradation

Holding the learned controller **completely unchanged**, we evaluated a family of classical configurations obtained by scaling the adaptation, proportional and damping gains around the validation-selected setting.

The ratio metric rises steadily as the classical arm degrades: Pearson +0.810 on one task distribution and +0.683 on an independent one. **This is algebra, not an empirical property of learned control.** With the learned arm fixed,

$$\underline{L}(c) = \underbrace{log_{10}\left(RMSE_{classical,c,t}\right)} - \underbrace{log_{10}\left(RMSE_{learned,t}\right)}_{constant\ in\ c}$$

so the slope of $\underline{L}$ against $\underline{log_{10}(RMSE_{classical})}$ is exactly one by construction; measured, **+1.0000**. Slopes fitted against a median-based abscissa (+0.537, +0.761) differ from one only because the mean of logarithms is not the logarithm of the median.

The decisive check is a statistic with no such algebraic dependence on the reference's own error level: across the same configurations the **win rate** shows no relationship with baseline quality (Pearson +0.036, p = 0.90), winning 58.3% of tasks against the most accurate classical configuration and 62.2% against the least accurate.

We deliberately do **not** quote a rank effect size alongside the win rate: Cliff's delta of the paired log-ratio against zero equals $2 \times (win\ rate) - 1$ identically (0.3292 and -0.3417 on the primary comparison), so reporting both would present one statistic twice. Where a separate effect size is wanted, the matched-pairs rank-biserial correlation (-0.298) is the appropriate quantity.

The corresponding figure, showing both replications and the rank-based null side by side, is Figure S4 in the supplement. What this supports is a caution about measurement, not a finding about learning:

> *Log-ratio improvement metrics are mechanically inflated when the reference controller is degraded, by an amount predictable from the reference's own error. Improvement ratios are therefore not comparable across studies with differently configured baselines, and rank-based statistics should be reported alongside them.*

We previously framed this relationship as an empirical dose-response. **That framing was wrong and is withdrawn**; the algebraic identity above accounts for it.

4.2 Offline-data sensitivity

Nested task subsets, three seeds, 120 fixed test tasks. Mean L was -0.1122, -0.2307, -0.1738, -0.1153 and -0.0328 at 0%, 10%, 25%, 50% and 100% of the offline data (0 to 216,000 samples); every budget lies below zero.

Learning did not outperform the classical baseline at any tested offline-data budget up to 216,000 samples. The deficit shrinks monotonically above the smallest budget; we do not extrapolate beyond the range tested, and we make no claim about what a larger budget would give.

One observation is worth recording cautiously: the smallest non-zero budget performed worse than no offline data at all. A small, inaccurately fitted prior appears to be worse than starting from zero and adapting online. This rests on a single crossing at three seeds and should not be generalised.

4.3 Residual-family robustness

4.3.1 Mismatch-family dependence within the primary matrix

Per-family intervals are task-clustered; n is the number of independent tasks.

family	n tasks	median rho	mean L	95% CI	win %
inertial/payload/friction	54	0.0000	-0.0648	[-0.133, +0.025]	19.6
+ Stribeck excess	43	0.0154	+0.0084	[-0.081, +0.117]	28.4
+ ripple, load-coupled	37	0.1281	**+0.2739**	**[+0.120, +0.438]**	**52.4**
+ drift	42	0.2100	**-0.2924**	**[-0.569, -0.035]**	32.9
+ LuGre, backlash, lag	30	0.2976	-0.2059	[-0.403, +0.007]	36.0
+ synthetic field	34	0.3512	-0.2159	[-0.491, +0.041]	35.9

Under task-clustered intervals, learning significantly helped on one family (ripple with load-coupled friction) **and significantly hurt on one** (drift); the remaining four span zero. An earlier row-bootstrapped version reported four significant negatives, three of which do not survive correct clustering — the corrected picture is weaker. What survives is the ordering problem: the family where learning helps has median ρ_{unexp} = 0.128, *lower* than three families where it does not, and its range overlaps theirs almost entirely. Learning benefit here was effect-dependent and not accounted for by ρ_{unexp} alone.

4.3.2 An independently pre-registered ladder of residual mechanisms

To test whether the outcome depends on *which* residuals were chosen, a second benchmark varied residual **mechanism** at fixed magnitude — seven mechanisms spanning linear-in- parameters, smooth memoryless, configuration-periodic, backlash, actuator lag, correlated hidden state and work-coupled drift, each calibrated to about 0.30 N·m of non-parameterizable torque, at three severities — with protocol and qualifying conditions frozen before any test task ran. Over 1,512 paired evaluations **0 of 21 cells** qualified as a positive regime under seven pre-declared conditions; pooled win rate **21.8%**, pooled mean L -0.312, task-clustered CI [-0.363, -0.263], and ρ_{unexp} again null against win rate (Spearman +0.166, p = 0.47). The only axis that tracked the outcome was **severity — magnitude, not structure**: win rate was strictly higher at severity 4 than severity 1 in 7 of 7 families (sign test p = 0.008) while every structural axis was null, and the family the original hypothesis most favoured (a correlated hidden state, ρ_{unexp} = 0.56) was the **worst** in the study. Per-cell detail is in the supplement.

4.3.3 Second architecture and cost

Architecture B (feed-forward MLP) lost against every classical configuration tested, by 0.62 to 1.12 log10. **The negative result was not unique to the primary learned architecture in the tested comparison, although the secondary architecture was not independently optimized to the same extent** — it reused architecture A's gains, which disadvantages it.

Per-step cost, parameter counts and training time are reported in the supplement.

4.4 Falsification of ρ_unexp as a predictor

Correlations are computed on the independent unit — one point per task, N = 240 — because the 1,200 evaluations are five repeats of 240 tasks:

per-task correlation	value	p		5-fold CV model	CV MSE
Pearson	-0.0342	0.598		constant (no relationship)	0.44754

Spearman	-0.0438	0.499		linear	0.44778
Kendall	-0.0293	0.497		quadratic	0.44844
				isotonic	0.44879
				segmented	0.44657

All three are near zero, none is close to significance, and each point estimate carries the sign opposite to the hypothesis. (Over rows the Pearson coefficient is -0.0519 with a nominal p = 0.073; that p-value is anticonservative for the same clustering reason as in Section 4.1, so the task-level value is the one we report.)

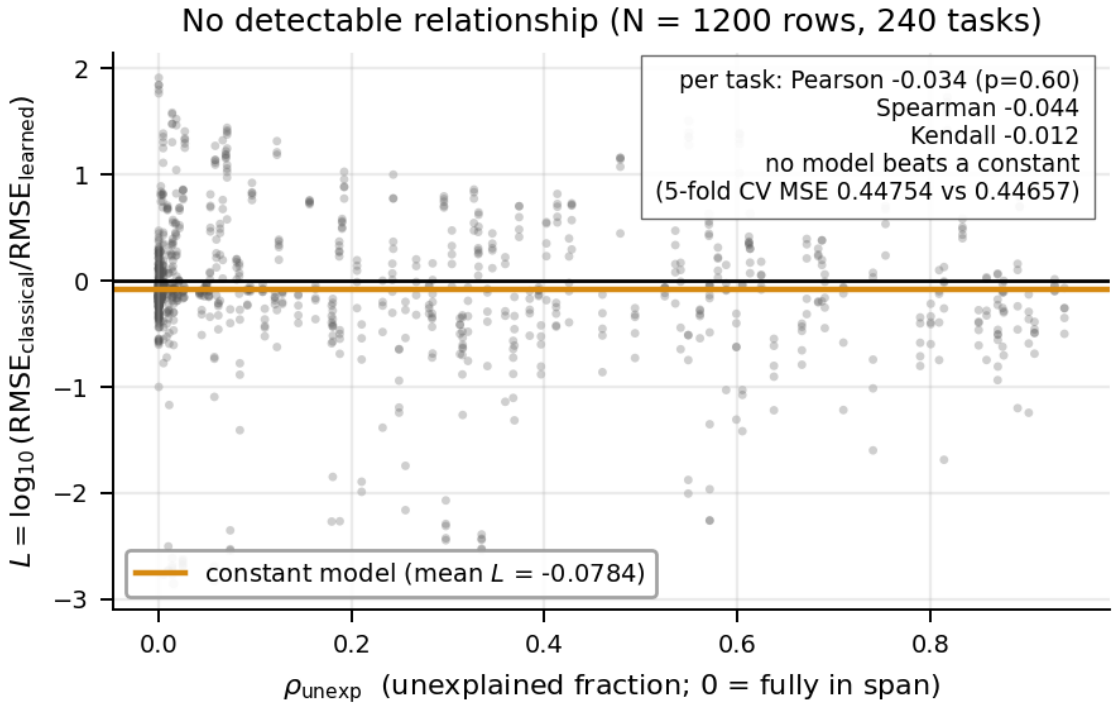


Figure 3: The tested hypothesis, and its rejection. Learning benefit against ρ_{unexp} for all N = 1200 evaluations (240 tasks). Correlations are computed per task: Pearson -0.034 (p = 0.60), Spearman -0.044, Kendall -0.029; 5-fold CV mean squared error is 0.44754 for a constant model against 0.44657 for the best of five candidates. Only the constant model is drawn: no trend line, no threshold and no segmented fit are shown, because none is supported. This figure records a hypothesis that was tested and rejected; it does not show that ρ_{unexp} is uninformative about residual geometry.

The five candidate models lie within 0.5% of one another; **no model improves on a constant.** We had hypothesised a threshold in ρ_{unexp} above which learning would overtake the classical controller. **That hypothesis is withdrawn: the data support no threshold, no monotone relationship, and no relationship.** Under a cluster-robust regression with mismatch family as a factor ρ_{unexp} is not significant (p = 0.193, partial R^2 0.005) while residual magnitude (p = 0.001, partial R^2 0.021) and severity (p < 0.001, partial R^2 0.035) are, both with negative coefficients; the full model reaches R^2 = 0.163, so most variance remains unexplained.

ρ_{unexp} is relative to model class and excitation.** Varying only the regressor across six bases moved measured ρ_{unexp} by up to 0.888 (gear ripple falls from 0.848 to 0.0000 once two joint-angle harmonics per joint are

added), and varying only the excitation across four trajectory classes moved it by up to 0.909 — in both cases by margins exceeding the differences between the mechanisms the metric was meant to distinguish. The effect ordering was stable among moderate-to-rich excitation (Spearman 0.89-0.94) and degraded at low excitation (0.49-0.54). This is a property of the metric, not the controllers, and one reason a scalar ρ_{unexp} was never going to suffice; the full sweep is in the supplement.

4.5 Mechanistic analysis: why a scalar criterion could not have worked

A factorial over offline initialization {none, random, trained} x online residual adaptation {off, on}, with **every arm sharing one set of gains** so the gain contribution is exactly zero, isolates what the learned component contributes. Two results bear on Section 4.4, and the full diagnostics are in the supplement. **First**, the conventional target satisfies $\tau - ID_{nom} = Y\delta a - \tau_{nonlip}$, whose first term lies in the span the adaptive law is concurrently identifying; where ρ_{unexp} = 0 the correct residual is identically zero yet the target has RMS 0.70 N·m and the arm loses (31.7% win), and disabling the parametric adaptation flips that head's effect to 100.0% win (p = 0.001). This projection is **not new** — it is [kon2022]'s construction, with the coupling identified by [smith2019] — and correcting the target makes the learned arm stop hurting without making it help: **no target, including one built from ground truth, beat the classical arm in the two non-trivial regimes. Second**, where ρ_{unexp} > 0 the benefit comes from *online* weight adaptation over a basis that need not be trained (an untrained random basis beat the mechanism-matched trained head on 7 of 7 transfer targets), while where ρ_{unexp} = 0 it is faster parameter convergence (warm-starting the classical arm won 83.3% of tasks, p = 0.0035, more than any residual arm). **Two mechanistically distinct effects wear the name "learning benefit"** at opposite ends of the axis, so no scalar criterion should be expected to order them.

4.6 7-DOF explainability replication (supporting)

On a 7-DOF chain with an exactly-constructed 42-column regressor, over 7 mechanisms x 4 trajectory families = 28 matched configurations, the ρ_{unexp} ordering transfers from 2 DOF:

correlation with 2-DOF ordering	value	95% CI
Pearson	+0.9623	[+0.9088, +0.9992]
Spearman	+0.9643	[+0.6981, +1.0000]
Kendall	+0.9048	[+0.5556, +1.0000]

The ordering transfers; the values do not — every mid-range mechanism measures lower at 7 DOF, because 42 columns span more than 9, which is itself an instance of the model-relative behaviour quantified in Section 4.4. **Only the residual-explainability ordering is claimed at 7 DOF.** A learned-versus-classical controller comparison was subsequently run at 7 DOF and is reported in the supplement, not here: its tuner drove the residual scale to ~0.0017, its log-ratio interval included zero, and its rank statistics showed no learning advantage, so it is bounded secondary evidence rather than a replication of Section 4.1.

5. Limitations

Scope. One simulated platform, 2 DOF for the primary controller experiment, no hardware validation. One additive joint-torque residual interface with a linear adaptive output layer; other interfaces (full learned dynamics, learned-Lyapunov-shielded residuals, residual RL) are untested and the mechanism results bound only this one. One principal learned architecture plus a secondary one not independently tuned. Six

benchmark-specific mismatch families plus seven ladder mechanisms. No claim beyond 216,000 offline samples. All conclusions are bounded by this benchmark.

The classical baseline is strong but not globally optimal. Tuning is a fixed finite budget on a validation split; the selected configuration ranked 17th of 36 on an independent split, and a re-centred search found a more accurate one. The negative result strengthened against that better baseline, so the limitation is conservative — but the baseline is not optimal.

Statistics. The rank direction is robust; the *magnitude* is not — the mean log-ratio on the independent unit has an interval including zero, so we claim direction only. An earlier draft bootstrapped rows rather than tasks, halving every primary interval and making three per-family results in Section 4.3.1 appear significant when they are not; all intervals here are task-clustered and the superseded values are in the repository. Multiple metrics were examined (Section 3) and no p-value survives a correction for the full set.

Attribution. The 7-DOF result validates the explainability ordering only; the 7-DOF controller comparison is supporting evidence in the supplement. The target correction of Section 4.5 is prior art [kon2022; smith2019], used diagnostically with no novelty claimed. Two sources could not be obtained in full text (Guo and Pan; Armstrong) and claims depending on them are stated as measurements.

6. Conclusion

We asked whether residual non-parameterizability predicts when a learned residual helps. Under a controlled benchmark with a bit-identical control law, a shared tuning protocol, and a learned search space containing the classical controller as an interior point, **within this benchmark the tested learned residual controllers were inferior to classical composite adaptive control**: 33.8% of task-level comparisons won (95% task-clustered CI [27.9%, 39.6%], Wilcoxon over task medians p = 6.0e-05, sign test p = 5.4e-07), losing on all five seeds individually despite an offline-data advantage of 216,000 samples. The size of the deficit is not well determined and we do not claim it. The claim is scoped to the platform, controller formulation, residual interface, mismatch families, training protocol, architectures and data budgets tested.

The criterion we proposed did not work. ρ_{unexp} showed no threshold, no monotone trend and no relationship with learning benefit, and the result reproduced on an independently pre-registered ladder of seven mechanisms. It behaves exactly as specified on synthetic cases and separates mechanisms cleanly by their relationship to a model class, yet does not predict what a controller will do with them — Section 4.5 supplies the reason, that two mechanistically distinct effects share the name and sit at opposite ends of the axis. Separately, the log-ratio improvement metric common in this literature is mechanically inflated by degradation of the reference controller, with slope exactly one by construction, while rank statistics over the same comparisons are unaffected; this is a caution about measurement, not a finding about learning, and it does not show baseline quality is unimportant.

These results motivate paired evaluation with rank-based reporting, clustering at the level at which tasks are actually independent, and explicit baseline equalization, before attributing performance gains to learned residuals. They are bounded by one simulated 2-DOF manipulator, one residual interface and two architectures, and we make no claim beyond that scope.

References

- **[armstrong1988]** Armstrong, B. *On finding exciting trajectories for identification experiments involving systems with non-linear dynamics.* (Full text not obtained — unverified full text.)
- **[camoriano2016]** Camoriano, R., Traversaro, S., Rosasco, L., Metta, G., Nori, F. *Incremental semiparametric inverse dynamics learning.* ICRA 2016. doi:10.1109/icra.2016.7487177
- **[cirrincione2026a]** Cirrincione, G., Fagiolini, A. *Learned Lyapunov Shielding for Adaptive Control.* arXiv:2605.06934, 2026.
- **[gautier1992]** Gautier, M., Khalil, W. *Exciting trajectories for the identification of base inertial parameters of robots.* (Full text not obtained — unverified full text.)
- **[giacomuzzo2024]** Giacomuzzo, G., Carli, R., Romeres, D., Dalla Libera, A. *A Black-Box Physics-Informed Estimator Based on Gaussian Process Regression for Robot Inverse Dynamics Identification.* IEEE Transactions on Robotics, 2024. doi:10.1109/tro.2024.3474851
- **[guo2022]** Guo, K., Pan, Y. *Composite adaptation and learning for robot control: A survey.* Annual Reviews in Control, 2022. (Full text not obtained — unverified full text.)
- **[gupta2020]** Gupta, J. K., Menda, K., Manchester, Z., Kochenderfer, M. J. *Structured Mechanical Models for Robot Learning and Control.* arXiv:2004.10301, 2020.
- **[kon2022]** Kon, J., Bruijnen, D., van de Wijdeven, J., Heertjes, M., Oomen, T. *Physics-Guided Neural Networks for Feedforward Control: An Orthogonal Projection-Based Approach.* American Control Conference (ACC), 2022. arXiv:2201.03308
- **[kunapuli2025]** Kunapuli, P., Welde, J., Jayaraman, D., Kumar, V. *Leveling the Playing Field: Carefully Comparing Classical and Learned Controllers for Quadrotor Trajectory Tracking.* Robotics: Science and Systems (RSS), 2025. arXiv:2506.17832.
- **[lutter2019]** Lutter, M., Ritter, C., Peters, J. *Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning.* ICLR 2019. arXiv:1907.04490
- **[nguyentuong2010]** Nguyen-Tuong, D., Peters, J. *Using model knowledge for learning inverse dynamics.* ICRA 2010. doi:10.1109/robot.2010.5509858
- **[oconnell2022]** O'Connell, M., Shi, G., Shi, X., Azizzadenesheli, K., Anandkumar, A., Yue, Y., Chung, S.-J. *Neural-Fly enables rapid learning for agile flight in strong winds.* Science Robotics 7(66), 2022. doi:10.1126/scirobotics.abm6597
- **[richards2021]** Richards, S. M., Azizan, N., Slotine, J.-J. E., Pavone, M. *Adaptive-Control-Oriented Meta-Learning for Nonlinear Systems.* Robotics: Science and Systems XVII, 2021. doi:10.15607/rss.2021.xvii.056
- **[romeres2020]** Romeres, D., Zorzi, M., Camoriano, R., Traversaro, S., Chiuso, A. *Derivative-Free Online Learning of Inverse Dynamics Models.* IEEE Transactions on Control Systems Technology 28(3), 2020. doi:10.1109/tcst.2019.2891222
- **[shi2019]** Shi, G., Shi, X., O'Connell, M., Yu, R., Azizzadenesheli, K., Anandkumar, A., Yue, Y., Chung, S.-J. *Neural Lander: Stable Drone Landing Control Using Learned Dynamics.* ICRA 2019. doi:10.1109/icra.2019.8794351
- **[shi2021]** Shi, G., Azizzadenesheli, K., O'Connell, M., Chung, S.-J., Yue, Y. *Meta-Adaptive Nonlinear Control: Theory and Algorithms.* NeurIPS 2021. arXiv:2106.06098
- **[slotine1987]** Slotine, J.-J. E., Li, W. *On the Adaptive Control of Robot Manipulators.* International Journal of Robotics Research 6(3), 49-59, 1987. doi:10.1177/027836498700600303
- **[slotine1989]** Slotine, J.-J. E., Li, W. *Composite Adaptive Control of Robot Manipulators.* Automatica 25(4), 509-519, 1989. doi:10.1016/0005-1098(89)90094-0
- **[smith2019]** Smith, J., Mistry, M. *Online Simultaneous Semi-Parametric Dynamics Model Learning.* arXiv:1910.04297, 2019.
- **[sutanto2020]** Sutanto, G., Wang, A., Lin, Y., Mukadam, M., Sukhatme, G., Rai, A., Meier, F. *Encoding Physical Constraints in Differentiable Newton-Euler Algorithm.* arXiv:2001.08861, 2020.