COMPACT CLOSED-LOOP RESPONSE REPRESENTATIONS FOR CONTROLLER–TASK MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reservoir computing enables robot controllers to exploit temporal information while training only an output readout. However, different trained reservoirs can interact differently with the same plant, making their suitability dependent on the requested motion. We introduce Closed-Loop Response Representations (CLRR) to compare trained nonlinear echo state network controllers through compact, reusable descriptions of their feedback behavior. For each plant, weighted principal component analysis learns a shared basis for local closed-loop response residuals from development controllers, without nonlinear performance labels. In the complex-response formulation, each finite-window multisine tracking task defines an analytical quadratic functional that evaluates the reconstructed response directly in its learned coordinates. After response characterization and encoding within a fixed operating regime, the same controller code supports new task queries without retraining or additional task-specific nonlinear rollouts. A distortion bound relates compression error to response-based task-cost error and controller-comparison margins. In near-equilibrium Cart-Pole simulations, eight coordinates achieve macro Kendall correlation 0.936 across 43 held-out controllers and 36 tasks, outperforming the tested compact spectral summaries. On a separate 16-controller test bank for a simulated seven-joint Panda arm, eight response coordinates plus seven equilibrium offsets achieve correlation 0.538, compared with 0.919 for the full response, yet select the best or second-best controller on all eight tasks. Full and sparse responses select the best throughout. These findings support compact response representations for task-dependent reservoir-controller comparison, while showing that useful controller choices can survive substantial losses in overall ranking fidelity. The quality of the representation therefore depends on the controller distinctions it preserves, not reconstruction accuracy alone.

1 INTRODUCTION

An important goal in learning-based robot control is to exploit temporal information while keeping controller training manageable. Reservoir computing (RC) offers a practical route. In an echo state network (ESN), a fixed nonlinear recurrent network transforms observations and reference signals into temporal features, while only an output readout is trained (Lukoševičius & Jaeger, 2009). This provides memory without optimizing the recurrent weights, but different trained realizations can produce different feedback behavior with the same robot (Ye et al., 2025a;b). As the requested motion changes, their relative suitability can change as well. Readout-only fitting therefore leaves the complementary problem of comparing the resulting controllers across tracking tasks without repeating a complete nonlinear evaluation for every query.

We study this problem for banks of already-trained nonlinear ESN controllers. A useful description should capture each controller's behavior with the plant, rather than its internal parameters alone, and remain reusable when the tracking objective changes. We ask whether their characterized closed-loop responses can be represented in shared, compact coordinates, with each task evaluated through a separate functional.

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.

Reusable behavioral descriptions also underlie policy representations for value generalization and performance prediction (Barreto et al., 2017; Tang et al., 2022; Scarpellini et al., 2024), and active offline policy selection combines offline estimates with limited interaction (Konyushkova et al., 2021). DeepONet and Fourier Neural Operators learn mappings between input and output functions (Lu et al., 2021; Li et al., 2021). These approaches provide context for reusing behavioral information; here, that information is the local response of a trained ESN in feedback with its plant.

A local frequency response describes how reference components change in amplitude and phase through the closed loop. Shared structure across reservoir controllers suggests a common low-dimensional basis. Projection-based model reduction exploits structure in dynamical behavior (Rowley, 2005; Benner et al., 2015); principal-component methods also support candidate assessment and frequency-domain representation (Wang, 2025; He et al., 2025). The challenge is to retain differences between controllers, not merely their common response. Small reconstruction errors can reverse close comparisons, while an imperfect ranking can still identify a good controller. Task-based and predict-then-optimize learning likewise distinguish prediction accuracy from decision quality (Donti et al., 2017; Elmachtoub & Grigas, 2022).

We introduce Closed-Loop Response Representations (CLRR) for trained reservoir controllers, pairing a reusable code with an analytical tracking-task functional. Weighted principal component analysis (PCA) fits the response basis to development ESNs without nonlinear performance labels (Jolliffe & Cadima, 2016). H-CLRR compresses complex tracking-response residuals; K-CLRR compresses a scalar response-error kernel. For H-CLRR, the shared decoder and multisine reference yield a quadratic tracking cost in the code. Held-out ESNs are projected into the fixed basis; new tasks change the functional, not the controller weights. Exactness concerns the reconstructed local response, with nonlinear tracking assessed separately.

Prior RC work characterizes reservoir dynamics and task dependence (Ozturk et al., 2007; Dambre et al., 2012; Hülser et al., 2023), including physical-adaptation representations (Giannetto et al., 2025). Here, the learned object is a shared representation of already-trained plant–reservoir responses. Appendix J distinguishes this compression problem from the related local-response ranking study (Anonymous authors, 2026).

On nonlinear Cart-Pole tracking, compact learned coordinates preserve strong ranking agreement and outperform the tested compact spectral summaries. A fresh simulated seven-joint arm study reveals a different trade-off: compact codes select high-performing controllers but lose substantially more ordering information than full and sparse responses. Analyses of retained dimension, development coverage, and controller-specific cost error examine which distinctions the representation loses.

Our main contributions are:

- A task-separated representation of trained nonlinear ESN controllers through compact coordinates of their local plant–reservoir responses, reused across tracking tasks with fixed controller weights.
- An analytical multisine tracking functional for the compressed responses, with a distortion bound relating reconstruction to response-based task-cost error and controller-comparison margins.
- A held-out study of ESN-controlled Cart-Pole and seven-joint Panda tracking that distinguishes response reconstruction, population ranking, and selected-controller quality.

2 Response characterization and task compatibility

Before learning a representation, we characterize the tracking behavior of each trained reservoir controller in feedback with the plant. The readout supplies actuation commands, while the resulting plant motion drives the recurrent state. Linearizing this coupled system about a joint equilibrium gives a local description of how small reference oscillations appear in the tracked outputs.

Consider a fixed plant with p tracked outputs and a bank of trained ESN controllers indexed by R. Let $H_R(\omega) \in \mathbb{C}^{p\times p}$ denote this local reference-to-output frequency response in matched coordinates. Each complex entry describes the amplitude ratio and phase shift from one reference channel to one output at angular frequency ω. We write $H_R(\omega)$ for $H_R(j\omega)$ in continuous time or $H_R(e^{j\omega h})$ for

a sampled loop with interval h. The local model includes the plant, reservoir state, trained readout, and any actuator or filter dynamics. The interface and operating regime remain fixed within each experiment, as do all controller weights.

Ideal tracking would reproduce each reference channel with unit gain, zero phase shift, and no cross-channel response. Its frequency-response matrix is the identity, so define the complex residual

$$E_R(\omega) = H_R(\omega) - I_p. \tag{1}$$

A task q specifies a reference excursion $\rho_q(t) = \sum_{j=1}^{m_q} a_{qj} \sin(\omega_{qj} t + \phi_{qj})$, with $a_{qj} \in \mathbb{R}^p$, output weighting $W_q \succeq 0$, evaluation window $\mathcal{T}_q$, and normalization $D_q > 0$. With an equilibrium output offset o_R, its harmonic error and response-based cost are

$$e_q^{\mathrm{resp}}(t; R) = o_R + \mathrm{Im}\left[\sum_{j=1}^{m_q} E_R(\omega_{qj}) a_{qj} e^{\mathrm{i}(\omega_{qj} t + \phi_{qj})}\right],$$

$$J_q^{\mathrm{resp}}(R) = \frac{1}{D_q} \int_{\mathcal{T}_q} e_q^{\mathrm{resp}}(t; R)^\top W_q e_q^{\mathrm{resp}}(t; R)\, dt. \tag{2}$$

The scalar Cart-Pole formulation uses $p = 1$ and its original zero-offset response convention. The arm uses $p = 7$ and retains each measured offset. D_q is fixed by the task, not by the candidate. A finite integration window retains harmonic interactions but does not model arbitrary startup transients. The same construction applies to a fixed nonnegative numerical quadrature.

Let $J_q^{\mathrm{NL}}(R)$ denote the nonlinear evaluation outcome. We test both the ordering induced by a compressed response score and the observed cost of its selected controller. These are distinct from reconstructing H_R or certifying stability. The experiments condition on locally stable, viable ESN controllers. Our representation and matching claims concern these reservoir-controller populations; applicability to other controller architectures is not evaluated. Response characterization and eventual nonlinear validation remain separate from repeated task queries.

3 Closed-loop response representations

We first learn response coordinates from development controllers, then construct a task functional for their shared decoder. Figure 1 depicts the H-CLRR construction for scalar and multichannel tracking. Held-out controllers are projected without refitting the representation; nonlinear full-bank outcomes evaluate the frozen scores and controller choices. K-CLRR is evaluated on Cart-Pole only.

3.1 Weighted response coordinates

On a common M-frequency grid, stack the real and imaginary parts of all entries of E_R into $v_R \in \mathbb{R}^{2Mp^2}$. Let S be a fixed invertible weighting operator, μ the development mean, and U_d the leading orthonormal directions of the weighted centered development responses. The encoder and decoder are

$$z_R = U_d^\top S(v_R - \mu), \qquad \widehat{v}_R = \mu + B_d z_R, \qquad B_d = S^{-1} U_d. \tag{3}$$

This is weighted least-squares reconstruction in the metric $S^\top S$. Mean, weights, and basis are fitted using development responses only. A new controller is projected with these fixed quantities. The code describes a controller-induced response, not a reservoir hidden state or a reduced dynamical realization. The arm retains all 49 complex input–output channels at each frequency, including the 42 off-diagonal channels.

Representation size is chosen or prespecified without nonlinear test outcomes. Cart-Pole uses development-only nested selection; the main arm study evaluates a prespecified bank-size-by-dimension design. Section 5 gives the procedures for each system.

3.2 A task functional in the learned coordinates

Under a fixed linear response-evaluation rule, the reconstructed harmonic error is affine in the code. The arm additionally stores $o_R \in \mathbb{R}^p$ outside PCA. Write $w_R = [z_R^\top, o_R^\top]^\top$ and $\Psi_q(t) = [\Phi_q(t), I_p]$, giving

$$\widehat{e}_q(t; w_R) = e_{0,q}(t) + \Psi_q(t) w_R. \tag{4}$$

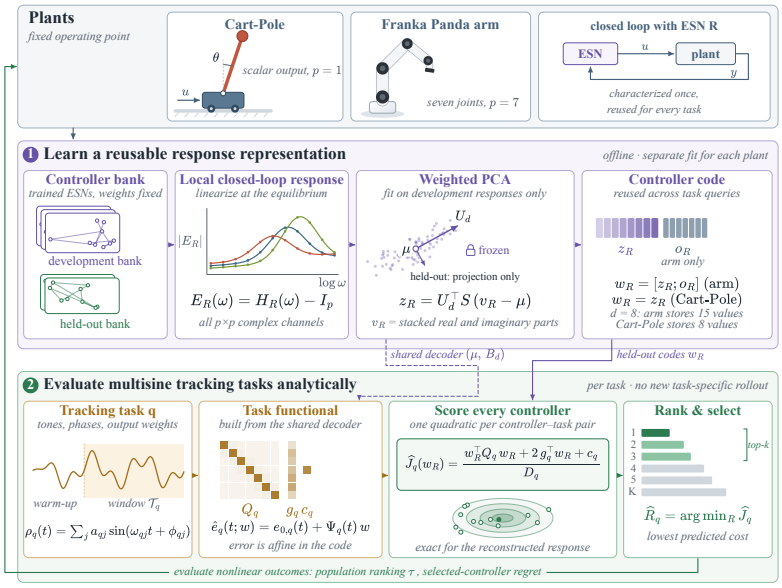

Figure 1: **H-CLRR overview. (1)** Plant-specific response characterization and representation learning. **(2)** Task-conditioned scoring and controller selection. The dashed arrow carries the shared decoder; the solid cross-stage arrow carries held-out controller codes. The arm retains equilibrium offsets alongside the latent code. Top-k marks a predicted shortlist, whereas the minimum-score controller defines the single selected candidate. Curves, point clouds and score bars are schematic.

Here $e_{0,q}$ is the task response of the shared mean and Φ_q contains the task responses of the decoder directions. When offsets are omitted, use $w_R = z_R$ and $\Psi_q = \Phi_q$. Expanding Equation (2) yields

$$\widehat{J}_q^H(w_R) = \frac{w_R^\top Q_q w_R + 2 g_q^\top w_R + c_q}{D_q}, \tag{5}$$

$$Q_q = \int_{\mathcal{T}_q} \Psi_q^\top W_q \Psi_q \, dt, \quad g_q = \int_{\mathcal{T}_q} \Psi_q^\top W_q e_{0,q} \, dt, \quad c_q = \int_{\mathcal{T}_q} e_{0,q}^\top W_q e_{0,q} \, dt.$$

One functional therefore applies to every controller code. Its coefficients depend on the task and fixed shared decoder, not the queried controller's identity. Once constructed, the functional requires $O((d+p)^2)$ arithmetic per controller, or $O(d^2)$ without offsets. Characterization, encoding, and coefficient construction are separate costs. Eight arm coordinates require 15 stored values including offsets.

Equation (5) is exact for the reconstructed harmonic error, not the nonlinear rollout. No nonlinear task-performance regression is fitted. Appendix A derives the identity and its response-evaluation assumptions; Appendix E specifies the sampled arm implementation.

3.3 Scalar response-error kernels

For scalar tracking, $K_R(\omega) = |E_R(\omega)|^2$ gives an objective-specific alternative. K-CLRR fits a separate weighted PCA, reconstructs the sampled kernel, projects it to nonnegative values, interpolates, and applies task weights:

$$\widehat{k}_R = [\mu_K + B_K z_R^K]_+, \qquad \widehat{J}_q^K(R) = \sum_j \alpha_{qj} \, \mathcal{I}_q(\widehat{k}_R)(\omega_{qj}). \tag{6}$$

This is a distinct pipeline, not the general quadratic in Equation (5). It retains phase error relative to ideal tracking because $|H-1|^2 = (|H|-1)^2 + 2|H|(1-\cos\arg H)$, but not the complete relative orientation of complex residuals across frequencies. The full kernel suffices when the task Gram is diagonal and isotropic within each real/imaginary frequency pair. Generic finite windows can retain additional interactions. Appendix B gives the condition and a counterexample. A scalar norm of a multichannel response is not substituted for the arm's full matrix residual.

4 WHAT COMPRESSION PRESERVES

Task geometry determines how response error affects compatibility. Let v denote the real response coordinates, augmented with retained offsets when required. A fixed response-evaluation map induces a positive-semidefinite task Gram G_q.

Proposition 1 (Response-cost distortion)**.** *For $J_q^{\mathrm{resp}}(v) = v^\top G_q v / D_q$, with $G_q \succeq 0$ and $D_q > 0$, let $v = \widehat{v} + \delta v$. Then*

$$\left| J_q^{\mathrm{resp}}(v) - J_q^{\mathrm{resp}}(\widehat{v}) \right| \le \frac{\|G_q\|_2}{D_q} \left(2 \|\widehat{v}\|_2 \|\delta v\|_2 + \|\delta v\|_2^2 \right). \tag{7}$$

The proof expands the quadratic forms and applies Cauchy–Schwarz (Appendix A). It separates compression error from the additional discrepancy $|J_q^{\mathrm{NL}} - J_q^{\mathrm{resp}}|$ caused by the local-response approximation. If two response-cost errors are bounded by ϵ_i and ϵ_j, a full-response cost gap exceeding $\epsilon_i + \epsilon_j$ preserves their ordering. Small average reconstruction error alone does not ensure this margin condition.

For comparisons, the distribution of cost error across controllers also matters. Decompose the error for a fixed task into a common shift b_q and controller-specific residual ε_{iq}:

$$\widehat{J}_{iq} - J_{iq}^{\mathrm{FULL}} = b_q + \varepsilon_{iq} \quad \Longrightarrow \quad (\widehat{J}_{iq} - \widehat{J}_{jq}) - (J_{iq}^{\mathrm{FULL}} - J_{jq}^{\mathrm{FULL}}) = \varepsilon_{iq} - \varepsilon_{jq}. \tag{8}$$

A common shift cancels, whereas differential errors can reverse an ordering. This motivates measuring controller-specific cost error relative to the spread of full-response costs. Section 6.4 applies this diagnostic to the stored response predictions, while the nonlinear experiments assess both complete rankings and the quality of the selected controller.

5 EXPERIMENTAL DESIGN

We use nonlinear ESN controller populations and separate response-representation development from nonlinear evaluation. Each plant has its own fitted mean and basis. Held-out controllers are already trained controllers excluded from representation fitting, not untrained policies. Cart-Pole and the arm are evaluated separately; their tasks and raw costs are not pooled.

5.1 CART-POLE COMPATIBILITY STUDY

Thirty-six development controllers and 440 development tasks support nested controller cross-validation with six outer folds and five inner folds. Each inner split uses 24 fitting and six validation controllers. From $d \in \{2, 4, 8, 12, 16\}$, the smallest dimension within 0.10 validation Kendall of the $d = 16$ reference and with median reconstruction NRMSE at most 0.05 is selected. All outer folds choose eight for both variants. Selection uses analytical response-derived compatibility, not nonlinear rollout labels (Appendix C).

The fixed representations are evaluated on 43 new locally stable controllers from the same generation distribution. Task sets A, B, and C contain four, eight, and 24 near-equilibrium multisine references, with two or three simultaneous tones and $\mathrm{Var}(r) = 0.005$. They reuse the same 43-controller bank; each task set is fixed before its corresponding outcomes. Of 516 trajectories in A, 514 complete; B and C contain 1,032 and 3,096 complete trajectories. The two solver-truncation cases remain recorded separately (Appendix D).

5.2 MULTIJOINT ARM STUDY

The main arm experiment evaluates a simulated seven-joint Panda in MuJoCo (Todorov et al., 2012) at one contact-free posture. Forty-unit tanh reservoirs output nominal position commands through the model's inner servos and a common gravity-compensation adapter. The complete loop is sampled every 10 ms, with five 2 ms simulator substeps. All 7×7 responses are retained at 32 logarithmically spaced frequencies from 0.1 to 1 Hz. Scored joint excursions are 2–5 mrad. The actuator interface, normalization, equilibrium offsets, and local-response checks are detailed in Appendix E.

We fit representations to nested banks of the first 24, 48, and 96 admitted development controllers, and evaluate all on a separate new bank of 16 controllers. Twelve development tasks and eight final tasks are newly generated. The latter contain three proximal, two wrist-dominant, and three distributed multijoint references, using new tone combinations, phases, and directions within the characterized band. The 3×3 design crosses development-bank size with prespecified dimensions $d \in \{2, 8, 16\}$. The primary contrast is the change in nonlinear-ranking correlation from 24 to 96 development controllers at fixed $d = 8$. An earlier held-out arm evaluation motivated this controlled study; its selection-rule limitation and results are retained in Appendix H.

Each of the eight tasks uses three matched initial conditions, producing 384 candidate trajectories; 24 additional trajectories evaluate the compensated servo-only baseline. The reference ramps in during a 10 s warm-up and is scored over $[10, 30)$ s. The outcome is unshifted, offset-inclusive joint tracking error normalized by reference-excursion energy, using $W_q = I_7/7$, and averaged over the three conditions. Every planned trajectory completes. These perturbations share the same plant and largely decay during warm-up; they are not independent environment replications.

5.3 Comparators, decisions, and uncertainty

Full uses the uncompressed complex response and the same task objective. Cart-Pole additionally compares K-CLRR, magnitude-only coordinates, a frozen random ensemble, a task-independent score, sampled spectra, and a descriptor-to-code predictor. Its extra dimension, sparse-summary, and 600-subspace comparisons are retrospective analyses of fixed outcomes. Arm sparse-eight retains complete complex matrices at eight grid nodes and interpolates real/imaginary parts in log frequency. It stores 784 response scalars, not eight. The arm's held-out-loss comparator uses complete reserved teacher episodes; 1,000 fixed random permutations provide descriptive selection references.

Macro Kendall is the equal-task mean of controller-ranking correlations. Arm results use Kendall τ_b and also report the rank and additive regret of the controller minimizing the predicted score. The chosen identity is fixed before its outcome is used. Best-in-shortlist coverage instead measures whether the observed optimum lies in a retained list. Cart-Pole H/K intervals on C resample its 12 matched task pairs; sparse-summary intervals resample controllers conditional on the fixed 36 tasks. Arm intervals use 2,000 paired controller-cluster bootstrap draws, preserving task and perturbation sharing. They condition on the fitted development banks and eight tasks, not on arbitrary future task families. Appendix F gives the arm's predeclared decision rule and secondary comparisons.

6 Results

The experiments ask which controller comparisons survive response compression. Cart-Pole tests whether learned coordinates preserve held-out compatibility better than simple compact descriptions. The arm then separates full-ranking agreement from the quality of a selected controller. We examine development coverage and controller-specific cost errors to interpret the lost information, before comparing the storage and computation required by each representation.

6.1 Held-out task compatibility

The Cart-Pole representation preserves strong ranking agreement across all three task sets (Table 1). H-CLRR reaches 0.9574, 0.9028, and 0.9439, close to the full response. On C, H-minus-Full is -0.01320 with a matched-pair 95% interval of $[-0.02233, -0.00406]$: the compression loss is small but detectable. Its slight advantage over Full on B also shows that exact reproduction of the response-based ordering and agreement with nonlinear outcomes are different criteria.

K-CLRR is comparable on A and C but weaker on B. On C, H-minus-K is -0.00129 with interval $[-0.00794, 0.00637]$. A matched 12-pair test of the hypothesized positive cross-frequency advantage gives mean interaction -0.00498 and does not support that explanation. The matched test therefore provides no evidence of a systematic H-over-K advantage despite their different information content. Appendix B gives the mechanism analysis and scalar-kernel conditions.

Table 1: Macro Kendall correlation with nonlinear rollout rankings.

Representation	Set A (4 tasks)	Set B (8 tasks)	Set C (24 tasks)
Full complex response	0.9657	0.8931	0.9571
H-CLRR, $d = 8$	0.9574	0.9028	0.9439
K-CLRR, $d = 8$	0.9629	0.8176	0.9452
Magnitude-only, $d = 8$	-0.1323	0.2276	-0.2366
Frozen random ensemble	0.5703	0.1905	0.5535
Task-independent	—	-0.3494	0.1941

All sets reuse one held-out 43-controller bank. The dash denotes an unreported comparison, not a zero score. These are task-level macro correlations, not selection accuracies.

Table 2: Compact descriptions on the fixed 36-task set.

Description	Per-controller values	Macro τ
H-CLRR, $d = 8$	8 real coordinates	0.9363
Eight sampled kernel values	8 real samples	0.8995
Eight complex response samples	16 real values	0.9044
Latent code predicted from descriptors	13 descriptor inputs	0.7529

Shared basis, mean, grid, and model overhead are excluded from per-controller counts. Sparse-summary and descriptor comparisons are retrospective. Full per-set values are in Appendix D.

6.2 Learned coordinates and compact alternatives

Sparse spectral summaries test whether learning a basis preserves more compatibility information at a small per-controller storage budget. Eight H-CLRR coordinates achieve all-task $\tau = 0.9363$, compared with 0.8995 for eight sampled kernel values and 0.9044 for eight complex response samples (Table 2). The paired controller-bootstrap advantages are 0.0368 (95% interval $[0.0244, 0.0485]$) and 0.0319 ($[0.0144, 0.0548]$). These comparisons condition on the fixed 36 tasks and exclude shared representation overhead from per-controller counts.

The leading coordinate explains 79.13% of development variance, yet alone gives all-task $\tau = 0.4871$. Two and four coordinates raise it to 0.8632 and 0.9078. None of 600 sampled eight-dimensional subspaces matches 0.9363; the strongest reaches 0.8603. The selected directions therefore matter beyond dimensionality alone, without establishing optimality among all encoders. Appendix D preserves the full coordinate and random-subspace analyses, and Appendix C the dimension sensitivity.

6.3 Multijoint ranking and controller selection

The fresh arm study separates population ordering from the quality of the selected controller (Table 3). At $N_{\mathrm{dev}} = 96$ and $d = 8$, H-CLRR achieves $\tau = 0.5375$ but selects the best controller on five tasks and the second-best on the remaining three, for mean rank 1.375. Full and sparse-eight preserve much more of the ordering, with correlations 0.9188 and 0.9250, and select the bank-best controller on all eight tasks. The compact code retains useful information near the optimum on this bank, not near-complete ranking fidelity. Figure 4 shows the selected rank for every task.

With 24 development controllers and eight coordinates, the mean selected rank is already 1.50, and five tasks select the exact optimum. The 24- and 96-controller versions choose identical winners on seven tasks; the remaining choice improves from third to second. Thus, both development sizes support useful choices on this bank. Sixteen coordinates at $N_{\mathrm{dev}} = 96$ increase correlation to 0.5813 but select the same winners as eight. Figure 2 and Appendix F report the complete within-bank comparison.

All candidate rollouts complete, with best-to-worst costs differing by 1.4–12.7% of the bank-best cost across tasks. The ranking problem therefore resolves differences among viable controllers rather than separating successful runs from failures. Every admitted reservoir controller also outperforms the compensated servo-only comparator, which lacks the teacher's corrective feedback; this is not a comparison against an equally configured nonreservoir controller.

Table 3: Controller ranking and choice on the fresh arm bank.

Method	τ_b [95% interval]	Winner rank ↓	Regret† ↓	Best
Full response	0.9188 [0.851, 0.975]	1.000	0	8/8
Sparse-eight	0.9250 [0.838, 0.982]	1.000	0	8/8
H-CLRR, $N = 24, d = 8$	0.4854 [0.324, 0.631]	1.500	5.31	5/8
H-CLRR, $N = 96, d = 8$	0.5375 [0.423, 0.655]	1.375	5.17	5/8
Held-out loss	0.1688 [0.030, 0.319]	7.875	37.81	0/8

N is the development-bank size; all rows use the same 16 test controllers and eight tasks. †Additive regret is shown in units of 10^{-5} normalized task cost. Best counts exact bank-optimum selections.

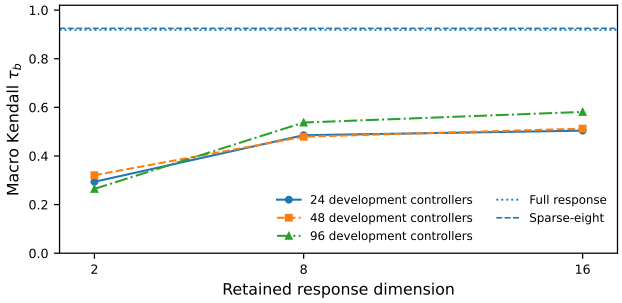

Figure 2: Development-bank size and retained dimension on one fresh arm test. Full and sparse-eight use the same 16 controllers and eight tasks. The primary bank-size comparison fixes $d = 8$.

6.4 Development coverage and comparison error

The predeclared primary contrast asks whether more development controllers improve ranking at fixed $d = 8$. Increasing N_{dev} from 24 to 96 changes correlation from 0.4854 to 0.5375, a paired difference of 0.0521 with 95% interval $[-0.0636, 0.1659]$. This interval includes zero and the prespecified practical threshold of 0.10, leaving the primary hypothesis inconclusive. By contrast, the secondary two-to-eight-coordinate contrasts are positive at every bank size, with separate intervals excluding zero. These contrasts share the same test bank, and their intervals are not adjusted for multiple comparisons.

Full-span coverage improves more than fixed-dimensional reconstruction. The median squared fraction outside the complete development span decreases from 0.372 to 0.223 to 0.090 as the bank grows. Yet median centered NRMSE at $d = 8$ decreases only from 0.778 to 0.721 to 0.710. A 95-dimensional development span can cover responses that its first eight directions still fail to reconstruct. Appendix G defines these distinct normalizations. The improved coverage has not translated into a conclusive fixed-dimensional ranking gain.

Controller-specific cost error provides a closer view of the lost ordering information (Equation (8); Figure 3). Relative to the within-task spread of Full costs, sparse-eight has a median common shift of 1.007 but controller-specific error of only 0.026. For H-CLRR with 96 development controllers and eight coordinates, the corresponding values are -0.090 and 0.632. Sparse-eight preserves 98.0% of Full's pairwise orders, versus 77.0% for H-CLRR. The sparse approximation can therefore preserve comparisons despite a larger shared shift: it introduces much less error into the differences between controllers. These are analytical comparisons against Full, separate from the nonlinear evaluation.

6.5 Storage and task-query cost

An eight-coordinate arm code stores 15 scalars including offsets, versus 791 for sparse-eight and 3,143 for Full. Shared overhead changes the comparison at the tested bank size: total stored values for 16 controllers are 33,568, 12,664, and 50,320, respectively, under the implemented accounting. The compact code is smaller per controller, but sparse-eight uses less total storage for this bank.

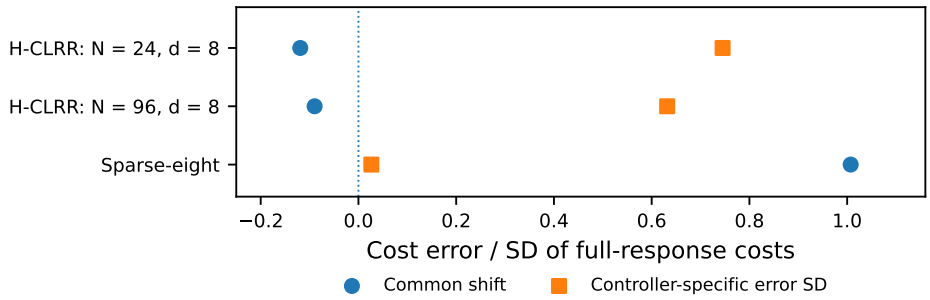

Figure 3: Analytical error components normalized by the within-task spread of Full costs, summarized by task medians. A common shift cancels in controller comparisons; differential error does not.

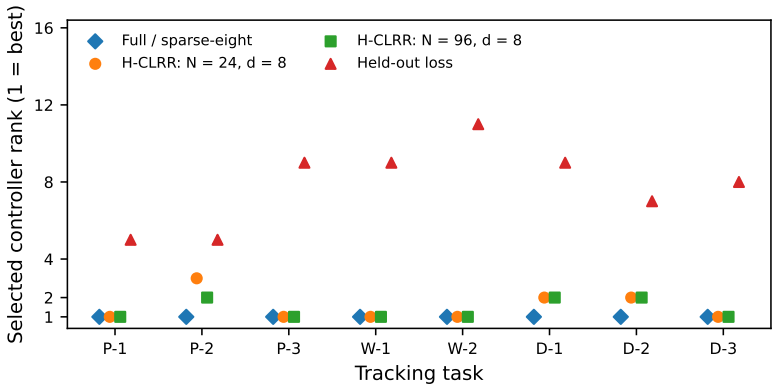

Figure 4: Rank of each predicted winner on the fresh arm test. Full and sparse-eight both select rank one in every task. The two eight-coordinate codes differ only on P-2.

The reported 16-controller-by-eight-task quadratic-query batch takes 0.256 ms after coefficients exist, compared with 24.3 ms for Full. Constructing the eight latent task functionals takes about 174 ms, however, and response characterization and fitting are additional costs. These measurements support inexpensive repeated queries after preprocessing, not an end-to-end speed advantage for a one-off batch. Appendix I gives stage-separated timing, repetitions, and storage accounting for the tested implementations.

7 Discussion and Conclusion

CLRR turns characterized plant–reservoir responses into reusable coordinates for trained ESN controllers and pairs them with analytical tracking-task functionals. The learned basis captures variation across closed loops, while the task functional specifies how that variation enters a comparison. This separates response characterization from repeated task queries without fitting a predictor to nonlinear performance labels. The experiments assess the resulting representation by the controller distinctions preserved, alongside its reconstruction and computational costs.

The two systems reveal different compression trade-offs. Cart-Pole supports strong ranking agreement at a small code size. On the fresh arm bank, compact codes retain good selected-controller performance while Full and sparse responses preserve substantially more of the ordering. Larger development banks expand response-span coverage, but the ranking gain at fixed dimension remains inconclusive. The common-versus-controller-specific error analysis connects this gap to the quantity needed for a comparison: relative cost differences. Smaller per-controller codes and inexpensive prepared queries must therefore be weighed against lost ordering information and shared preprocessing costs.

AI USE STATEMENT

ChatGPT (OpenAI) assisted with research discussion, literature checks, mathematical exposition, manuscript drafting and revision, numerical consistency checks, and LaTeX/figure code. The authors retain responsibility for the research and verification of AI-assisted material.

REPRODUCIBILITY STATEMENT

Sections 2–5 define the response representation, task functional, and evaluation design. Appendices A–B give assumptions and proofs; the remaining appendices provide implementation details, complete numerical comparisons, and historical sensitivity results. The accompanying source package contains result tables and scripts for reproducing the presented summary figures. Those materials reproduce the reported summaries; they are distinct from the full controller-training and nonlinear-simulation pipeline.

REFERENCES

Anonymous authors. Ranking and shortlisting nonlinear reservoir controllers using local closed-loop dynamics. Related manuscript submitted to ICRA 2027, 2026.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015. doi: 10.1137/130932715.

Joni Dambre, David Verstraeten, Benjamin Schrauwen, and Serge Massar. Information processing capacity of dynamical systems. *Scientific Reports*, 2:514, 2012. doi: 10.1038/srep00514.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.04529.

Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022. doi: 10.1287/mnsc.2020.3922.

Cross Giannetto, Ibragim R. Atadjanov, Fumiya Iida, and Arsen Abdulali. Reservoir computing encodes physical adaptations for reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pp. 6005–6011, 2025.

Chengan He, Xin Sun, Zhixin Shu, Fujun Luan, Sören Pirk, Jorge Alejandro Amador Herrera, Dominik L. Michels, Tuanfeng Y. Wang, Meng Zhang, Holly Rushmeier, and Yi Zhou. Perm: A parametric representation for multi-style 3D hair modeling. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2407.19451.

Tobias Hülser, Felix Köster, Kathy Lüdge, and Lina Jaurigue. Deriving task specific performance from the information processing capacity of a reservoir computer. *Nanophotonics*, 12(5):937–947, 2023. doi: 10.1515/nanoph-2022-0415.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065):20150202, 2016. doi: 10.1098/rsta.2015.0202.

Ksenia Konyushkova, Yutian Chen, Thomas Paine, Caglar Gulcehre, Cosmin Paduraru, Daniel J. Mankowitz, Misha Denil, and Nando de Freitas. Active offline policy selection. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009. doi: 10.1016/j.cosrev.2009.03.005.

Mustafa C. Ozturk, Dongming Xu, and José C. Príncipe. Analysis and design of echo state networks. *Neural Computation*, 19(1):111–138, 2007. doi: 10.1162/neco.2007.19.1.111.

Clarence W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(3):997–1013, 2005. doi: 10.1142/S0218127405012429.

Gianluca Scarpellini, Ksenia Konyushkova, Claudio Fantacci, Thomas Paine, Yutian Chen, and Misha Denil. π2vec: Policy representation with successor features. In *International Conference on Learning Representations*, 2024.

Hongyao Tang, Zhaopeng Meng, Jianye Hao, Chen Chen, Daniel Graves, Dong Li, Changmin Yu, Hangyu Mao, Wulong Liu, Yaodong Yang, Wenyuan Tao, and Li Wang. What about inputting policy in value function: Policy representation and policy-extended value function approximator. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8441–8449, 2022. doi: 10.1609/aaai.v36i8.20820.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012. doi: 10.1109/IROS.2012.6386109.

Shang Wang. W-PCA based gradient-free proxy for efficient search of lightweight language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2504.15983.

Fan Ye, Arsen Abdulali, Kai-Fung Chu, Xiaoping Zhang, and Fumiya Iida. Reservoir controllers design though robot–reservoir timescale alignment. *Communications Engineering*, 4:81, 2025a. doi: 10.1038/s44172-025-00418-1.

Fan Ye, Timothy Bonner, Arsen Abdulali, Kai-Fung Chu, and Fumiya Iida. Reservoir computing for torque-restricted pendulum control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3823–3830, 2025b.

A Compatibility construction and proofs

A.1 Finite-window task geometry

For a scalar reference component, write $E_j = E_R(\omega_{qj}) = E_{j,\mathrm{r}} + \mathrm{i}E_{j,\mathrm{i}}$ and $\psi_j(t) = \omega_{qj}t + \phi_{qj}$. Its real harmonic error is

$$e_j(t) = a_{qj}\big[E_{j,\mathrm{r}} \sin\psi_j(t) + E_{j,\mathrm{i}} \cos\psi_j(t)\big]. \tag{9}$$

Stack the real and imaginary residuals at task frequencies into η_R and their amplitude-weighted sine and cosine functions into $b_q(t)$. Then the zero-offset scalar error is $b_q(t)\eta_R$, with task Gram

$$G_q^{\mathrm{tone}} = \int_{\mathcal{T}_q} b_q(t)^\top b_q(t)\, dt \succeq 0, \qquad J_q^{\mathrm{resp}}(R) = \eta_R^\top G_q^{\mathrm{tone}} \eta_R / D_q. \tag{10}$$

Cross-frequency terms are retained. On a generic finite window, distinct tones need not be orthogonal, and sine/cosine terms within one tone need not have equal energy or zero inner product. Diagonal power weighting is therefore a special case rather than an identity for all multisine tasks.

For multichannel tracking, each tone contributes $E_{j,\mathrm{r}}a_{qj}\sin\psi_j + E_{j,\mathrm{i}}a_{qj}\cos\psi_j$. Stacking all matrix entries gives a real matrix-valued evaluation map in place of b_q. The Gram uses W_q between the transposed and untransposed evaluation maps. Appending o_R adds an identity block to the map, so the cost is still a positive-semidefinite quadratic in the complete response-and-offset vector.

When interpolation from the stored grid is linear and fixed, write the task-frequency coordinates as $\eta_R = L_q v_R$. The sampled-response Gram is $G_q = L_q^\top G_q^{\mathrm{tone}} L_q$. With affine reconstruction $\widehat{v}_R = \mu + B_d z_R$, the zero-offset cost becomes

$$\widehat{J}_q^H(z_R) = \frac{z_R^\top B_d^\top G_q B_d z_R + 2(B_d^\top G_q \mu)^\top z_R + \mu^\top G_q \mu}{D_q}. \tag{11}$$

Retained offsets are handled by augmenting the feature vector and decoder. A nonlinear, code-dependent interpolation rule need not preserve this quadratic form. The arm uses grid-node task frequencies; sparse baselines use a fixed linear interpolation in $\log_{10} f$, separately for real and imaginary parts.

A.2 Proof of the quadratic identity

Substitute $\widehat{e}_q = e_{0,q} + \Psi_q w$ into Equation (2). Since W_q is symmetric,

$$\int \widehat{e}_q^\top W_q \widehat{e}_q\, dt = \int e_{0,q}^\top W_q e_{0,q}\, dt + 2\int w^\top \Psi_q^\top W_q e_{0,q}\, dt + \int w^\top \Psi_q^\top W_q \Psi_q w\, dt.$$

The code and offsets are constant in time. Moving w outside the integrals gives $c_q + 2g_q^\top w + w^\top Q_q w$, proving Equation (5). For any a, $a^\top Q_q a = \int (\Psi_q a)^\top W_q (\Psi_q a)\, dt \geq 0$. The same algebra holds for fixed nonnegative quadrature weights. It is an identity for the reconstructed-response model, not for arbitrary nonlinear trajectories.

A.3 Proof of Proposition 1

Because G_q is symmetric and $v = \widehat{v} + \delta v$,

$$v^\top G_q v - \widehat{v}^\top G_q \widehat{v} = 2\widehat{v}^\top G_q \delta v + \delta v^\top G_q \delta v.$$

The triangle inequality and induced Euclidean norm give Equation (7) after division by $D_q > 0$. PCA supplies the reconstruction; the inequality does not require PCA-specific optimality. With offsets reproduced exactly, their reconstruction discrepancy is zero in the augmented vector.

If $|\widehat{J}_i - J_i^{\mathrm{FULL}}| \leq \epsilon_i$ and $|\widehat{J}_j - J_j^{\mathrm{FULL}}| \leq \epsilon_j$, then the error of the predicted cost difference is at most $\epsilon_i + \epsilon_j$. A larger full-response margin therefore preserves its sign. More specifically, decomposing the cost errors into a common shift and centered residuals gives Equation (8). The common shift is defined separately for each task and method as the controller mean error; subtracting it is diagnostic, not an outcome-calibrated scoring correction.

The discrepancy to nonlinear outcomes separates as

$$|J_q^{\mathrm{NL}}(R) - \widehat{J}_q^H(R)| \le |J_q^{\mathrm{NL}}(R) - J_q^{\mathrm{resp}}(R)| + |J_q^{\mathrm{resp}}(R) - \widehat{J}_q^H(R)|. \tag{12}$$

Only the second term is controlled by the reconstruction bound. Local characterization, finite-amplitude effects, and transients influence the first.

A.4 Implementation-level checks

For Cart-Pole, the latent identity was checked on 1,376 unique controller–task pairs from B and C with maximum discrepancy 1.4×10^{-14}; A contributes 172 additional pairs. The off-diagonal Gram perturbation calculation was checked on 2,752 full/reconstructed-response cases from B and C. These are algebraic checks, not independent nonlinear experiments.

For the fresh arm data, refitting weighted PCA from each stored development prefix reproduces all nine representations up to SVD signs and roundoff. Direct harmonic costs from the decoded responses match 1,280 stored Full/PCA predictions to approximately 3.4×10^{-16}. Recomputing all 112 development/test responses from the archived sampled matrices gives maximum discrepancy approximately 6.7×10^{-14}. These checks establish consistency of the matrix-to-response-to-score calculation; they do not independently certify the simulator derivatives or nonlinear trajectory costs.

B When the scalar kernel is sufficient

Arrange scalar task-frequency coordinates into real/imaginary pairs. If G_q^{tone} is block diagonal with blocks $\beta_{qj} I_2$, then

$$J_q^{\mathrm{resp}}(R) = \sum_j \frac{\beta_{qj}}{D_q} \left[E_{j,\mathrm{r}}^2 + E_{j,\mathrm{i}}^2\right] = \sum_j \alpha_{qj} |E_R(\omega_{qj})|^2.$$

The uncompressed scalar kernel suffices for this objective. The condition includes orthogonal tones with equal sine/cosine energy under suitable windows or averaging; it is stronger than merely using a finite sum of sinusoids. Compressed K-CLRR remains approximate and includes nonnegativity projection and interpolation.

An elementary example illustrates the limitation. Two scalar response vectors $(1, 1)$ and $(1, -1)$ at two distinct frequencies have identical kernels. Their zero-phase, unit-amplitude errors are $s_1(t) + s_2(t)$ and $s_1(t) - s_2(t)$, where $s_j(t) = \sin \omega_j t$. The cost difference is $4 \int_{\mathcal{T}_q} s_1(t) s_2(t)\, dt / D_q$, which is nonzero when the cross term is nonzero. This is a mathematical illustration, not another simulation result. For multichannel tasks, a scalar Frobenius norm can additionally discard directional interactions between input channels; the arm study therefore uses complex matrix responses rather than transferring scalar K-CLRR unchanged.

The Cart-Pole matched experiment contains 12 high/low task pairs designed to test a systematic complex-response advantage. The observed mean interaction is -0.004983, with one substantive positive pair, seven negative pairs, and four numerical zeros. The H-minus-K correlation difference on C is -0.0012919897 with 95% interval $[-0.0079365, 0.0063677]$. Neither result supports the proposed positive mechanism, nor proves that all finite-window interaction terms are irrelevant.

C Cart-Pole dimension selection

Nested selection uses 36 development controllers and 440 analytical development tasks. Six outer folds and five inner folds keep validation controllers out of fitting; inner splits contain 24 fitting and six validation controllers. The criterion selects the smallest tested dimension with a Kendall gap no larger than 0.10 to $d = 16$ and median reconstruction NRMSE at most 0.05. Table 4 records the selected values. The retrospective test-dimension sweep does not change that choice (Table 5, Figure 5). A pooled out-of-fold development audit gives Kendall 0.9111 for H-CLRR and 0.8674 for K-CLRR. This is distinct from the held-out 43-controller evaluation. Eight is the smallest tested dimension meeting the specified criterion, not a universal intrinsic dimension or an accuracy-maximizing test choice.

Table 4: Development adequacy at the selected dimension.

Quantity	H-CLRR, $d = 8$	K-CLRR, $d = 8$
Inner-validation Kendall	0.917869	0.880953
Gap to $d = 16$	0.060323	0.084478
Median reconstruction NRMSE	0.014470	0.028164
Development variance captured	99.8592%	99.9264%
Held-out variance captured	94.9717%	94.0376%
Outer folds selecting eight dimensions	6/6	6/6

NRMSE and captured variance describe response reconstruction, not nonlinear tracking-error accuracy.

Table 5: Complete H-CLRR dimension sweep.

d	Dev. τ	Gap	NRMSE	Set A	Set B	Set C
2	.755020	.223172	.062591	.935770	.766334	.883444
4	.809556	.168636	.052605	.935216	.840532	.925618
8	.917869	.060323	.014470	.957364	.902824	.943891
12	.956242	.021949	.007659	.965670	.907807	.955795
16	.978192	.000000	.003309	.961794	.899225	.957641

Dimensions 8, 12 and 16 satisfy the development rule; 8 is the smallest. Test-set columns are retrospective checks and did not determine this choice.

D Additional Cart-Pole comparisons

D.1 Coordinate and sparse-summary comparisons

Table 6 retains the per-set comparisons. The all-task column weights the sets by their task counts (4, 8, and 24), not equally by set. Sparse spectral and descriptor comparisons are retrospective on these stored outcomes. Eight complex samples require 16 real values, while eight scalar samples require eight; shared basis and grid overhead is excluded from these counts. The first coordinate explains

Table 6: Per-set compression comparisons and descriptive aggregates.

Description	Set A	Set B	Set C	All 36
H-CLRR, first coordinate	.767996	.379845	.476006	.487080
H-CLRR, first 2	.935770	.766334	.883444	.863234
H-CLRR, first 4	.935216	.840532	.925618	.907777
H-CLRR, all 8	.957364	.902824	.943891	.936262
8-node sampled kernel	.938538	.840255	.912698	.899471
8 complex response nodes	.866556	.868217	.922757	.904393
Latent from 13 descriptors	.885382	.674695	.756829	.752861

79.1317% of development variance. High-frequency response-error energy is its strongest tested single-descriptor predictor, with held-out $R^2 = 0.905978$. This predicts a coordinate, not nonlinear cost. Predicting the code from 13 descriptors does not exhaust all direct task-conditioned uses of those descriptors.

D.2 Random subspaces

The frozen random ensemble in Table 1 is separate from the 600 later single-subspace controls. The latter fix $d = 8$ and draw 200 subspaces from each of the ambient feature space, development-response span, and leading-16-PC span. Table 7 and Figure 6 report their medians, 95th percentiles, and maxima. These are summaries over sampled representations, not confidence intervals over independent nonlinear experiments.

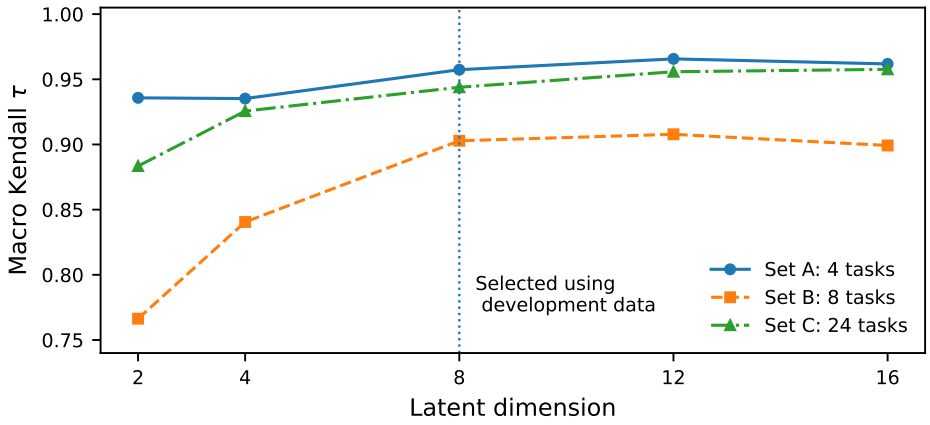

Figure 5: Cart-Pole dimension sensitivity. The primary $d = 8$ choice uses development-only validation; other test dimensions are retrospective.

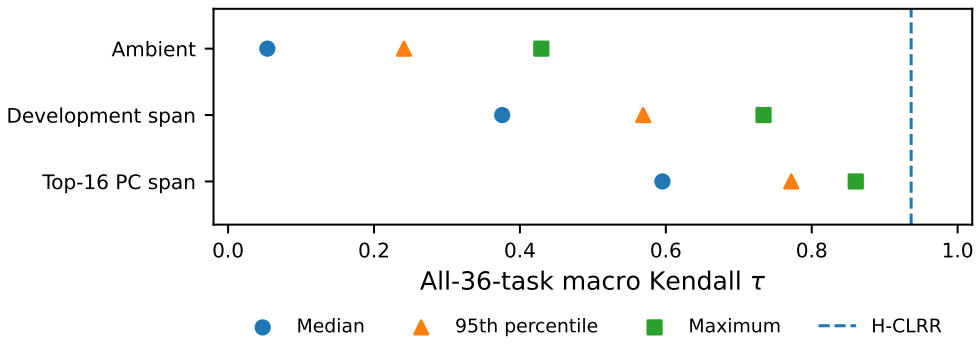

Figure 6: Cart-Pole random-subspace controls. Markers are medians, 95th percentiles, and maxima over 200 draws per family, not confidence intervals.

Table 7: Eight-dimensional random-subspace sensitivity.

Sampling space	Draws	Median	95th percentile	Maximum
Ambient response space	200	.053802	.241015	.429310
Development-response span	200	.375723	.568925	.733912
Top-16 development PC span	200	.595330	.771939	.860342
Fixed H-CLRR	—	.936262	—	.936262

All entries use the fixed 36-task aggregate. No random draw equals or exceeds the H-CLRR value.

Table 8: Development and held-out evaluation.

Stage	Controllers	Tasks	Controller–task pairs	Valid / total trajectories
Development	36	440	—	No labels used for selection
Set A	43	4	172	514 / 516
Set B	43	8	344	1,032 / 1,032
Set C	43	24	1,032	3,096 / 3,096

Sets A–C use the same held-out controller bank. Their tasks were specified before each set's nonlinear outcomes were evaluated. The 440 development tasks serve response-based validation.

D.3 Uncertainty and completion counts

The H-minus-Full interval on C is $[-0.022333, -0.004061]$ around -0.0131967516. H/K and H/Full inference on C resamples matched task pairs; its 3,096 trajectories do not replace the 12 pairs as mechanism-test units. For the all-task aggregate, controller resampling gives the H-minus-sampled-kernel interval $[0.0244, 0.0485]$ around 0.036791, and H-minus-complex-nodes interval $[0.0144, 0.0548]$ around 0.031869. These condition on the fixed task set. Set A includes two solver-truncated trajectories, with 514/516 completing. All 1,032 in B and 3,096 in C complete. Completion counts are distinct from the controller and task-pair units used in the statistical analyses.

E Arm implementation and evaluation protocol

E.1 Robot and trained controller interface

The arm uses the seven-joint, no-hand Panda model, with position-servo actuation, at nominal posture

$$r^\star = (0, 0, 0, -1.57079, 0, 1.57079, -0.7853)^\top \text{ rad}.$$

The model is integrated with `implicitfast` at 2 ms. Five substeps make one 10 ms controller interval. The simulated model has 14 physical state coordinates and no dynamic actuator states. A 40-unit reservoir therefore gives a 54-dimensional sampled closed-loop state $\xi = [\theta^\top, \dot\theta^\top, s^\top]^\top$. Output is joint position at the beginning of the interval, before the new command acts.

The controller observes normalized position and reference excursions, $x = [\theta - r^\star; r - r^\star]$, but not joint velocity. Its readout is $u_{\text{nom}} = W_{\text{out}}[1; x; s]$. The reservoir follows

$$s_{n+1} = s_n + \frac{h}{\tau_s}\left[-s_n + \tanh(W_{\text{in}} x_n + W_r s_n + b)\right], \qquad h = 0.010\,\text{s}, \quad \tau_s = 0.05\,\text{s}. \tag{13}$$

The recurrent spectral-radius recipe is 1.1, connection probability $10/40$, bias scale 0.1, and input-scale multiplier 4. Ridge regularization is 10^{-8}. Candidate recurrent weights and biases vary under the fixed generator, and one readout is fitted per candidate. Per-channel feature scales come from the common readout-training data.

The common teacher command is $u_{\text{teach}} = r + 1.0(r - \theta) - 0.03\dot\theta$. Eight complete demonstration episodes train the readout; three other complete episodes support the held-out-loss comparator. The teacher sees velocity, while the RC must infer its effect from history. The same nominal position interface is used throughout. A gravity adapter applies

$$u_{\text{applied}} = u_{\text{nom}} + K^{-1} g(\theta)$$

before each simulator substep. Here K is the position-command-to-generalized-force map from the model, checked against a numerical derivative, and $g(\theta)$ is the zero-velocity gravity term. Nominal RC commands remain held over the five substeps. The adapter, servos, and reservoir are included in both characterization and nonlinear evaluation; there is no separate direct torque-command interface.

E.2 Equilibria and response construction

The complete sampled map is $\xi_{n+1} = F_R(\xi_n, r_n)$. The implementation resets numerical warm starts, computes the output and current command, updates the reservoir, and advances the held-command plant substeps. At constant $r^\star$, a joint equilibrium is obtained from constant-reference settling and numerical polishing. The measured output offset $o_R = \theta_R^\star - r^\star$ is retained.

Central differences of the full sampled step give A_R and B_R, with output map $C_R = [I_7\ 0\ 0]$ and $D_R = 0$ at the recorded timestamp. The response is

$$H_R(e^{\mathrm{i}\omega h}) = C_R(e^{\mathrm{i}\omega h} I - A_R)^{-1} B_R + D_R. \tag{14}$$

The inverse denotes a linear solve. Discrete-time matrices are not substituted into a continuous-time resolvent. The 32 grid frequencies are $f_k = 10^{-1+k/31}$ Hz, $k = 0, \ldots, 31$, with $\omega = 2\pi f$. The feature order concatenates the row-major real entries E_{kij} followed by the row-major imaginary entries. Normalized trapezoidal weights in $\log_{10} f$ are repeated across all 49 entries in both blocks. The encoder uses their square roots, and the decoder divides by those square roots.

Eligibility is evaluated before final task outcomes: finite fits and responses, a consistent stable equilibrium, inactive local constraints, multistart checks, perturbation recovery, and a modal-settling condition. The new development bank admits 96 of 124 attempts; its 24- and 48-controller prefixes define the smaller banks. The test bank admits 16 of 21 attempts. No favorable final tracking cost is used to replenish either bank. These checks condition the population on viability, not on arbitrary random reservoirs.

E.3 References, outcomes, and comparators

The eight final tasks are three proximal, two wrist-dominant, and three distributed multisines. Each has two or three tones, with fixed seven-component amplitude vectors and phases at grid frequencies. Peak excursions are 2, 3.5, or 5 mrad according to the task schedule. The exact vectors and frequency indices are included in `data/arm_final_tasks.json`. Reference phase remains continuous through a two-second raised-cosine ramp and the warm-up/scoring boundary.

Each controller is evaluated from the nominal physical state and two matched position perturbations $\pm\delta\theta$, with maximum absolute coordinate 0.5 mrad; reservoir states start at zero. The scored grid is $t_n \in [10, 30)$ s at $h = 0.01$ s. For each trajectory,

$$J_q^{\mathrm{NL}}(R) = \frac{\sum_{t_n \in [10,30)} h\,[\theta_R(t_n) - r_q(t_n)]^\top (I_7/7)[\theta_R(t_n) - r_q(t_n)]}{\sum_{t_n \in [10,30)} h\,[r_q(t_n) - r^\star]^\top (I_7/7)[r_q(t_n) - r^\star]}. \tag{15}$$

The reported controller–task cost is the mean over the three conditions. Neither demeaning, phase alignment, nor dynamic time warping is applied. All 384 candidate and 24 servo-only jobs complete, so no failure penalty changes the reported comparisons. The recorded maximum actuator-force ratio is 0.345, with no control-range or force-limit ticks in this evaluation.

Full evaluates the entire response and the same offsets. Sparse-eight selects grid indices $(0, 4, 9, 13, 18, 22, 27, 31)$ and linearly interpolates real and imaginary components in log frequency. The initial-study two- and four-node variants use $(0, 31)$ and $(0, 10, 21, 31)$. The held-out-loss score is a global mean of channelwise teacher-command MSE normalized by the squared teacher-correction RMS on three reserved episodes. It does not choose a new loss function for each test task. Random lists use 1,000 fixed-seed permutations without replacement from the same admitted bank.

E.4 Local-regime verification

Engineering pilots preceded the original final evaluation. At the initial larger proposed amplitudes, local cost discrepancies were substantial; uncompensated small-amplitude costs were dominated

by gravity offsets. The executed protocol therefore added common gravity compensation and reduced scored amplitudes from the proposed 20–50 to 2–5 mrad before final outcomes. A separate compensated pilot reported response-error/separation ratios 0.022, 0.076, and 0.122, and worst-family cost-error/separation ratios 0.05, 0.15, and 0.33 at 2, 3.5, and 5 mrad. These passed the pilot's 0.5 gate. The added modal-settling condition rejected no controller in that pilot and is not established as the cause of improved fidelity.

This evaluates a narrow local regime, not large-excursion manipulation. An initial uncompensated complex-response plot is not used as direct evidence for the final compensated loop. The full-response agreement with nonlinear outcomes in the executed regime is reported independently of the compression comparisons.

F Fresh arm results and statistical comparisons

F.1 Factorial design and primary contrast

All nine representation conditions use the same fresh 16-controller test bank and eight tasks. Development banks are nested, not independent repetitions. Four development folds contain 18/6, 36/12, and 72/24 fitting/validation controllers for banks of 24, 48, and 96. The primary nonlinear contrast fixes $d = 8$ in advance, even though applying the original relative dimension rule to those banks would select 8, 8, and 12. Support for the predeclared primary hypothesis required a point improvement of

Table 9: Prespecified arm bank-size and dimension comparison.

N_{dev}	$d = 2$	$d = 8$	$d = 16$
24	0.2937 [0.186, 0.393]	0.4854 [0.324, 0.631]	0.5042 [0.339, 0.643]
48	0.3208 [0.210, 0.417]	0.4792 [0.342, 0.612]	0.5125 [0.365, 0.643]
96	0.2646 [0.131, 0.401]	0.5375 [0.423, 0.655]	0.5813 [0.454, 0.678]

Entries are macro Kendall correlations with nonlinear outcomes and conditional 95% controller-bootstrap intervals. All nine conditions share the same test bank.

at least 0.10, a positive lower 95% paired-bootstrap bound, and improvements in the three analytical coverage/error diagnostics. A below-threshold conclusion required an upper bound below 0.10. The observed difference 0.0520833 and interval $[-0.0635975, 0.1659292]$ satisfy neither criterion. The result is inconclusive, not evidence of equivalence or proof that larger development banks cannot help.

Bootstrap draws resample controller identities while keeping all tasks and perturbations for each identity together; 2,000 draws use seed 20261001. The development populations and task set are fixed. All 2,000 primary draws are defined. Secondary contrasts are paired on the same draws and are not corrected for multiple comparisons. No bank-size contrast excludes zero. The two-to-eight-coordinate gains are 0.192 ($[0.051, 0.341]$), 0.158 ($[0.045, 0.273]$), and 0.273 ($[0.149, 0.390]$) for development sizes 24, 48, and 96. Eight-to-sixteen-coordinate gains are smaller, and their intervals include zero.

F.2 Controller choices and all-method results

For each task, the predicted winner is selected before inspecting nonlinear costs. Its empirical regret is its mean nonlinear cost minus the lowest mean cost in the admitted bank. Its rank is one plus the number of strictly lower-cost candidates, with the executed tie convention retained. Shortlist coverage checks whether that observed bank optimum is among the first k predicted candidates; it is not a separately retested final choice from the shortlist. Full and sparse-eight choose the optimum on all eight tasks. The $N_{\mathrm{dev}} = 96, d = 8$ and $d = 16$ versions choose identical controllers: best on five tasks, second-best on three. The $N_{\mathrm{dev}} = 24, d = 8$ version differs on only the second proximal task, choosing rank three instead of rank two. Figure 4 shows every task rather than only favorable examples.

The compensated servo-only comparator is not a ranked candidate. All admitted RC controllers outperform it, and its cost is 2.07–2.56 times the bank-best cost. Because it lacks the same corrective

Table 10: All representations on the fresh arm test.

Method	τ_b	Winner rank	Regret†	Top-1	Top-3	Top-5
Full response	0.9188	1.000	0	8/8	8/8	8/8
Sparse-eight	0.9250	1.000	0	8/8	8/8	8/8
H-CLRR, $N = 24, d = 2$	0.2937	7.250	35.04	1/8	2/8	4/8
H-CLRR, $N = 24, d = 8$	0.4854	1.500	5.31	5/8	6/8	7/8
H-CLRR, $N = 24, d = 16$	0.5042	1.500	6.46	5/8	5/8	7/8
H-CLRR, $N = 48, d = 2$	0.3208	6.250	24.31	2/8	4/8	5/8
H-CLRR, $N = 48, d = 8$	0.4792	2.125	8.14	5/8	6/8	8/8
H-CLRR, $N = 48, d = 16$	0.5125	1.375	5.17	5/8	7/8	7/8
H-CLRR, $N = 96, d = 2$	0.2646	6.125	20.23	2/8	4/8	5/8
H-CLRR, $N = 96, d = 8$	0.5375	1.375	5.17	5/8	7/8	8/8
H-CLRR, $N = 96, d = 16$	0.5813	1.375	5.17	5/8	6/8	8/8
Historical fixed $d = 2$	0.1708	6.250	24.51	3/8	4/8	5/8
Historical fixed $d = 8$	0.4375	2.875	11.54	3/8	5/8	5/8
Held-out loss	0.1688	7.875	37.81	0/8	0/8	0/8

†Regret is in units of 10^{-5}. Top-k counts tasks whose predicted list contains the observed optimum. Historical codes are unchanged encoders from the separate initial study, applied to this new bank.

feedback as the training teacher, this comparison does not establish a unique advantage of the reservoir architecture.

F.3 Cross-bank comparisons

The initial arm bank and fresh factorial bank contain different controllers and task instances. Even the unchanged historical eight-dimensional code rises from 0.258 on the initial study to 0.438 on the fresh one. Thus, an old-to-new headline difference does not identify a method improvement. Table 10 labels the fixed historical-code controls separately from the newly fitted development-bank representations.

G Response coverage and comparison diagnostics

G.1 Reconstruction relative to controller variation

Let v_R be a response vector and use the same weighted norm as PCA. Full-norm reconstruction error divides $\|S(v_R - \widehat{v}_R)\|_2$ by $\max(\|Sv_R\|_2, \epsilon)$. Centered NRMSE instead divides by $\max(\|S(v_R - \mu)\|_2, \epsilon)$, with scale floor $\epsilon = 10^{-6}$. The latter measures reconstruction error relative to deviation from the development mean, rather than the common response level.

Let P_{span} project onto the complete weighted centered development span. The out-of-span statistic is

$$\eta_R = \frac{\|(I - P_{\text{span}})S(v_R - \mu)\|_2^2}{\|S(v_R - \mu)\|_2^2}.$$

This is a squared relative residual for the full available span, not the centered NRMSE of an eight-coordinate code. The development mean changes with bank size; the observed decrease in normalized η_R is not guaranteed by span nesting alone. Full-span ranks are 23, 47, and 95. At fixed $d = 8$,

Table 11: Full-span coverage and eight-coordinate reconstruction.

N_{dev}	Span rank	Median η	Centered NRMSE	Dev. variance
24	23	0.3723	0.7784	80.34%
48	47	0.2231	0.7209	71.23%
96	95	0.0900	0.7101	64.40%

η is a squared relative residual for the entire span. NRMSE and variance refer to the retained $d = 8$ subspace; the two error columns do not share a squared/norm convention.

the larger bank improves centered reconstruction much less than full-span coverage. The first eight components explain approximately 80.3%, 71.2%, and 64.4% of their respective development variance. A larger observed population can expose variation beyond eight leading directions without making PCA itself inconsistent.

G.2 Common and controller-specific cost errors

For each task and method, let b_q be the mean analytical error $\widehat{J}_{iq} - J_{iq}^{\mathrm{FULL}}$ across test controllers. The centered residual is ε_{iq}. Table 12 reports the common shift and residual standard deviation divided by the within-task standard deviation of Full costs, then summarized by task medians. These are analytical diagnostics from stored responses; no nonlinear outcome is used to fit a correction. Sparse-eight has larger common bias but much smaller differential error than the tested PCA codes.

Table 12: Analytical prediction errors relative to Full.

Method	Common shift	Differential error	τ_b vs. Full
H-CLRR, $N = 24, d = 8$	-0.119	0.745	0.479
H-CLRR, $N = 48, d = 8$	-0.157	0.717	0.473
H-CLRR, $N = 96, d = 8$	-0.090	0.632	0.540
H-CLRR, $N = 96, d = 16$	-0.060	0.630	0.588
Sparse-eight	+1.007	0.026	0.960

Shift and differential-error SD are divided by the task-specific SD of Full costs and summarized by medians over eight tasks. Correlation is the task mean. No nonlinear outcomes enter this decomposition.

Because common bias cancels in a pairwise comparison, its absolute cost offset does not prevent strong ordering. The task-relative margin analysis likewise shows better preservation of large Full cost gaps: DEV96 $d = 8$ preserves approximately 53% of orders with normalized margin below 0.1 and 94% with margin above 2; sparse-eight preserves 88% and 100%, respectively. These pair counts share controllers and tasks and are descriptive, not independent samples.

G.3 Historical task-weighted reconstruction diagnostic

An exploratory metric family blended the original response weighting with a task-derived reconstruction metric using $\alpha \in \{0, 0.5, 0.9\}$. It was built and selected using development responses and separate development-task subsets. Selection chose $\alpha = 0, d = 8$, which is the original weighting rather than a successful new task-aware method. The associated test-bank evaluations were already revealed and remain exploratory.

The development component analysis reports Spearman correlation 0.99 between component variance and task-window reconstruction energy, and 0.77 between variance and cost sensitivity. Some later directions matter to comparisons, but these results do not support a blanket claim that leading PCA variance is task-irrelevant. The evaluated metric family does not settle whether other decision-oriented learning objectives would preserve more information.

H Initial arm evaluation

An earlier held-out study used 24 development and 16 test controllers with eight tasks. Its four-fold development rule compared $d \in \{2, 4, 8, 12, 16\}$ to the largest feasible dimension and selected the smallest within a 0.10 Kendall gap and below 0.05 median full-norm NRMSE. The $d = 16$ reference achieved development correlation only 0.367 with Full; $d = 2$ achieved 0.306 and passed the relative criterion. This selected dimension was fixed before the original nonlinear outcomes. The resulting $d = 2$ correlation was 0.175, versus 0.931 for Full. It illustrates a limitation of a relative adequacy rule whose reference representation is itself weak. A small reconstruction error relative to the whole response also concealed large error relative to controller variation: median test full-norm NRMSE was 0.0146, but centered NRMSE was 0.960. The PCA itself was centered; the adequacy denominator was the issue.

Later $d = 4, 8, 12, 16$ analyses reused the already-revealed initial test bank and are exploratory. Their correlations were 0.225, 0.258, 0.235, and 0.302. Increasing dimension did not restore near-Full

Table 13: Earlier held-out arm evaluation.

Method	τ_b	Mean winner rank	Regret[†]
Full response	0.9313	1.500	1.14
H-CLRR (d=2)	0.1750	7.250	17.41
SPARSE-2	0.4667	4.375	13.38
SPARSE-4	0.8062	1.750	3.00
SPARSE-8	0.9188	1.625	1.41
Held-out loss	-0.0479	10.750	37.58

[†]Regret is in units of 10^{-5}. This bank is separate from the fresh factorial study. The selected $d = 2$ result remains the frozen primary result of this earlier experiment.

ranking, and the exploratory $d = 8$ mean regret exceeded the primary $d = 2$ value despite its higher correlation. This initial study is separate from the fresh prespecified bank-size/dimension experiment, not replaced by its stronger selected-controller result.

I TIMING AND STORAGE ACCOUNTING

Table 14: Stored values and measured task-query costs.

Method	Per controller	Total ($K = 16$)	Query (ms)	Task setup (ms)
H-CLRR, $N = 96, d = 2$	9	13,456	0.245	75.3
H-CLRR, $N = 96, d = 8$	15	33,568	0.256	174.3
H-CLRR, $N = 96, d = 16$	23	61,280	0.250	324.5
Sparse-eight	791	12,664	35.850	—
Full response	3,143	50,320	24.319	—

Query denotes all 16 controllers and eight tasks after any latent coefficients exist. Task setup denotes construction of eight latent functionals. Dashes mean the corresponding latent preparation stage does not apply; they are not measured zero end-to-end costs.

All times are reported wall-clock means from the experiment environment (macOS ARM64, one BLAS/OpenMP thread), not measurements repeated on another machine. Characterization is measured over five repetitions; encoding and query batches over 20; task-coefficient construction over three. Standard deviations and every timed condition are retained in `data/Reported_Timing_Storage.json`. The exact CPU model is not identified by that environment record.

For DEV96 $d = 8$, fitting takes 15.223 ms, encoding 16 characterized responses 0.0568 ms, and constructing eight task functionals 174.261 ms. The subsequent 128 controller–task queries take 0.256 ms. Characterization of one controller's finite-difference Jacobian and 32-node response takes 11.180 ms in the reported timing setup. Equilibrium search, demonstration generation, candidate rejection, and training are not included in that characterization timing label. Full and sparse-eight query batches take 24.319 and 35.850 ms in their tested implementations.

Storage counts include the controller code and seven offsets, plus stored means, basis arrays, feature weights, frequency information, and eight cached task functionals where applicable. For $d = 8$, shared overhead is 33,328 scalar values and the total is $33,328 + 15K$ for K candidates. Sparse-eight uses the reported total $8 + 791K$; Full uses $32 + 3,143K$. These are implementation-specific scalar inventories, not a claim that those exact arrays or full symmetric matrices are information-theoretically necessary. The raw experiment archives and original controller checkpoints are outside this descriptor accounting.

At $K = 16$, sparse-eight has the smallest total among these response descriptors. Under the same stored-array convention and eight-task cache, the $d = 8$ total becomes smaller than sparse-eight for $K \geq 43$. This arithmetic concerns storage only. The task-preparation cost and the large ranking gap must still be considered when choosing a representation; query-only timing does not establish end-to-end speed/accuracy dominance.

J RELATIONSHIP TO THE RELATED CONTROLLER-RANKING SUBMISSION

The related manuscript (Anonymous authors, 2026) ranks and shortlists fixed nonlinear reservoir controllers using local closed-loop response scores. It includes requested-frequency conditioning, phase-sensitive comparisons, shared plant identification, local-stability diagnostics, disjoint evaluation banks, and amplitude analyses. Its principal empirical object is an ordering or shortlist under a prescribed response score.

CLRR instead fits a shared response representation across development controllers and pairs each encoded controller with a task-specific functional. The central question is which task-dependent comparisons survive compression on controllers excluded from fitting. Development-only selection, compact spectral comparisons, coordinate and random-subspace controls, and the arm bank-size/dimension study evaluate that representation. The quadratic identity and distortion bound characterize reconstructed-response costs rather than introducing a new controller-training method.

Both studies use closed-loop response information, but their result tables address different questions. ICRA single-frequency and shortlist outcomes are not pooled into CLRR. The fresh arm selection metrics assess the compact representation rather than relabeling Full response ranking as a new compression result.