mathematics

MDPI

Article

Information-Geometric Selection for Validation-Guided Ensembles of Extreme Learning Machines: The LEA-EELM Framework

Amir Abbas Sabzevari [1,2], Hamid Tabatabaee [1,2,*] and Mehrdad Jalali [3,*]

[1] Department of Computer Engineering, Ma.C., Islamic Azad University, Mashhad 9187147578, Iran
[2] Department of Artificial Intelligence and Data Science, Intelligent Financial Innovation Research Center, Ma.C., Islamic Azad University, Mashhad 9187147578, Iran
[3] Applied Data Science and Artificial Intelligence, SRH University Heidelberg, 69123 Heidelberg, Germany
* Correspondence: h_tabatabaee@mshdiau.ac.ir (H.T.); mehrdad.jalali@srh.de (M.J.)

Abstract

Extreme learning machines (ELMs) solve output weights in closed form, so an optimized population of ELMs leaves open which members should form the ensemble. We report a controlled comparison of five rules for this selection step inLEA-EELM, where a Lotus Effect Algorithm (LEA) variant generates fuzzy-activation ELM candidates and $M = 10$ members are chosen from a shared top-2M validation shortlist. Two rules are derived from the Fisher information $I(\beta) = H^\top H/\sigma^2$ of a candidate's linear-Gaussian output stage. Because $H^\top H$ is only positive semidefinite and candidates have different hidden layers, F-complementary compares candidates by the Kullback–Leibler divergence between their predictive distributions, not by a Fisher–Rao distance. For F-norm, we prove that, with least-squares output weights, $\|H\beta\|^2 = \|T\|^2 - \|T - H\beta\|^2$, so the rule selects the worst training fits; it is significantly worse than β-norm on three of four regression benchmarks. F-complementary performs like greedy validation selection and is never significantly better. The information-geometric rules therefore add no measured benefit; the gains come from validation-guided greedy selection, which lowers test RMSE on three of four regression datasets and raises accuracy only on Satellite and Letter, but not on Letter after duplicate rows are removed. All findings are conditional on the candidate pool of a non-canonical (subtractive-sign) LEA variant.

Keywords: extreme learning machine; information geometry; Fisher information; Kullback–Leibler divergence; selective ensemble; validation-guided selection; metaheuristic optimization; fuzzy activation function

MSC: 68T07; 62B11; 62B10

Check for updates

Academic Editor: Paul W. Vos

Received: 23 September 2026
Revised: 21 October 2026
Accepted: 28 October 2026
Published: 2 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Extreme Learning Machines (ELMs) [1] are single-hidden-layer feedforward networks in which hidden-layer weights and biases are assigned randomly, and only the output weights are solved analytically, via a Moore–Penrose pseudoinverse. The resulting training speed is a genuine advantage over gradient-trained networks, but it comes with a well-known cost: two independently drawn hidden layers on the same data can generalize very differently. A single, fixed activation applied uniformly across the input space adds a second source of mismatch whenever local noise or class overlap would favor a different response.

Metaheuristic search can replace the random draw with an optimized population of hidden-layer encodings. That step improves candidate quality, yet it does not by itself produce a reliable ensemble. Directed optimization tends to concentrate the population around similar high-fitness solutions, so member errors become correlated. Candidate quality alone is therefore insufficient: an ensemble also needs complementary errors and a selection rule that can identify them without looking at the test set.

Much of the ELM ensemble literature emphasizes how candidates are generated—independent random draws, genetic search, or other population-based updates—and treats the final subset as a secondary detail, often a simple ranking by fitness or output-weight norm. Less attention has been given to constructing a compact ensemble from an already optimized pool under a strict validation-only protocol.

The Lotus Effect Algorithm (LEA), previously introduced by members of the present research team [2], is a nature-inspired population-based optimizer designed to balance global exploration and local exploitation. Building on that optimizer, LEA-EELM applies LEA to optimize the hidden-layer weights and biases of a population of fuzzy ELM candidates. The present contribution lies in integrating this previously introduced LEA with adaptive fuzzy activation, analytical output-weight estimation, and validation-guided construction of a compact ensemble. The fuzzy adaptive activation function (FAF) of our earlier FAF-ELM study [3] modulates steepness from a Gaussian-membership estimate of each pre-activation; output weights are fitted analytically on the training split; and a selective ensemble of $M = 10$ members is taken from the $2M$ fittest candidates in the LEA-optimized candidate population. Three selection rules are compared on that same pool, followed by two rules derived from the Fisher information of the output stage (Section 3.6). The default LEA-EELM configuration selects $M = 10$ members with the smallest output-weight norms from the 20 candidates with the lowest validation RMSE. The greedy-validation configuration sequentially selects the candidate that maximizes partial-ensemble validation performance. The greedy-complementary configuration sequentially selects candidates using a specified combination of validation performance and error complementarity. Aggregation is the member-output mean (regression) or majority vote (classification). Test inputs enter only at prediction.

The paper asks four questions, none of which requires LEA-EELM to be best overall:

1. Does a moderate ensemble of optimized ELM candidates improve held-out performance relative to a single optimized candidate?
2. Does validation-guided or diversity-aware selection improve on the default β-norm configuration when both are applied to the same LEA-optimized candidate population, and do rules derived from the Fisher information of the output stage change that comparison?
3. Does ridge regularization of the output weights help the tested pipeline?
4. How does FAF compare with a conventional Gaussian activation on the same backbone, and how does the full pipeline compare with task-specific external baselines?

LEA-EELM is a framework for generating and selectively combining optimized ELM candidates. None of its components is new on its own: LEA, fuzzy activation, analytical output weights, and greedy ensemble selection [4] are all established. The contribution is a controlled study of the ensemble-construction stage on one shared candidate pool, including two selection rules derived from the Fisher information of the output stage, an exact explanation of why one of them fails, and an honest account of what does and does not help. The contributions are:

- An integrated learning framework that employs the previously introduced LEA to optimize fuzzy ELM candidates and subsequently constructs a compact ensemble through validation-guided member selection.

- A controlled comparison of output-weight-norm selection, established greedy ensemble selection [4], and its complementarity-weighted variant over the same LEA-optimized candidate population.
- Evidence that moderate ensembling improves over a single optimized candidate on the evaluated M-sweep datasets.
- Evidence that, relative to the default β-norm configuration, validation-guided selection improves regression test RMSE on three of four regression datasets (Boston Housing is not significant after Holm adjustment), and classification accuracy on Satellite, the dataset with the largest duplicate-free validation split. The two greedy variants perform similarly.
- An information-geometric analysis of the selection step: the Fisher quadratic form of the output stage is defined only for a shared hidden layer and is a semimetric; candidates with distinct hidden layers are compared by a predictive Kullback–Leibler divergence (F-complementary), and an output-space norm rule (F-norm) is proved to select the worst training fits in the shortlist (Proposition 1). Empirically, neither rule improves on established greedy selection.
- A reproducible evaluation that identifies the useful operating conditions—regression ensemble construction—and the remaining classification challenge of selecting from small validation sets.

Against complete external baseline families (five regression baselines; seven classification baselines), LEA-EELM does not uniformly outperform every comparator. That external outcome is reported in full; it is not the claim this paper is written to establish. Section 2 situates the design; Section 3 specifies the method; Section 4 records the evaluation protocol; Section 5 reports results in the order of the questions above; and Section 6 interprets them.

2. Related Work

The ELM framework [1] trains a single-hidden-layer network by drawing input weights and biases at random and solving for output weights in closed form. That random assignment is the source of ELM's training speed, but it also makes two independently initialized models on the same data generalize differently. Hidden-layer sensitivity is therefore a structural property of the standard ELM, not an incidental implementation detail.

Replacing the random draw with a metaheuristic search is now a standard response to that sensitivity. Genetic-ensemble ELM (GE-ELM) [5] searches hidden-layer parameters with a genetic algorithm and assembles an ensemble from the resulting population. Recent siblings use particle-swarm or related search for application-specific ELMs [6–9]. Eshtay et al. review the broader family of metaheuristic ELM design formulations, covering both hidden-layer optimization and related encoding choices [10]. Across this line of work, the optimizer is used to generate improved hidden-layer encodings rather than to replace the analytical output-weight step.

Within this broader class of nature-inspired optimizers, Dalirinia et al. introduced the Lotus Effect Algorithm (LEA), with members of the present research team contributing to its original development [2]. LEA combines exploration mechanisms with lotus-inspired exploitation and reinforcement operations to balance global and local search. The original study evaluated LEA on benchmark functions and constrained engineering-design problems. LEA has since been extended to multimodal optimization (M-LEA), in which independently evolving subpopulations maintain several optima simultaneously [11]. In the present work, LEA is not proposed as a new optimizer; instead, it is employed as the candidate-generation mechanism for searching the hidden-layer weights and biases of fuzzy ELMs. This use of LEA establishes the LEA-optimized candidate population,

whereas the principal methodological question addressed here is how that population should be reduced to a compact and effective ensemble using validation data alone.

Independently trained ELM ensembles reduce draw-to-draw variance by combining members that need not have been jointly optimized. Liu and Wang average independently initialized ELMs (EE-ELM) [12]; Chen et al. likewise construct a committee of ELMs rather than relying on a single random draw [13]. Evolutionary search has also been applied at the *selection* stage: Wang and Alhamdoosh form ELM ensembles under an explicit size constraint, using diversity as a fitness term to keep the committee compact [14]. Lu et al. score candidate ELMs by a disagreement measure before aggregation, which is a diversity-aware alternative to ranking by individual accuracy [15]. Later designs gate or reweight members dynamically [16,17] rather than using a uniform average. Bagging [18] and boosting remain the natural non-metaheuristic ensemble controls. Ensembles of multiple classifiers have also been used for multi-label prediction, where combining several base classifiers improved multi-label classification performance [19]; that work combines classifiers to predict label sets, whereas the question here is which members of an optimized ELM population to retain, using validation data alone. Across this family, the dominant design effort is usually candidate *generation*. The last step—choosing a compact subset from an already optimized population, using only validation data—is often a fitness or output-weight-norm ranking. LEA-EELM keeps LEA-based candidate generation and validation-guided selective ensembling as distinct stages and compares five selection rules on a shared pool.

A parallel line of work changes the hidden mapping rather than the ensemble. NFELM uses fuzzy membership to down-weight outliers in regression [20]; RAFELM modifies the activation for robustness in a kernel-ELM setting [21]. Related robustness mechanisms include residual-aware weighting and correntropy losses [22,23]. Fuzzy modeling on an ELM backbone also appears in hierarchical designs [24]. The fuzzy adaptive activation function (FAF) used here belongs to this family and was established in our earlier FAF-ELM study [3], which combined it with firefly optimization of a single ELM: its steepness is modulated by a Gaussian-membership certainty score of the pre-activation. FAF is therefore not a contribution of the present manuscript; the contribution here is the LEA-optimized candidate population and the validation-guided selective-ensemble framework built on it.

Kolmogorov–Arnold networks (KANs) with Gaussian bases raise a closely related numerical question. Noorizadegan et al. [25] analyze their accuracy and stability through the geometry and conditioning of the first-layer feature matrix and find both highly sensitive to the Gaussian scale parameter, which controls how distinguishable the basis features are. LEA-EELM depends on its hidden matrix H in the same way twice: the analytical output weights $\beta = H^{\dagger}T$ inherit the conditioning of H, and the Fisher information of Section 3.6 is $H^{\top}H/\sigma^2$. When hidden features are strongly correlated or poorly resolved, $H^{\top}H$ is ill-conditioned; $\|\beta\|$ becomes large and unstable, while the fitted output $H\beta$, the projection of T, does not. This is exactly where the parameter-space rule (β-norm) and the output-space rule (F-norm) part ways (Section 3.6 and Appendix A).

Averaging weakly correlated estimators reduces the variance of the ensemble mean relative to a single member [26,27]. Once candidates are produced by a directed optimizer, that correlation structure is no longer incidental: high-fitness members can share errors. Selective ensemble learning—retaining only a subset of a larger pool—is the standard response to that redundancy [28]. Selection rules that look only at individual fitness or $\|\beta\|$ do not measure complementarity. Evolutionary weighting followed by thresholded pruning is one established alternative [29]. Greedy forward selection on partial-ensemble validation performance is the established ensemble-selection method of Caruana et al. [4]; our greedy-validation rule is its fixed-size, without-replacement variant applied to an

optimized ELM population, not a new method. Diversity-aware pruning that scores members in output space, such as DREP [30], motivates the error-complementarity term of greedy-complementary. On the information-geometric side, the Fisher information has been used to analyze and train neural networks through the natural gradient [31,32], to measure capacity through the Fisher–Rao norm [33], and to prune network units by their estimated effect on the loss [34]. The two information-geometric rules studied here (Section 3.6) borrow these ideas for member selection; their value is tested, not assumed. All are validation-only or training-only; all keep ensemble size fixed at $M = 10$.

Existing work therefore supplies many mechanisms for generating or optimizing ELM candidates, but less attention has been given to how a compact, complementary ensemble should be selected from an LEA-optimized candidate population under a strict validation-only protocol. That gap motivates the controlled selection comparison in Section 3 and the experiments in Section 5.

LEA-EELM therefore connects three previously distinct research directions: meta-heuristic optimization of ELM hidden layers, adaptive fuzzy activation, and selective ensemble construction. Its distinctive contribution is the validation-guided construction of a compact ensemble from a shared LEA-optimized candidate population, enabling the effect of alternative selection strategies to be studied independently of candidate generation.

3. Method

LEA-EELM employs the previously introduced LEA for candidate generation and then constructs a compact ensemble from the resulting LEA-optimized candidate population by validation-guided selective ensembling. Figure 1 and Algorithm 1 specify the same workflow: $P = 100$ initial candidates; an initial training fit and validation score for every candidate; $L = 20$ LEA iterations with analytical β on the training split and acceptance by validation RMSE; a top-$2M = 20$ shortlist; selection of $M = 10$ members; and test data used only for the final prediction.

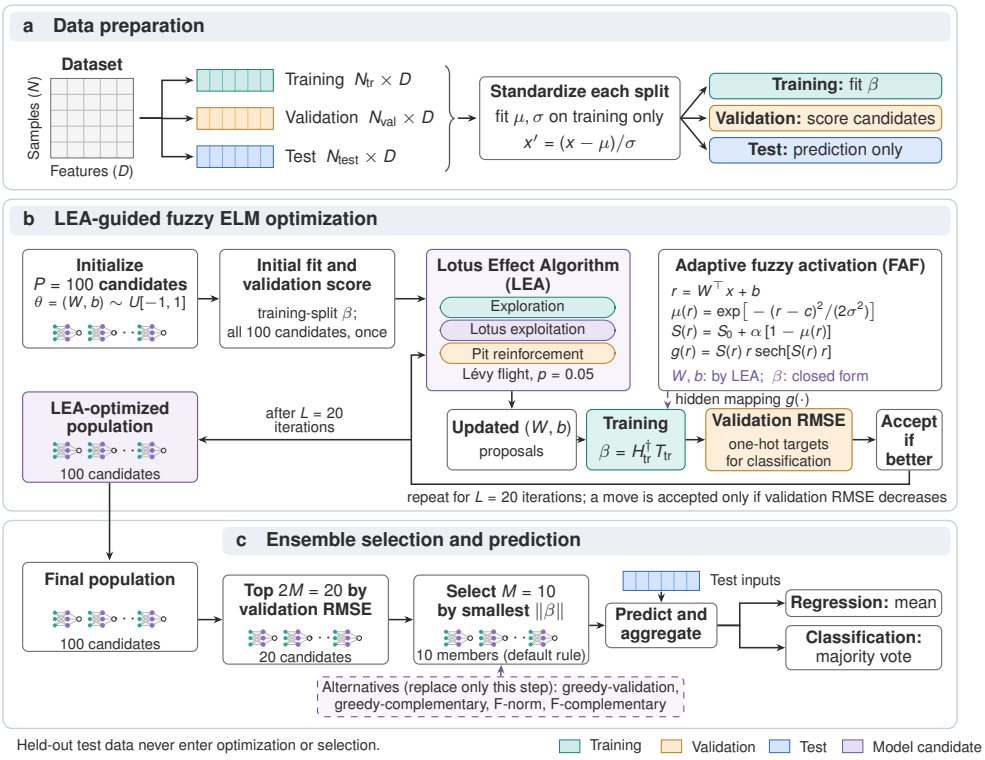


Figure 1. LEA-EELM workflow (Algorithm 1). (**a**) Data preparation: a disjoint train/validation/test split; z-score statistics are fit on the training split only; the test split is unused until prediction. (**b**) LEA-guided

fuzzy ELM optimization: $P = 100$ candidates are initialized and scored once (training-split β, validation RMSE); the previously introduced LEA then updates hidden-layer (W, b) for $L = 20$ iterations by exploration, lotus exploitation, and pit reinforcement (Lévy flight with $p = 0.05$), accepting a move only when validation RMSE decreases. Output weights β are solved analytically; FAF is the hidden mapping. (**c**) Selective ensemble: the final population is shortlisted to the $2M = 20$ lowest validation-RMSE candidates; the default configuration selects $M = 10$ members by smallest $\|\beta\|$. The four alternative rules (greedy-validation, greedy-complementary, F-norm, F-complementary) replace only this step. Held-out test inputs enter solely at aggregation (mean or majority vote).

Algorithm 1 LEA-EELM training and prediction. The default (reference) selection used in the main external comparison is β-norm; the four alternative rules replace only the ensemble-selection step.

Require: dataset $\{(x_j, y_j)\}$; task $\in$ {regression, classification}; hidden width $\tilde{N}$; population $P = 100$; iterations $L = 20$; ensemble size $M = 10$; selection rule $\in$ {β-norm, greedy-val., greedy-compl., F-norm, F-compl.}
Ensure: ensemble $\mathcal{E} = \{(\omega^{(m)}, b^{(m)}, \beta^{(m)})\}_{m=1}^{M}$; test predictions $\hat{y}$
Partition and train-only preprocessing
Draw disjoint $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{va}}, \mathcal{D}_{\text{te}}$ (random if regression; class-stratified if classification)
Fit feature z-score mean/std on $\mathcal{D}_{\text{tr}}$ only; if regression, also fit target z-score mean/std on $\mathcal{D}_{\text{tr}}$ only; apply both transforms to all three splits
Set targets $T_{\text{tr}}, T_{\text{va}}$ to y (regression) or one-hot(y) (classification)
Initialize population
Sample $\theta_i \sim \text{Unif}[-1, 1]^{n\tilde{N}+\tilde{N}}$ for $i = 1, \ldots, P$ (Equation (3))
for each θ_i **do**
    Decode (ω, b); form H_{tr} with FAF $g(\cdot)$ (Equations (5)–(7))
    $\beta \leftarrow H_{\text{tr}}^{\dagger} T_{\text{tr}}$ ▷ training split only (Equation (2))
    fitness$(\theta_i) \leftarrow \text{RMSE}(T_{\text{va}}, H_{\text{va}}\beta)$ ▷ Equation (8); non-finite $\rightarrow +\infty$
LEA optimization (fixed iteration budget; no other stopping rule)
for $t = 1$ to L **do**
    Propose θ_i' for every candidate by LEA exploration, exploitation with $R = 2e^{-(4t/L)^2}$ (Equation (4)), and pit update (Section 3.2); apply a Lévy flight independently with probability 0.05
    Recompute β and fitness(θ_i') as above (train β, validation RMSE)
    Accept $\theta_i \leftarrow \theta_i'$ iff fitness$(\theta_i') <$ fitness(θ_i)
Ensemble selection (validation only; test unused)
$\mathcal{P} \leftarrow$ the $2M$ candidates in the final population with lowest validation RMSE
if rule $= \beta$-norm **then** ▷ default/reference configuration
    $\mathcal{E} \leftarrow$ the M members of $\mathcal{P}$ with smallest $\|\beta\|$
else if rule = greedy-validation **then**
    From $\emptyset$, add M times the member of $\mathcal{P}$ that most improves partial-ensemble validation RMSE (regression) or accuracy (classification)
else if rule = greedy-complementary **then**
    Same greedy loop, scoring a fixed 0.5/0.5 mix of that validation term and error complementarity
else if rule $= F$-norm **then**
    $\mathcal{E} \leftarrow$ the M members m of $\mathcal{P}$ with smallest $\|H_{\text{tr},m}\beta_m\|_F$ ▷ Equation (17); Proposition 1
else if rule $= F$-complementary **then**
    Same greedy loop as greedy-complementary, with the complementarity term replaced by the min–max normalized mean predictive divergence $D(c, m)$ on $\mathcal{D}_{\text{va}}$, m ranging over the members already selected ▷ Equation (14)
Prediction aggregation
if regression **then**
    $\hat{y}(x) \leftarrow \frac{1}{M}\sum_{m=1}^{M} f^{(m)}(x)$ ▷ member-output mean
else
    $\hat{y}(x) \leftarrow$ majority vote of arg max class outputs ▷ ties: lowest class index

3.1. Problem Setting and ELM Formulation

Extreme learning machines [1,35,36] are single-hidden-layer feedforward networks in which the input-layer weights and biases are assigned randomly, and only the output-layer weights are solved analytically. For a single-hidden-layer feedforward network (SLFN) with $\tilde{N}$ hidden nodes, the network output is

$$f_{\tilde{N}}(x) = \sum_{i=1}^{\tilde{N}} \beta_i \, G(\omega_i, b_i, x), \tag{1}$$

for training pairs $\{(x_j, t_j)\}_{j=1}^{N}$, $x_j \in \mathbb{R}^n$, $t_j \in \mathbb{R}^m$, with matrix form $H\beta = T$ and closed-form output weights

$$\beta = H^{\dagger} T \tag{2}$$

via the Moore–Penrose pseudoinverse. Random (ω_i, b_i) make two otherwise identical ELMs generalize differently; a single fixed $G(\cdot)$ ignores local structure in the input. LEA-EELM replaces the random draw with LEA-based candidate generation and validation-guided selective ensembling, and replaces a static G with the input-adaptive FAF of Section 3.3.

3.2. LEA-Based Candidate Generation

Each candidate is a flattened hidden-layer encoding

$$\theta = [\omega_{11}, \ldots, \omega_{1\tilde{N}}, \, b_1, \ldots, b_{\tilde{N}}], \tag{3}$$

initialized uniformly in $[-1, 1]$. Output weights β are never encoded; they are recomputed from θ by Equation (2) whenever a candidate is evaluated.

The previously introduced LEA [2] supplies the search over θ. Three phases are used as implemented:

- **Exploration** (dragonfly-inspired): separation, alignment, cohesion, attraction-to-best, and repulsion-from-worst combine into a per-candidate step. A Lévy-flight replacement is applied independently to each candidate with probability 0.05. The implementation defines no neighborhood radius, so this move is not gated on a "no neighbors" condition.
- **Exploitation** (lotus self-pollination): a growth step of radius R is applied along the vector from the current best candidate to the individual,

$$x_i^{t+1} = x_i^t + R \, (x_i^t - x_{\text{best}}^t), \qquad R = 2\, e^{-(4t/L)^2}. \tag{4}$$

  R shrinks from 2 toward 0 as $t \to L$. Equation (4) has the same sign as the growth step printed in the original LEA paper [2], which we follow. With $R > 0$, the displacement points *away from* the current best, so in this pipeline the step acts as a shrinking diversification move around the incumbent, not as an attraction; a proposal is kept only if it lowers validation RMSE (Section 3.4).
- **Reinforced exploitation** (water-droplet/pit model): a velocity update toward the deepest known pit, used as a second local-search mechanism late in the search.

The implementation departs from the original LEA paper in the sign of three updates: the exploration step, the Lévy replacement, and the pit move are applied in subtractive form $(x_i - \Delta x_i, \, x_i - \text{Levy}(d)\, x_i, \, x_i - V_i)$, where the original prints them additively. Together with Equation (4), every move therefore displaces a candidate away from, not toward, the incumbent, and progress comes from the accept-if-better rule. A validation-only check on the three development datasets ($N = 5$) shows how weak this search is: the best validation RMSE in the population moves from 0.691 to 0.687 on Abalone, from 0.3602 to 0.3600

on Vehicle, and from 0.477 to 0.466 on Diabetes over the $L = 20$ iterations, against 0.676, 0.325, and 0.435 with the additive form. All reported results use the subtractive form; the selection-stage comparisons below are conditional on the candidate pool it produces (Section 6).

Population size $P = 100$, iteration count $L = 20$, inertia $0.9 \to 0.4$, speed-increment $q = 0.1$, and Lévy parameter $\beta = 1.5$ follow the canonical LEA recommendations and are held fixed. LEA is treated as a candidate generator; it is not proposed as a new optimizer in this manuscript, and validating it as a general-purpose metaheuristic is outside the present scope [2].

3.3. Adaptive Fuzzy Activation

The hidden mapping is the fuzzy adaptive activation function (FAF) established in our earlier FAF-ELM study [3]; it is reused here unchanged as a component, and the present work does not claim it as new. The hidden mapping adapts its steepness to a Gaussian-membership estimate of each pre-activation $r = b + w^\top x$. Certainty, steepness, and activation are

$$\mu(r) = \exp\left(-\frac{(r-c)^2}{2\sigma^2}\right), \tag{5}$$

$$S(r) = S_0 + \alpha\,(1 - \mu(r)), \tag{6}$$

$$g(r) = S(r) \cdot r \cdot \operatorname{sech}\big(S(r) \cdot r\big). \tag{7}$$

This is an odd, bounded, adaptive-steepness function, vanishing at $\pm\infty$. The numeric defaults $S_0 = 1.0$, $\alpha = 0.5$, $c = 0.0$, $\sigma = 1.0$ are used throughout: under z-scored inputs and $[-1, 1]$ weights, pre-activations are approximately zero-centered and unit-scale, so a standard-normal certainty bump is a natural choice. These values are not tuned.

As drop-in replacements on the same backbone, we also implement sigmoid, hard-limit, Gaussian ($\exp(-b\|x - \omega\|^2)$), multiquadric, and ReLU. The Gaussian comparison uses a positivity-constrained bandwidth $|b|$ as the well-defined variant; an unconstrained $b \in [-1, 1]$ is reported alongside it because a negative bandwidth turns the RBF into an exploding function of distance (Section 5.5).

3.4. Candidate Evaluation and Data Separation

Fitness is validation-set RMSE,

$$\text{fitness}(\theta) = \text{RMSE}_{\text{val}}(\theta), \tag{8}$$

computed after β has been fitted on the training split only. The test set is never passed into a fitness or selection routine. Classification still uses validation RMSE on one-hot targets during LEA search; accuracy appears later, in the greedy selection rules and as the reported test metric. A proposed move is accepted if and only if validation RMSE strictly decreases. Non-finite fitness is treated as $+\infty$.

3.5. Selective Ensemble Construction

After L iterations, the $2M = 20$ candidates with lowest validation RMSE form a shortlist $\mathcal{P}$. Exactly $M = 10$ members are then chosen by one of five rules: the three described in this subsection, and two additional information-geometric rules, F-norm and F-complementary, introduced in Section 3.6. All five operate on $\mathcal{P}$ only, break remaining ties by smaller $\|\beta\|$, then by smaller pool index, and never see test data.

Default LEA-EELM configuration (β-norm).

It selects $M = 10$ members with the smallest output-weight norms from the 20 candidates with the lowest validation RMSE. This is the default, or reference, configuration used in the main external comparison. It uses no validation-performance term and no diversity term.

Greedy-validation configuration.

It sequentially selects the candidate that maximizes partial-ensemble validation performance: starting from an empty ensemble S, for M steps add the remaining candidate $c \in \mathcal{P}$ that most improves $S \cup \{c\}$ on the validation split (lowest RMSE for regression, highest accuracy for classification). Ensemble size is always M; the search does not stop early.

Greedy-complementary configuration.

It sequentially selects candidates using the specified combination of validation performance and error complementarity: the same greedy loop, with each step maximizing a fixed 0.5/0.5 mix of (a) the min–max normalized validation-performance term of greedy-validation and (b) an error-complementarity term against already-selected members. For regression, complementarity is the mean of $(1 - \mathrm{corr})/2$ of validation residuals (1 for perfectly anti-correlated errors, 0.5 uncorrelated, 0 perfectly correlated). For classification, it is the mean pairwise disagreement rate on validation labels. The complementarity term is 0.5 (neutral) for the first member. The 0.5/0.5 weights were set a priori and not tuned.

Ridge output weights (auxiliary test).

As a separate modification, the training-split fit can be replaced by

$$\beta = \arg\min_{\beta} \|H\beta - T\|_F^2 + \lambda\|\beta\|_F^2, \tag{9}$$

solved via the SVD. One λ per run is chosen from $\{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$ by the validation performance of a single randomly initialized probe hidden layer, before LEA starts, and is then used for every candidate throughout the search. With $\lambda = 0$, the solver reproduces the unregularized fit exactly. Ridge is not part of the proposed method; it is a screened auxiliary change (Section 5.4).

Averaging M weakly correlated estimators reduces the variance of the ensemble mean [26,27],

$$\mathrm{Var}\left(\frac{1}{M}\sum_{m=1}^{M}\varepsilon_m\right) \approx \overline{\mathrm{Var}}(\varepsilon)/M \tag{10}$$

under zero-mean, mutually uncorrelated member errors ε_m. Equation (10) does not directly describe RMSE or majority-vote classification error. Section 5.1 reports an empirical M-sweep of those metrics, not a numerical test of the $1/M$ rate.

3.6. Information-Geometric Selection Criteria

The three rules above score candidates either in parameter space ($\|\beta\|$) or by pairwise correlation of validation residuals. Neither uses the fact that β influences a prediction only through the candidate's hidden matrix H. This subsection gives the output stage of a single candidate its Fisher-information reading, states where the resulting Fisher quadratic form is defined, and defines two further post-hoc selection rules, F-norm and F-complementary, applied to the same shortlist $\mathcal{P}$. Every shortlisted candidate c has its own hidden-layer parameters (ω_c, b_c) and therefore its own hidden matrices $H_{\mathrm{tr},c}$ and $H_{\mathrm{va},c}$; the subscript is kept explicit wherever two candidates are compared.

Fisher information of the output stage.

For a single candidate with fixed hidden-layer parameters, and hence a fixed hidden-output matrix H, the ELM output stage (Equation (1)) is a linear-Gaussian observation model in β,

$$T = H\beta + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2 I), \tag{11}$$

under which the log-likelihood of β given (H, T) is quadratic in β and the Fisher information matrix is

$$I(\beta) = \frac{1}{\sigma^2} H^\top H, \tag{12}$$

constant in β because Equation (11) is linear in β with fixed H.

Fisher quadratic form for a shared hidden layer.

For two coefficient vectors β, β' *of the same hidden layer*, Equation (12) defines the quadratic form

$$q_I(\beta, \beta') = (\beta - \beta')^\top I(\beta)\,(\beta - \beta') = \frac{1}{\sigma^2}\,\|H(\beta - \beta')\|^2, \tag{13}$$

a Mahalanobis-type form with weight matrix $H^\top H/\sigma^2$. It measures separation of the *predicted outputs* $H\beta$ and $H\beta'$, scaled by the noise level, not separation of the raw coefficient vectors. Because $H^\top H$ is only positive semidefinite, q_I vanishes whenever $\beta - \beta'$ lies in the null space of H: $\sqrt{q_I}$ is a seminorm of $\beta - \beta'$ and induces only a pseudometric on coefficient space, which becomes the Fisher–Rao distance of this (flat) model only when H has full column rank. We therefore call Equation (13) a quadratic form rather than a distance. In the shared-H setting, it equals twice the Kullback–Leibler divergence between $\mathcal{N}(H\beta, \sigma^2 I)$ and $\mathcal{N}(H\beta', \sigma^2 I)$, the classical link between Fisher information and KL divergence [37–39]. It is not defined between two shortlisted candidates: their coefficient vectors belong to different models, there is no common $I(\beta)$, and $H_c\beta_m$ is not a prediction of either candidate for $c \neq m$.

Candidates with distinct hidden layers.

Two candidates c and m are therefore compared directly through their predictive distributions on the validation inputs, $\mathcal{N}(H_{\mathrm{va},c}\beta_c, \bar{\sigma}^2 I)$ and $\mathcal{N}(H_{\mathrm{va},m}\beta_m, \bar{\sigma}^2 I)$, whose KL divergence is

$$D(c, m) = \frac{\|H_{\mathrm{va},c}\beta_c - H_{\mathrm{va},m}\beta_m\|_F^2}{2\,\bar{\sigma}^2_{cm}}, \qquad \bar{\sigma}^2_{cm} = \tfrac{1}{2}\big(\hat{\sigma}_c^2 + \hat{\sigma}_m^2\big). \tag{14}$$

D is symmetric, $D(c, c) = 0$, and it is well-defined for any pair of candidates because it involves only their predictions. When $H_{\mathrm{va},c} = H_{\mathrm{va},m} = H_{\mathrm{va}}$ and the two variances coincide, it reduces to $\frac{1}{2}q_I(\beta_c, \beta_m)$ of Equation (13) evaluated with the validation hidden matrix H_{va}, i.e., with the Fisher information of the validation design, not with H_{tr}; in general, it is a divergence between predictive distributions, not a quantity on a single parameter manifold.

The common variance $\bar{\sigma}^2_{cm}$ is a deliberate modeling choice. With candidate-specific variances, the symmetric (Jeffreys) divergence between $\mathcal{N}(\mu_c, \hat{\sigma}_c^2 I_n)$ and $\mathcal{N}(\mu_m, \hat{\sigma}_m^2 I_n)$, with μ the n stacked validation outputs, is

$$J(c, m) = \frac{1}{2}\Big(\frac{1}{\hat{\sigma}_c^2} + \frac{1}{\hat{\sigma}_m^2}\Big)\|\mu_c - \mu_m\|^2 + \frac{n}{2}\Big(\frac{\hat{\sigma}_c^2}{\hat{\sigma}_m^2} + \frac{\hat{\sigma}_m^2}{\hat{\sigma}_c^2} - 2\Big). \tag{15}$$

Its second term compares the two candidates' residual levels, which the validation-performance term of the greedy score already accounts for, and it would dominate for large

n. Equation (14) keeps only a location term, with one pooled variance, so the diversity term measures disagreement between predictions alone; it is symmetric by construction and is the exact KL divergence between the two equal-variance predictive Gaussians it names. For classification, β is fitted to one-hot targets (Algorithm 1, line 3), so $H\beta$ is a continuous matrix and Equation (14) applies with the Frobenius norm.

Estimating σ^2.

σ^2 is not known and is not treated as a free hyperparameter. For each candidate m in the shortlist $\mathcal{P}$, $\hat{\sigma}_m^2$ is the pooled residual variance of that candidate's own validation predictions,

$$\hat{\sigma}_m^2 = \mathrm{Var}\big(\,\mathrm{vec}(H_{\mathrm{va},m}\beta_m - T_{\mathrm{va}})\big), \tag{16}$$

i.e., the sample variance of every entry of the validation residual matrix, pooled across validation examples and output dimensions. Each $\hat{\sigma}_m^2$ is floored away from zero for numerical stability, and the pair variance $\bar{\sigma}_{cm}^2$ in Equation (14) is the arithmetic mean of the two floored estimates.

F-norm rule.

It scores each shortlisted candidate by the norm of its *predicted training output* rather than its raw coefficients,

$$s_{\text{F-norm}}(m) = \|H_{\mathrm{tr},m}\beta_m\|_F = \Big(\sigma^2\ \mathrm{tr}\big(\beta_m^\top I_m(\beta_m)\,\beta_m\big)\Big)^{1/2}, \qquad I_m(\beta_m) = \frac{1}{\sigma^2}\,H_{\mathrm{tr},m}^\top H_{\mathrm{tr},m}, \tag{17}$$

the Fisher-weighted length of β_m within its own model (Equation (12) with $H = H_{\mathrm{tr},m}$; the trace sums over output columns when β_m is a matrix), and selects the M candidates in $\mathcal{P}$ with the smallest score, with the same tie-breaking as β-norm. Up to the factor σ, this is the Fisher–Rao norm that Liang et al. [33] propose as a capacity measure for neural networks, restricted to the output stage. The rule was designed by analogy with the β-norm, as a small-solution preference measured with Equation (12) instead of the Euclidean norm. Proposition 1 shows that the analogy fails for least-squares output weights: $H_{\mathrm{tr},m}\beta_m$ is the projection of the training targets onto the hidden-feature space, so its norm measures training fit, not capacity.

Proposition 1. *Let $H \in \mathbb{R}^{N\times\tilde{N}}$, $T \in \mathbb{R}^{N\times m}$, and $\beta = H^\dagger T$. Then*

$$\|H\beta\|_F^2 = \|T\|_F^2 - \|T - H\beta\|_F^2. \tag{18}$$

Consequently, over any set of candidates fitted to the same targets T, ordering by increasing $\|H\beta\|_F$ is identical to ordering by decreasing training residual $\|T - H\beta\|_F$: selecting the M smallest F-norm scores selects the M worst training fits.

Proof. $H\beta = HH^\dagger T = \Pi T$, where $\Pi = HH^\dagger$ is the orthogonal projector onto the column space of H. Hence $T - H\beta = (I - \Pi)T$ is orthogonal to $H\beta$ column by column, and the Pythagorean identity gives $\|T\|_F^2 = \|\Pi T\|_F^2 + \|(I - \Pi)T\|_F^2$, which is Equation (18). Within a run, $T = T_{\mathrm{tr}}$ is the same for every candidate, so $\|H\beta\|_F^2$ is a strictly decreasing function of the residual. □

Proposition 1 holds for any hidden mapping and any data; it does not depend on FAF or on the conditioning of $H_{\mathrm{tr}}^\top H_{\mathrm{tr}}$. Conditioning does, however, separate the two norm rules. Writing $H_{\mathrm{tr},m} = U\Sigma V^\top$, $\|\beta_m\|^2 = \sum_i (u_i^\top T)^2/s_i^2$ grows without bound as small singular values s_i approach zero, whereas $\|H_{\mathrm{tr},m}\beta_m\|^2 = \sum_i (u_i^\top T)^2$ does not depend on the s_i at all. β-norm therefore penalizes candidates whose hidden features are nearly collinear or

poorly resolved, and the F-norm cannot see this; Appendix A reports the measured spectra. F-norm is thus a theoretically motivated rule whose output-space form provably inverts its intent on a least-squares shortlist: among the $2M$ candidates with the best validation RMSE, it keeps the M that fit the training targets least well. The identity does not hold for β-norm, because $\|\beta\|$ is not a function of the residual. It would also not hold for ridge-regularized output weights (Equation (9)), for which $H\beta$ is no longer an orthogonal projection of T. Section 5.3 verifies Equation (18) numerically on every shortlisted candidate and reports its test-set consequence.

F-complementary rule.

It replaces only the complementarity term of greedy-complementary (Section 3.5) with the predictive divergence of Equation (14). The greedy loop, the fixed 0.5/0.5 mix with the min–max normalized validation-performance term, the neutral first-member complementarity value of 0.5, and the tie-breaking rule are otherwise identical to greedy-complementary. For a remaining candidate c and the already-selected set S, the complementarity term at each greedy step is the mean of $D(c, m)$ over $m \in S$, min–max normalized across the remaining candidates. Where the correlation term of greedy-complementary asks whether two candidates' validation *errors* move together, D asks how far apart two candidates' validation *predictions* are relative to their noise level; correlation is scale-invariant in the residuals while D is not, so the two terms are not interchangeable in value.

Motivation and scope.

In the shared-H special case, Equation (12) does not depend on which β is retained, so choosing among coefficient vectors of one hidden layer is a choice of output-space locations $H\beta$ at fixed information. That reading is what motivated scoring candidates in output space. It is motivation only: shortlisted LEA-EELM candidates do not share a hidden layer, no common Fisher information exists across them, and neither rule minimizes a bound on discarded information. What the two rules actually compute is stated by Proposition 1 for F-norm and by Equation (14) for F-complementary.

3.7. Prediction and Aggregation

For a test input x, regression predictions are the mean of the M member outputs. Classification predictions are a majority vote of arg max class outputs, with ties broken by the lowest class index. Member β values used at test time are those fitted on the training split during candidate evaluation; they are not refit on validation or test data.

3.8. Implementation Summary

Each of the P candidates is evaluated $L + 1$ times (initial scoring plus L iterations). An evaluation forms the hidden matrix on the training split and computes β by pseudoinverse through FAF (or the chosen alternative activation). Greedy selection adds a validation pass over a pool of size $2M$ for M steps. All experiments use CPU NumPy/SciPy (Section 4.4).

The equations above are the implemented method and are used for every reported result. The sign departures from the original LEA paper described in Section 3.2 are part of that method, and every result is conditional on them.

4. Experimental Setup

4.1. Datasets, Splits, and Metrics

Nine public benchmarks are used: four regression tasks (Sinc, Abalone, Boston Housing, California Housing) and five classification tasks (Vehicle, Diabetes, Image Segmenta-

tion, Satellite Image, Letter). Table 1 records the source of each. Retrieval and preprocessing live in the accompanying repository (see the Data Availability Statement).

Boston Housing contains a variable constructed from the proportion of Black residents, which has documented ethical problems, and the dataset has been deprecated and removed from scikit-learn for that reason; it is retained here only for comparability with prior ELM literature.

Rows with missing values are dropped. Abalone's categorical `Sex` column is dropped (7 input features). Features are z-score standardized with mean and standard deviation fit on the training split only. Classification labels are integer-encoded and one-hot encoded for majority vote. Regression targets are z-score standardized as the primary convention; a min-max $[0, 1]$ check leaves ranking conclusions unchanged. Train/validation/test splits are redrawn independently each repetition (random for regression; class-stratified for classification). There is no single split ratio: each dataset uses fixed nominal train/validation/test sizes that partition all of its rows (Table 1), with training fractions between 50% and 75%. For classification, the nominal sizes are applied as class-wise proportions, so the realized counts differ from the nominal ones by a few rows (nominal validation sizes 410, 1609, and 4000 for Image Segmentation, Satellite, and Letter; the Image Segmentation sizes are first rescaled to its 2310 rows). The reported test metrics are RMSE (regression) and majority-vote accuracy (classification).

Table 1. Public benchmark datasets with the realized train/validation/test sizes of the primary benchmark (Regr.: regression; Class.: classification). Percentages are rounded and need not sum to 100.

Dataset	Task	Source	Rows (Train/Val/Test)	Split (%)
Sinc	Regr.	Synthetic $y = \sin(x)/x$, generated in code (seeded)	5000/2500/2500	50/25/25
Abalone	Regr.	UCI Abalone	2177/1000/1000	52/24/24
Boston Housing	Regr.	Classic Boston Housing (StatLib mirror)	256/100/150	51/20/30
California Housing	Regr.	scikit-learn California Housing loader	12,640/4000/4000	61/19/19
Vehicle	Class.	UCI Statlog (Vehicle Silhouettes)	446/100/300	53/12/35
Diabetes	Class.	UCI Pima Indians Diabetes	576/96/96	75/13/13
Image Segmentation	Class.	UCI Image Segmentation (concatenated data + test; 2310 rows)	1491/413/406	65/18/18
Satellite Image	Class.	UCI Statlog (Landsat Satellite)	3216/1609/1610	50/25/25
Letter	Class.	UCI Letter Recognition	12,001/3997/4002	60/20/20

Exact duplicate rows occur in Image Segmentation (9.70%; 224 of 2310) and Letter (6.66%; 1332 of 20,000). A split that does not drop them first can place copies on both sides of a boundary. The primary benchmark keeps them for comparability with prior work; Section 5.7 repeats both datasets with duplicates removed before splitting.

4.2. Baselines

Regression and classification use different baseline families, with the same split protocol, seeds, and N as LEA-EELM on each dataset. Regression uses five baselines (GE-ELM, EE-ELM, ELM, Bagging, AdaBoost; six methods including LEA-EELM). Classification uses seven (RAFELM, NFELM, GE-ELM, EN-ELM, ELM, Bagging, AdaBoost; eight methods including LEA-EELM). NFELM, RAFELM, and EN-ELM are classification-only here; EE-ELM is regression-only. GE-ELM [5], EE-ELM [12], NFELM [20], RAFELM [21], and EN-ELM are reconstructions from their source papers: no public reference implementations were available. In particular, EE-ELM is implemented as a validation-selected ensemble of random ELMs (the LEA-EELM pipeline with no search iterations) and EN-ELM as an unselected ensemble of independently initialized ELMs; neither is a verified reimplementation of a published method. Rankings against these five baselines are therefore rankings against our reconstructions, and the external-comparison conclusions rest partly on them. Bagging [18] and AdaBoost use ELM weak learners. Plain ELM follows Huang et al. [1].

4.3. Repetitions, Statistics, and Experiment Map

Table 2 maps every experiment to its saved sample size, configuration, and purpose. The primary benchmark (default LEA-EELM configuration) and the selection confirmation use $N = 15$ independent repetitions on all nine datasets. Seeds are seed $= 42 +$ run_index.

For each reported cell, we give the mean, sample standard deviation, and a 95% t interval. External comparisons use paired Wilcoxon signed-rank tests of the default LEA-EELM configuration against each external baseline, with Holm–Bonferroni correction within the dataset's baseline family. Selection comparisons use the same test of each greedy configuration against the default β-norm configuration, Holm-adjusted within the two-rule family of that dataset ($\alpha = 0.05$); when the two information-geometric rules are added (Section 5.3), the family is the four non-baseline rules of that dataset.

Table 2. Experiment map. The primary benchmark and the selection confirmation use $N = 15$ paired runs on every dataset. The greedy and F-rule logs of the six cheaper datasets contain 20 runs; the selection comparisons of Sections 5.2 and 5.3 use only the 15 paired with the default configuration. The F-norm identity check covers the 20/8/4/2 runs (six cheaper datasets/Image Segmentation/Satellite/Letter) saved before the repetition count of the three largest datasets was raised to 15, and was not repeated on the added runs; Proposition 1 is an exact identity, so its outcome does not depend on which runs are checked.

Experiment	Datasets	N	Configuration and Purpose
Primary benchmark	All nine	15	Default LEA-EELM configuration (FAF, β-norm, non-deduplicated loaders); external baselines
Ensemble-size sweep	Abalone, Vehicle, Diabetes	5	Same backbone; $M \in \{1, 5, 10, 20\}$; matched seeds
Selection confirmation	All nine	15	Greedy-val., greedy-compl., F-norm, and F-compl. vs. the default β-norm configuration on the same shortlist, paired by run index
F-norm identity check	All nine	20/8/4/2	Equation (18) on every shortlisted candidate (134 runs; Section 5.3)
Ridge/selection pilot	Abalone, Vehicle, Diabetes	5	Validation-only screen of ridge and both greedy rules
Activation ablation	Four regression; Vehicle, Diabetes	5/3	FAF vs. Gaussian, with and without the bandwidth constraint
Deduplication check	Image Seg., Letter	15	Default configuration, external baselines, and all five rules with exact duplicate rows removed

4.4. Software and Hardware

All experiments use Python 3.9 with NumPy 1.26 and SciPy 1.13 on CPU only. Output weights use NumPy's `linalg.pinv`; random draws use seeded NumPy `Generator` objects.

4.5. Use of AI Tools

Large language model tools were used substantially: for source code, experiment configuration, analysis and plotting scripts, and manuscript drafting beyond copy-editing, including the implementation of the F-norm and F-complementary selection rules (Section 3.6), the experiment script and statistical analysis that produced the information-geometric selection results (Section 5.3), and the drafting of Section 3.6 and its results and discussion text. The corrected F-complementary implementation (Equation (14)), the numerical check of the F-norm identity (Proposition 1), and the corresponding rerun were likewise produced with LLM assistance and are verified by unit tests for symmetry, zero self-divergence, the shared-H reduction to Equation (13), and the least-squares identity. The authors reviewed the generated code, experiment outputs, and text, checked the code

against an automated test suite and the saved per-run files, and take responsibility for the content, method, and conclusions. No LLM is listed as an author.

5. Results

Results follow the questions in Section 1. Unless noted, reported LEA-EELM results use the default configuration (FAF and β-norm selection). Sample sizes are those in Table 2.

5.1. Effect of Ensemble Size

Table 3 and Figure 2 report an $M \in \{1, 5, 10, 20\}$ sweep on Abalone (regression), Vehicle, and Diabetes (classification), $N = 5$ matched seeds. This is an empirical observation of test RMSE and accuracy, not a numerical test of Equation (10).

Moving from a single optimized ELM ($M = 1$) to a moderate ensemble improves every dataset in the sweep. Vehicle shows the largest effect: test accuracy rises from 0.654 at $M = 1$ to 0.729 at $M = 10$. Abalone RMSE improves from 0.705 ($M = 1$) to 0.690 ($M = 5$) and then plateaus through $M = 20$. Diabetes improves more gradually from $M = 1$ through $M = 20$. Gains do not follow a clean $1/M$ decline: they plateau (Abalone, Diabetes) or reverse slightly (Vehicle: 0.729 at $M = 10$ to 0.725 at $M = 20$). Vehicle training accuracy continues to rise with M after test accuracy peaks, which Equation (10) does not model.

The default $M = 10$ used elsewhere is therefore a moderate operating point, not a claim of optimality.

Table 3. Ensemble-size (M) ablation on the LEA-EELM backbone, $M \in \{1, 5, 10, 20\}$, $N = 5$ per cell. Lower RMSE/higher accuracy is better.

Dataset	Task	M	Test Mean	Test Std	N
Abalone	regression	1	0.7054	0.0309	5
Abalone	regression	5	0.6905	0.0261	5
Abalone	regression	10	0.6898	0.0285	5
Abalone	regression	20	0.6919	0.0281	5
Vehicle	classification	1	0.6540	0.0324	5
Vehicle	classification	5	0.7153	0.0313	5
Vehicle	classification	10	0.7287	0.0259	5
Vehicle	classification	20	0.7247	0.0285	5
Diabetes	classification	1	0.7458	0.0350	5
Diabetes	classification	5	0.7458	0.0291	5
Diabetes	classification	10	0.7583	0.0341	5
Diabetes	classification	20	0.7625	0.0474	5

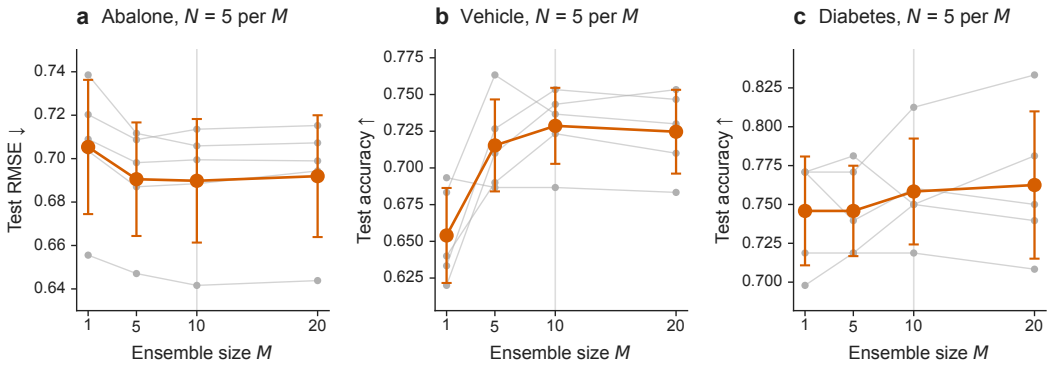


Figure 2. Ensemble-size (M) ablation of LEA-EELM, $M \in \{1, 5, 10, 20\}$. Gray lines and points are five matched seeds; the orange marker and error bar are the mean ± 1 sample standard deviation ($N = 5$ per cell). The light vertical line marks the default $M = 10$. Panel (**a**) is test RMSE (lower is better); panels (**b**,**c**) are test accuracy (higher is better).

5.2. Effect of Ensemble Selection Strategy

We compare three strategies for selecting $M = 10$ ensemble members from the 20 candidates with the lowest validation RMSE: β-norm selection, greedy-validation selection, and greedy-complementary selection. All other components of LEA-EELM are held fixed, allowing the observed differences to be attributed specifically to the ensemble-selection strategy. The comparison is paired by run index (Table 4, Figure 3).

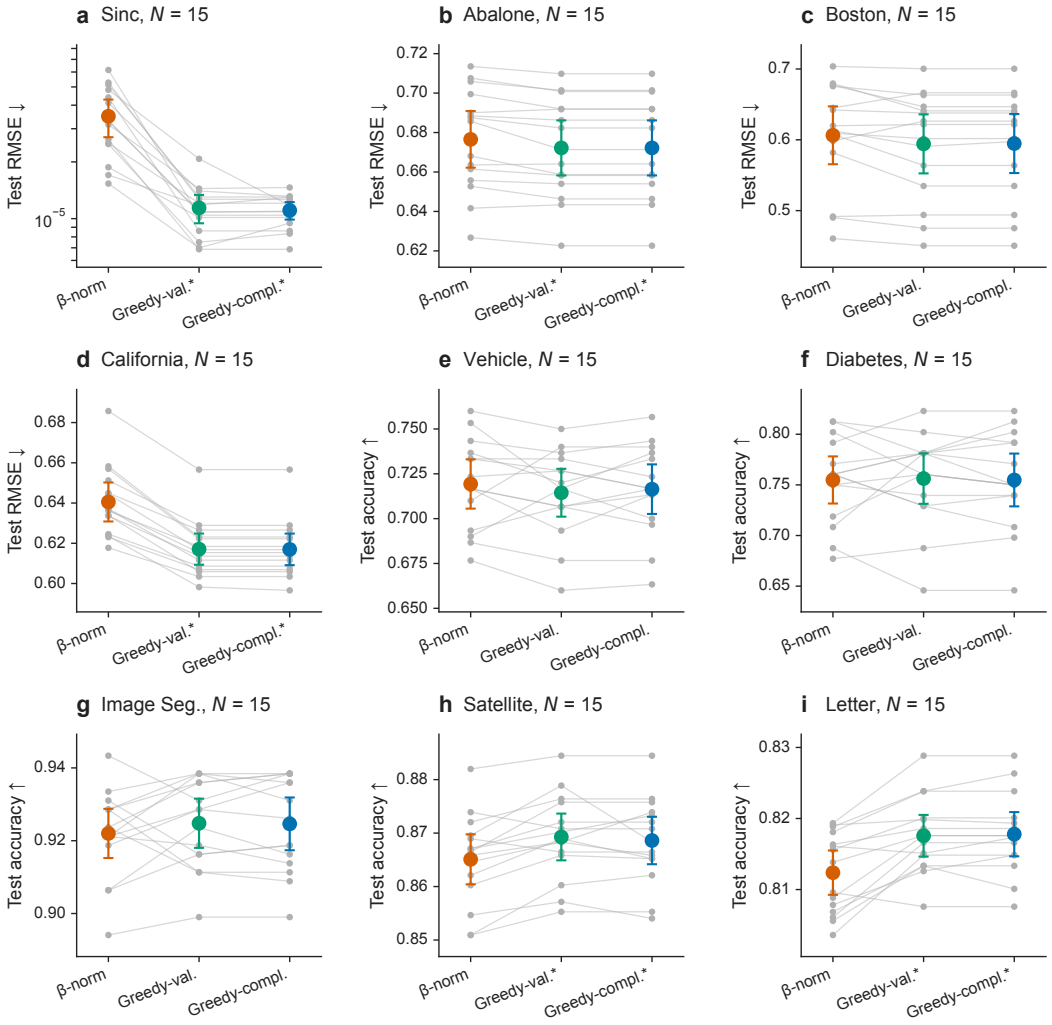


Figure 3. Held-out test confirmation of greedy ensemble selection versus the default β-norm configuration. Gray lines connect matched run indices. Colored markers are means; error bars are 95% t intervals on the paired subset ($N = 15$ on every dataset). An asterisk marks a paired Wilcoxon test versus the default β-norm configuration that remains significant after Holm correction across the two greedy rules ($\alpha = 0.05$). Panels (**a**–**d**): test RMSE; (**e**–**i**): test accuracy. Unpaired greedy logs contain 20 runs on the cheaper datasets and are not mixed in.

Relative to the default β-norm configuration, validation-guided greedy selection significantly reduces test RMSE on three of the four regression datasets. The two greedy strategies produce nearly identical results, suggesting that most of the improvement comes from validation-guided forward selection rather than from the additional complementarity term. Holm-adjusted tests within the two-rule family give Sinc $3.49 \times 10^{-5} \rightarrow 1.14 \times 10^{-5}$, Abalone $0.677 \rightarrow 0.672$, and California Housing $0.641 \rightarrow 0.617$. Boston Housing moves in the same direction ($0.606 \rightarrow 0.594$; raw Wilcoxon $p = 0.035$–0.041) but is not significant after Holm adjustment ($p = 0.071$).

Table 4. Greedy-validation and greedy-complementary selection (no ridge) versus the default β-norm configuration, held-out test mean RMSE/accuracy, paired by run index. N is the matched subset: 15 on every dataset (the greedy logs of the six cheaper datasets contain 20 runs; only the 15 paired with the default configuration are used). Holm–Bonferroni p is the step-down adjusted value across the two greedy rules within each dataset ($\alpha = 0.05$); the larger of the two adjusted values is reported in every row (they differ only on Satellite: 0.0031 for greedy-validation and 0.0067 for greedy-complementary). Boston Housing raw Wilcoxon $p = 0.035$–0.041 becomes Holm-adjusted $p = 0.071$ and is not significant. Lower RMSE/higher accuracy is better. The validation-only pilot is reported in Section 5.4.

Dataset	Task	β-Norm	Greedy-Val.	Greedy-Compl.	Holm-Adj. p	Result	N
Sinc	Regression	3.49×10^{-5}	1.14×10^{-5}	1.11×10^{-5}	1.2×10^{-4}	Significant	15
Abalone	Regression	0.6765	0.6722	0.6722	0.0023	Significant	15
Boston Housing	Regression	0.6064	0.5943	0.5947	0.071	Not significant	15
California Housing	Regression	0.6405	0.6171	0.6170	1.2×10^{-4}	Significant	15
Vehicle	Classification	0.7193	0.7144	0.7164	0.26	Not significant	15
Diabetes	Classification	0.7549	0.7562	0.7549	1.00	Not significant	15
Image Segmentation	Classification	0.9220	0.9248	0.9246	0.78	Not significant	15
Satellite	Classification	0.8651	0.8693	0.8686	0.0067	Significant	15
Letter	Classification	0.8124	0.8176	0.8178	6.1×10^{-4}	Significant	15

On classification, the outcome depends on the dataset. Both greedy rules significantly improve held-out accuracy on Satellite (0.8651 $\rightarrow$ 0.8693 and 0.8686) and Letter (0.8124 $\rightarrow$ 0.8176 and 0.8178), the two datasets with the largest validation splits (1609 and 4000 rows); the gains are about 0.4–0.5 percentage points, and the Letter gain does not survive duplicate removal (Section 5.7). Vehicle trends slightly negative, Diabetes is indistinguishable ($p > 0.9$), and Image Segmentation is not significant (Holm-adjusted $p = 0.78$). A validation-only pilot on Abalone, Vehicle, and Diabetes (Section 5.4) showed a clear classification gain for both greedy configurations. That pilot signal does not replicate on test data for Vehicle and Diabetes (Section 6).

5.3. Effect of Information-Geometric Selection

Table 5 extends the comparison of Section 5.2 with the two information-geometric rules of Section 3.6, F-norm and F-complementary, under the identical protocol: same 9 datasets, same N per dataset, same top-$2M$ shortlist, paired by run index against the default β-norm configuration. All five rules are applied to the one shortlist produced by a single LEA search per run. Because two more rules are now compared against the same baseline, Holm–Bonferroni correction is applied within a four-method family per dataset rather than the two-method family of Table 4; this does not change which datasets are significant for greedy-validation/greedy-complementary (Sinc, Abalone, California Housing, Satellite, and Letter are significant under both family sizes; the other four are significant under neither).

F-complementary behaves like the two existing greedy rules on regression: it significantly reduces test RMSE relative to β-norm on Sinc ($3.49 \times 10^{-5} \rightarrow 1.13 \times 10^{-5}$), Abalone (0.6765 $\rightarrow$ 0.6728), and California Housing (0.6405 $\rightarrow$ 0.6197), and moves in the same direction on Boston Housing without reaching significance (0.6064 $\rightarrow$ 0.5975, raw $p = 0.064$). Its means sit within the spread of the two greedy rules on every regression dataset. F-norm does not replicate this pattern: it is significantly worse than β-norm on Abalone, Boston Housing, and California Housing, and non-significantly worse on Sinc. On classification, F-complementary significantly improves accuracy on Satellite (0.8651 $\rightarrow$ 0.8683) and Letter (0.8124 $\rightarrow$ 0.8178), as the two greedy rules do, and on no other dataset; F-norm does not differ significantly from β-norm on any classification dataset.

Table 5. Selection rules versus the default β-norm configuration, held-out test mean RMSE/accuracy, paired by run index. Greedy-validation and greedy-complementary reproduce Table 4; F-norm (Equation (17)) and F-complementary (Equation (14)) are the information-geometric rules of Section 3.6. All five rules are applied to the same top-2M shortlist of the same LEA search in each run. $N = 15$ paired runs on every dataset. Significance uses a paired Wilcoxon test, Holm–Bonferroni step-down adjusted within a four-method family per dataset (all four non-baseline rules against the shared β-norm baseline, $\alpha = 0.05$)—a wider family than the two-method correction in Table 4. An asterisk (*) marks a statistically significant *improvement* over β-norm; a dagger (†) marks a statistically significant *deterioration* relative to β-norm. Unmarked entries are not significant after correction. Lower RMSE/higher accuracy is better.

Dataset	Task	β-Norm	Greedy-Val.	Greedy-Compl.	F-Norm	F-Compl.	N
Sinc	Regression	3.49×10^{-5}	1.14×10^{-5} *	1.11×10^{-5} *	4.16×10^{-5}	1.13×10^{-5} *	15
Abalone	Regression	0.6765	0.6722*	0.6722 *	0.6796 †	0.6728 *	15
Boston Housing	Regression	0.6064	0.5943	0.5947	0.6220 †	0.5975	15
California Housing	Regression	0.6405	0.6171 *	0.6170 *	0.6457 †	0.6197 *	15
Vehicle	Classification	0.7193	0.7144	0.7164	0.7124	0.7271	15
Diabetes	Classification	0.7549	0.7562	0.7549	0.7375	0.7472	15
Image Segmentation	Classification	0.9220	0.9248	0.9246	0.9194	0.9236	15
Satellite	Classification	0.8651	0.8693 *	0.8686 *	0.8658	0.8683 *	15
Letter	Classification	0.8124	0.8176 *	0.8178 *	0.8133	0.8178 *	15

Proposition 1 accounts for the F-norm result, and the saved runs confirm it numerically. The check covers the 134 runs saved before the repetition count of the three largest datasets was raised: 20 runs on each of the six cheaper datasets and 8, 4, and 2 runs on Image Segmentation, Satellite, and Letter (2680 shortlisted candidates). It was not repeated on the $N = 15$ runs behind Table 5. Proposition 1 is an exact algebraic identity, so its outcome does not depend on which runs are checked; the check only confirms that the floating-point implementation follows it. On these runs, Equation (18) holds with a maximum relative error of 1.9×10^{-5} (the median of the per-run maxima is 3×10^{-15}). On the eight non-synthetic datasets, the F-norm ranking of the shortlist coincides exactly with the ranking by decreasing training RMSE (Spearman correlation 1 in all 114 runs), and the M members F-norm selects are exactly the M candidates with the highest training RMSE. On Sinc, the hidden matrix is numerically singular, and the scores differ only at rounding level, so the agreement is slightly imperfect (Spearman 0.93–1.00, mean 0.990; at least 9 of the 10 selected members coincide). F-norm is therefore worse than β-norm because it keeps the worse-fitting half of the shortlist, not because of any property of FAF. Appendix A reports the spectra of $H_{\mathrm{tr}}^{\top} H_{\mathrm{tr}}$: on the three real regression datasets, FAF candidates are far better conditioned than Gaussian ones and their hidden units are not more correlated.

A direct paired comparison answers whether F-complementary improves on greedy-validation at an equal computational budget. Both rules select from the same shortlist of the same LEA search in every run, so the candidate-generation budget is identical, and the selection step costs well under a second for every rule (Appendix A). On the 15 paired runs per dataset, F-complementary is not significantly better than greedy-validation on any of the nine datasets after Holm correction across datasets, and it is significantly worse on California Housing (0.6197 vs. 0.6171; worse in 14 of 15 runs). The only dataset where it is ahead in most runs, Vehicle (0.7271 vs. 0.7144, better in 12 of 15 runs, raw $p = 0.026$), does not survive the correction. The same holds against greedy-complementary. The divergence term therefore adds no measurable benefit over established greedy selection.

The asymmetry between the two new rules follows from their construction. F-complementary retains the greedy validation-performance term that drives most of the improvement in greedy-validation/greedy-complementary and replaces only the complementarity term; F-norm has no validation-performance term at all, exactly like β-norm, and its score is tied to training fit by Equation (18).

5.4. Effect of Output-Weight Regularization

Table 6 and Figure 4 screen ridge (Equation (9)) on the three development datasets, $N = 5$, validation metrics only. Adding ridge to the default β-norm configuration is worse than no ridge on all three datasets; adding it on top of either greedy configuration is worse than the matching no-ridge rule. This is not a contradiction of $\lambda = 0$ being in the grid. The tuning step chooses λ on one random probe hidden layer, not on the LEA candidates, and that λ then changes every fitness value and, hence, the whole LEA trajectory, so the ridge and no-ridge pipelines select from different candidate pools. We verified the procedure directly on the pilot runs: with $\lambda = 0$, the pipeline reproduces the unregularized validation metric and the same ensemble members in all 15 runs, and in the 4 runs where the probe chose $\lambda = 0$, the ridge result equals the no-ridge result exactly; the probe chose $\lambda > 0$ in the other 11. The screen therefore tests this particular tuning procedure, not ridge tuned per candidate, which remains untested. Ridge was not carried into the test-set confirmation and is not recommended for this pipeline in this form.

Table 6. Validation-only pilot (3 development datasets, 5 seeds) for ridge regularization and greedy ensemble selection. Adding ridge to the default LEA-EELM configuration is worse than the matching no-ridge variant on every dataset. Greedy selection without ridge improves validation RMSE/accuracy relative to the default β-norm configuration on all three datasets; those no-ridge greedy configurations were advanced to the held-out test confirmation in Table 4.

Method (Validation Metric)	Abalone (RMSE)	Vehicle (Acc.)	Diabetes (Acc.)
Plain ELM	0.6993	0.7620	0.7208
Ridge ELM	0.6892	0.7760	0.7417
Ensemble of random ridge ELMs	0.6827	0.7900	0.7333
LEA-EELM (β-norm)	0.6786	0.7360	0.7562
LEA-EELM + ridge	0.6822	0.7220	0.7312
LEA-EELM + greedy-val. (no ridge)	0.6713	0.8040	0.8063
LEA-EELM + greedy-val. + ridge	0.6749	0.7880	0.8000
LEA-EELM + greedy-compl. (no ridge)	0.6713	0.8140	0.8042
LEA-EELM + greedy-compl. + ridge	0.6749	0.7960	0.7958
Random search, matched budget	0.6697	0.7620	0.7688

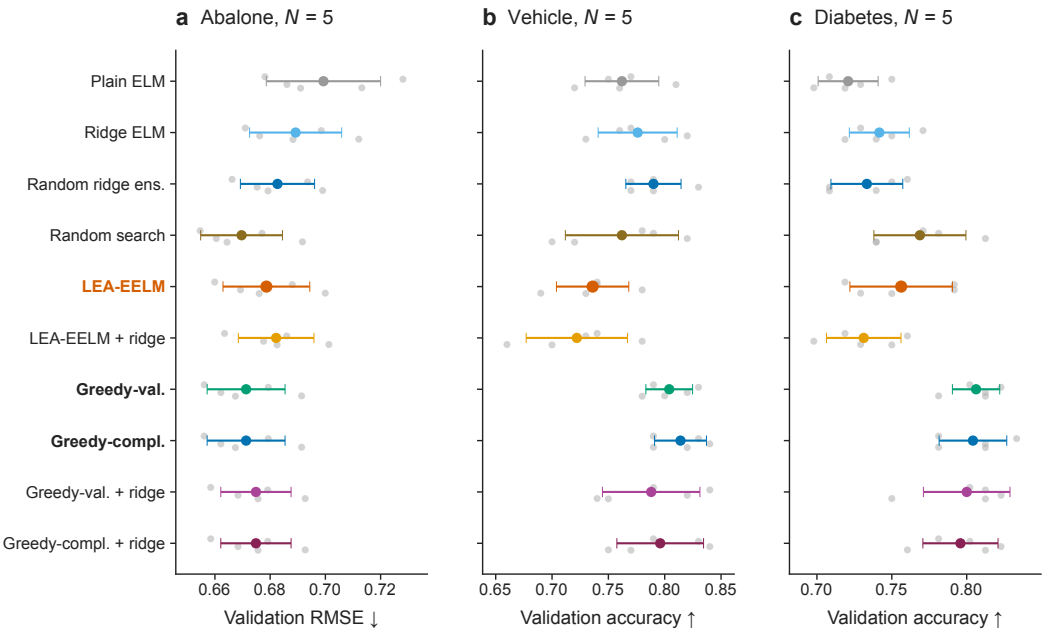


Figure 4. Validation-only pilot on the three development datasets ($N = 5$). Markers are means; error bars are ± 1 sample standard deviation; small gray points are individual seeds. Each color identifies

one method (row label); LEA-EELM (β-norm) is drawn in orange. Panel (**a**): validation RMSE; (**b**,**c**): validation accuracy. Arrows on the axis labels indicate the direction of better performance (↓ lower is better; ↑ higher is better). These are not held-out test results. Ridge is worse than the matching no-ridge variant in every panel. Both greedy configurations without ridge improve the validation metric relative to the default β-norm configuration, including on Vehicle and Diabetes, where the test confirmation does not reproduce the gain.

5.5. Activation-Function Analysis

Table 7 and Figure 5 compare FAF with a bandwidth-constrained Gaussian activation on an otherwise identical LEA-EELM backbone. FAF is input-adaptive and well-defined (Equations (5)–(7)); the empirical comparison is task-dependent and does not show FAF dominating.

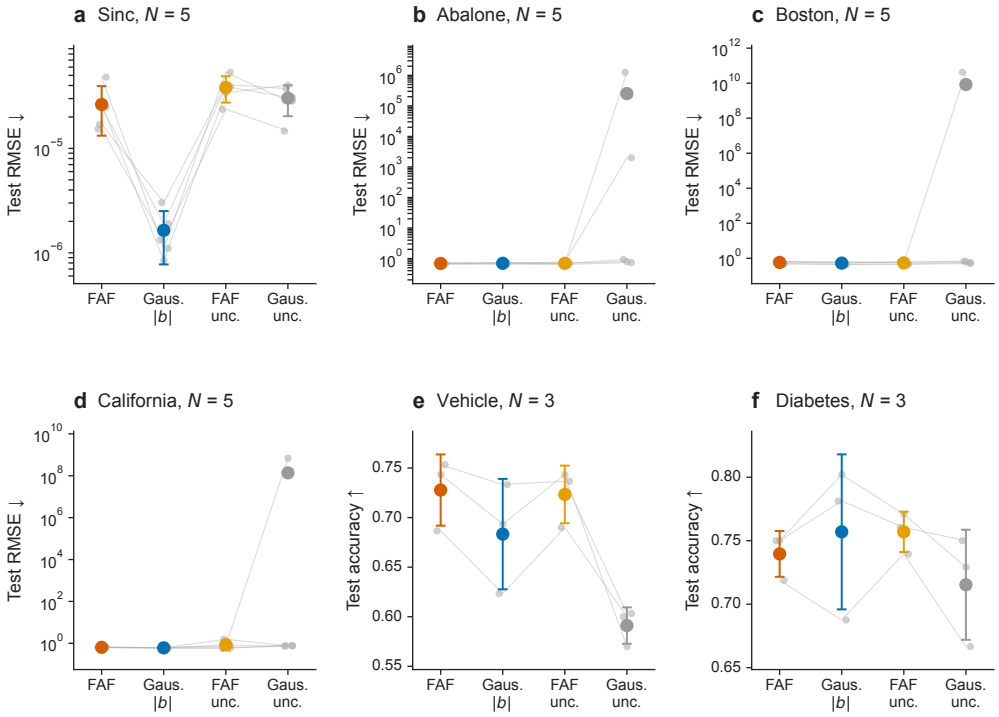


Figure 5. Activation ablation on an otherwise identical LEA-EELM backbone. Every run enters the plotted mean, matching Table 7. Filled markers are means; error bars are ± 1 sample standard deviation when the lower bound is positive. Gray lines connect the same seed. Regression panels use a logarithmic RMSE axis; classification uses linear accuracy. Regression $N = 5$; classification $N = 3$.

Constrained Gaussian attains lower test RMSE on Sinc, Boston Housing (0.529 vs. 0.578), and California Housing (0.607 vs. 0.644); FAF is slightly better on Abalone (0.690 vs. 0.698). On classification, FAF is higher on Vehicle (0.728 vs. 0.683) and constrained Gaussian is higher on Diabetes (0.757 vs. 0.740). Constraining Gaussian bandwidth to $|b|$ prevents unconstrained-Gaussian RMSE blow-ups on Abalone (mean 2.5×10^5), Boston Housing (8.4×10^9), and California Housing (1.4×10^8); Sinc remains finite without the constraint (mean 3.0×10^{-5}). Every run is included in the plotted means.

5.6. Comparison with External Baselines

Tables 8 and 9 and Figures 6 and 7 report the default LEA-EELM configuration against the task-specific external baseline families of Section 4.2. LEA-EELM is within the spread of the baselines on Sinc, Abalone, and Diabetes and ranks in the lower half elsewhere. Full per-method means, standard deviations, and paired Holm-adjusted Wilcoxon tests are in those tables and in Table 10: 37 of 55 comparisons (67.3%) are significant. Of these, 29 favor the external baseline and 8 favor LEA-EELM (NFELM on Image Segmentation, Satellite,

and Letter; ELM on Sinc and Letter; Bagging on Sinc; AdaBoost on Sinc and Abalone). The default configuration ranks sixth of eight on Image Segmentation, Satellite, and Letter.

Table 7. Activation ablation on the LEA-EELM backbone: FAF vs. Gaussian, each with and without the implementation constraint described in Section 3.8. Test mean (std); lower RMSE/higher accuracy is better.

Dataset	Task	Activation	Test Mean	Test Std	N
Sinc	regression	FAF (unconst.)	3.825×10^{-5}	1.083×10^{-5}	5
Sinc	regression	FAF	2.636×10^{-5}	1.313×10^{-5}	5
Sinc	regression	Gaussian (unconst.)	3.028×10^{-5}	9.922×10^{-6}	5
Sinc	regression	Gaussian ($\|b\|$)	1.643×10^{-6}	8.68×10^{-7}	5
Abalone	regression	FAF (unconst.)	0.6956	0.04121	5
Abalone	regression	FAF	0.6898	0.02845	5
Abalone	regression	Gaussian (unconst.)	2.521×10^{5}	5.626×10^{5}	5
Abalone	regression	Gaussian ($\|b\|$)	0.6975	0.02847	5
Boston Housing	regression	FAF (unconst.)	0.5416	0.07579	5
Boston Housing	regression	FAF	0.5783	0.0835	5
Boston Housing	regression	Gaussian (unconst.)	8.36×10^{9}	1.869×10^{10}	5
Boston Housing	regression	Gaussian ($\|b\|$)	0.5287	0.07812	5
California Housing	regression	FAF (unconst.)	0.8639	0.4025	5
California Housing	regression	FAF	0.6444	0.01213	5
California Housing	regression	Gaussian (unconst.)	1.374×10^{8}	3.073×10^{8}	5
California Housing	regression	Gaussian ($\|b\|$)	0.6067	0.008079	5
Vehicle	classification	FAF (unconst.)	0.7233	0.0291	3
Vehicle	classification	FAF	0.7278	0.0360	3
Vehicle	classification	Gaussian (unconst.)	0.5911	0.0184	3
Vehicle	classification	Gaussian ($\|b\|$)	0.6833	0.0557	3
Diabetes	classification	FAF (unconst.)	0.7569	0.0159	3
Diabetes	classification	FAF	0.7396	0.0180	3
Diabetes	classification	Gaussian (unconst.)	0.7153	0.0434	3
Diabetes	classification	Gaussian ($\|b\|$)	0.7569	0.0610	3

Table 8. Regression benchmark, test RMSE (lower is better). Each cell is the mean across N independent runs; the sample standard deviation is in parentheses. LEA-EELM uses the default β-norm configuration (Algorithm 1). External baselines do not use the FAF activation or the LEA growth-radius term. Primary benchmark: default configuration, non-deduplicated loaders, $N = 15$.

Dataset	LEA-EELM	GE-ELM	EE-ELM	ELM	Bagging	AdaBoost	N
Sinc	3.50×10^{-5} (1.43×10^{-5})	1.87×10^{-5} (8.20×10^{-6})	3.11×10^{-5} (8.70×10^{-6})	8.68×10^{-4} (8.18×10^{-4})	7.37×10^{-4} (3.79×10^{-4})	4.66×10^{-4} (4.04×10^{-4})	15
Abalone	0.677 (0.026)	0.701 (0.077)	0.669 (0.022)	0.679 (0.026)	0.671 (0.023)	0.730 (0.024)	15
Boston Housing	0.606 (0.074)	0.419 (0.045)	0.424 (0.041)	0.517 (0.050)	0.449 (0.044)	0.446 (0.046)	15
California Housing	0.641 (0.017)	0.555 (0.011)	0.574 (0.010)	0.603 (0.014)	0.583 (0.010)	0.645 (0.020)	15

Table 9. Classification benchmark, test accuracy (higher is better). Each cell is the mean across N independent runs; the sample standard deviation is in parentheses. LEA-EELM uses the default β-norm configuration (Algorithm 1). Primary benchmark: default configuration, non-deduplicated loaders, $N = 15$ on every dataset.

Dataset	LEA-EELM	RAFELM	NFELM	GE-ELM	EN-ELM	ELM	Bagging	AdaBoost	N
Vehicle	0.719 (0.025)	0.806 (0.023)	0.739 (0.021)	0.792 (0.020)	0.790 (0.020)	0.758 (0.025)	0.784 (0.024)	0.771 (0.024)	15
Diabetes	0.755 (0.042)	0.763 (0.044)	0.738 (0.046)	0.758 (0.033)	0.768 (0.034)	0.756 (0.051)	0.767 (0.037)	0.760 (0.050)	15
Image Seg.	0.922 (0.012)	0.952 (0.010)	0.880 (0.016)	0.928 (0.009)	0.931 (0.007)	0.917 (0.012)	0.930 (0.012)	0.924 (0.012)	15
Satellite	0.865 (0.008)	0.873 (0.009)	0.822 (0.007)	0.880 (0.006)	0.877 (0.009)	0.868 (0.008)	0.875 (0.008)	0.865 (0.007)	15
Letter	0.812 (0.006)	0.822 (0.006)	0.754 (0.006)	0.829 (0.008)	0.837 (0.008)	0.798 (0.005)	0.833 (0.008)	0.886 (0.007)	15

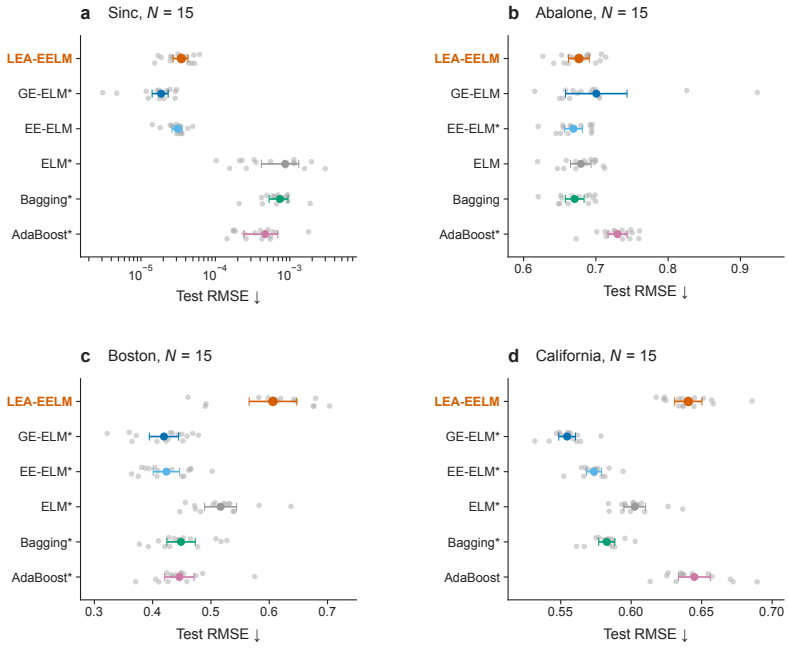


Figure 6. Held-out test RMSE on the four regression datasets (lower is better). Small gray points are independent runs; the colored marker and error bar are the mean and a 95% t interval ($N = 15$). LEA-EELM uses the default β-norm configuration and is drawn in orange. An asterisk marks a paired Wilcoxon test versus LEA-EELM that remains significant after Holm correction within that dataset's external baseline family ($\alpha = 0.05$). Each color identifies one method (row label). The downward arrow (↓) on the axis label indicates that lower RMSE is better. Sinc is logarithmic. Values are not averaged across datasets.

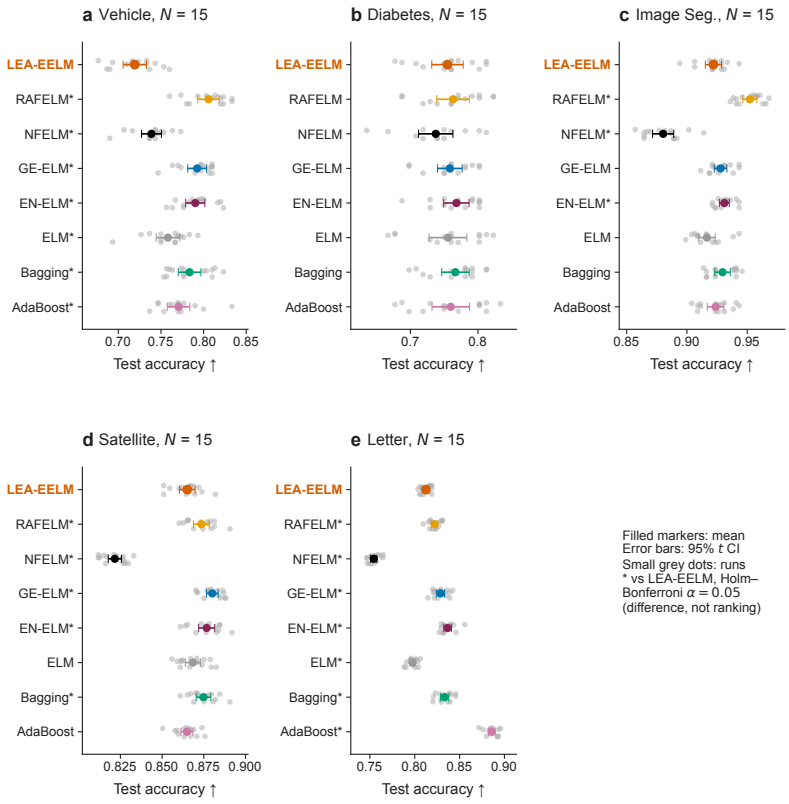


Figure 7. Held-out test accuracy on the five classification datasets (higher is better). Markers, error bars, colors, and asterisks follow Figure 6; LEA-EELM is drawn in orange, and each other color identifies one external baseline (row label). The upward arrow (↑) on the axis label indicates that higher accuracy is better. $N = 15$ on every dataset.

Table 10. Paired Wilcoxon signed-rank tests of the default LEA-EELM configuration (β-norm selection) against each external baseline, Holm–Bonferroni corrected within each dataset's baseline family, $N = 15$ paired runs per dataset. Significance does not imply that LEA-EELM is better: several rejections favor a higher-ranked external baseline (Tables 8 and 9).

Datasets ($N = 15$ Each)	Comparisons	Significant	%
Regression (four datasets)	20	15	75.0
Vehicle, Diabetes	14	7	50.0
Image Segmentation, Satellite, Letter	21	15	71.4
All nine	55	37	67.3

5.7. Robustness and Data-Quality Checks

Table 11 repeats Image Segmentation and Letter with exact duplicate rows removed before splitting, using the final implementation, the default configuration, $N = 15$, and the same seeds; the external baselines and all five selection rules were rerun on the same deduplicated splits. Duplicate removal lowers the accuracy of every method except AdaBoost on Image Segmentation (unchanged) by up to about one percentage point, which is the expected effect of no longer placing copies of a row on both sides of a split. LEA-EELM stays sixth of eight on both datasets; on Image Segmentation, AdaBoost and Bagging swap places, and no other rank changes. After deduplication, the default configuration is still significantly worse than RAFELM, GE-ELM, EN-ELM, and Bagging on Image Segmentation, and than every baseline except ELM and NFELM on Letter.

Duplicate removal does change one selection-stage result. With duplicates removed, no selection rule differs significantly from β-norm on Letter (greedy-validation 0.8103 vs. 0.8074, raw $p = 0.12$; F-complementary 0.8108, raw $p = 0.073$) or on Image Segmentation. The Letter accuracy gain of Section 5.2 therefore does not survive duplicate removal, and Satellite, which contains no exact duplicates, is the only classification dataset on which validation-guided selection is significantly better.

Table 11. Test accuracy and within-dataset rank of the default LEA-EELM configuration and the seven external baselines with and without exact-duplicate-row removal before splitting, Image Segmentation and Letter, $N = 15$ each, same final implementation, seeds, and split sizes (rescaled to the deduplicated row count). Rank Δ = dedup rank − non-dedup rank (positive = fell in rank).

Dataset (Rows)	Method	Non-Dedup Mean	Non-Dedup Rank	Dedup Mean	Dedup Rank	Rank Δ
Image Segmentation (2310→2086)	RAFELM	0.9522	1	0.9420	1	0
	EN-ELM	0.9310	2	0.9256	2	0
	AdaBoost	0.9238	5	0.9238	3	−2
	GE-ELM	0.9279	4	0.9225	4	0
	Bagging	0.9296	3	0.9222	5	+2
	LEA-EELM	0.9220	6	0.9141	6	0
	ELM	0.9166	7	0.9115	7	0
	NFELM	0.8803	8	0.8784	8	0
Letter (20,000→18,668)	AdaBoost	0.8859	1	0.8793	1	0
	EN-ELM	0.8367	2	0.8300	2	0
	Bagging	0.8333	3	0.8285	3	0
	GE-ELM	0.8287	4	0.8201	4	0
	RAFELM	0.8221	5	0.8178	5	0
	LEA-EELM	0.8124	6	0.8074	6	0
	ELM	0.7976	7	0.7880	7	0
	NFELM	0.7544	8	0.7461	8	0

6. Discussion

A single LEA-optimized ELM is still one random-started, analytically fitted model. Averaging several accepted members reduces idiosyncratic error whenever those members are not perfectly correlated. The M-sweep supports that mechanism on the reported metrics:

the step from $M = 1$ to $M = 5$–10 is where the gain appears, and further members add little or slightly reverse it. That plateau is what one expects once remaining candidates share errors, consistent with Equation (10), which serves as motivation, not as a fitted $1/M$ curve. Ensembling is therefore a load-bearing stage of LEA-EELM, not an optional wrapper around an already sufficient point estimate.

β-norm ranks the shortlist by output-weight magnitude. That proxy is cheap and test-blind, but it does not ask whether a candidate improves the ensemble on unseen-within-training data. Greedy-validation does: each added member is chosen to improve partial-ensemble validation RMSE. Relative to the default β-norm configuration, validation-guided greedy selection significantly reduces test RMSE on three of the four regression datasets; Boston Housing moves in the same direction without reaching $\alpha = 0.05$ after adjustment. Because both greedy configurations see the same pool as the default β-norm configuration, the comparison isolates the selection stage.

Greedy-complementary was designed to reward error complementarity on top of the same validation term. In the held-out confirmation, the two greedy traces are almost superimposed. With a $0.5/0.5$ mix that was never tuned, and with a shortlist already restricted to the $2M$ fittest members, the performance term appears to dominate the step-wise choice. Complementary-error scoring remains a well-defined option in the implementation; it is not, on this evidence, an additional source of test-set gain.

The two information-geometric rules of Section 3.6 give a mixed result that is fully explained. F-complementary, which keeps the same greedy validation-performance term as greedy-validation/greedy-complementary and only substitutes the predictive divergence of Equation (14) for the correlation-based complementarity term, reproduces their regression-side significant gains over the β-norm on Sinc, Abalone, and California Housing, and their classification gains on Satellite and Letter (Table 5). This is consistent with the reading in Section 5.2 that the validation-performance term, not the specific form of the diversity term, drives most of the improvement: correlation of residuals and divergence of predictions are different quantities that play the same secondary role in the same greedy loop. F-norm is significantly worse than β-norm on Abalone, Boston Housing, and California Housing, and Proposition 1 gives the reason. With least-squares output weights, $\|H_{\mathrm{tr}}\beta\|^2 = \|T_{\mathrm{tr}}\|^2 - \|T_{\mathrm{tr}} - H_{\mathrm{tr}}\beta\|^2$, so the smallest output-space norms belong to the largest training residuals, and the rule keeps the half of the shortlist that fits the training data worst. The numerical check of Section 5.3 confirms this run by run. The explanation involves neither FAF nor correlation among hidden units: the identity holds for any hidden mapping, and the measured Gram-matrix statistics do not single FAF out. The lesson is specific. An output-space norm is not a regularizer for an unregularized least-squares fit, because its magnitude is fixed by the fit quality; $\|\beta\|$ is not determined by the residual, which is why the β-norm remains a usable heuristic. A meaningful output-space analogue would need ridge output weights or a norm evaluated on data not used for the fit, and is left for future work.

The classification pilot improved validation accuracy under both greedy rules on Vehicle and Diabetes, then lost that gain on the test set, including a slight reversal on Vehicle. Greedy selection makes $M = 10$ sequential choices on a single validation split. Vehicle and Diabetes validation sets have on the order of 100 rows, so a discrete accuracy signal is noisy relative to continuous RMSE on the larger regression splits. With $N = 15$ repetitions, the greedy rules and F-complementary do significantly improve test accuracy on Satellite and Letter, whose validation splits have 1609 and 4000 rows, by about 0.4–0.5 percentage points; Image Segmentation (410 validation rows) shows no significant change. With exact duplicates removed, the Letter gain is no longer significant, so the support for a benefit on large validation sets rests on Satellite alone. This pattern is consistent with

selection overfitting on small validation sets and a real but small benefit on large ones. It is not a controlled test of that explanation: validation-set size is confounded with every other property of five datasets, and direct confirmation (for example, selection scoring by cross-validation or a held-out inner split) is still needed.

FAF supplies a bounded, odd, input-adaptive hidden mapping whose steepness tracks a simple certainty score of the pre-activation. That is a coherent design for regions where a fixed sigmoid or RBF is a poor match. Empirically, a positivity-constrained Gaussian activation is stronger on three of the four regression datasets in the ablation, while FAF is stronger on Vehicle. The activation is therefore a specified component of the framework whose benefit is task-dependent, not a claim of uniform superiority over conventional hidden mappings.

GE-ELM, EE-ELM, and related methods already generate or average multiple ELMs. LEA-EELM employs the previously introduced LEA as that generator. Its distinctive evidence is not that this optimizer beats every such method on every dataset—the external comparison shows that it does not. It is that, given one LEA-optimized candidate population, the way a compact ensemble is selected changes regression test error by a statistically supported margin, and that a moderate ensemble improves on the single best-so-far candidate.

The evidence about the generator itself is unfavorable and should be read plainly. On Boston Housing and Vehicle, the default configuration is significantly worse than a plain ELM (Tables 8 and 9; test RMSE 0.606 vs. 0.517 and test accuracy 0.719 vs. 0.758), and the same holds on California Housing (0.641 vs. 0.603). In the validation pilot (Table 6), random search with a matched evaluation budget is comparable to LEA with β-norm selection on all three development datasets. The implemented search also moves the best validation RMSE of the initial population only slightly (Section 3.2). The contribution of this paper is therefore the selection stage, which is generator-agnostic, not LEA as the best available generator. The selection-stage results are conditional on the candidate pool this search produces, and they remain available to other generators that expose a validation-scored candidate pool, including structure-aware population optimizers such as SOCIAL [40]. A multimodal generator such as M-LEA [11], which preserves several distinct optima, is a natural route to a less correlated shortlist and is left for future work.

This study has several limitations. All primary and selection comparisons use $N = 15$ repetitions; the ensemble-size sweep, ridge pilot, and activation ablation use $N = 5$ or $N = 3$. This bounds the size of effect that can be detected, and some literature baselines had to be reconstructed because reference implementations were unavailable. The benchmark does not include deliberately corrupted or strongly imbalanced datasets, and the FAF hyperparameters and LEA search budget were not exhaustively tuned. The LEA update signs used here differ from the original LEA paper in three places (Section 3.2). On the three development datasets, the canonical additive form reached better validation fitness than the implemented subtractive form (best validation RMSE 0.676 vs. 0.687 on Abalone, 0.325 vs. 0.360 on Vehicle, and 0.435 vs. 0.466 on Diabetes). A rerun of the benchmark with the canonical form is left for future work; the external-baseline results therefore do not characterize LEA as published. Ensemble selection also depends on validation data and may be sensitive to small validation splits, particularly for classification. The primary benchmark keeps the exact duplicate rows; removing them (Section 5.7) leaves every external ranking of LEA-EELM unchanged but removes the Letter selection gain. These limits bound the scope of the supported ensemble-construction findings; they do not reverse them.

7. Conclusions

LEA-EELM is a framework for generating and selectively combining optimized ELM candidates: the previously introduced LEA searches over hidden-layer encodings, FAF supplies an input-adaptive hidden mapping, output weights are fitted analytically, and a compact ensemble is taken from the LEA-optimized candidate population under a validation-only protocol. The principal contribution is that ensemble-construction stage.

A moderate ensemble improves on a single optimized candidate on every dataset in the M-sweep. Relative to the default β-norm configuration, validation-guided greedy selection significantly reduces test RMSE on three of the four regression datasets. The two greedy strategies produce nearly identical results, suggesting that most of the improvement comes from validation-guided forward selection rather than from the additional complementarity term. Casting the output stage of a single candidate as a linear-Gaussian model gives a Fisher-information reading (Section 3.6) whose Fisher quadratic form is defined only for a shared hidden layer and is a semimetric; across candidates, the comparison is a Kullback–Leibler divergence between predictive distributions. Substituting that divergence for the correlation term (F-complementary) reproduces the greedy rules' regression-side gains. Substituting an output-space norm for $\|\beta\|$ (F-norm) is significantly worse than β-norm on three of four regression datasets, and Proposition 1 shows why: for least-squares output weights, the score is a decreasing function of the training residual, so the rule selects the worst training fits in the shortlist. Ridge-regularized output weights do not help. On classification, the greedy rules and F-complementary significantly improve held-out accuracy on the two datasets with the largest validation splits (Satellite and Letter, by about half a percentage point) and on none of the other three, and the Letter gain does not survive removal of duplicate rows; greedy selection can look strong on a small validation split and fail to generalize. Against complete external baseline families, the default configuration is significantly worse than most comparators on five of nine datasets, including plain ELM on Boston Housing, California Housing, and Vehicle; the supported contribution is the selection stage, not the generator. All of these results are conditional on the candidate pool produced by the implemented, non-canonical variant of LEA (Section 3.2); they do not characterize LEA as published.

Ensemble construction is a central determinant of optimized-ELM performance. LEA-based candidate generation and adaptive activation produce a pool; the process by which that pool is reduced to M members is where this pipeline's supported gains appear, particularly for regression.

Author Contributions: Conceptualization, A.A.S., H.T. and M.J.; Methodology, A.A.S., H.T. and M.J.; Software, A.A.S., H.T. and M.J.; Validation, A.A.S., H.T. and M.J.; Formal analysis, H.T. and M.J.; Investigation, M.J.; Data curation, M.J.; Writing—original draft, H.T. and M.J.; Writing—review and editing, H.T. and M.J.; Visualization, A.A.S., H.T. and M.J.; Supervision, H.T. and M.J.; Project administration, M.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This study analyzes only pre-existing, publicly available, de-identified benchmark datasets (UCI Machine Learning Repository, StatLib, and a seeded synthetic generator; Section 4.1); it does not involve human participants or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available on GitHub at https://github.com/MehrdadJalali-AI/LEA-EELN (accessed on 27 October 2026).

Acknowledgments: During the preparation of this manuscript, the authors used general-purpose large language model tools for source code (including the F-norm and F-complementary selection

rules of Section 3.6, the corrected F-complementary implementation, and the F-norm identity check, all verified by unit tests), experiment configuration, statistical analysis, and manuscript drafting beyond copy-editing, as detailed in Section 4.5. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Hidden-Matrix Spectra and Selection Cost

Table A1 summarizes the spectrum of the training Gram matrix $H_{\mathrm{tr}}^{\top} H_{\mathrm{tr}}$ over the $2M = 20$ shortlisted candidates of the first three runs of each regression dataset (same seeds, splits, and LEA configuration as the main experiments), for FAF and for the bandwidth-constrained Gaussian activation. The effective rank is the participation ratio $(\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$ of the eigenvalues λ_i. FAF candidates are far better conditioned than Gaussian ones on the three real regression datasets and spread their energy over more directions, and their hidden units are not more correlated than Gaussian ones. On the synthetic Sinc task, both activations give numerically singular Gram matrices, and FAF units are the more correlated. These measurements contradict the earlier attribution of the F-norm result to FAF-induced correlation and support the conditioning argument of Section 3.6: they matter for $\|\beta\|$ but, by Proposition 1, not for $\|H\beta\|$.

Table A1. Spectrum of $H_{\mathrm{tr}}^{\top} H_{\mathrm{tr}}$ over shortlisted candidates (medians over 60 candidates per row; mean absolute off-diagonal correlation of hidden-unit outputs is a mean). ∞: numerically singular (smallest eigenvalue zero in double precision). Source: `results/gram_spectra.csv`.

Dataset	Activation	$\tilde{N}$	cond$(H^{\top}H)$	Eff. Rank	Top-λ Share	Mean $\lvert$corr$\rvert$
Sinc	FAF	15	2.1×10^{16}	2.1	0.60	0.73
	Gaussian	15	∞	1.4	0.85	0.63
Abalone	FAF	25	902	3.7	0.42	0.38
	Gaussian	25	1.3×10^{8}	1.2	0.93	0.45
Boston Housing	FAF	50	136	13.6	0.19	0.17
	Gaussian	50	∞	1.1	0.95	0.22
California Housing	FAF	20	58	6.9	0.27	0.23
	Gaussian	20	4.6×10^{6}	1.1	0.93	0.27

Table A2 reports the wall-clock cost of each selection rule on the same shortlists. All rules select from the output of one LEA search, so the candidate-generation budget is identical by construction; the selection step itself costs at most about a tenth of a second, one to three orders of magnitude less than the search.

Table A2. Selection cost in milliseconds (median over three runs, each the median of five timings) and LEA search time in seconds, measured on a shared machine while other jobs were running, so absolute values are indicative only. Source: `results/selection_cost.csv`.

Dataset	β-Norm	Greedy-Val.	Greedy-Compl.	F-Norm	F-Compl.	LEA Search (s)
Sinc	0.03	15.3	93.9	32.5	52.3	9.3
Abalone	0.03	18.5	68.9	16.4	24.1	8.6
Boston Housing	0.03	3.4	42.4	3.7	9.5	3.0
California Housing	0.03	35.4	102.4	111.1	82.4	35.3

References

1. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme Learning Machine: Theory and Applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
2. Dalirinia, E.; Jalali, M.; Yaghoobi, M.; Tabatabaee, H. Lotus effect optimization algorithm (LEA): A lotus nature-inspired algorithm for engineering design optimization. *J. Supercomput.* **2024**, *80*, 761–799. [CrossRef]

3. Tabatabaee, H.; Sabzevari, A.A.; Jalali, M. FAF-ELM: Robust ELM with Fuzzy Adaptive Activation Function and Firefly Optimization. *Artif. Intell. Tools Softw. Data Eng.* **2026**, *12*, 73. [CrossRef]
4. Caruana, R.; Niculescu-Mizil, A.; Crew, G.; Ksikes, A. Ensemble selection from libraries of models. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML'04)*; ACM: New York, NY, USA, 2004; p. 18. [CrossRef]
5. Xue, X.; Yao, M.; Wu, Z.; Yang, J. Genetic ensemble of extreme learning machine. *Neurocomputing* **2014**, *129*, 175–184. [CrossRef]
6. Raju, V.A.G.; Dash, P.B.; Mishra, M. A Robust Framework for Smart Grid Stability Prediction: Optimization of Extreme Learning Machine Algorithm. In *Computational Intelligence in Pattern Recognition (CIPR 2024)*; Lecture Notes in Networks and Systems; Springer Nature: Singapore, 2025. [CrossRef]
7. Rasa, G.R.R.; Ebrahimi Mood, S.; Abbasi, E. Increasing the Accuracy of an Extreme Learning Machine Using a Chaos Particle Swarm Optimization Algorithm. *Evol. Syst.* **2026**, *17*, 3. [CrossRef]
8. Syamala Rao, P.; Parthasaradhi Varma, G.; Chinta, D.P.; Gottapu, K.; Hyma Lakshmi, T.V.; Appala Naidu, K.; Saritha, M. Financial Time Series Prediction Using Pelican Optimized Extreme Learning Machine with Reduced Weights. *Comput. Econ.* **2025**, *66*, 4763–4780. [CrossRef]
9. Hammadi, I.; Belhassen, L.; Ayed, L.; Bouabidi, A.; Ameen, A. Experimental Data-Driven Hybrid PSO-ELM Model for Accurate Prediction of Hydraulic Turbine Parameters. *Water* **2026**, *18*, 1446. [CrossRef]
10. Eshtay, M.; Faris, H.; Obeid, N. Metaheuristic-based extreme learning machines: A review of design formulations and applications. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 1543–1561. [CrossRef]
11. Dalirinia, E.; Yaghoobi, M.; Tabatabaee, H.; Chandna, S.; Jalali, M. Multimodal Lotus Effect Algorithm for Engineering Optimization Problems. *Eng. Rep.* **2025**, *7*, e70137. [CrossRef]
12. Liu, N.; Wang, H. Ensemble Based Extreme Learning Machine. *IEEE Signal Process. Lett.* **2010**, *17*, 754–757. [CrossRef]
13. Chen, H.; Chen, H.; Nian, X.; Liu, P. Ensembling extreme learning machines. In *Proceedings of the Advances in Neural Networks—ISNN 2007*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; pp. 1069–1076. [CrossRef]
14. Wang, D.; Alhamdoosh, M. Evolutionary extreme learning machine ensembles with size control. *Neurocomputing* **2013**, *102*, 98–110. [CrossRef]
15. Lu, H.J.; An, C.L.; Ma, X.P.; Zheng, E.H.; Yang, X.B. Disagreement measure based ensemble of extreme learning machine for gene expression data classification. *Jisuanji Xuebao/Chin. J. Comput.* **2013**, *36*, 341–348. [CrossRef]
16. Han, X.; Yuan, H.; Wu, L. Kalman Filter Anomaly Values Processing Meta-Model Ensemble Learning Framework for Lithium-Ion Battery Capacity Prediction. *Energy* **2025**, *322*, 135625. [CrossRef]
17. Zhang, X.; Hu, T.; Wu, Y.; Yang, L.; Wang, J.; He, J. ACDE2LM: Autocorrelation-Constrained Dynamic Ensemble Extreme Learning Machine for Real-Time Fault Diagnostics of Bearings. *Measurement* **2025**, *245*, 116608. [CrossRef]
18. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
19. Mahdavi-Shahri, A.; Houshmand, M.; Yaghoobi, M.; Jalali, M. Applying an ensemble learning method for improving multi-label classification performance. In *Proceedings of the 2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)*; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6. [CrossRef]
20. Zheng, E.; Liu, J.; Lu, H.; Wang, L.; Chen, L. A New Fuzzy Extreme Learning Machine for Regression Problems with Outliers or Noises. In *Proceedings of the Advanced Data Mining and Applications*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; pp. 524–534. [CrossRef]
21. Liu, S.; Feng, L.; Xiao, Y.; Wang, H. Robust Activation Function and its Application: Semi-Supervised Kernel Extreme Learning Method. *Neurocomputing* **2014**, *144*, 318–328. [CrossRef]
22. Wang, L.; Xie, J. Enhancing Extreme Learning Machine Robustness via Residual-Variance-Aware Dynamic Weighting and Broyden-Fletcher-Goldfarb-Shanno Optimization: Application to Metro Crowd Flow Prediction. *Systems* **2025**, *13*, 349. [CrossRef]
23. Xiao, Y.; Adegoke, M.; Leung, C.S.; Leung, K.W. Robust Fault-Aware Extreme Learning Machine Based on Maximum Correntropy. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 19964–19972. [CrossRef] [PubMed]
24. Zhou, G.; Zhang, X.; Wang, T.; Wang, B. A Spatiotemporal Fuzzy Modeling Approach Combining Automatic Clustering and Hierarchical Extreme Learning Machines for Distributed Parameter Systems. *Mathematics* **2025**, *13*, 364. [CrossRef]
25. Noorizadegan, A.; Wang, S.; Ling, L. Making Gaussian Kolmogorov–Arnold networks reliable and accurate. *Neurocomputing* **2026**, *703*, 134771. [CrossRef]
26. Opitz, D.W.; Shavlik, J.W. Actively Searching for an Effective Neural Network Ensemble. *Connect. Sci.* **1996**, *8*, 337–353. [CrossRef]
27. Brown, G.; Wyatt, J.; Harris, R.; Yao, X. Diversity creation methods: A survey and categorisation. *Inf. Fusion* **2005**, *6*, 5–20. [CrossRef]
28. Zhang, C.X.; Zhang, J.S. A survey of selective ensemble learning algorithms. *Jisuanji Xuebao/Chin. J. Comput.* **2011**, *34*, 1399–1410. [CrossRef]
29. Zhang, Y.; Liu, B.; Yu, J. A selective ensemble learning approach based on evolutionary algorithm. *J. Intell. Fuzzy Syst.* **2017**, *32*, 2365–2373. [CrossRef]

30. Li, N.; Yu, Y.; Zhou, Z.H. Diversity Regularized Ensemble Pruning. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; pp. 330–345. [CrossRef]
31. Amari, S.i. Natural Gradient Works Efficiently in Learning. *Neural Comput.* **1998**, *10*, 251–276. [CrossRef]
32. Amari, S.I.; Karakida, R.; Oizumi, M. Fisher Information and Natural Gradient Learning in Random Deep Networks. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (AISTATS), Naha, Okinawa, Japan, 16–18 April 2019*; PMLR: Cambridge, MA, USA, 2019; Volume 89, pp. 694–702.
33. Liang, T.; Poggio, T.; Rakhlin, A.; Stokes, J. Fisher-Rao Metric, Geometry, and Complexity of Neural Networks. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (AISTATS), Naha, Okinawa, Japan, 16–18 April 2019*; PMLR: Cambridge, MA, USA, 2019; Volume 89, pp. 888–896.
34. Theis, L.; Korshunova, I.; Tejani, A.; Huszár, F. Faster gaze prediction with dense networks and Fisher pruning. *arXiv* **2018**, arXiv:1801.05787.
35. Huang, G.; Huang, G.B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Netw.* **2015**, *61*, 32–48. [CrossRef] [PubMed]
36. Wang, J.; Lu, S.; Wang, S.H.; Zhang, Y.D. A review on extreme learning machine. *Multimed. Tools Appl.* **2022**, *81*, 41611–41660. [CrossRef]
37. Fisher, R.A. Theory of Statistical Estimation. *Math. Proc. Camb. Philos. Soc.* **1925**, *22*, 700–725. [CrossRef]
38. Rao, C.R. Information and the Accuracy Attainable in the Estimation of Statistical Parameters. *Bull. Calcutta Math. Soc.* **1945**, *37*, 81–91.
39. Amari, S.I.; Nagaoka, H. *Methods of Information Geometry*; Translations of Mathematical Monographs; American Mathematical Society/Oxford University Press: Providence, RI, USA, 2000; Volume 191.
40. Jalali, M.; Vu, B.; Chandna, S.; Nadimi-Shahraki, M.H. SOCIAL: Social Network Optimization Algorithm via Centrality and Influence-Aware Learning. *Appl. Soft Comput.* **2026**, *194*, 114914. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.