Newton Divided Difference Optimization for Fingerprint-Based Neural Virtual Screening against Avian Influenza A/H9N2

Siti Amiroch[1,*], Mohammad Jamhuri[2], Awawin Mustana Rohmah[3], Mohammad Hamim Zajuli Al Faroby[4], Chairul Anwar Nidom[5], Reviany Vibrianita Nidom[6]

[1,3]*Department of Mathematics, Faculty of Science and Technology, Universitas Islam Darul 'Ulum, Lamongan 62253, Indonesia*

[2]*Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim, Malang 65144, Indonesia*

[4]*Data Science Study Program, Telkom University, Surabaya 60231, Indonesia*

[5]*Faculty of Veterinary Medicine, Airlangga University, Surabaya 60115, Indonesia*

[6]*Pharmacy Study Program, Universitas PGRI Adi Buana Surabaya, Surabaya 60234, Indonesia*

[5,6]*Professor Nidom Foundation, Jl. Jemur Andayani XVIII No. 8 Surabaya, Surabaya 60237, Indonesia*

(Received: April 15, 2026; Revised: May 20, 2026; Accepted: June 22, 2026; Available online: July 12, 2026)

Abstract

Avian influenza A/H9N2 poses persistent zoonotic and veterinary threats, yet efficient computational tools for antiviral compound prioritization remain underdeveloped, particularly with respect to optimizer behavior in high-dimensional neural screening. This study proposes Newton Divided Difference (NDD) optimization, a lightweight positive diagonal curvature-aware training strategy, as a novel optimizer for fingerprint-based neural virtual screening against avian influenza A/H9N2, with the objective of evaluating its performance across aligned molecular fingerprint representations and chemically structured validation protocols. An aligned benchmark of 1,459 molecules consisting of 615 candidate active compounds and 844 decoys was represented by EState (79 features), PubChem (881 features), and Klekota–Roth (4,860 features) fingerprints, sharing identical molecule identities, labels, and split assignments. A fixed multilayer perceptron (MLP) classifier was trained with NDD and seven baseline optimizers under stratified, scaffold-key, and similarity-cluster split protocols across five repeated seeds. NDD achieved the highest descriptive ROC-AUC (Receiver Operating Characteriztic – Area Under the Curve) and PR-AUC (Precision-Recall Area Under the Curve) on Klekota–Roth fingerprints under scaffold-key and similarity-cluster protocols, and remained competitive under the stratified split with ROC-AUC of 0.9872. Architecture-sensitivity tests confirmed stable NDD performance across multiple network configurations, with ROC-AUC values ranging from 0.9876 to 0.9890. Compared with Hessian-free optimization, NDD reduced per-run runtime from approximately 50–54 seconds to approximately 8 seconds on Klekota–Roth under the same CPU-only configuration while achieving comparable ranking performance. The novelty of this work lies in the first systematic assessment of NDD for H9N2 neural virtual screening, demonstrating that positive diagonal curvature-aware scaling provides a practical, stable, and computationally efficient optimization alternative in sparse high-dimensional ligand-based screening settings, although external validation and prospective experimental confirmation remain necessary before practical antiviral prioritization.

Keywords: H9N2 Influenza, Virtual Screening, Molecular Fingerprints, Divided Difference, Curvature-Aware Optimization

1. Introduction

Avian influenza A/H9N2 remains a persistent veterinary and zoonotic threat. The virus is enzootic in poultry across Asia, the Middle East, and parts of Europe and Africa, causing flock mortality, reduced productivity, and substantial economic losses in the poultry sector [1]. Its continuing evolution also raises concern because low-pathogenic avian influenza viruses can contribute to reassortment and cross-species transmission risk [2], [3]. In Indonesia, the detection and molecular characterization of H9N2 further highlight the relevance of continued surveillance and preparedness [4].

Surveillance, vaccination, and biosecurity remain essential control measures, but antiviral candidate discovery is also

***Corresponding author: author (siti.amiroch@unisda.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i3.1416

needed to support preparedness. Direct experimental screening across large chemical spaces is expensive and time-consuming. Computational virtual screening therefore provides an efficient route for prioritizing promising compounds before structural analysis and biological validation [5], [6]. Recent reviews also emphasize that virtual screening is most useful when integrated with downstream mechanistic or experimental confirmation [7], [8].

Ligand-based virtual screening has become a useful strategy when known active and inactive molecules are available. In this setting, predictive performance depends not only on the classifier, but also on molecular representation, validation design, and optimization behavior. These factors are tightly coupled in neural screening models. Compact fingerprints may simplify the learning problem, whereas sparse high-dimensional fingerprints may encode richer chemical fragments but create more difficult optimization landscapes. Thus, an apparent gain in screening performance may arise from the representation, the data split, or the optimizer rather than from the classifier alone.

Previous studies have demonstrated that molecular fingerprint-based machine learning models are capable of discriminating active compounds from inactive molecules or decoys in virtual screening tasks. In particular, gradient boosting approaches such as XGBoost have demonstrated strong predictive performance and computational efficiency for general molecular bioactivity prediction and compound prioritization across diverse target spaces [9]. Fingerprint-based virtual screening has demonstrated applicability in antiviral compound prioritization [10], [11], and network-level analyzes of H9N2 viral protein interactions have further informed the identification of candidate screening targets [11]. These findings collectively demonstrate the feasibility and relevance of computational approaches for antiviral compound discovery. However, no study has systematically evaluated how curvature-aware neural-network optimization behaves across aligned molecular fingerprint representations and chemically structured validation protocols in ligand-based H9N2 virtual screening.

This gap is important because random splitting, representation mismatch, and unstable training can produce optimiztic or difficult-to-interpret results. A robust evaluation should therefore separate the effects of molecular representation, data partitioning, and optimizer behavior. This is especially relevant for sparse high-dimensional fingerprints, where many input coordinates are inactive for most molecules and only a subset of molecular fragments may drive the classification boundary.

Here, we evaluate NDD optimization as a lightweight positive diagonal curvature-aware method for fingerprint-based neural virtual screening against avian influenza A/H9N2. NDD estimates coordinate-wise curvature from successive parameter and gradient differences, constructs a positive diagonal Hessian surrogate via divided-difference quotients, and applies curvature-scaled parameter updates across all p coordinates. This mechanism captures local curvature information in a lightweight manner: it requires only one additional vector of divided-difference values per iteration, avoids full Hessian construction and Hessian-vector products, and does not store any dense curvature matrix. Its additional computational cost is O(p) beyond gradient evaluation, where p denotes the number of trainable parameters.

To the best of our knowledge, this is the first systematic assessment of NDD for H9N2 neural virtual screening across aligned molecular fingerprints and chemically structured validation settings. We construct an aligned benchmark in which EState, PubChem, and Klekota--Roth fingerprints share the same molecules, labels, and split assignments. We compare NDD with seven established optimizers under stratified, scaffold-key, and similarity-cluster protocols. We further evaluate ranking performance, probabilistic error, calibration, runtime, architecture sensitivity, dimensionality scaling, and ablation behavior. The results show that NDD is not universally dominant, but provides its clearest advantage in the sparse high-dimensional Klekota--Roth setting, particularly under chemically structured evaluation.

The remainder of this paper is organized as follows. Section 2 reviews related work on machine-learning-based virtual screening, molecular fingerprints, validation design, and neural optimization. Section 3 presents the dataset, fingerprint construction, split protocols, neural classifier, optimizers, and evaluation metrics. Section 4 reports and discusses the experimental findings. Section 5 concludes the paper.

2. Literature Review

The present study builds on three connected lines of work: machine-learning-based antiviral virtual screening, fingerprint representation and chemical validation, and neural-network optimization. We review these areas to clarify the methodological basis of the study and to position the contribution of NDD in H9N2 compound prioritization.

2.1. Machine Learning in Antiviral Virtual Screening

Virtual screening is widely used to reduce the number of compounds that must be evaluated experimentally. Structure-based approaches use docking, pharmacophore modeling, binding-site analysis, and molecular dynamics to assess

target--ligand compatibility [5], [12]. Ligand-based approaches, by contrast, learn activity patterns from known active and inactive compounds [6]. These approaches are often complementary. Machine learning can rank candidates rapidly, while structure-based modeling and biological assays provide mechanistic and experimental confirmation [7].

Influenza-related studies increasingly use such multi-stage computational workflows. Pharmacophore modeling, docking, and ADME (Absorption, Distribution, Metabolism, and Excretion) analysis have been used to prioritize neuraminidase inhibitor candidates [13]. Other studies have explored virtual screening and validation workflows for influenza antiviral discovery [14]. Machine-learning-based approaches have also introduced attention-based and explainable ensemble models for influenza compound prediction [15], [16].

In the virtual screening setting, fingerprint-based classifiers have been shown to identify candidate active compounds and separate them from decoys with competitive performance. Gradient boosting approaches such as XGBoost have demonstrated strong predictive performance and computational efficiency for molecular bioactivity prediction and compound prioritization [9]. Fingerprint-based virtual screening methods have also demonstrated value in antiviral compound prioritization and target-focused screening campaigns [10], [11], and target-level analyzes of H9N2 viral protein interaction networks have further clarified which protein targets merit inclusion in ligand-based screening [11]. However, these studies primarily emphasized predictive modeling, classifier comparison, and target identification. None directly evaluates how neural-network optimization behaves across aligned fingerprint spaces and chemically structured validation protocols.

2.2. Molecular Fingerprints and Validation Design

Molecular fingerprints convert chemical structures into numerical vectors that can be processed by machine-learning models. EState fingerprints encode electrotopological atom-level information [17]. PubChem fingerprints represent predefined substructural patterns and are widely used in chemical informatics workflows[18]. Klekota--Roth fingerprints describe a large collection of chemical fragments, producing sparse high-dimensional vectors [19], [20]. These representations differ in compactness, sparsity, and fragment coverage, and these differences can directly affect neural learning.

A reliable comparison across fingerprints requires the same molecules, labels, and split assignments. Otherwise, performance differences may reflect sample composition rather than representation quality. This issue is especially relevant in active--decoy virtual screening designs, where decoys are selected to challenge the model while preserving physicochemical comparability [21].

Validation design is equally important. Stratified random splits are useful for baseline evaluation, but they may place structurally similar molecules in both training and testing sets. Scaffold-based splitting provides a more demanding test by separating molecules according to chemical scaffolds [22]. Similarity-cluster splitting offers another structured evaluation strategy by grouping molecules according to molecular similarity before partitioning [23]. These protocols are important because prospective virtual screening often requires generalization to compounds that are not close analogs of the training molecules.

2.3. Neural Optimization and Research Gap

First-order optimizers such as Adam and AdamW are widely used because they are efficient and robust in many neural-network settings [26]. However, they rely mainly on gradient information and may not fully address uneven scaling across sparse fingerprint dimensions. Curvature-aware methods can improve update scaling, but classical Newton-type and quasi-Newton methods often require expensive curvature computation, matrix storage, or inverse-Hessian approximation [24], [25]. Standard numerical optimization references also highlight the computational trade-offs between first-order, Newton-type, and quasi-Newton strategies [26].

Several modern approaches seek to reduce the cost of curvature-aware training. Hessian-free and inexact second-order strategies avoid explicit full Hessian construction but may still require additional computation that is non-trivial in repeated experiments [27]. Gauss--Newton variants provide another route for curvature-informed neural optimization, although they remain more involved than purely diagonal update scaling [28]. These trade-offs motivate lightweight alternatives that retain some curvature information without imposing full second-order cost.

Sparse high-dimensional molecular fingerprints provide a clear motivation for lightweight curvature-aware optimization. Many coordinates are inactive for most molecules, whereas a smaller subset of fragments may strongly influence the decision boundary. NDD targets this setting by estimating positive diagonal curvature from divided differences between consecutive parameter and gradient values. This mechanism enables coordinate-wise update

scaling without full second-order computation.

3. Materials and Methods

The study was designed as a controlled fingerprint-based neural virtual screening experiment. EState, PubChem, and Klekota–Roth experiments used identical molecules, labels, ordering, and split assignments. This alignment reduces sample-composition effects. It does not, however, create identical optimization landscapes. The fingerprint dimension changes the first-layer input size and the number of trainable parameters. Performance differences are therefore interpreted as the combined effect of molecular representation, induced parameter dimension, and optimizer behavior. The complete experimental workflow is summarized in figure 1.

3.1. Dataset Construction and Fingerprint Alignment

The positive class consisted of candidate active or target-associated compounds compiled from H9N2-related experimental studies and publicly reported antiviral screening results in the literature. The negative class consisted of decoys generated using a DUD-E (Directory of Useful Decoys, Enhanced)-style strategy [21]. We use the term *candidate active* because this study evaluates retrospective ligand-based classification and optimizer behavior, not direct experimental confirmation of antiviral activity. Three fingerprint representations were used: $p_{\mathrm{EState}} = 79$, $p_{\mathrm{PubChem}} = 881$, $p_{\mathrm{KR}} = 4860$.

These representations were selected to examine optimizer behavior across increasing input dimension and different descriptor structures. EState is compact, PubChem encodes broader substructure patterns, and Klekota–Roth provides a high-dimensional fragment-based representation [17], [18], [19], [20]. The pipeline consists of fingerprint extraction and benchmark alignment, split protocol construction, fixed MLP classification, NDD optimization, and performance–efficiency evaluational.

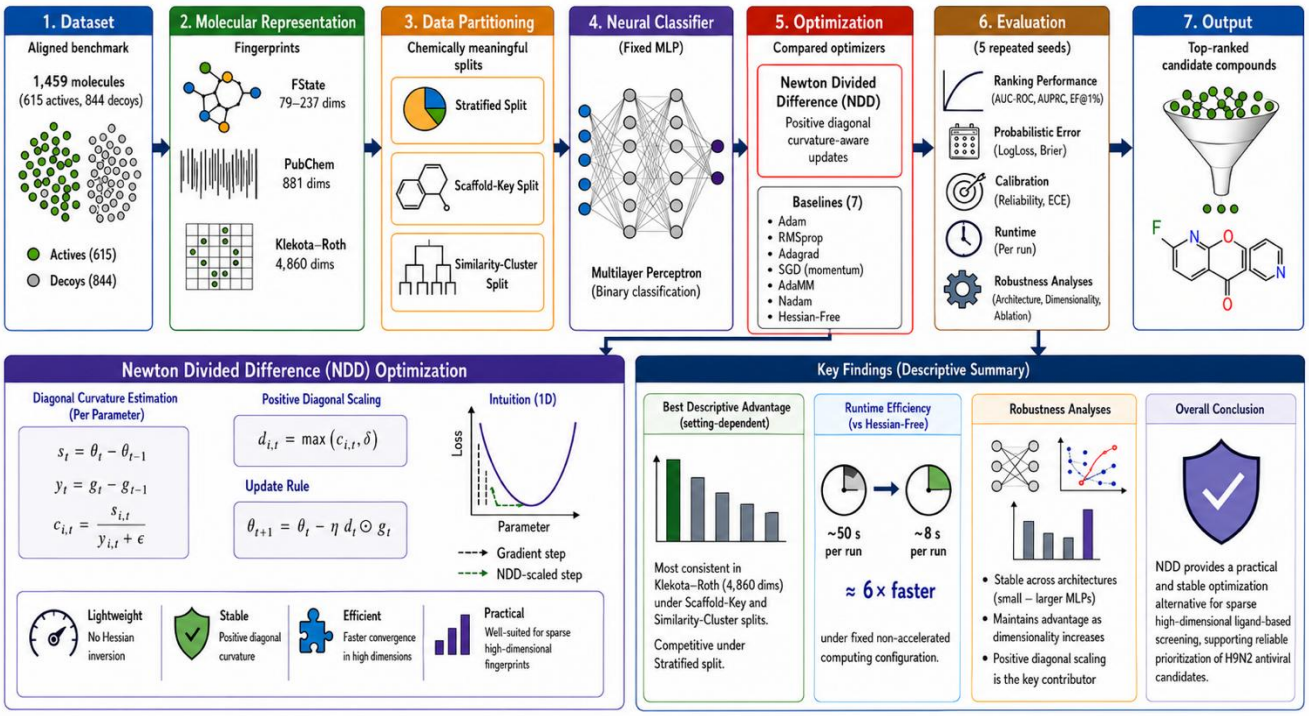

Figure 1. Overall methodological workflow of the proposed NDD-based neural virtual screening framework.

Molecular strings were standardized before alignment. When RDKit parsing was available, SMILES (Simplified Molecular-Input Line-Entry System) strings were canonicalized into canonical isomeric SMILES. If parsing failed, the standardized text form was retained. For each molecule, an alignment key was constructed from the canonical SMILES string and binary label:

$$\kappa(m_i) = \mathrm{SMILES}_{\mathrm{canon}}(m_i) \| y_i, \tag{1}$$

$\|$ denotes string concatenation. Let $\mathcal{K}^{(r)}$ be the set of alignment keys in fingerprint representation r. The common

 ISSN 2723-6471

aligned key set was

$$\mathcal{K}_{\text{align}} = \bigcap_{r\in\{\text{EState,PubChem,KR}\}} \mathcal{K}^{(r)}. \quad (2)$$

Molecules with conflicting labels across representations were removed. Duplicate records were removed using the alignment key. The remaining molecules were sorted by the same key so that all fingerprint matrices shared identical row order.

The standardization procedure was limited to RDKit parsing, canonical isomeric SMILES generation when parsing succeeded, duplicate removal, and conflicting-label removal. Canonical isomeric SMILES preserves explicitly encoded stereochemical information, but it does not infer missing or ambiguous stereochemistry. No additional tautomer canonicalization, salt stripping, charge neutralization, or stereoisomer imputation was applied unless already handled by the input representation. Thus, the alignment removes exact standardized-key mismatches and label conflicts, but it does not collapse all tautomeric, salt, or stereochemical equivalents into one chemical identity.

Table 1 shows that all representations contain the same 1459 molecules, 615 candidate active compounds, 844 decoys, and positive-class ratio. Figure 2 illustrates the fixed class composition and the increasing feature dimension from EState to Klekota–Roth. This design fixes molecular identity, labels, row order, and split assignments across fingerprint representations. It controls sample-composition effects, but it does not make the induced neural optimization problems identical. Larger fingerprint spaces also induce larger first-layer weight matrices and larger trainable parameter spaces.

Table 1. Aligned fingerprint benchmark summary.

Fingerprint	Samples	Candidate active	Decoy	Positive ratio	Features	Unique SMILES	Removed label conflicts
EState	1459	615	844	0.4215	79	1459	0
PubChem	1459	615	844	0.4215	881	1459	0
Klekota–Roth	1459	615	844	0.4215	4860	1459	0

3.2. Split Protocols

We evaluated three split protocols: stratified split, scaffold-key split, and similarity-cluster split. The stratified split used an exact 60:20:20 train–validation–test allocation. The grouped protocols used the same target allocation, but partition sizes were allowed to deviate to preserve group integrity. For each seed, split assignments were generated on the aligned reference dataset and reused across all fingerprint spaces. The stratified split preserved the class ratio across training, validation, and test partitions. The aligned dataset was first divided into train–validation and test subsets using a test proportion of 0.20. The train–validation subset was then divided using the adjusted validation proportion $\frac{0.20}{1-0.20} = 0.25$, yielding the intended 60:20:20 allocation. Class composition is identical across EState, PubChem, and Klekota–Roth, while dimensionality increases from 79 to 881 and 4860 features.

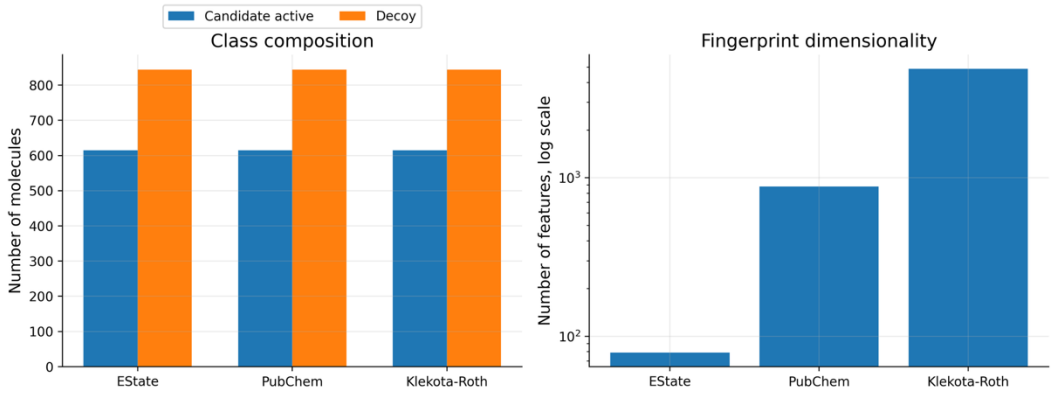

Figure 2. Aligned fingerprint benchmark overview.

The scaffold-key split used the Bemis–Murcko scaffold abstraction implemented in RDKit. For each canonical SMILES string, a scaffold key was generated using MurckoScaffoldSmiles. If parsing failed or an empty scaffold was

returned, the canonical SMILES string was retained as the grouping key. Group-based partitioning used GroupShuffleSplit, first for the test set and then for the validation set. Molecules sharing the same scaffold key were assigned to the same partition. If a grouped split failed to contain both classes in all partitions, the pipeline reverted to a stratified fallback and recorded the effective split mode. Thus, the scaffold-key split refers specifically to RDKit Bemis–Murcko scaffold-string grouping, not hashed scaffold identifiers or alternative scaffold abstractions. The similarity-cluster split used RDKit Morgan fingerprints with radius 2 and 2048 bits. Pairwise molecular similarity was computed using the Tanimoto coefficient and converted into distance:

$$d_{\text{Tanimoto}}(m_i, m_j) = 1 - \text{sim}_{\text{Tanimoto}}(m_i, m_j). \tag{3}$$

Clusters were generated using the Butina algorithm with distance cutoff 0.40. This approximately groups molecules with Tanimoto similarity at least 0.60, depending on cluster structure. The cluster labels were then used as grouping variables in GroupShuffleSplit. This protocol reduces overly optimiztic evaluation caused by highly similar molecules appearing in both training and test sets [22], [23]. Fallback behavior was recorded when grouped splitting failed to preserve both classes in all partitions. Thus, the similarity-cluster split refers specifically to Butina clustering of Morgan-fingerprint Tanimoto distances with distance cutoff 0.40, followed by group-based partitioning.

Table 2 shows that the stratified split preserves the global positive-class ratio. The scaffold-key and similarity-cluster splits show larger variation in partition size and positive-class ratio because chemical group integrity was preserved. This behavior is expected in grouped chemical evaluation.

Table 2. Split quality summary across repeated runs.

Split protocol	Partition	Size	Positive ratio
Stratified	Train	875	0.4217 ± 0.0000
Stratified	Validation	292	0.4212 ± 0.0000
Stratified	Test	292	0.4212 ± 0.0000
Scaffold-key	Train	836	0.4045 ± 0.0274
Scaffold-key	Validation	288	0.4011 ± 0.0820
Scaffold-key	Test	335	0.4579 ± 0.0883
Similarity-cluster	Train	819	0.3786 ± 0.0462
Similarity-cluster	Validation	309	0.4504 ± 0.0439
Similarity-cluster	Test	331	0.4775 ± 0.1124

3.3. Neural Classifier

Let

$$\mathcal{D}_r = \left\{\left(\mathbf{x}_i^{(r)}, y_i\right)\right\}_{i=1}^{n}, \qquad \mathbf{x}_i^{(r)} \in \mathbb{R}^{p_r}, \qquad y_i \in \{0,1\}, \tag{4}$$

denote the aligned binary classification dataset under fingerprint representation r. The label $y_i = 1$ denotes a candidate active molecule, and $y_i = 0$ denotes a decoy. Across fingerprint representations, molecules and labels were identical; only the fingerprint dimension p_r changed. A fixed two-hidden-layer feed-forward neural classifier mapped each fingerprint vector to a scalar logit. The model was

$$\mathbf{h}_1 = \tanh\left(\mathbf{W}_1^{(r)}\mathbf{x}^{(r)} + \mathbf{b}_1\right), \quad \mathbf{W}_1^{(r)} \in \mathbb{R}^{128\times p_r}, \quad \mathbf{b}_1 \in \mathbb{R}^{128}, \tag{5}$$

$$\mathbf{h}_2 = \tanh(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2), \quad \mathbf{W}_2 \in \mathbb{R}^{64\times 128}, \quad \mathbf{b}_2 \in \mathbb{R}^{64}, \tag{6}$$

$$z = g_{\boldsymbol{\theta}}^{(r)}\left(\mathbf{x}^{(r)}\right) = \mathbf{w}_3^{\top}\mathbf{h}_2 + b_3, \quad \mathbf{w}_3 \in \mathbb{R}^{64}. \tag{7}$$

The predicted active-class probability and threshold-dependent prediction were

 ISSN 2723-6471

$$\hat{y} = \sigma(z) = \frac{1}{1+\exp(-z)}, \qquad \hat{c}_\tau = \mathbb{I}[\hat{y} \geq \tau], \qquad \tau \in (0,1). \tag{8}$$

The 128–64 architecture was used as a fixed base classifier to provide a consistent optimizer-comparison setting across fingerprints. It was not selected through exhaustive architecture search and was not intended to be optimal for every representation. Architecture-sensitivity tests with smaller and larger networks and a ReLU variant were therefore included. The trainable parameter vector was

$$\boldsymbol{\theta}^{(r)} = \mathrm{vec}\left(\mathbf{W}_1^{(r)}, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2, \mathbf{w}_3, b_3\right). \tag{9}$$

Because hidden-layer widths were fixed, the number of trainable parameters depended on the fingerprint dimension:

$$\begin{aligned} p_\theta^{(r)} &= \underbrace{128p_r}_{w_1^{(r)}} + \underbrace{128}_{b_1} + \underbrace{64 \cdot 128}_{w_2} + \underbrace{64}_{b_2} + \underbrace{64}_{w_3} + \underbrace{1}_{b_3} \\ &= 128p_r + 8449. \end{aligned} \tag{10}$$

Thus, the classifier contained 18,561 parameters for EState, 121,217 for PubChem, and 630,529 for Klekota–Roth. This scaling keeps the classifier structure consistent, but it also couples fingerprint dimension with neural parameter dimension. Therefore, improved Klekota–Roth performance should be interpreted as evidence that NDD is effective in the larger and sparser parameterized problem induced by Klekota–Roth fingerprints, not as an optimizer-only or dimension-only effect. For the i-th sample,

$$z_i = g_{\boldsymbol{\theta}}^{(r)}\left(\mathbf{x}_i^{(r)}\right), \qquad \hat{y}_i = \sigma(z_i). \tag{11}$$

The binary cross-entropy objective was

$$\mathcal{L}\left(\boldsymbol{\theta}^{(r)}\right) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)], \tag{12}$$

or equivalently, in logit form,

$$\mathcal{L}\left(\boldsymbol{\theta}^{(r)}\right) = \frac{1}{n}\sum_{i=1}^{n}\left[\log\left(1+\exp(z_i)\right) - y_i z_i\right]. \tag{13}$$

The implementation used the numerically stable form

$$\ell(z_i, y_i) = \max(z_i, 0) - y_i z_i + \log\left(1+\exp(-|z_i|)\right), \qquad \mathcal{L}\left(\boldsymbol{\theta}^{(r)}\right) = \frac{1}{n}\sum_{i=1}^{n}\ell\,(z_i, y_i). \tag{14}$$

The full-batch gradient was

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}\left(\boldsymbol{\theta}^{(r)}\right) = \frac{1}{n}\sum_{i=1}^{n}(\sigma(z_i) - y_i)\,\nabla_{\boldsymbol{\theta}} z_i. \tag{15}$$

All optimizers minimized the same full-batch objective, so optimizer differences reflected update mechanisms rather than minibatch sampling noise. At iteration k, the generic update was

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha_k \mathbf{d}_k, \tag{16}$$

$\mathbf{d}_k$ is the update direction and $\alpha_k > 0$ is either selected by line search or determined by the optimizer-specific learning-rate rule. In the reported NDD implementation, curvature clipping, damping, direction clipping, and Armijo line search were used. Direction clipping replaced $\mathbf{d}_k$ by

$$\tilde{\mathbf{d}}_k = \mathbf{d}_k \min\left\{1, \frac{c_d}{\| \mathbf{d}_k \|_2}\right\}. \tag{17}$$

No separate logit-clipping bound or parameter-projection bound was exported in the reported configuration.

3.4. Newton Divided Difference Optimization

NDD constructs a positive diagonal curvature surrogate from local gradient-displacement information. Let

$$\mathbf{g}_k = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_k)$$

be the full-batch gradient at iteration k. Consecutive parameter and gradient displacements are

$$\mathbf{s}_k = \boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}, \qquad \mathbf{y}_k = \mathbf{g}_k - \mathbf{g}_{k-1}. \tag{18}$$

For coordinate j, the raw divided-difference estimate is

$$\tilde{b}_{k,j} = \frac{|y_{k,j}|}{|s_{k,j}| + \varepsilon}, \tag{19}$$

$\varepsilon > 0$ prevents division by zero. The positive diagonal scaling coefficient is

$$b_{k,j} = \mathrm{clip}\left(\tilde{b}_{k,j} + \lambda, b_{\min}, b_{\max}\right), \tag{20}$$

$\lambda \geq 0$ is damping and $0 < b_{\min} < b_{\max}$. The NDD direction is

$$d_{k,j} = -\frac{g_{k,j}}{b_{k,j}}, \qquad j = 1, \ldots, p. \tag{21}$$

Equivalently,

$$\mathbf{d}_k = -\mathbf{B}_k^{-1}\mathbf{g}_k, \qquad \mathbf{B}_k = \mathrm{diag}\left(b_{k,1}, \ldots, b_{k,p}\right). \tag{22}$$

The absolute values in Equation (15) are intentional. NDD does not enforce the signed secant relation used in standard quasi-Newton updates and should not be interpreted as diagonal BFGS. Instead, it uses the magnitude of local gradient variation as a positive coordinate-wise scaling surrogate. This discards signed curvature information, including negative curvature, so the method cannot exploit negative-curvature directions near saddle regions. The trade-off is stable positive scaling, preserved descent, and low computational cost. Because $b_{k,j} > 0$, NDD gives a descent direction whenever $\mathbf{g}_k \neq \mathbf{0}$:

$$\mathbf{g}_k^{\top}\mathbf{d}_k = -\sum_{j=1}^{p} \frac{g_{k,j}^2}{b_{k,j}} < 0. \tag{23}$$

Direction clipping multiplies $\mathbf{d}_k$ by a positive scalar no larger than one. It therefore preserves the sign of $\mathbf{g}_k^{\top}\mathbf{d}_k$.

The direction was used in an Armijo line search. Given $c \in (0,1)$ and $\rho \in (0,1)$, the step size was selected from $\{1, \rho, \rho^2, \ldots\}$ until

$$\mathcal{L}(\boldsymbol{\theta}_k + \alpha_k \mathbf{d}_k) \leq \mathcal{L}(\boldsymbol{\theta}_k) + c\alpha_k \mathbf{g}_k^{\top}\mathbf{d}_k. \tag{24}$$

The update was

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha_k \mathbf{d}_k. \tag{25}$$

The method uses only coordinate-wise arithmetic beyond gradient evaluation, so the additional memory and arithmetic cost is $O(p)$.

Proposition 1. (*Descent and stationarity under Armijo line search*). Assume that $\mathcal{L}$ is continuously differentiable, bounded below on the generated level set, and has Lipschitz-continuous gradient on that level set. Assume also that

$$0 < b_{\min} \leq b_{k,j} \leq b_{\max} < \infty$$

for all k and j, and that α_k is chosen by the Armijo rule in Equation (20). Then each nonstationary NDD direction is a descent direction. Moreover, under the standard Armijo backtracking lower-bound condition for accepted step sizes,

$$\lim_{k \to \infty} \|\nabla \mathcal{L}(\boldsymbol{\theta}_k)\|_2 = 0.$$

Proof. The descent property follows from Equation (19). Since $b_{k,j} \leq b_{\max}$,

 ISSN 2723-6471

$$-\mathbf{g}_k^{\top}\mathbf{d}_k = \sum_{j=1}^{p} \frac{g_{k,j}^2}{b_{k,j}} \geq \frac{1}{b_{\max}} \| \mathbf{g}_k \|_2^2.$$

The Armijo condition gives

$$\mathcal{L}(\boldsymbol{\theta}_{k+1}) \leq \mathcal{L}(\boldsymbol{\theta}_k) - c\alpha_k \frac{1}{b_{\max}} \| \mathbf{g}_k \|_2^2.$$

If accepted step sizes are bounded below by a positive constant on the generated level set, summing this inequality and using lower boundedness of $\mathcal{L}$ yields

$$\sum_{k=0}^{\infty} \| \mathbf{g}_k \|_2^2 < \infty.$$

Hence $\| \mathbf{g}_k \|_2 \to 0$. This is a stationarity result for a nonconvex objective and does not imply convergence to a global minimizer. □

3.5. Baseline Optimizers and Training Protocol

NDD was compared with SGD (Stochastic Gradient Descent), momentum SGD, RMSprop, Adam, AdamW, L-BFGS, and Hessian-free optimization (HFO). All optimizers minimized the same binary cross-entropy objective using the same neural classifier and full-batch training. The checkpoint with the best validation ROC-AUC was selected for test evaluation, and early stopping was not used.

SGD, momentum SGD, RMSprop, Adam, and AdamW were trained for 120 epochs without line search, damping, conjugate-gradient steps, or direction clipping. L-BFGS was trained for 60 epochs using a strong-Wolfe line search, history size 20, and maximum four internal iterations per epoch. HFO and NDD were trained for 120 epochs using Armijo line search, fixed damping 10^{-2}, and direction clipping norm 5.0. HFO used maximum conjugate-gradient iteration 20 and conjugate-gradient tolerance 10^{-4}, whereas NDD used coordinate-wise divided-difference scaling with $b_{\min} = 10^{-3}$, $b_{\max} = 10^{3}$, and $\varepsilon = 10^{-6}$. The numerical safeguards and optimizer parameters are summarized in table 3.

Table 3. Numerical safeguards and optimizer parameters used in the reported NDD and HFO experiments.

Parameter	Value	Role
α_0	1.0	Initial step size for Armijo-based NDD and HFO updates
λ	10^{-2}	NDD damping coefficient
ε	10^{-6}	Stabilization term in the divided-difference denominator
$b_{\min}$	10^{-3}	Lower bound for NDD diagonal curvature scaling
$b_{\max}$	10^{3}	Upper bound for NDD diagonal curvature scaling
c_d	5.0	Direction-clipping norm for NDD and HFO
HFO damping	10^{-2}	Damping used in Hessian-free optimization
HFO CG max iter	20	Maximum conjugate-gradient iterations in HFO
HFO CG tolerance	10^{-4}	Conjugate-gradient tolerance in HFO

The learning-rate settings were fixed from the exported experimental configuration: SGD used 0.05, momentum SGD 0.02, RMSprop 0.005, Adam 0.01, AdamW 0.005, and L-BFGS 0.8. Weight decay was 10^{-4} for SGD, momentum SGD, RMSprop, and Adam; 10^{-3} for AdamW; and zero for L-BFGS.

Runtime comparison was interpreted as an implementation-specific wall-clock comparison under the reported CPU-only, full-batch setting. It was not intended to prove universal algorithmic superiority over HFO. The HFO comparison should therefore be read as a transparent fixed-configuration baseline, not an exhaustive HFO tuning study.

3.6. Computational Environment and Reproducibility

All experiments were executed in a CPU-only environment. Hardware and software specifications are reported in table 4 because runtime comparisons depend on the computational environment, implementation framework, and GPU availability.

Table 4. Computational environment used for the reported experiments.

Item	Value
Platform	Kaggle (Linux-6.6.122+-x86_64-with-glibc2.35)
Python version	3.12.12
PyTorch version	2.10.0+cpu
CUDA available	False
Device	CPU
CPU count	4
Total RAM	31.35 GB
Available RAM at run time	30.14 GB

All runtime values should be interpreted under this CPU-only configuration. They are implementation-specific wall-clock measurements and may change with different processors, memory bandwidth, software versions, GPU acceleration, parallelization settings, or HFO hyperparameters. The runtime comparison characterizes the reported setting and should not be read as hardware-independent computational dominance. The scripts recorded seeds, split modes, optimizer settings, fingerprint dimensions, split assignments, repeated-run metrics, optimizer diagnostics, calibration outputs, and figure-generation outputs. Five seeds were used: $\mathcal{S} = \{42,43,44,45,46\}$.The same seed-specific split assignments were reused across fingerprint representations.

3.7. Evaluation Metrics and Statistical Analysis

The primary threshold-independent metrics were ROC-AUC (Receiver Operating Characteriztic – Area Under the Curve) and PR-AUC (Precision-Recall Area Under the Curve). Threshold-dependent metrics included accuracy, balanced accuracy, sensitivity, specificity, F1 score, and Matthews correlation coefficient (MCC). LogLoss was used as a probabilistic error metric. Enrichment-factor and hit-rate diagnostics were computed to assess early retrieval behavior. Calibration was evaluated as a secondary diagnostic because virtual screening may depend on probability reliability as well as ranking. Brier score, expected calibration error, and reliability diagrams were used. Expected calibration error was computed by partitioning predicted probabilities into M bins:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|, \tag{26}$$

B_m is the set of samples in bin m, $\mathrm{acc}(B_m)$ is the observed positive-class frequency, and $\mathrm{conf}(B_m)$ is the mean predicted probability. Threshold sensitivity was examined by varying the decision threshold and recording balanced accuracy, F1 score, and MCC.

For each metric, mean and standard deviation were reported across five seeds. Pairwise comparisons between NDD and baseline optimizers used paired tests over the same seeds. Both paired t-tests and Wilcoxon signed-rank tests were used as diagnostic comparisons. Because only five seeds were used, all statistical comparisons were interpreted conservatively. Wilcoxon tests with $p > 0.05$ were treated as non-significant. Small ROC-AUC differences, especially 0.002–0.005, were described as competitive or directional when standard deviations overlapped. For the main paired comparisons, the paired effect size was summarized as

$$d_z = \frac{\overline{\Delta}}{s_\Delta}, \tag{27}$$

$\overline{\Delta}$ is the mean paired difference and s_Δ is the standard deviation of paired differences across seeds. Confidence intervals were treated as descriptive because the number of seeds was small.

3.8. Additional Robustness and Diagnostic Experiments

Additional diagnostics tested whether the main optimizer interpretation depended on calibration, architecture, or nominal fingerprint dimension. Calibration was evaluated using Brier score, expected calibration error, and reliability diagrams. Threshold sensitivity was evaluated by varying the decision threshold. Architecture sensitivity was tested on Klekota–Roth using small, base, and large hidden-layer configurations, together with tanh and ReLU variants. Dimensionality scaling used Klekota–Roth feature subsets with 79, 881, 2000, and 4860 features. Fingerprint sparsity, nonconstant features, duplicate columns, and principal-component explained variance were also measured. These analyzes were robustness diagnostics, not exhaustive hyperparameter or representation searches.

4. Results and Discussion

We report the experiments in four steps. First, benchmark validity, chemical diagnostics, fingerprint capacity, and split quality are examined. Second, predictive performance is compared across fingerprints and split protocols. Third, architecture sensitivity, dimensionality scaling, calibration, and threshold behavior are assessed. Finally, optimization stability, runtime, and ablation results are analyzed.

4.1. Aligned Benchmark and Chemical Diagnostics

The aligned benchmark provides a controlled basis for comparison. EState, PubChem, and Klekota–Roth contain identical molecules and labels. As shown in table 1 and figure 2, each representation contains 1459 molecules, consisting of 615 candidate active compounds and 844 decoys. The positive-class ratio is 0.4215 in all three fingerprint spaces. Thus, performance differences across fingerprints cannot be attributed to different sample sets. They should not, however, be interpreted as optimizer-only effects because each fingerprint dimension induces a different first-layer parameter dimension and a different neural optimization landscape.

RDKit-enabled preprocessing revealed measurable chemical differences between candidate active molecules and decoys. Candidate active molecules had larger mean molecular weight, higher hydrogen-bond donor and acceptor counts, and higher topological polar surface area (TPSA). Decoys had higher mean logP and more aromatic rings. Selected diagnostics are summarized in table 5. These differences do not invalidate the benchmark, but they should be considered when interpreting ligand-based classification performance.

Table 5. Selected RDKit chemical-property diagnostics for candidate active and decoy molecules.

Property	Active mean	Decoy mean	Standardized mean difference	Mann–Whitney p-value
Molecular weight	439.36	373.43	0.371	5.37×10^{-3}
logP	-0.18	3.08	-1.131	1.98×10^{-91}
Hydrogen-bond donors	4.54	1.89	1.263	1.89×10^{-123}
Hydrogen-bond acceptors	7.41	3.91	1.019	5.09×10^{-61}
TPSA	164.10	72.27	1.439	9.50×10^{-134}
Rotatable bonds	7.76	4.92	0.625	8.31×10^{-28}
Heavy atoms	30.60	25.43	0.410	5.45×10^{-6}
Aromatic rings	1.50	2.26	-0.622	2.24×10^{-32}

Scaffold diagnostics showed that candidate active molecules and decoys occupy largely different scaffold spaces. RDKit Bemis–Murcko analysis identified 226 active scaffolds, 720 decoy scaffolds, and only 8 shared scaffolds. The active–decoy scaffold Jaccard index was 0.0085. The Morgan-fingerprint/Butina procedure produced 977 similarity clusters. These results support the use of scaffold-key and similarity-cluster splits as chemically meaningful grouped evaluation protocols.

The fingerprint representations differ in both dimensionality and sparsity. EState has 79 features, PubChem has 881 features, and Klekota–Roth has 4860 features. Klekota–Roth is also highly sparse. As shown in table 6, it has density 0.0136 and sparsity 0.9864, with 2005 nonconstant features. This property matters because NDD scales parameter directions coordinate-wise in the large neural parameter space induced by this fingerprint.

Table 6. Fingerprint sparsity and capacity diagnostics.

Fingerprint	Samples	Features	Density	Sparsity	Constant features	Effective nonconstant features
EState	1459	79	0.1312	0.8688	38	41
PubChem	1459	881	0.1679	0.8321	253	628
Klekota–Roth	1459	4860	0.0136	0.9864	2855	2005

These diagnostics do not prove that Klekota–Roth is intrinsically more informative than EState or PubChem. Klekota–Roth has a much larger nominal dimension and many constant or sparse features. Its stronger predictive performance may reflect fragment coverage, representational capacity, sparsity, redundancy, and architecture-induced parameter scaling. The dimensionality-scaling experiment partially probes this issue, but it does not fully separate chemical informativeness from feature expansion.

The split-quality diagnostics clarify the evaluation setting. The stratified split preserves the global positive-class ratio almost exactly. The scaffold-key and similarity-cluster splits produce larger variation in partition size and class ratio because chemical group integrity is preserved. This behavior is expected in grouped chemical evaluation.

Figure 3 is only an exploratory two-dimensional PCA (Principal Component Analysis) projection of the Klekota–Roth space. Candidate active compounds and decoys show visible structure, but this should not be interpreted as evidence of strong class separability in the original 4860-dimensional space. PCA is unsupervised and variance-based, not class-discriminative. The first two components explain 28.15% of the variance, and the first ten explain 47.21%. The figure is therefore a qualitative diagnostic, not evidence of predictive superiority. The first two components explain 28.15% of the variance. Because principal component analysis is unsupervised and variance-based, this plot should be interpreted

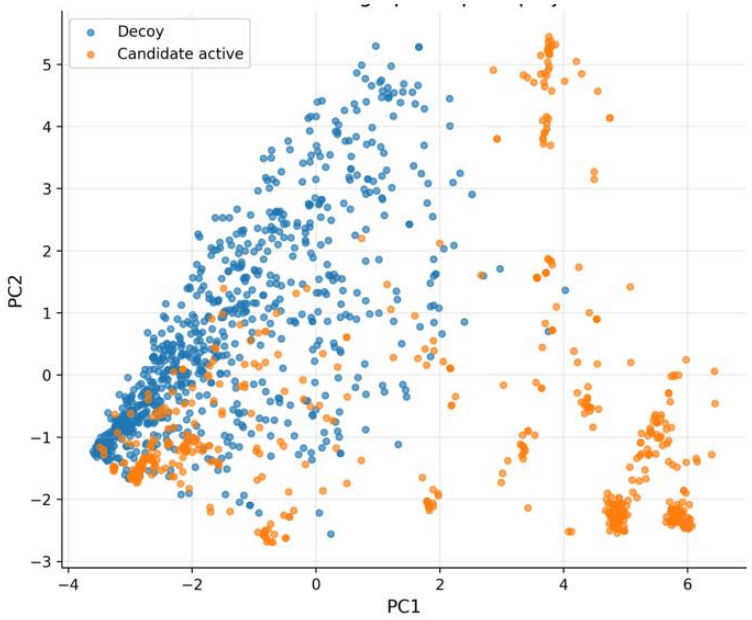

Figure 3. Exploratory Klekota–Roth fingerprint-space projection using principal component analysis.

4.2. NDD Shows Its Strongest Performance in Sparse High-Dimensional Klekota–Roth Representations

We first asked whether NDD consistently improves neural virtual screening across fingerprints and split protocols. The answer is setting-dependent. As summarized in table 7, NDD achieved the best mean ROC-AUC in three settings: scaffold-key EState, scaffold-key Klekota–Roth, and similarity-cluster Klekota–Roth. Under stratified Klekota–Roth, it ranked second and was only 0.0005 below Momentum. These results suggest that NDD is most competitive in sparse high-dimensional Klekota–Roth settings. The ROC-AUC margins, however, are often small and should be interpreted cautiously because standard deviations overlap and only five seeds were used.

Table 7. Evidence matrix comparing NDD with the best optimizer in each split–fingerprint setting.

Split protocol	Fingerprint	Best optimizer	Best ROC-AUC	NDD ROC-AUC	NDD minus best	Interpretation
Stratified	EState	RMSprop	0.9780 ± 0.0074	0.9728 ± 0.0047	-0.0052	NDD below best
	PubChem	HFO	0.9811 ± 0.0092	0.9736 ± 0.0034	-0.0075	NDD below best
	Klekota–Roth	Momentum	0.9901 ± 0.0029	0.9896 ± 0.0044	-0.0005	NDD competitive
Scaffold-key	EState	NDD	0.9654 ± 0.0092	0.9654 ± 0.0092	$+0.0000$	NDD best
	PubChem	Momentum	0.9607 ± 0.0241	0.9574 ± 0.0232	-0.0033	NDD below best
	Klekota–Roth	NDD	0.9717 ± 0.0145	0.9717 ± 0.0145	$+0.0000$	NDD best
Similarity-cluste	EState	AdamW	0.9697 ± 0.0115	0.9695 ± 0.0041	-0.0002	NDD competitive
	PubChem	Momentum	0.9820 ± 0.0085	0.9815 ± 0.0079	-0.0005	NDD competitive
	Klekota–Roth	NDD	0.9868 ± 0.0099	0.9868 ± 0.0099	$+0.0000$	NDD best

The evidence map in Figure 4 shows the same descriptive pattern. NDD is not uniformly superior, but its most competitive results cluster in the Klekota–Roth column, especially under scaffold-key and similarity-cluster evaluation. This pattern is consistent with the hypothesis that diagonal curvature-aware scaling is useful in large sparse parameter spaces. It should not be read as definitive statistical evidence of dominance. NDD is descriptively best for scaffold-key EState, scaffold-key Klekota–Roth, and similarity-cluster Klekota–Roth, while remaining competitive but not uniformly dominant in other settings.

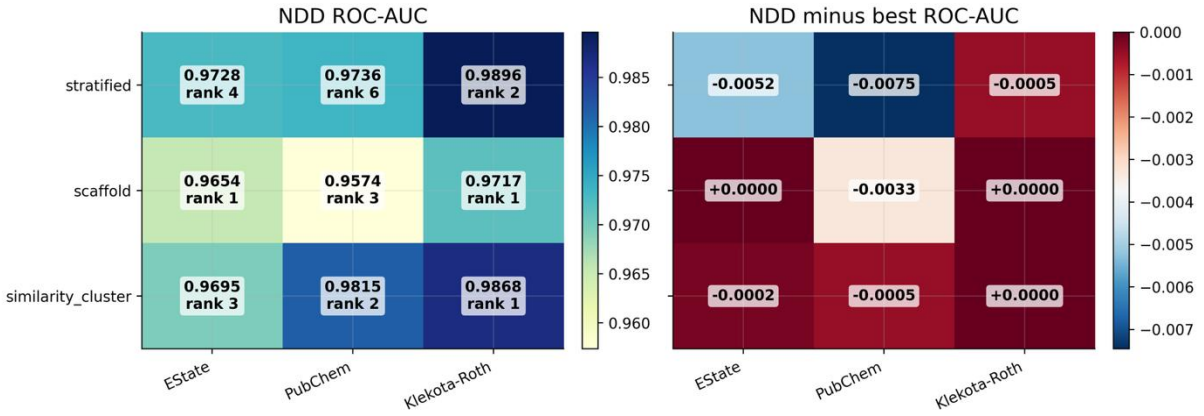

Figure 4. Evidence map showing where NDD provides the most competitive mean ROC-AUC results.

We then focused on Klekota–Roth because this representation most directly tests the proposed high-dimensional optimization setting. Detailed Klekota–Roth results are summarized in table 8 and visualized in figure 5. Under the stratified split, Momentum achieved the highest ROC-AUC, while NDD was nearly tied with Adam and only 0.0005 below Momentum. NDD also achieved the highest PR-AUC and MCC among the key optimizers. Under the scaffold-key split, NDD achieved the highest ROC-AUC, PR-AUC, and MCC. Under the similarity-cluster split, NDD achieved the highest ROC-AUC and PR-AUC, while HFO and Momentum remained strong competitors.

Table 8. Klekota–Roth performance of key optimizers across split protocols.

Split protocol	Optimizer	ROC-AUC	PR-AUC	MCC	LogLoss	Runtime (s)
Stratified	Adam	0.9896 ± 0.0043	0.9891 ± 0.0038	0.9135 ± 0.0034	0.1980 ± 0.0651	5.869 ± 0.221
	AdamW	0.9854 ± 0.0072	0.9863 ± 0.0053	0.9144 ± 0.0153	0.1665 ± 0.0535	6.095 ± 0.158
	Momentum	$\mathbf{0.9901 \pm 0.0029}$	0.9892 ± 0.0035	0.9103 ± 0.0201	$\mathbf{0.1176 \pm 0.0218}$	5.675 ± 0.129
	HFO	0.9858 ± 0.0058	0.9857 ± 0.0048	0.9039 ± 0.0180	0.2278 ± 0.0809	52.644 ± 1.583

	NDD	0.9896 ± 0.0044	**0.9894 ± 0.0042**	**0.9302 ± 0.0108**	0.1873 ± 0.1015	8.194 ± 0.121
Scaffold-key	Adam	0.9684 ± 0.0154	0.9706 ± 0.0246	0.8616 ± 0.0739	0.3593 ± 0.2336	5.632 ± 0.141
	AdamW	0.9621 ± 0.0203	0.9667 ± 0.0260	0.8533 ± 0.0858	0.2659 ± 0.1071	5.939 ± 0.136
	Momentum	0.9615 ± 0.0348	0.9626 ± 0.0491	0.8508 ± 0.1117	0.2489 ± 0.1673	5.601 ± 0.142
	HFO	0.9699 ± 0.0207	0.9704 ± 0.0318	0.8742 ± 0.0458	0.2960 ± 0.1145	50.380 ± 5.517
	NDD	**0.9717 ± 0.0145**	**0.9727 ± 0.0226**	**0.8788 ± 0.0401**	**0.2466 ± 0.0598**	8.173 ± 0.484
Similarity-cluster	Adam	0.9794 ± 0.0116	0.9821 ± 0.0118	0.8950 ± 0.0480	0.1956 ± 0.0823	5.628 ± 0.290
	AdamW	0.9779 ± 0.0119	0.9790 ± 0.0156	0.8973 ± 0.0409	0.1990 ± 0.0955	5.941 ± 0.316
	Momentum	0.9842 ± 0.0101	0.9855 ± 0.0101	0.9111 ± 0.0412	**0.1334 ± 0.0466**	5.553 ± 0.268
	HFO	0.9847 ± 0.0138	0.9873 ± 0.0114	**0.9115 ± 0.0482**	0.1756 ± 0.1037	53.835 ± 8.677
	NDD	**0.9868 ± 0.0099**	**0.9875 ± 0.0085**	0.9103 ± 0.0376	0.1937 ± 0.1648	8.015 ± 0.384

Paired tests provided limited statistical support for optimizer superiority. Most Wilcoxon signed-rank tests were not significant at the 0.05 level, and several ROC-AUC or PR-AUC margins were small relative to the standard deviations. Therefore, the paired comparisons were used only as diagnostic evidence. We use the terms *best*, *competitive*, and *directional* descriptively, not as claims of statistically established dominance. Additional paired-test outputs generated by the reproducibility scripts are available in the public code repository. NDD is descriptively strongest under scaffold-key and similarity-cluster protocols and remains competitive under the stratified protocol. Small margins should be interpreted together with standard deviasions and paired test.

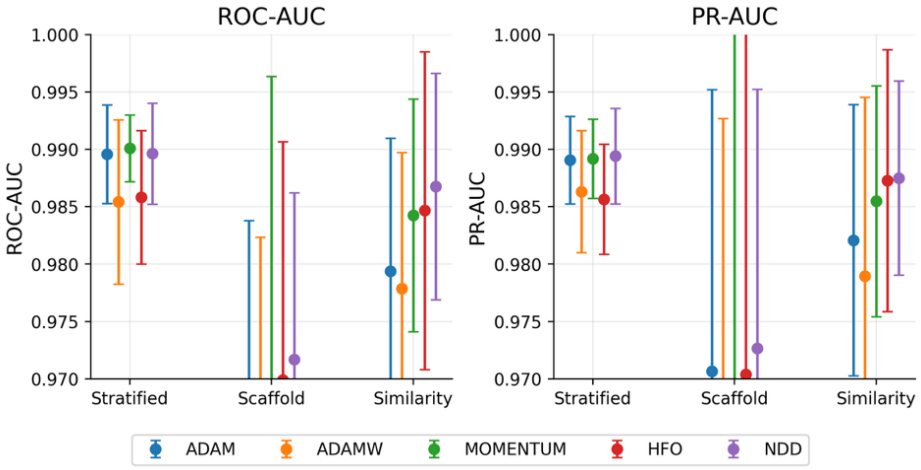

Figure 5. Klekota–Roth ROC-AUC and PR-AUC comparison for key optimizers.

4.3. NDD Remains Stable Across Architectures and Becomes More Competitive with Increasing Klekota–Roth Dimensionality

We next examined whether the main NDD pattern was an artefact of the selected neural architecture or one high-dimensional fingerprint configuration. These analyzes are diagnostic rather than definitive. They assess whether the observed behavior persists across changes in model capacity, activation function, and Klekota–Roth feature dimension. Table 9 shows that NDD remained stable across small, base, and large architectures, as well as tanh and ReLU variants. Mean ROC-AUC values were tightly grouped between 0.9876 and 0.9890. Increasing the model size from 313,217 to 1,277,441 parameters did not systematically improve ROC-AUC. Thus, the result is not merely an artefact of one network size or larger model capacity.

Table 9. NDD architecture sensitivity on Klekota–Roth.

Architecture	Activation	Parameters	ROC-AUC	PR-AUC	MCC	Runtime (s)
Small, 64–32	tanh	313217	0.9878 ± 0.0076	0.9884 ± 0.0057	0.9142 ± 0.0198	5.37 ± 0.18

Base, 128–64	tanh	630529	0.9882 ± 0.0074	0.9885 ± 0.0064	0.9202 ± 0.0281	7.92 ± 0.21
Large, 256–128	tanh	1277441	0.9876 ± 0.0084	0.9882 ± 0.0063	0.9128 ± 0.0263	13.22 ± 0.41
Base, 128–64	ReLU	630529	0.9890 ± 0.0068	0.9897 ± 0.0056	0.9228 ± 0.0262	7.86 ± 0.22

The dimensionality-scaling analysis in table 10 provides partial support for the high-dimensional interpretation. At 79 Klekota–Roth features, NDD was slightly below the best baseline. At 881, 2000, and 4860 features, NDD achieved the highest mean ROC-AUC or was essentially tied with the strongest baseline. This experiment partially isolates nominal feature dimension within the same fingerprint family. It does not prove that dimension alone causes the NDD advantage. Feature sparsity, redundancy, chemical coverage, and architecture-induced parameter scaling may still contribute. We therefore interpret the result as evidence that NDD becomes more competitive in sparse high-dimensional Klekota–Roth settings, not as proof of a universal dimensionality-only effect.

Table 10. Dimensionality-scaling analysis using Klekota–Roth feature subsets.

Feature subset size	Best optimizer	NDD ROC-AUC	Best baseline ROC-AUC	NDD minus best baseline	NDD runtime (s)
79	AdamW	0.7650 ± 0.0834	0.7665	-0.0016	2.10
881	NDD	0.9647 ± 0.0280	0.9644	+0.0003	3.24
2000	NDD	0.9843 ± 0.0077	0.9816	+0.0027	4.56
4860	NDD	0.9882 ± 0.0074	0.9872	+0.0010	7.88

4.4. Calibration and Threshold Sensitivity

Because log loss evaluates probabilities, we examined whether NDD produced reasonably calibrated predictions. In the Klekota–Roth setting, NDD achieved expected calibration error values of 0.0246 under stratified split, 0.0307 under scaffold-key split, and 0.0170 under similarity-cluster split. These values indicate reasonable calibration, although NDD was not always the best calibrated optimizer. A representative reliability diagram is shown in figure 6. Reliability diagrams were used as qualitative diagnostics, not evidence of universal calibration superiority.

Threshold-sensitivity analysis showed that the MCC-optimal threshold varied across split protocols. The selected thresholds were 0.75 for stratified split, 0.40 for scaffold-key split, and 0.65 for similarity-cluster split. This variation supports using ROC-AUC and PR-AUC as primary threshold-independent metrics. Threshold-dependent metrics should be interpreted as operating-point-specific diagnostics. The diagram is used as a qualitative calibration diagnostic and should be interpreted together with Brier score and expected calibration error.

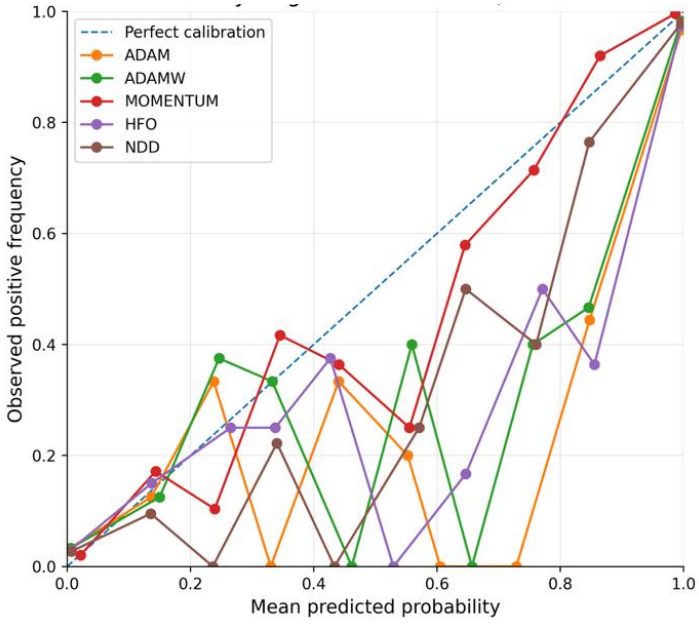

Figure 6. Reliability diagram for Klekota–Roth under the stratified split.

4.5. NDD Provides Stable Descent and a Favourable Runtime Trade-Off Relative to Hessian-Free Optimization

We then examined optimization behavior. Table 11 shows an accepted-step rate of 1.0 in every fingerprint–split setting. On Klekota–Roth, the mean step size stayed close to 1.0, and the mean number of backtracking reductions was exactly 1.0 across all split protocols. The final gradient norm was approximately zero at reporting precision, and the mean value of $g^{\top}d$ remained negative. These diagnostics are consistent with the descent analysis in Equation (19).

Table 11. NDD optimizer diagnostics across split protocols and fingerprint spaces.

Split protocol	Fingerprint	Runtime (s)	Best epoch	Accepted-step rate	Mean step size	Mean backtracks
Stratified	EState	2.130 ± 0.088	89.6 ± 22.6	1.0000 ± 0.0000	0.9748 ± 0.0164	1.45 ± 0.39
	PubChem	3.250 ± 0.133	12.0 ± 6.1	1.0000 ± 0.0000	0.9652 ± 0.0153	1.25 ± 0.23
	Klekota–Roth	8.194 ± 0.121	37.6 ± 46.3	1.0000 ± 0.0000	0.9958 ± 0.0042	1.00 ± 0.00
Scaffold-key	EState	2.069 ± 0.054	47.0 ± 42.8	1.0000 ± 0.0000	0.9752 ± 0.0138	1.31 ± 0.31
	PubChem	3.151 ± 0.027	9.4 ± 3.6	1.0000 ± 0.0000	0.9746 ± 0.0153	1.08 ± 0.07
	Klekota–Roth	8.173 ± 0.484	13.0 ± 9.5	1.0000 ± 0.0000	0.9958 ± 0.0029	1.00 ± 0.00
Similarity-cluster	EState	2.077 ± 0.114	70.0 ± 24.6	1.0000 ± 0.0000	0.9727 ± 0.0149	1.29 ± 0.24
	PubChem	3.095 ± 0.096	13.0 ± 3.2	1.0000 ± 0.0000	0.9829 ± 0.0081	1.12 ± 0.18
	Klekota–Roth	8.015 ± 0.384	21.4 ± 30.0	1.0000 ± 0.0000	0.9967 ± 0.0035	1.00 ± 0.00

Figure 7 provides additional convergence diagnostics for NDD on Klekota–Roth under the stratified split. The gradient-norm panel in figure 7 shows that the gradient norm decreases by several orders of magnitude, while the descent-diagnostic panel in figure 7 shows that $g^{\top}d$ remains nonpositive and approaches zero as training converges. These diagnostics support stable curvature-scaled descent. The gradient norm decreases by several orders of magnitude, while the mean value of remains nonpositive and approaches zero as optimization converges.

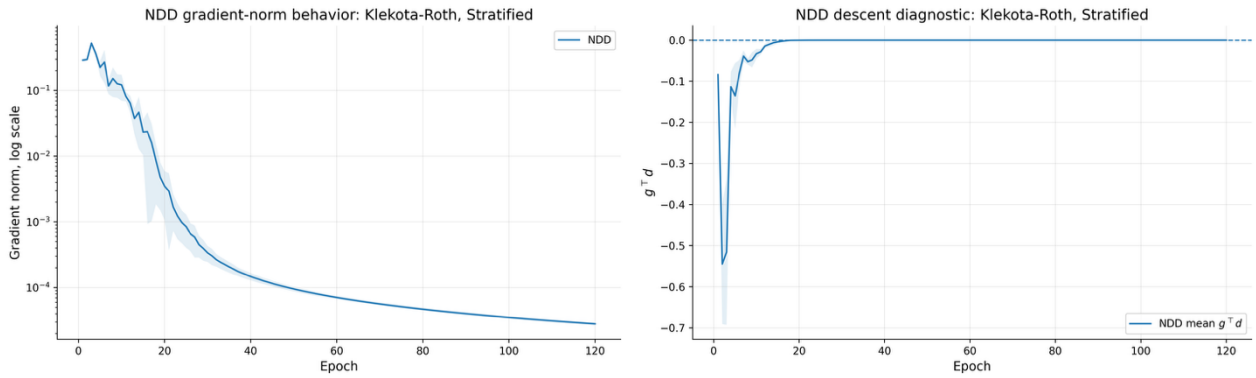

Figure 7. NDD convergence diagnostics on Klekota–Roth under the stratified split.

Validation and training-loss curves provide complementary evidence. The validation ROC-AUC panel in figure 8 and the training-loss panel in figure 8 are summarized together in figure 8. NDD reaches high validation ROC-AUC early on Klekota–Roth and remains stable. Training loss should be interpreted together with validation and test metrics because lower training loss does not always imply better generalization.

ISSN 2723-6471

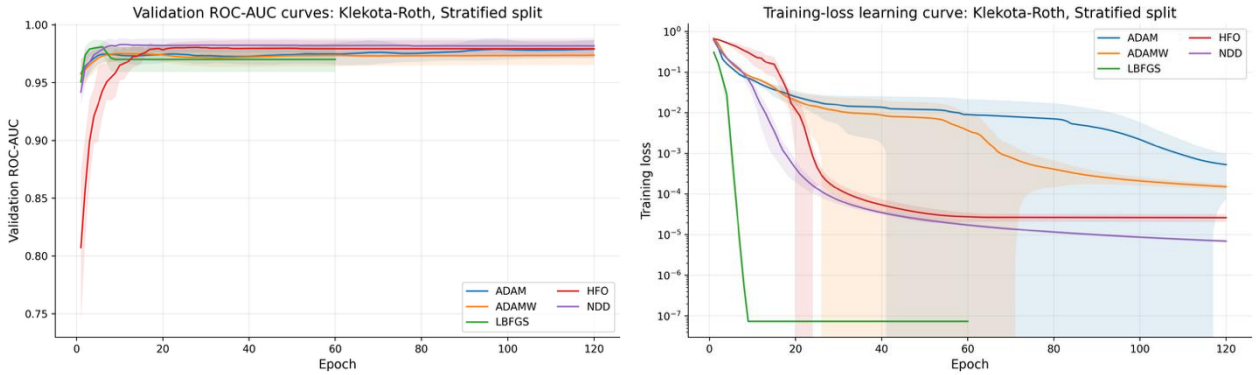

Figure 8. Validation and training-loss behavior on Klekota–Roth under the stratified split.

Figure 9 summarizes the performance–runtime trade-off. HFO is competitive in several settings but requires approximately 50–54 seconds per Klekota–Roth run. NDD requires approximately 8 seconds per run while achieving comparable or better grouped-split ROC-AUC. Momentum and Adam are faster than NDD. Thus, NDD should not be described as the fastest optimizer overall. Its main advantage is a lower-runtime curvature-aware alternative to HFO.

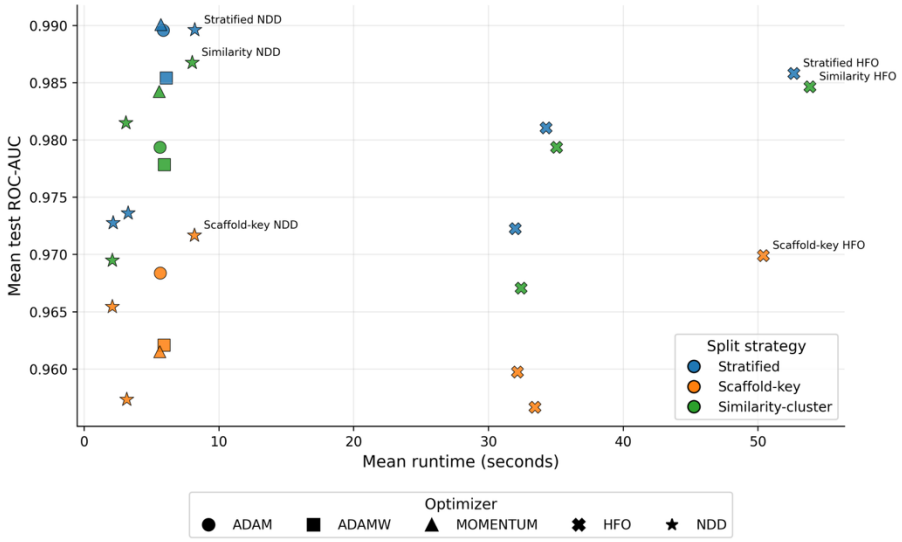

Figure 9. Performance–runtime trade-off across key optimizer and split combinations.

Under the reported CPU-only, full-batch, fixed-configuration setting, NDD provides a lower-runtime curvature-aware alternative to HFO while remaining competitive in Klekota–Roth ranking performance. The comparison should not be interpreted as a universal runtime ranking across all possible HFO tuning choices. The ablation results in table 12 show a trade-off between numerical safeguards and empirical generalization. Clipping, damping, and Armijo line search are intended to prevent unstable or excessively large updates. They preserve descent behavior but do not always maximize validation or test ROC-AUC. Removing a safeguard can improve ROC-AUC or log loss when the unprotected trajectory remains stable and permits less conservative movement. Thus, the ablation results do not show that every safeguard always improves predictive performance. They show that safeguards reduce instability risk while introducing possible conservativeness.

Table 12. NDD ablation results on Klekota–Roth.

Split protocol	Variant	ROC-AUC	PR-AUC	MCC	LogLoss
Stratified	Full NDD	0.9896 ± 0.0044	0.9894 ± 0.0042	$\mathbf{0.9302 \pm 0.0108}$	0.1873 ± 0.1015
Stratified	No Armijo	$\mathbf{0.9901 \pm 0.0034}$	$\mathbf{0.9902 \pm 0.0034}$	0.9286 ± 0.0212	$\mathbf{0.1188 \pm 0.0149}$
Stratified	No clipping	0.9897 ± 0.0043	0.9897 ± 0.0038	0.9273 ± 0.0040	0.1641 ± 0.0693
Stratified	No damping	0.9862 ± 0.0040	0.9868 ± 0.0032	0.9217 ± 0.0182	0.2143 ± 0.1202
Scaffold-key	Full NDD	0.9717 ± 0.0145	0.9727 ± 0.0226	0.8788 ± 0.0401	0.2466 ± 0.0598

Split protocol	Variant	ROC-AUC	PR-AUC	MCC	LogLoss
Scaffold-key	No Armijo	0.9714 ± 0.0135	0.9733 ± 0.0195	0.8762 ± 0.0549	0.2797 ± 0.1669
Scaffold-key	No clipping	0.9727 ± 0.0140	0.9742 ± 0.0224	$\mathbf{0.8916 \pm 0.0334}$	$\mathbf{0.2277 \pm 0.0411}$
Scaffold-key	No damping	$\mathbf{0.9748 \pm 0.0150}$	$\mathbf{0.9754 \pm 0.0241}$	0.8907 ± 0.0371	0.2420 ± 0.0904
Similarity-cluster	Full NDD	0.9868 ± 0.0099	0.9875 ± 0.0085	0.9103 ± 0.0376	0.1937 ± 0.1648
Similarity-cluster	No Armijo	0.9865 ± 0.0098	0.9880 ± 0.0083	$\mathbf{0.9178 \pm 0.0346}$	0.1411 ± 0.0607
Similarity-cluster	No clipping	0.9877 ± 0.0080	0.9880 ± 0.0085	0.9108 ± 0.0410	0.1393 ± 0.0551
Similarity-cluster	No damping	$\mathbf{0.9897 \pm 0.0074}$	$\mathbf{0.9904 \pm 0.0057}$	0.9126 ± 0.0222	$\mathbf{0.1326 \pm 0.0616}$

Taken together, the results support five observations. First, the aligned benchmark controls sample-composition effects, but it does not fix the induced optimization landscape. Second, Klekota–Roth is a sparse high-dimensional representation, not simply a uniformly richer information source. Third, NDD is most competitive in the Klekota–Roth setting, especially under grouped splits. Fourth, NDD provides stable descent behavior, with accepted-step rate 1.0 and consistently negative $g^{\top}d$. Fifth, under the reported CPU-only fixed implementation, NDD offers a lower-runtime curvature-aware alternative to HFO, although it is not always faster or more accurate than every first-order optimizer.

4.6. Comparison with Previous Virtual Screening Studies

Previous studies have demonstrated the utility of molecular fingerprints and machine learning for antiviral compound prioritization. Fingerprint-based gradient boosting approaches such as XGBoost have shown strong predictive performance and computational efficiency for bioactivity prediction and compound prioritization tasks [9], [10]. Fingerprint representations have also been applied in conjunction with nonlinear classifiers for antiviral compound screening, demonstrating competitive performance across different evaluation settings [11]. These findings establish the feasibility of fingerprint-based machine learning for antiviral screening, with emphasis on molecular representation and classifier selection.

The present study shifts the focus from classifier comparison to optimizer behavior in high-dimensional neural virtual screening. The MLP was fixed in the main experiments, while the optimizer was varied under aligned fingerprints and repeated split protocols. This design evaluates optimizers on identical molecules, labels, architecture, and split assignments within each setting.

Recent influenza-related virtual screening studies also show the growing role of machine learning and advanced computational workflows. Structure-based studies have combined pharmacophore modeling, docking, ADME analysis, and biological validation to identify neuraminidase inhibitor candidates [13], [14]. Other studies have explored attention-based architectures and explainable ensemble learning for influenza compound prediction [15], [16]. The present study complements these approaches by focusing on optimization efficiency and stability in fingerprint-based neural screening.

4.7. Scope, Limitations, and Future Directions

This study evaluates NDD as a curvature-aware optimizer for fingerprint-based neural virtual screening. The aligned benchmark, grouped splits, optimizer diagnostics, architecture sensitivity, dimensionality scaling, calibration analysis, and ablation study support its stability and usefulness in sparse high-dimensional Klekota–Roth settings. The proposed model should therefore be interpreted as a ligand-based prioritization framework for H9N2-associated candidate compounds, not as experimental confirmation of antiviral activity.

Several limitations define the scope of the findings. Benchmark alignment fixes molecular identity, labels, ordering, and split assignments, but it does not make the induced neural optimization landscapes identical. Because the same hidden-layer widths were used across fingerprints, the number of trainable parameters increases from EState to PubChem and Klekota–Roth. Thus, the observed NDD advantage may reflect the interaction between sparse high-dimensional fingerprints and architecture-induced parameter scaling. The runtime comparison with HFO is also specific to the reported CPU-only, full-batch, fixed-configuration setting, and different HFO settings may change the runtime–performance trade-off. In addition, several ROC-AUC differences are small relative to standard deviations, and the five-seed design supports descriptive rather than definitive statistical interpretation.

Further limitations concern representation, preprocessing, and validation. Klekota–Roth performance cannot be fully separated from feature capacity, sparsity, redundancy, and input-induced parameter size. The dimensionality analysis

partially supports the high-dimensional interpretation, but it does not isolate dimension from chemical feature composition. SMILES standardization did not resolve all tautomeric, salt, or stereochemical equivalences. Finally, no external H9N2 validation dataset or independent assay condition was used, so the reported results should not be interpreted as guaranteed generalization to novel H9N2 chemical spaces or prospective screening campaigns.

Future work should include controlled architecture studies, fixed-parameter or bottlenecked models, larger repeated-run budgets, broader fingerprint-subset experiments, more extensive molecular standardization, and external validation. NDD-based screening may be integrated into multi-stage antiviral discovery workflows involving docking, molecular dynamics, binding-energy estimation, pharmacokinetic filtering, and experimental assays. Methodological extensions may also explore adaptive damping, adaptive clipping, and split-aware line search within the proposed positive diagonal curvature-scaling framework.

5. Conclusion

This study proposed Newton Divided Difference optimization for fingerprint-based neural virtual screening of candidate compounds against Avian Influenza A/H9N2. NDD uses coordinate-wise curvature estimates from consecutive parameter and gradient differences, without constructing or inverting a full Hessian matrix. With positive diagonal scaling, clipping, and damping, it preserves descent for nonzero gradients while adding only $O(p)$ cost beyond gradient evaluation.

Using an aligned benchmark of 1459 molecules represented by EState, PubChem, and Klekota–Roth fingerprints, NDD was most useful in sparse high-dimensional Klekota–Roth settings rather than uniformly superior across all cases. It achieved the highest descriptive Klekota–Roth ROC-AUC and PR-AUC under the scaffold-key split, 0.9717 ± 0.0145 and 0.9727 ± 0.0226, and under the similarity-cluster split, 0.9868 ± 0.0099 and 0.9875 ± 0.0085. Under the stratified split, Momentum gave the highest ROC-AUC, while NDD remained competitive with ROC-AUC 0.9896 ± 0.0044. These differences should be interpreted conservatively because standard deviations overlap and only five seeds were used.

NDD also provided a favourable curvature-aware runtime trade-off relative to HFO, requiring about 8 seconds per Klekota–Roth run compared with about 50–54 seconds for HFO under the reported CPU-only configuration. Architecture and dimensionality diagnostics further supported its stability in sparse high-dimensional fingerprint spaces. Overall, NDD is a practical positive diagonal curvature-aware optimizer for neural virtual screening, although external H9N2 validation and prospective experimental confirmation remain necessary before antiviral prioritization.

6. Declarations

6.1. Author Contributions

Conceptualization: S.A., M.J., A.M.R., M.H.Z.A.F., C.A.N., and R.V.N.; Methodology: M.J.; Software: S.A.; Validation: S.A., M.J., A.M.R., M.H.Z.A.F., C.A.N., and R.V.N.; Formal Analysis: S.A., M.J., A.M.R., M.H.Z.A.F., C.A.N., and R.V.N.; Investigation: S.A.; Resources: M.J.; Data Curation: M.J.; Writing Original Draft Preparation: S.A., M.J., A.M.R., M.H.Z.A.F., C.A.N., and R.V.N.; Writing Review and Editing: M.J., S.A., A.M.R., M.H.Z.A.F., C.A.N., and R.V.N.; Visualization: S.A.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The datasets, code, experimental results, figures, diagnostic outputs, and reproducibility metadata supporting this study are available at https://www.kaggle.com/code/edumath/paper-jads-version-of-may-20-2026.

6.3. Funding

This work was supported by Kemdiktisaintek under the National Competitive Grant – Fundamental Research scheme, grant number from DPPM DIKTI to LLDIKTI7: 286/C3/DT.05.00/PL-BARU/2026 and grant number from LLDIKTI7 to Universitas Islam Darul ‘Ulum: 049/LL7/DT.05.00/PL-BARU/2026. The authors gratefully acknowledge the institutional support and research facilities that contributed to this study.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] W. Song and K. Qin, "Human-infecting influenza A (H9N2) virus: A forgotten potential pandemic strain?," *Zoonoses and Public Health*, vol. 67, no. 3, pp. 203–212, 2020. doi: 10.1111/zph.12685.

[2] T. P. Peacock, J. James, J. E. Sealy, and M. Iqbal, "A global perspective on H9N2 avian influenza virus," *Viruses*, vol. 11, no. 7, p. 620, pp. 1-28, 2019. doi: 10.3390/v11070620.

[3] F. Bonfante, E. Mazzetto, M. Zanardello, V. Fortin, C. Gobbo, M. Maniero, M. Drigo, G. Terregino, and I. Monne, "A G1-lineage H9N2 virus with oviduct tropism causes chronic pathological changes in the infundibulum and a long-lasting drop in egg production," *Veterinary Research*, vol. 49, no. August, p. 83, pp. 1-16, 2018. doi: 10.1186/s13567-018-0575-1.

[4] S. Indrasari, D. A. Hewajuli, N. L. P. I. Dharmayanti, W. Asmara, and K. T. Putri, "The First Pathogenicity Analysis Report in Mice with Two H9N2 Subtype Avian Influenza Viruses Isolated from Indonesia," *Biochemical and Cellular Archives*, vol. 21, no. 1, pp. 593–598, 2021.

[5] M. Kontoyianni, "Docking and Virtual Screening in Drug Discovery," in *Proteomics for Drug Discovery: Methods and Protocols*, vol. 1647, *Methods in Molecular Biology*, Humana Press, vol. 2017, no. August, pp. 255–266, 2017, doi: 10.1007/978-1-4939-7201-2_18.

[6] S. F. L. da Silva Rocha, C. G. Olanda, H. H. Fokoue, and C. M. R. Sant'Anna, "Virtual Screening Techniques in Drug Discovery: Review and Recent Applications," *Current Topics in Medicinal Chemistry*, vol. 19, no. 19, pp. 1751–1767, 2019. doi: 10.2174/1568026619666190816101948.

[7] S. Moshawih, M. S. M. Arshad, S. Ming, S. H. E. Lau, C. Y. Ming, C. S. Tan, W. K. Chan, S. H. Goh, H. A. Wahab, and K. S. Liew, "Consensus holistic virtual screening for drug discovery: A novel machine learning model approach," *Journal of Cheminformatics*, vol. 16, no. May, p. 62, pp. 1-17, 2024. doi: 10.1186/s13321-024-00855-8.

[8] T. A. de Oliveira, M. P. da Silva, E. H. B. Maia, A. M. da Silva, and A. G. Taranto, "Virtual Screening Algorithms in Drug Discovery: A Review Focused on Machine and Deep Learning Methods," *Drugs and Drug Candidates*, vol. 2, no. 2, pp. 311–334, 2023. doi: 10.3390/ddc2020017.

[9] D. Boldini, F. Grisoni, D. Kuhn, L. Friedrich, A. Volkamer, and G. Schneider, "Practical guidelines for the use of gradient boosting for molecular property prediction," *Journal of Cheminformatics*, vol. 15, no. 73, pp. 1-13, 2023. doi: 10.1186/s13321-023-00743-7.

[10] A. V. Fassio and L. Shub, "Prioritizing Virtual Screening with Interpretable Interaction Fingerprints," *Journal of Chemical Information and Modeling*, vol. 62, no. 18, pp. 4300–4318, 2022. doi: 10.1021/acs.jcim.2c00695.

[11] L. Jiang, H. Chen, and C. Li, "Advances in deciphering the interactions between viral proteins of influenza A virus and host cellular proteins," *Cell Insight*, vol. 2, no. 2, Art. no. 100079, pp. 1-10, 2023. doi: 10.1016/j.cellin.2023.100079.

[12] J. M. Rollinger, H. Stuppner, and T. Langer, "Virtual Screening for the Discovery of Bioactive Natural Products," in *Progress in Drug Research*, vol. 65, Basel, Switzerland: Birkhäuser, vol. 65, 2008, pp. 213–249. doi: 10.1007/978-3-7643-8117-2_6.

[13] D. Nurjanah, F. Fadilah, and N. L. P. I. Dharmayanti, "Virtual Screening of Indonesian Herbal Compounds with Neuraminidase Inhibitor Activity against N2 Influenza Virus Protein: An in silico Study," *Molecular and Cellular Biomedical Sciences*, vol. 8, no. 2, pp. 58-126, 2024. doi: 10.21705/mcbs.v8i2.468.

[14] R. Gitto, M. Brindisi, A. Aiello, S. Castellano, F. Corelli, M. T. Chimenti, A. Caruso, and G. De Luca, "Discovery of Influenza Neuraminidase Inhibitors: Structure-Based Virtual Screening and Biological Evaluation of Novel Chemotypes," *Molecules*, vol. 30, no. 23, p. 4636, pp. 1-22, 2025. doi: 10.3390/molecules30234636.

[15] J. Li, Y. Zhang, H. Wang, X. Liu, Z. Wang, and Y. Wang, "SMCseeker: An attentive virtual screening model for antiviral discovery," *iScience*, vol. 28, no. 9, p. 113452, pp. 1-17, 2025. doi: 10.1016/j.isci.2025.113452.

[16] I. Meewan, N. Schaduangrat, L. Mookdarsanit, P. Mookdarsanit, and W. Shoombuatong, "Explainable AI-driven prediction of influenza neuraminidase inhibitors using a stacked ensemble-learning framework," *Computers in Biology and Medicine*, vol. 199, no. December, p. 111313, pp. 1-23, 2025. doi: 10.1016/j.compbiomed.2025.111313.

[17] L. H. Hall and L. B. Kier, "The electrotopological state: An atom index for QSAR," *Journal of Chemical Information and Computer Sciences*, vol. 35, no. 6, pp. 1039–1045, 1995. doi: 10.1021/ci00028a014.

[18] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, and S. H. Bryant, "PubChem substance and compound databases," *Nucleic Acids Research*, vol. 44, no. D1, pp. D1202–D1213, 2016. doi: 10.1093/nar/gkv951.

[19] J. Klekota and F. P. Roth, "Chemical substructures that enrich for biological activity," *Bioinformatics*, vol. 24, no. 21, pp. 2518–2525, 2008. doi: 10.1093/bioinformatics/btn479.

[20] C. W. Yap, "PaDEL-Descriptor: An open source software to calculate molecular descriptors and fingerprints," *Journal of Computational Chemistry*, vol. 32, no. 7, pp. 1466–1474, 2011. doi: 10.1002/jcc.21707.

[21] M. M. Mysinger, M. Carchia, J. J. Irwin, and B. K. Shoichet, "Directory of Useful Decoys, Enhanced (DUD-E): Better ligands and decoys for better benchmarking," *Journal of Medicinal Chemistry*, vol. 55, no. 14, pp. 6582–6594, 2012. doi: 10.1021/jm300687e.

[22] Q. Guo, S. Hernandez-Hernandez, and P. J. Ballester, "Scaffold Splits Overestimate Virtual Screening Performance," in *Artificial Neural Networks and Machine Learning – ICANN 2024*, *Lecture Notes in Computer Science*, vol. 2024, no. September, pp. 58–72, 2024, doi: 10.1007/978-3-031-72359-9_5.

[23] Q. Guo, S. Hernandez-Hernandez, and P. J. Ballester, "UMAP-based clustering split for rigorous evaluation of AI models for virtual screening on cancer cell lines," *Journal of Cheminformatics*, vol. 17, no. June, p. 94, pp. 1-18, 2025. doi: 10.1186/s13321-025-01039-8.

[24] C. G. Broyden, "A Class of Methods for Solving Nonlinear Simultaneous Equations," *Mathematics of Computation*, vol. 19, no. 92, pp. 577–593, 1965. doi: 10.1090/S0025-5718-1965-0198670-6.

[25] J. E. Dennis and J. J. Moré, "Quasi-Newton Methods, Motivation and Theory," *SIAM Review*, vol. 19, no. 1, pp. 46–89, 1977. doi: 10.1137/1019005.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[27] J. Martens, "Deep learning via Hessian-free optimization," in *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, Omnipress, vol. 2010, no. June, pp. 735–742.

[28] M. Korbit, A. D. Adeoye, A. Bemporad, and M. Zanon, "Exact Gauss-Newton optimization for training deep neural networks," *Neurocomputing*, vol. 658, no. December, p. 131738, pp. 1-14, 2025. doi: 10.1016/j.neucom.2025.131738.