REWARD-GUIDED LATENT DIFFUSION FOR SCALABLE FEATURE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Feature selection (FS) is a cornerstone of high-dimensional data analysis, aimed at identifying informative feature subsets that enhance model generalizability and interpretability. Despite the emergence of latent-space generative feature-selection frameworks, two challenges remain: candidate generation can rely on direct optimization of selected latent points, and dense interaction modeling becomes prohibitively expensive as dimensionality grows. We propose Reward-Guided Feature Selection (RGFS), which reformulates FS as a reward-guided latent denoising process. By integrating a Latent Diffusion Model (LDM) with a Variational Autoencoder (VAE) and a Reward Evaluator, RGFS models the empirical distribution of latent embeddings from collected reward-labeled masks. During inference, evaluator gradients steer independently initialized reverse trajectories toward regions of higher predicted utility. To further scale interaction modeling to ultra-high-dimensional data, we introduce RGFS+, which incorporates Graph-Induced Sparse Attention. RGFS+ restricts each feature to a correlation-defined neighborhood, reducing sparse attention computation and attention-weight storage from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nk)$ for fixed neighborhood size k. Across 14 standard benchmarks, RGFS or RGFS+ achieves the highest or tied-highest reported score. On two additional datasets with approximately 60,000 features, RGFS+ remains runnable when the evaluated dense alternatives encounter out-of-memory errors. These results support reward-guided diffusion for feature-mask generation and graph-induced sparse attention for high-dimensional feasibility under the evaluated settings.

1 INTRODUCTION

Feature selection (FS) aims to identify the most discriminative subset from an original high-dimensional feature set (Guyon & Elisseeff, 2003). As a fundamental component of data mining, effective FS mitigates the curse of dimensionality (Li et al., 2017), reduces computational overhead, and enhances the generalization and interpretability of downstream models. This process is essential in critical domains such as biomarker discovery (Zhang et al., 2024) and financial risk assessment (Jemai & Zarrad, 2023).

Prior feature selection literature spans three main paradigms: **(1) Filter methods** (Yang & Pedersen, 1997; Yu & Liu, 2003) utilize univariate metrics but assess features in isolation, failing to capture complex synergies between features. **(2) Embedded methods** (Tibshirani, 1996; Sugumaran et al., 2007) are constrained by the learner's functional form, where linear assumptions often struggle to model non-linear data manifolds. **(3) Wrapper methods** (Narendra & Fukunaga, 1977; Kohavi & John, 1997; Kim et al., 2000; Yang & Honavar, 1998) evaluate feature subsets directly in a search space containing 2^N candidate subsets, making exhaustive search computationally intractable for high-dimensional data. To bypass these limitations, recent works have explored Deep Generative Feature Selection (e.g., VTFS (Ying et al., 2024)), which attempts to optimize selection decisions by learning a mapping in a continuous latent space.

However, despite their promise, existing generative frameworks still encounter two critical challenges that hinder their effectiveness in high-dimensional scenarios: **(1) Limited Distributional Exploration:** Feature selection is essentially a complex search problem aimed at identifying the optimal feature combination from a vast solution space. Current generative frameworks typically rely

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

on sequential decision-making or direct gradient-based optimization to find a solution. These approaches are often highly sensitive to initialization, frequently converging to sub-optimal local minima without adequately exploring the solution space. Consequently, they lack a robust mechanism to model the global distribution of task-optimal subsets. This raises a critical question: *How can we move beyond static local optimization to effectively model and traverse the global distribution of high-utility feature configurations in a coarse-to-fine manner?* **(2) The Expressiveness-Efficiency Dilemma in Interaction Modeling:** To overcome the limitations of filter methods, advanced FS models must capture complex dependencies between features. Deep generative models typically employ self-attention mechanisms (Vaswani et al., 2017) to model these global interactions. However, standard self-attention requires computing an $N \times N$ interaction matrix, resulting in a quadratic complexity $\mathcal{O}(N^2)$ in terms of both memory and computation. For datasets with tens of thousands of features, this complexity is prohibitive. Yet, simply discarding these interactions would cause the model to lose critical high-order dependency information among features. This leads to another pivotal question: *How can we capture essential feature interactions for high-dimensional data while reducing the computational complexity?*

Along this line, we propose **Reward-Guided Feature Selection (RGFS)**. Recognizing that diffusion models learn empirical data distributions through iterative refinement (Ho et al., 2020), we reformulate feature selection as a reward-guided denoising process rather than a single deterministic search trajectory. During training, we jointly optimize a Variational Autoencoder (VAE) (Kingma & Welling, 2013) and a Reward Evaluator to construct a latent space aligned with predictive utility. The diffusion model learns the latent distribution of the collected feature masks, while inference begins from Gaussian noise and uses gradients from the Reward Evaluator to steer each stochastic trajectory toward higher-utility regions. This guidance mechanism effectively combines probabilistic global exploration with targeted local refinement, enabling the generation of high-performance feature masks from noise.

Furthermore, to address the scalability bottleneck, we hypothesize that meaningful feature synergies are typically concentrated within a sparse dependency topology, making the explicit modeling of all N^2 pairwise relationships unnecessary. We therefore introduce a **Graph-Induced Sparse Attention** mechanism, denoting the resulting variant as **RGFS+**. Specifically, we construct a feature correlation graph to identify the top-k most relevant neighbors for each feature and apply a structural mask during the attention calculation. This design reduces the attention complexity to nearly $\mathcal{O}(N)$, ensuring scalability for high-dimensional datasets while preserving the model's ability to capture critical feature interactions. The main contributions of this paper can be summarized as follows:

- We formulate feature selection as reward-guided latent denoising, providing a stochastic alternative to methods that optimize a single latent trajectory.
- We propose RGFS and its sparse-attention variant RGFS+. RGFS uses evaluator gradients to guide latent samples toward higher-utility regions, while RGFS+ restricts attention to a correlation-induced graph with $\mathcal{O}(Nk)$ attention complexity for fixed k under a sparse implementation.
- We evaluate the methods on 14 standard benchmarks and conduct additional analyses of reward guidance, feature-subset diversity across random seeds, subset compactness, cross-model behavior, inference time, and feasibility on two datasets with approximately 60,000 features.

2 Related Work

2.1 Feature Selection

Feature selection methods are commonly grouped into filter, wrapper, and embedded families. Filter methods use statistical criteria such as mutual information, dependency, or instance-based relevance (Battiti, 1994; Peng et al., 2005; Kira & Rendell, 1992; Kononenko, 1994); they are efficient but often score features in isolation. Wrapper methods evaluate subsets with a predictive model, including sequential search, recursive elimination, and genetic algorithms (Pudil et al., 1994; Kohavi & John, 1997; Guyon & Elisseeff, 2003; Yang & Honavar, 1998), but face a combinatorial (2^N) search space. Embedded methods incorporate selection into model training, from Lasso (Tibshirani, 1996) to differentiable gates such as L2X and STG (Chen et al., 2018; Yamada et al., 2020).

Reinforcement-learning approaches formulate selection as sequential decision making (Malialis et al., 2016; Liu et al., 2019), while generative methods learn to construct subsets in a latent space, as in VTFS (Ying et al., 2024). LLM-Select (Jeong et al., 2025) uses feature descriptions and task prompts when available. RGFS follows the generative direction but models a distribution of high-utility subsets and samples from it under reward guidance, rather than optimizing a single latent point as in VTFS.

2.2 Diffusion Model

Diffusion models learn a data distribution through forward noising and reverse denoising (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b; Nichol & Dhariwal, 2021). Their stochastic reverse trajectories support distribution-level exploration, while reward-guided diffusion uses evaluator gradients to steer samples toward higher-utility regions (Gong et al., 2025). Related applications span architecture, molecule, protein, text, and audio generation (An et al., 2024; Watson et al., 2023; Hoogeboom et al., 2022; Li et al., 2022; Kong et al., 2021; Liu et al., 2023; Croitoru et al., 2023). However, diffusion models remain largely unexplored for feature selection. RGFS adapts latent diffusion to feature-mask generation, combining stochastic global exploration with reward-based refinement.

3 Preliminaries

3.1 Feature Selection Mask

We formulate a feature subset as a fixed-order binary mask that can be encoded by a sequence model. Specifically, we represent each feature subset as $F = (m_1, m_2, \ldots, m_N) \in \{0,1\}^N$, where N is the total number of features and each position corresponds to a fixed feature identity. For example, given a feature subset $\{f_1, f_3\}$, we convert it to a mask $F = (1, 0, 1, 0, \ldots, 0)$, where $m_j = 1$ if the j-th feature is selected and $m_j = 0$ otherwise.

3.2 Mask-based Training Data

To construct the training data, we generate K candidate feature masks and evaluate each on the downstream predictive task to obtain its reward score. Each training sample is a pair $(F, S) \in \mathcal{D}$, where F represents a specific feature combination and S is the reward score of this combination on the downstream task. Formally, the dataset is defined as:

$$\mathcal{D} = \{(F_i, S_i)\}_{i=1}^{K}. \tag{1}$$

3.3 Latent Diffusion Models

Diffusion models learn to approximate the data distribution by reversing a gradual noising process. Since our framework operates within the latent space constructed by the VAE, we specifically focus on Latent Diffusion Models (LDMs) (Rombach et al., 2022). The forward process is a fixed Markov chain that gradually corrupts a clean latent variable z_0 with Gaussian noise according to a variance schedule $\beta_1, \ldots, \beta_T$. Defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the noisy latent at any timestep t can be sampled directly from z_0 in closed form:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{2}$$

where $\bar{\alpha}_t$ controls the signal-to-noise ratio, ensuring that z_T approaches a standard Gaussian as $t \to T$. The reverse (generative) process recovers z_0 from pure noise through a learned Markov chain with Gaussian transitions $p_\zeta(z_{t-1} \mid z_t)$, where ζ denotes the learnable parameters of the denoiser; in practice, the denoiser is trained to predict the injected noise ϵ rather than the transition mean for better stability (Ho et al., 2020). This denoising objective, instantiated in our framework in Eq. (6), aligns with denoising score matching and serves as the foundation for our reward-guided generation process.

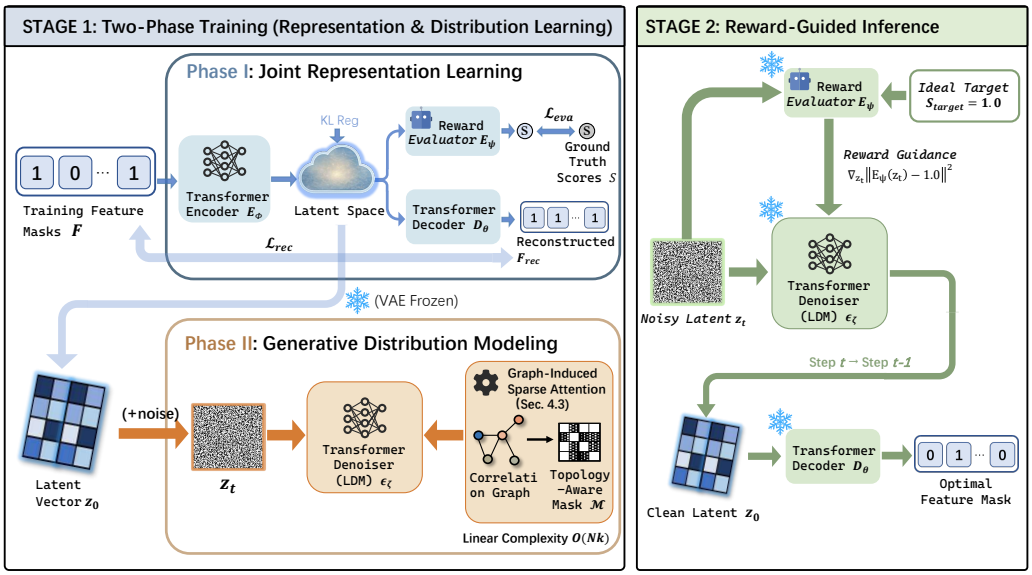


Figure 1: Framework overview. The framework operates in two main stages: 1) Two-Phase Training, which first learns a compact latent representation via a VAE jointly trained with a reward evaluator, followed by a Latent Diffusion Model (LDM) equipped with Graph-Induced Sparse Attention to model the distribution of collected embeddings; and 2) Reward-Guided Inference, where the frozen reward evaluator provides gradient guidance during iterative denoising to steer sampling toward high-utility feature masks.

4 METHOD

4.1 FRAMEWORK OVERVIEW

We propose a two-stage framework to generate feature mask sequences. In the training stage, a VAE encodes reward-labeled masks into a latent space and is trained concurrently with a latent reward evaluator. An LDM is then trained to model the empirical distribution of the resulting embeddings. During inference, we employ reward-guided sampling. Starting from Gaussian noise, the reverse process is iteratively refined: the evaluator predicts the performance associated with the current latent variable, and we compute a Mean Squared Error (MSE) loss against a prescribed target reward. The gradient of this loss guides the noise prediction toward latent regions associated with higher predicted performance. Finally, the resulting latent embedding is decoded by the VAE into a feature mask sequence.

4.2 STAGE 1: LEARNING THE PERFORMANCE-ALIGNED LATENT SPACE

4.2.1 LATENT SEQUENCE SPACE VIA VAE

Given a collection of reward-labeled feature mask sequences, we develop a Variational Autoencoder (VAE) to encode these discrete binary masks into a continuous embedding space. This architecture consists of a probabilistic encoder and a reconstruction decoder.

1) Transformer-Based Encoder. Given a discrete feature mask sequence $F = (m_1, m_2, \ldots, m_N)$ from a training pair $(F, S) \in \mathcal{D}$, the encoder $E_\phi(\cdot)$ encodes the mask sequence into a latent representation $\mathbf{z} \in \mathbb{R}^{N \times d}$, which is sampled from a latent distribution $q_\phi(\mathbf{z}|F) = \mathcal{N}(\mu, \sigma^2)$. To align the continuous manifold for effective diffusion, we impose a KL divergence to measure and minimize the distance between the embedding space posterior distribution and a standard Gaussian prior:

$$\mathcal{L}_{\text{KL}} = \text{KL}(q_\phi(\mathbf{z}|F)\|p(\mathbf{z})). \tag{3}$$

2) Transformer-Based Decoder. The goal of the decoder is to reconstruct the binary feature mask F from the learned latent representation $\mathbf{z}$. Rather than generating mask entries autoregressively,

the decoder predicts all N binary decisions in parallel. It forms a position-indexed query sequence from a repeated start-token embedding and positional encodings, applies full self-attention among the queries and cross-attention to $\mathbf{z}$, and maps each output position to a two-class probability distribution. Let $\mathbf{q}$ denote the resulting positional query sequence and let $m_j \in \{0, 1\}$ denote the decision at position j. The probability of the j-th decision is

$$p_\theta(m_j = c \mid \mathbf{z}) = [\mathrm{Softmax}(D_\theta(\mathbf{q}, \mathbf{z})_j)]_c, \quad c \in \{0, 1\}, \tag{4}$$

where D_θ is the parallel Transformer decoder parameterized by θ. Because no previously generated mask entries are fed back to the decoder, the conditional output likelihood factorizes across positions given $\mathbf{z}$. We minimize the mean negative log-likelihood of the ground-truth mask entries:

$$\mathcal{L}_{\text{rec}} = -\frac{1}{N} \sum_{j=1}^{N} \log p_\theta(m_j \mid \mathbf{z}). \tag{5}$$

4.2.2 Latent Diffusion for Sequence Generation

We employ a Latent Diffusion Model (LDM) (Rombach et al., 2022) to model the empirical distribution of encoded masks through denoising. Independent reverse trajectories generate diverse candidates, which reward guidance then steers toward higher predicted utility.

Diffusion Architecture and Training. Because masks are fixed-order sequences, we use a Transformer denoiser rather than a UNet. It projects $\mathbf{z}_t$ into a d-dimensional hidden state and conditions L residual blocks on a positional-MLP timestep embedding through AdaLN (Peebles & Xie, 2023).

Each block contains two self-attention layers and a feed-forward layer, with LayerScale and DropPath on the residual branches (Touvron et al., 2021; Larsson et al., 2017); Appendix C gives the block equations. The final block output is projected back to the original latent dimension to predict the noise ϵ by minimizing

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t, \mathbf{z}_0, \epsilon} \left[\|\epsilon - \epsilon_\zeta(\mathbf{z}_t, t)\|_2^2 \right]. \tag{6}$$

4.2.3 Learning the Latent Reward Evaluator

We normalize each collected score to $\widetilde{s} = (s - s_{\min})/(s_{\max} - s_{\min})$. Given $\mathbf{z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N] \in \mathbb{R}^{N \times d}$, the evaluator mean-pools the sequence and applies an MLP with a sigmoid output:

$$\overline{\mathbf{z}} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{z}_j, \qquad E_\psi(\mathbf{z}) = \sigma(g_\psi(\overline{\mathbf{z}})), \tag{7}$$

where g_ψ includes the MLP and linear output layer. We train it with

$$\mathcal{L}_{\text{eva}} = \|E_\psi(\mathbf{z}) - \widetilde{s}\|_2^2. \tag{8}$$

Both the evaluator output and guided-sampling target therefore lie in $[0, 1]$.

4.2.4 Training Strategy and Objectives

Phase I: Joint Representation Learning. We jointly train the VAE (E_ϕ, D_θ) and evaluator E_ψ, allowing $\mathcal{L}_{\text{eva}}$ to shape the encoder toward utility-predictive representations. The objective balances utility prediction, reconstruction, and latent regularity:

$$\mathcal{L}_{\text{VAE}} = \alpha \cdot \mathcal{L}_{\text{eva}} + (1 - \alpha) \cdot \mathcal{L}_{\text{rec}} + \beta \cdot \mathcal{L}_{\text{KL}}, \tag{9}$$

where $\alpha \in [0, 1]$ trades off utility prediction and reconstruction, and β weights KL regularization.

Phase II: Generative Distribution Modeling. After convergence, we freeze both models. This step fixes the semantic structure of the latent space, preventing distribution shifts that could destabilize the subsequent training. We then train ϵ_ζ on reparameterized samples from the fixed encoder by minimizing Eq. (6).

4.3 Scaling to Long Sequences via Graph-Induced Sparsity

Dense self-attention costs $O(N^2)$ in sequence length. We instead assume that dominant feature-selection dependencies form a sparse graph and restrict each query to k correlated neighbors, yielding $O(Nk)$ attention computation and storage under a sparse implementation.

Feature Correlation Graph Construction. We construct a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from the absolute Pearson correlations $R \in \mathbb{R}^{N \times N}$:

$$R_{ij} = \left| \frac{\mathrm{cov}(f_i, f_j)}{\sigma_{f_i} \sigma_{f_j}} \right|, \tag{10}$$

where f_i is the i-th feature vector. The absolute value captures correlation strength regardless of sign. For each query i, we select the top-k off-diagonal entries in row i and add a self-loop:

$$A_{ij} = \begin{cases} 1, & \text{if } j \in \mathcal{N}_i, \\ 0, & \text{otherwise.} \end{cases} \quad \mathcal{N}_i = \{i\} \cup \mathrm{TopK}_k(R_{i,\neg i}), \tag{11}$$

Here $R_{i,\neg i}$ excludes the diagonal. Row-wise rankings can differ, so $j \in \mathcal{N}_i$ need not imply $i \in \mathcal{N}_j$; each query retains one self-loop and at most k non-self outgoing neighbors.

Topology-Aware Indexed Attention. For each query $\mathbf{q}_i$, indexed attention gathers and aggregates only keys and values in $\mathcal{N}_i$:

$$\begin{aligned} e_{ij} &= \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d_k}}, \qquad j \in \mathcal{N}_i, \\ a_{ij} &= \frac{\exp(e_{ij})}{\sum_{\ell \in \mathcal{N}_i} \exp(e_{i\ell})}, \\ \mathrm{SparseAttention}(Q, K, V)_i &= \sum_{j \in \mathcal{N}_i} a_{ij} \mathbf{v}_j. \end{aligned} \tag{12}$$

Here, $Q \in \mathbb{R}^{N \times d_k}$, $K \in \mathbb{R}^{N \times d_k}$, and $V \in \mathbb{R}^{N \times d_v}$ denote the query, key, and value matrices. This gather-based implementation avoids dense $N \times N$ logits. Because each query accesses at most $k+1$ keys, attention computation and weight storage are $O(Nk)$ for fixed k, excluding graph construction and other pipeline costs.

4.4 Stage 2: Reward-Guided Inference

We sample with DDIM (Song et al., 2021a), using the frozen evaluator E_ψ to bias the learned latent distribution toward a target score S_{target}.

Starting from $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we seek at each reverse step t to maximize the probability of the target property. We model $p_\psi(S_{\text{target}} \mid \mathbf{z}_t) = \mathcal{N}(S_{\text{target}}; E_\psi(\mathbf{z}_t), \sigma_S^2)$. Its gradient is proportional to the prediction-error gradient (Chung et al., 2023):

$$\nabla_{\mathbf{z}_t} \log p_\psi(S_{\text{target}} \mid \mathbf{z}_t) = -\frac{1}{2\sigma_S^2} \nabla_{\mathbf{z}_t} \| E_\psi(\mathbf{z}_t) - S_{\text{target}} \|_2^2. \tag{13}$$

Following classifier guidance (Dhariwal & Nichol, 2021), we scale this gradient by the reward-guidance coefficient $\gamma \geq 0$ and modify the unconditional prediction:

$$\hat{\epsilon}_t(\mathbf{z}_t) := \epsilon_\zeta(\mathbf{z}_t, t) - \gamma \sqrt{1 - \bar{\alpha}_t} \nabla_{\mathbf{z}_t} \log p_\psi(S_{\text{target}} \mid \mathbf{z}_t). \tag{14}$$

We substitute $\hat{\epsilon}_t$ into the deterministic DDIM update given in Appendix C. Repeating this update steers the trajectory toward latents assigned higher reward by E_ψ.

5 Experiments

5.1 Experimental Setup

For the RGFS+ variant, we implement the graph-induced sparse mask within the self-attention layers of both the VAE encoder and the diffusion model.

Table 1: Overall comparison. Classification entries report weighted F1 score (%), whereas regression entries report $100 \times (1 - \mathrm{RAE})$. The best results are **bolded** and the second-best results are underlined. Our proposed RGFS and RGFS+ are highlighted in gray.

Dataset	Original	mRMR	GAN	MCDM	RFE	LASSO	LNet	GFS	MARLFS	GRACES	VTFS	LLM-Select	RGFS	RGFS+
SpectF	75.96	78.21	79.16	80.36	80.80	79.16	75.96	75.01	75.01	81.86	84.58	86.91	92.65	**92.72**
SVMGuide3	77.81	76.84	77.91	76.66	78.07	77.91	76.44	83.12	76.84	78.07	85.02	79.79	85.02	**85.39**
German Credit	64.88	66.79	66.31	70.85	64.86	66.40	63.97	67.54	66.31	69.85	73.50	66.09	75.36	**80.61**
UCI Credit	80.19	80.59	80.59	74.46	80.28	77.94	80.05	79.96	80.24	80.59	81.21	74.65	**82.04**	**82.04**
SpamBase	92.68	92.34	92.34	88.95	91.68	91.81	91.67	92.25	92.35	92.68	**93.53**	85.65	**93.53**	**93.53**
Ionosphere	92.85	94.27	94.27	88.64	95.69	88.17	88.38	91.34	89.92	95.69	97.13	87.21	**98.55**	**98.55**
Mice-Protein	74.99	78.68	78.68	78.69	77.29	78.71	76.40	77.35	76.40	78.68	81.96	82.25	83.34	**83.86**
OpenML_586	54.95	57.64	60.74	57.95	58.10	60.67	58.28	62.27	58.27	62.27	**63.99**	52.17	**63.99**	**63.99**
OpenML_589	50.95	57.17	57.17	55.43	54.25	58.74	57.55	44.72	57.39	55.43	**61.13**	59.31	**61.13**	**61.13**
OpenML_607	51.73	55.17	57.88	55.56	54.39	58.10	55.38	45.70	54.99	57.88	62.72	61.26	63.88	**66.01**
OpenML_616	15.63	25.45	28.56	22.92	24.08	28.98	25.98	22.93	26.29	28.56	33.85	45.46	**55.31**	**55.31**
OpenML_618	46.89	51.08	52.40	50.90	50.64	47.41	51.11	52.40	51.87	51.73	55.91	49.23	**61.62**	60.15
OpenML_620	51.01	55.03	55.94	55.66	53.96	57.99	55.94	58.99	55.42	57.99	**62.58**	55.04	**62.58**	**62.58**
OpenML_637	14.95	20.49	21.12	22.16	17.82	26.02	19.43	39.12	20.75	26.02	**42.18**	35.68	**42.18**	**42.18**

Datasets. We perform experiments on 16 datasets from the UCI and OpenML repositories, spanning diverse domains. They consist of 14 standard benchmarks (21–77 features) used for the overall comparison, and 2 ultra-high-dimensional datasets (59,004–61,359 features) reserved for the feasibility study in Section 5.3. These datasets are classified based on their task types into two categories: (1) classification (C) and (2) regression (R). The statistical details of these datasets are presented in Table A.1.

Evaluation Design. For each dataset, we randomly partition the samples into an 80% search set and a 20% strictly held-out evaluation set. The search set is used to evaluate feature subsets and collect (feature mask, predictive score) pairs for constructing the performance-aligned latent space; the highest-scoring mask identified on the search set is then applied to the evaluation set. We employ Random Forest (RF) as the downstream model so that performance variations are primarily attributable to subset quality. On the 14 standard benchmarks, we use weighted F1 for classification and $1 -$ Relative Absolute Error (RAE) for regression. The two ultra-high-dimensional datasets use classification accuracy, reported as Acc. in Table 2. All tabulated scores are expressed as percentages, and higher values indicate better feature-subset performance.

Baselines. We compare RGFS and RGFS+ with 11 established selectors spanning filter (mRMR), generative knockoff (GAN), multi-criteria ranking (MCDM), wrapper (RFE), embedded (LASSO and LNet), reinforcement learning (GFS and MARLFS), graph-based (GRACES), latent generative (VTFS), and LLM-based (LLM-Select) methods (Peng et al., 2005; Jordon et al., 2019; Hashemi et al., 2022; Granitto et al., 2006; Tibshirani, 1996; Lemhadri et al., 2021; Fan et al., 2021; Liu et al., 2019; Chen et al., 2023; Ying et al., 2024; Jeong et al., 2025); full descriptions are provided in Appendix B.

Implementation details. See Appendix C for data collection, architecture, optimization, inference, hardware, and random-seed settings.

5.2 Overall Performance

Table 1 shows that RGFS or RGFS+ attains the best or tied-best score on all 14 standard benchmarks. The comparison spans filter, wrapper, embedded, reinforcement-learning, graph-based, latent-generative, and LLM-based selectors, indicating that the gains are not confined to one baseline family.

The largest margins occur on OpenML_616, where RGFS reaches 55.31% versus 45.46% for LLM-Select and 33.85% for VTFS, and on German Credit, where RGFS+ reaches 80.61% versus 73.50% for the strongest baseline. RGFS also improves over the latent-generative VTFS baseline on SpectF (92.65% vs. 84.58%). These results are consistent with independently initialized denoising trajectories covering useful candidate regions that may be missed when a latent candidate is optimized along a single trajectory; they do not imply that the global optimum is recovered.

RGFS+ improves on RGFS on German Credit, matches it on Ionosphere (both 98.55%), and slightly trails it on OpenML_618 (60.15% vs. 61.62%). By restricting each feature to its most strongly correlated neighbors, RGFS+ directly encodes the assumption that feature dependencies concentrate

in a sparse topology, focusing its capacity on the dominant interactions. Thus, the correlation graph acts as a data-dependent inductive bias rather than a uniform accuracy improvement. Its consistent benefit is computational: RGFS+ remains among the top two methods on every standard benchmark while enabling the high-dimensional runs studied next.

5.3 High-Dimensional Feasibility Study

Dense attention is the bottleneck RGFS+ targets: at the scale of the 21–77-feature benchmarks it is cheap, but on two classification datasets with 59,004 and 61,359 features, a single dense FP32 $N \times N$ attention matrix already requires 13–14 GiB. At this scale, the dense baselines and MARLFS all hit OOM. As Table 2 shows, RGFS+ completes both runs and beats LASSO by 5.0 and 13.3 accuracy points on OpenML_1082 and OpenML_1085. Under the reported per-output timing protocol, RGFS+ requires 0.18–0.19 s per mask, compared with 1.13–1.93 s for LASSO, corresponding to a $5.9\times$–$10.7\times$ reduction in per-output runtime. This comparison excludes RGFS+'s mask collection, representation learning, diffusion training, and graph construction costs.

Table 2: Ultra-high-dimensional feasibility. OOM: out of memory on one H100 80GB GPU; time: seconds per output mask.

Dataset (#Features)	Method	Acc. (%)	Time (s/mask)
OpenML_1082 (59,004)	MARLFS / VTFS / RGFS	OOM	
	LASSO	68.3	1.93
	RGFS+	**73.3**	**0.18**
OpenML_1085 (61,359)	MARLFS / VTFS / RGFS	OOM	
	LASSO	56.2	1.13
	RGFS+	**69.5**	**0.19**

5.4 Ablation Study

Figure 2 isolates reward guidance on four datasets. Removing it lowers the score in every case, by 1.48–2.86 points on three datasets and 18.80 points on OpenML_616, confirming that evaluator-guided denoising steers sampling toward higher-utility regions beyond what the unconditional diffusion prior already captures.

The gap is consistent across both classification and regression datasets, and is largest on OpenML_616 – the same dataset where RGFS shows its biggest margin over VTFS in the overall comparison.

5.5 Inference-Time Analysis

Figure 3 reports 11.3–17.7 ms per mask for RGFS and 13.5–21.8 ms for RGFS+, close to MARLFS (12.0–19.0 ms) and $75\times$–$97\times$ faster than VTFS. This gap reflects that one RGFS reverse-diffusion pass amortizes over a whole batch of candidates, whereas VTFS optimizes each candidate individually in latent space.

At these lower dimensions, RGFS+ incurs a small overhead over RGFS because sparse-neighborhood gathering is less hardware-efficient than dense kernels when N is small. The timing excludes mask collection, representation learning, training, and graph construction, so it complements rather than replaces the high-dimensional feasibility results.

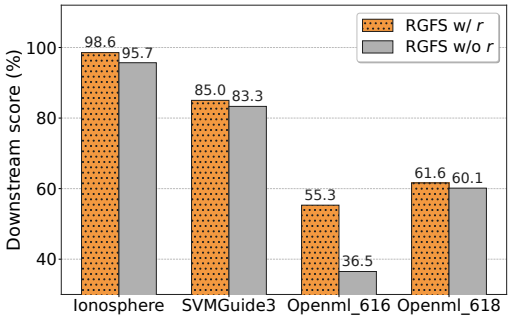


Figure 2: Ablation study of RGFS with and without reward guidance (r).

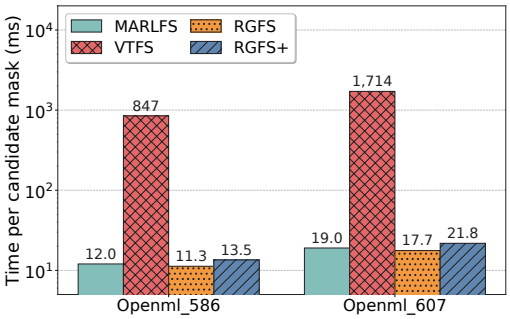


Figure 3: Inference cost per generated candidate mask (ms/mask).

Table 3: Subset compactness and stability over five random seeds. #Sel: number of selected features; Ratio: selection ratio; Stab.: pairwise Jaccard similarity across independent runs; Score: weighted F1 (%) for SpectF and $100 \times (1 - \mathrm{RAE})$ for OpenML_607.

Dataset	Method	#Sel	Ratio	Stab.	Score (%)
SpectF (44)	MARLFS	21.6±3.8	0.49	0.34	79.1±1.7
	VTFS	25.4±2.1	0.58	**0.75**	79.2±1.8
	RGFS+	**13.6±8.7**	**0.31**	0.28	**88.9±2.8**
OpenML_607 (50)	MARLFS	26.6±4.3	0.53	0.42	54.1±0.7
	VTFS	34.2±2.8	0.68	**0.84**	50.7±2.3
	RGFS+	**21.0±9.3**	**0.42**	0.29	**56.3±2.2**

5.6 Subset Compactness and Stability

To examine whether repeated runs collapse to nearly identical solutions, Table 3 reports pairwise Jaccard similarity, subset size, and predictive score over five random seeds. VTFS exhibits high cross-run similarity (0.75–0.84) but lower mean scores, indicating that its runs repeatedly return closely overlapping subsets across seeds. In contrast, RGFS+ attains the highest mean scores with substantially lower similarity (0.28–0.29), showing that independently initialized diffusion trajectories reach distinct high-utility subsets rather than a single recurring mask.

RGFS+ also selects the most compact subsets among the three methods: 13.6 features on SpectF and 21.0 on OpenML_607, corresponding to 31% and 42% of the inputs, respectively. Together with its low solution overlap across runs, this shows that RGFS+ explores a broader set of compact, high-utility solutions than VTFS rather than repeatedly converging to one region.

5.7 Robustness Check

We re-evaluate the selected subsets on OpenML_616 using five downstream regressors: Decision Tree, KNN, Linear Regression, SVR, and Random Forest. As shown in Table 4, RGFS+ obtains the best or tied-best value under all five models, reaching 54.14–55.31 while the strongest non-RGFS baseline in each column remains below 50. The narrow range, together with VTFS's drop to 33.85 under RF, indicates that the selected subsets carry model-agnostic predictive signal rather than being tuned to the reward-collection model. RGFS and RGFS+ are also close under DT, KNN, LR, and SVR, indicating that the sparse attention modification preserves this model-agnostic transfer.

Table 4: Robustness on OpenML_616 across five downstream regressors; entries report $100 \times (1 - \mathrm{RAE})$.

Method	DT	KNN	LR	SVR	RF
MARLFS	35.25	36.16	37.45	36.93	26.29
LLM-Select	46.14	47.59	48.94	48.29	45.46
VTFS	46.83	48.12	49.27	45.52	33.85
RGFS	53.06	53.36	53.18	55.24	**55.31**
RGFS+	**54.14**	**54.78**	**55.04**	**55.31**	**55.31**

6 Conclusion

We presented RGFS, which formulates feature selection as reward-guided latent denoising, and RGFS+, which constrains attention using a feature-correlation graph. Under the reported setting, RGFS or RGFS+ obtains the highest or tied-highest score on the 14 standard benchmarks, and the reward-guidance ablation supports evaluator-guided sampling on four datasets. RGFS+ also remains runnable on two datasets with approximately 60,000 features where the evaluated dense alternatives encounter OOM. Together, these results support reward-guided diffusion for feature selection and graph-induced sparse attention for high-dimensional feasibility under the evaluated conditions.

AI USE STATEMENT

In this work, the authors used generative AI tools only for language polishing, including improvements to grammar, clarity, and readability. Generative AI tools were not used to generate research ideas, design the methodology or experiments, implement the methods, generate or clean data, formulate mathematical claims, or analyze or interpret the results. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

Sohyun An, Hayeon Lee, Jaehyeong Jo, Seanie Lee, and Sung Ju Hwang. Diffusionnag: Predictor-guided neural architecture generation with diffusion models, 2024. URL https://arxiv.org/abs/2305.16943.

R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994. doi: 10.1109/72.298224.

Can Chen, Scott T Weiss, and Yang-Yu Liu. Graph convolutional network-based feature selection for high-dimensional and low-sample size data. *Bioinformatics (Oxford, England)*, 39(4):btad135, April 2023. ISSN 1367-4811. doi: 10.1093/bioinformatics/btad135.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. https://arxiv.org/abs/1802.07814v2, February 2018.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *The Eleventh International Conference on Learning Representations*, 2023.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869, September 2023. ISSN 1939-3539. doi: 10.1109/tpami.2023.3261988. URL http://dx.doi.org/10.1109/TPAMI.2023.3261988.

Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis, June 2021.

Wei Fan, Kunpeng Liu, Hao Liu, Ahmad Hariri, Dejing Dou, and Yanjie Fu. AutoGFS: Automated group-based feature selection via interactive reinforcement learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 342–350. 2021. doi: 10.1137/1.9781611976700.39.

Nanxu Gong, Zijun Li, Sixun Dong, Haoyue Bai, Wangyang Ying, Xinyuan Wang, and Yanjie Fu. Sculpting Features from Noise: Reward-Guided Hierarchical Diffusion for Task-Optimal Feature Transformation, May 2025. URL https://arxiv.org/abs/2505.15152.

Pablo M. Granitto, Cesare Furlanello, Franco Biasioli, and Flavia Gasperi. Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems*, 83(2):83–90, 2006. ISSN 0169-7439. doi: 10.1016/j.chemolab.2006.01.007.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3(null):1157–1182, March 2003. ISSN 1532-4435.

Amin Hashemi, Mohammad Bagher Dowlatshahi, and Hossein Nezamabadi-pour. Ensemble of feature selection algorithms: A multi-criteria decision-making approach. *International Journal of Machine Learning and Cybernetics*, 13(1):49–69, January 2022. ISSN 1868-808X. doi: 10.1007/s13042-021-01347-z.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d, 2022. URL https://arxiv.org/abs/2203.17003.

Jaber Jemai and Anis Zarrad. Feature selection engineering for credit risk assessment in retail banking. *Information*, 14(3), 2023. ISSN 2078-2489. doi: 10.3390/info14030200. URL https://www.mdpi.com/2078-2489/14/3/200.

Daniel P Jeong, Zachary Chase Lipton, and Pradeep Kumar Ravikumar. LLM-select: Feature selection with large language models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=16f7ea1N3p.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Knockoffgan: Generating knockoffs for feature selection using generative adversarial networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ByeZ5jC5YQ.

YeongSeog Kim, W. Nick Street, and Filippo Menczer. Feature selection in unsupervised learning via evolutionary search. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pp. 365–369, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132336. doi: 10.1145/347090.347169. URL https://doi.org/10.1145/347090.347169.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. URL https://arxiv.org/abs/1312.6114.

Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, ML92, pp. 249–256, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. ISBN 15586247X.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(97)00043-X. URL https://www.sciencedirect.com/science/article/pii/S000437029700043X. Relevance.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, 2021. URL https://arxiv.org/abs/2009.09761.

Igor Kononenko. Estimating attributes: analysis and extensions of relief. In *Proceedings of the European Conference on Machine Learning*, ECML-94, pp. 171–182, Berlin, Heidelberg, 1994. Springer-Verlag. ISBN 3540578684.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals, 2017. URL https://arxiv.org/abs/1605.07648.

Ismael Lemhadri, Feng Ruan, Louis Abraham, and Robert Tibshirani. Lassonet: A neural network with feature sparsity. *Journal of Machine Learning Research*, 22(127):1–29, 2021. URL http://jmlr.org/papers/v22/20-848.html.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Comput. Surv.*, 50(6), December 2017. ISSN 0360-0300. doi: 10.1145/3136625. URL https://doi.org/10.1145/3136625.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation, 2022. URL https://arxiv.org/abs/2205.14217.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models, 2023. URL https://arxiv.org/abs/2301.12503.

Kunpeng Liu, Yanjie Fu, Pengfei Wang, Le Wu, Rui Bo, and Xiaolin Li. Automating Feature Subspace Exploration via Multi-Agent Reinforcement Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 207–215, Anchorage AK USA, July 2019. ACM. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330868.

Kleanthis Malialis, Jun Wang, Gary Brooks, and George Frangou. Feature selection as a multiagent coordination problem, 2016. URL https://arxiv.org/abs/1603.05152.

Narendra and Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26(9):917–922, 1977. doi: 10.1109/TC.1977.1674939.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. URL https://arxiv.org/abs/2102.09672.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers, March 2023.

Hanchuan Peng, Fuhui Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005. doi: 10.1109/TPAMI.2005.159.

Pavel Pudil, Jana Novovicová, and Josef Kittler. Floating search methods in feature selection. *Pattern Recognit. Lett.*, 15:1119–1125, 1994. URL https://api.semanticscholar.org/CorpusID:270333833.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021b. URL https://arxiv.org/abs/2011.13456.

V. Sugumaran, V. Muralidharan, and K.I. Ramachandran. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. *Mechanical Systems and Signal Processing*, 21(2):930–942, 2007. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2006.05.004. URL https://www.sciencedirect.com/science/article/pii/S0888327006001142.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1996.tb02080.x.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers, 2021. URL https://arxiv.org/abs/2103.17239.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8.

Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature Selection using Stochastic Gates. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 10648–10659. PMLR, November 2020.

J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications*, 13(2):44–49, 1998. doi: 10.1109/5254.671091.

Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pp. 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1558604863.

Wangyang Ying, Dongjie Wang, Haifeng Chen, and Yanjie Fu. Feature Selection as Deep Sequential Generative Learning. *ACM Transactions on Knowledge Discovery from Data*, 18(9):1–21, November 2024. ISSN 1556-4681, 1556-472X. doi: 10.1145/3687485.

Lei Yu and Huan Liu. Feature selection for high-dimensional data: a fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Learning*, ICML'03, pp. 856–863. AAAI Press, 2003. ISBN 1577351894.

Weiliang Zhang, Zhen Meng, Dongjie Wang, Min Wu, Kunpeng Liu, Yuanchun Zhou, and Meng Xiao. Enhanced gene selection in single-cell genomics: Pre-filtering synergy and reinforced optimization, 2024. URL https://arxiv.org/abs/2406.07418.

A Dataset Statistics

Table A.1: Summary of Dataset Statistics. The bottom group contains the ultra-high-dimensional datasets used in the feasibility study (Section 5.3).

Dataset	Task	#Samples	#Features
SpectF	C	267	44
SVMGuide3	C	1,243	21
German Credit	C	1,000	24
UCI Credit	C	30,000	25
SpamBase	C	4,601	57
Ionosphere	C	351	34
Mice-Protein	C	1,080	77
OpenML_586	R	1,000	25
OpenML_589	R	1,000	25
OpenML_607	R	1,000	50
OpenML_616	R	500	50
OpenML_618	R	1,000	50
OpenML_620	R	1,000	25
OpenML_637	R	500	50
OpenML_1082	C	92	59,004
OpenML_1085	C	159	61,359

B Baseline Definitions

1. mRMR (Peng et al., 2005) selects a subset by maximizing feature–label relevance while minimizing redundancy among selected features.
2. GAN (Jordon et al., 2019) (KnockoffGAN) generates knockoff features via generative adversarial training, enabling feature importance estimation beyond Gaussian assumptions.
3. MCDM (Hashemi et al., 2022) formulates feature ranking as a multi-criteria decision-making problem and aggregates multiple selectors (e.g., via VIKOR) to obtain a consensus ranking.
4. RFE (Granitto et al., 2006) performs recursive feature elimination by iteratively removing the least important features under a chosen predictor.
5. LASSO (Tibshirani, 1996) induces sparsity through an ℓ_1 penalty, driving irrelevant feature coefficients toward zero.
6. LNet (Lemhadri et al., 2021) (LassoNet) integrates sparsity constraints into a neural network to encourage selecting only a compact subset of input features.
7. GFS (Fan et al., 2021) is an interactive reinforcement learning-based method that selects features in a group-wise manner.
8. MARLFS (Liu et al., 2019) assigns an agent to each feature and learns selection policies with rewards reflecting redundancy and downstream performance.
9. GRACES (Chen et al., 2023) leverages sample relationships by converting tabular data into a graph structure and selecting features via graph-aware scoring.
10. VTFS (Ying et al., 2024) learns a feature-subset embedding space with a variational Transformer and conducts selection through generative modeling in that space.
11. LLM-Select (Jeong et al., 2025) uses large language models to rank features based on textual descriptions and task prompts (when available), producing an importance ordering for selection.

C Implementation Details

Data Collector. We use the reinforcement data collector (MARLFS) to explore the feature space, generating 200–300 candidate subsets per experiment (depending on dataset dimensionality) to collect feature mask–reward pairs. For the ultra-high-dimensional datasets OpenML_1082 and OpenML_1085, the standard full-scale MARLFS evaluation encounters out-of-memory errors because of dense feature interactions. To obtain the training data nevertheless, we ran the MARLFS collector with the configured hyperparameters in a conservative, low-throughput setting and accepted substantially longer runtimes; the resulting candidate masks were scored on the downstream task to form the mask–reward pairs used to train RGFS and RGFS+. Thus, the OOM entries for MARLFS in Table 2 refer to the standard full-scale baseline evaluation, whereas the low-throughput MARLFS run was used only for data collection.

Training Data Expansion. For VAE training, we generate 25 random permutations of the selected-feature index sequence for each collected subset, retaining its reward label and the original sequence. This yields 26 training records per subset, or 5,200–7,800 records for 200–300 collected subsets. Each sequence is converted to a fixed-order binary mask before being passed to the VAE. Consequently, permutations of the same subset yield identical masks: this expansion repeats existing mask–reward pairs rather than introducing distinct feature subsets. The VAE batch size of 1024 refers to these expanded training records.

Model Architecture. Feature tokens are mapped to 64-dimensional embeddings. The VAE utilizes a Transformer-based encoder-decoder (2 layers, 8 heads, feed-forward dim 128, latent dim 64), while the estimator is a 2-layer MLP with 200 hidden units. For the RGFS+ variant, we set the neighbor count $k = 5$.

Denoiser Block. Each of the L blocks conditions two self-attention layers and a two-layer feed-forward network on the timestep embedding through AdaLN:

$$\begin{aligned}
\mathbf{h}^{(\ell,1)} &= \mathbf{h}^{(\ell-1)} + \mathrm{SA}_1\Big(\mathrm{AdaLN}(\mathbf{h}^{(\ell-1)}, \mathbf{t}_{\mathrm{emb}})\Big), \\
\mathbf{h}^{(\ell,2)} &= \mathbf{h}^{(\ell,1)} + \mathrm{SA}_2\Big(\mathrm{AdaLN}(\mathbf{h}^{(\ell,1)}, \mathbf{t}_{\mathrm{emb}})\Big), \\
\mathbf{h}^{(\ell)} &= \mathbf{h}^{(\ell,2)} + \mathrm{FFN}\Big(\mathrm{AdaLN}(\mathbf{h}^{(\ell,2)}, \mathbf{t}_{\mathrm{emb}})\Big).
\end{aligned} \tag{15}$$

LayerScale and DropPath are applied to each residual branch before addition.

Guided DDIM Update. At reverse step t, the guided prediction $\hat{\epsilon}_t$ replaces the unconditional noise estimate in the deterministic update:

$$\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left(\frac{\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}_t(\mathbf{z}_t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}_t(\mathbf{z}_t). \tag{16}$$

Training and Inference. The diffusion model is trained for 800 epochs (batch size 8, learning rate 10^{-4}, Adam optimizer, gradient clipping 5.0). The loss weights of the VAE training objective are fixed to $\alpha = 0.8$ and $\beta = 0.001$ across all datasets. The VAE is trained with a batch size of 1024. The number of inference steps is fixed at 1,410. During inference, we use a reward-guidance coefficient of $\gamma = 100$ ($\gamma = 500$ for Mice-Protein) with an evaluator target of $S_{\text{target}} = 1.0$, and select the best outcome. Setting $\gamma = 0$ disables reward guidance.

Environment and Random Seeds. All experiments are conducted on NVIDIA H100 GPUs using PyTorch 2.0.1. The overall comparison in Table 1 uses a fixed random seed of 42, while the subset compactness and stability analysis is conducted over five random seeds.