Hypernetworks for Dynamic Feature Selection

Javier Fumanal-Idocin
University of Essex
Wivenhoe Park, Colchester, United Kingdom
j.fumanal-idocin@essex.ac.uk

Raquel Fernandez-Peralta
Slovak Academy of Sciences
Štefánikova 49, 814 73 Bratislava, Slovakia
raquel.fernandez@mat.savba.sk

Javier Andreu-Perez
University of Essex
Wivenhoe Park, Colchester, United Kingdom
j.andreu-perez@essex.ac.uk

Abstract

Dynamic feature selection (DFS) is a machine learning framework in which features are acquired sequentially for individual samples under budget constraints. The exponential growth in the number of possible feature acquisition paths forces a DFS model to balance fitting specific scenarios against maintaining general performance, even when the feature space is moderate in size. In this paper, we study the structural limitations of existing DFS approaches to achieve an optimal solution. Then, we propose HYPER-DFS, a hypernetwork-based DFS approach that generates feature subset-specific classifier parameters on demand. We show that the use of hypernetworks compared to mask-embedding methods results in a smaller structural complexity bound. We also use a Set Transformer encoding to create a smooth conditioning space for the hypernetwork, so that functionally similar tasks are also geometrically close. In our benchmarks, HYPER-DFS performed best or second best compared to all state-of-the-art approaches on synthetic and real-life tabular data. It is also best or second best across all image datasets tested, and shows stronger zero-shot generalisation to feature subsets never seen during training than existing DFS approaches.

Code available in: https://github.com/Fuminides/hyper_DFS

1 Introduction

Machine learning (ML) has achieved remarkable success in many real-life scenarios [Paleyes et al., 2022]. However, the deployment of ML systems in budget-constrained environments still presents difficulties. For example, data is often incomplete and subject to acquisition costs that can vary dynamically with operational constraints [Erion et al., 2022]. Dynamic feature selection (DFS), also known as active feature learning, is a ML paradigm designed for such settings, in which a different set of features is chosen for each processed sample based on the information available [Karayev et al., 2013]. Because of this, the model can be adapted to different conditions where some features are more relevant to a particular case, or where economic and time constraints prevent access to some information.

Existing approaches to DFS are broadly categorised into three paradigms: generative models, reinforcement learning-based policies and greedy acquisition methods. Generative models learn the conditional distribution of the missing features [Ma et al., 2019]. Reinforcement learning-based approaches frame the sequential feature acquisition as a Markov Decision Process, learning a policy that maximises a long-term reward signal [Janisch et al., 2019]. Greedy methods query at each step the feature that optimises an immediate utility metric, with Conditional Mutual Information

(CMI) emerging as a particularly popular criterion [Gadgil et al., 2024], although some concerns have been raised about the validity of this metric [Norcliffe et al., 2025]. These approaches generally use a single shared predictor for all possible observation statuses, as having a dedicated predictor for each is impossible due to their exponential number. Approximated per-subset predictors have been proposed in the literature [Valancius et al., 2024], but they incur significant computational and memory overhead that prevents scaling to large feature spaces.

Although the value proposition of DFS is clear, existing approaches present several limitations: lack of clear superiority compared to static selection models [Erion et al., 2022]; the DFS models' decisions can be unreliable during the sequential acquisition phase [Fumanal-Idocin et al., 2025]; and tendency to overfit to small subsets of features, especially relevant if the acquisition policy is heavily biased towards selecting the same features for every sample [Liu et al., 2025].

In this paper, we address these issues by studying the conceptual limitations of existing DFS models. More precisely, we study how feasible it is for a DFS classifier to return optimal predictions for every possible feature subset. We study the difference between the ideal objective and the tractable training objective used in the literature to train DFS predictors. We also argue that mask-concatenation approaches, in which the observed subset S is encoded as a conditioning signal to a shared neural network, are instances of *embedding methods* as described in Galanti and Wolf [2020], which means that they incur a complexity requirement that scales with both the input dimension and the task-conditioning dimension simultaneously, rather than with each independently. Then, we propose HYPER-DFS, a hypernetwork-based framework that generates subset-specific classifier parameters on demand using a Set Transformer to encode the task into a continuous representation. We show how to train the DFS policy for HYPER-DFS and benchmark it against other previous single and multi-model DFS approaches.

In summary, our contributions are the following:

- We study the fundamental limitations of existing DFS approaches, and show how tractable objectives diverge from the ideal DFS solution and how a hypernetwork can close this gap.
- We propose HYPER-DFS, a hypernetwork-based framework that generates subset-specific classifiers on demand. We use Set Transformers to map discrete feature masks into a continuous representation that promotes smooth interpolation over different tasks, which also helps generalisation to feature subsets unseen during training.
- We identify and address two hypernetwork training issues particularly relevant in the DFS setting: variable subset cardinality causes variance drift in the generated weights, which we solve using a feature absence embedding vector; and multiple conditioning tasks make the hypernetwork training unstable, which we solve using a controlled distribution of masks per batch during training.
- We evaluate HYPER-DFS on a broad benchmark suite spanning synthetic and real data, in tabular and image domains, demonstrating consistent improvements over single-model and multi-model baselines.

2 Preliminaries

Notation. Let $\mathbf{x} = (x_1, \ldots, x_M) \in \mathcal{X} \subseteq \mathbb{R}^M$ denote a sample with M features, $y \in \mathcal{Y}$ its label, $S \subseteq [M] = \{1, \ldots, M\}$ a subset of observed features and $\bar{S} = [M] \setminus S$ for its complement. We denote as $\mathbf{I}_S \in \{0,1\}^M$ the binary mask that marks which features are present in S, which we also call the *knowledge status*. We use $\mathbf{x}_S = (x_j)_{j \in S} \in \mathbb{R}^{|S|}$ to denote the subvector of observed entries. We use S_t to indicate the feature set observed in the moment t of the sequential feature acquisition process.

2.1 Dynamic Feature Selection

In standard supervised learning, a fixed set of features is assumed to be available for every sample at both training and test time. DFS relaxes this assumption by allowing features to be acquired sequentially, on a per-instance basis and under budget constraints [Karayev et al., 2013]. A DFS system consists of two interacting components: a *selector* and a *predictor*. The selector $q_\psi \colon \mathbb{R}^M \times \{0,1\}^M \to \{1, \ldots, M\}$ selects the feature to query. The predictor $f_\theta \colon \mathbb{R}^M \times \{0,1\}^M \to \mathcal{Y}$ produces a prediction from

the observed features and knowledge status. Since accommodating variable-sized inputs in the predictor can be impractical, a common approach is the *impute-then-predict* paradigm [Le Morvan et al., 2020], which consists of imputing the missing features $\bar{S}$ before prediction. The imputation is typically done with the mean, although more sophisticated generative schemes have also been studied [Ma et al., 2019]; in non-linear problems, however, the choice of imputation has limited impact on predictive performance, as model expressivity largely compensates for imputation errors [Le Morvan and Varoquaux, 2025]. We denote by $\tilde{\mathbf{x}}_S$ the vector $\mathbf{x}_S$ with values in $\bar{S}$ replaced by their means. The predictor is then trained to minimise the expected loss over feature subsets:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_S \, \mathbb{E}_{x,y} \, \mathcal{L}(f_\theta(\tilde{\mathbf{x}}_S, \mathbf{I}_S), y), \tag{1}$$

where the distribution of the feature sets can be given by a uniform distribution, or by the selector. The selector is typically trained jointly with the predictor, either through reinforcement learning [Janisch et al., 2019, Yoon et al., 2018], greedy maximisation of an information-theoretic acquisition criterion, like CMI [Covert et al., 2023, Gadgil et al., 2024, Chattopadhyay et al., 2023], or hand-crafted scoring functions [Norcliffe et al., 2025].

2.2 Hypernetworks

The idea of using one neural network to generate the weights of another has roots in early work on fast weight programmers [Schmidhuber, 1992], where a "slow" network learned to produce context-dependent weight changes for a "fast" network. This concept was formalised and scaled to modern deep learning architectures by Ha et al. [2017], who introduced the term *hypernetwork* to describe a network that takes a conditioning input and outputs the parameters of a separate *primary* network.

Formally, let g_ϕ denote the hypernetwork with parameters ϕ, which produces the weights $\theta_z = g_\phi(z)$ of the primary network f_θ conditioned on an input z. The output for a data point x is then $\hat{y} = f_{g_\phi(z)}(x)$, and the hypernetwork parameters are optimised end-to-end:

$$\min_{\phi} \; \mathbb{E}_{x,y,z} \left[\mathcal{L}\left(f_{g_\phi(z)}(x), \, y \right) \right]. \tag{2}$$

Because the loss is backpropagated through both f_θ and g_ϕ, the hypernetwork learns weight-generation strategies directly optimised for the downstream task. A key property of hypernetworks is that they encode an entire family of related models within a single set of parameters ϕ, achieving a form of soft weight sharing across conditioning inputs [Chauhan et al., 2024].

3 Structural Risks in Dynamic Feature Selection

3.1 Compromises in DFS predictors

The ideal DFS predictor would require a Bayes-optimal parameter set, θ^*_S, for every subset, solving:

$$\theta^*_S = \arg\min_{\theta_S} \mathbb{E}_{x,y} \, \mathcal{L}(f_{\theta_S}(\mathbf{x}_S), y), \qquad \forall S \subseteq [M]. \tag{3}$$

However, this is not generally feasible in practice as it would require 2^M different parametrizations. Because of that, prior work in DFS has mostly focused on one shared parametrization that amortises the cost of learning the 2^M individual predictors. This shared predictor is normally trained first using Eq. (1) as target loss with a uniform distribution over S, so that it performs well on average on any feature subset. Then, it is trained using feature sets sampled from the selector q_ψ, so that: $S_{t+1} = S_t \cup \{i_t\}$ with $i_t \sim q_\psi(\tilde{\mathbf{x}}_{S_t}, \mathbf{I}_{S_t})$. This two-step solution is designed to mitigate the problems that arise because the number of possible knowledge statuses is 2^M. The first phase encourages f_θ to perform well on arbitrary subsets; the second phase concentrates the gradient signal on the subsets that q_ψ actually selects, which typically form a much smaller and more structured region of $2^{[M]}$.

This amortisation is what makes the DFS problem tractable, but it comes at a cost. Because a single θ must amortise the cost of such a big task space simultaneously, it depends on how well the Bayes-optimal predictors can be represented under the same shared structure. When this can be done, the compromise is mild. Otherwise, the shared θ is optimal on average but not necessarily close to optimal on any individual subset. Conflicts can also happen at the gradient level, so that updates

beneficial for one subset actively degrade performance on another [Liu et al., 2021]. Moreover, subsets that q_ψ rarely or never visits receive little or no gradient signal at all, so the predictor may systematically underperform on them, despite the uniform-subset pre-training phase [Fumanal-Idocin et al., 2025]. So, in practical settings, we can assume that there will be some degree of compromise for some feature sets [Muller et al., 2025, Lobo et al., 2025].

3.2 From Ideal to Tractable: Hypernetworks

If we replace the discrete lookup table $\{\theta_S\}_{S\subseteq[M]}$ in the ideal objective (Eq. (3)) with a function $g_\phi\colon \{0,1\}^M \to \Theta$ that generates model parameters from the knowledge status $\mathbf{I}_S$:

$$\phi^* = \arg\min_{\phi} \mathbb{E}_S \, \mathbb{E}_{x,y} \, \mathcal{L}\big(f_{g_\phi(\mathbf{I}_S)}(\tilde{\mathbf{x}}_S),\, y\big), \tag{4}$$

we naturally obtain a hypernetwork objective [Ha et al., 2017]. Variable g_ϕ generates the parameters of the primary network f, which then produces predictions from the input $\tilde{\mathbf{x}}_S$. Compared with the optimal solution (Eq. (3)), the cost of learning and storing an exponential set of parameters is replaced by the cost of learning a single network g_ϕ. Compared with the tractable solution (Eq. (1)), the predictor parameters are now a function of $\mathbf{I}_S$, so we are not forced to find a single primary network to fit all subsets. Galanti and Wolf [2020] formalise the structural advantage of hypernetworks over mask-conditioned predictors. A mask-conditioned predictor is an instance of an *embedding method*: the subset S enters the network as a conditioning signal that is fused with the data features. For data dimension m_1, task dimension m_2, and target-function smoothness parameter r (in the DFS setting, $m_1 = M$ and $m_2 = M$ for binary-mask conditioning), Theorem 4 in Galanti and Wolf [2020] shows that a hypernetwork primary network of complexity $\mathcal{O}(\varepsilon^{-m_1/r})$ is sufficient to achieve error $\leq \varepsilon$, while their Theorems 2 and 3 show that the primary network of any embedding method must have complexity at least $\Omega(\varepsilon^{-(m_1+m_2)})$.

Prior work also provides evidence that hypernetwork training with gradient descent has been successfully applied for amortised inference across large task families [Zhao et al., 2020, Beck et al., 2023a, von Oswald et al., 2020]. However, the amortisation advantage also depends on the geometry of the conditioning space because g_ϕ is a neural network with bounded Lipschitz constant [Bartlett et al., 2017]. This means that it cannot produce very different primary networks for nearby conditioning inputs. So, if the optimal parameters θ_S^* change abruptly between similar subsets, the hypernetwork is forced into a compromise. We mitigate this in our proposal by encoding S through a Set Transformer (Section 4.1). This produces a continuous representation of the knowledge state, in which feature subsets that induce similar Bayes-optimal predictors map to nearby points. In this way, the Lipschitz constraint helps generalisation rather than becoming a limitation.

4 Proposed Method

In this section, we introduce a new DFS approach: HYPER-DFS. Unlike prior work that relies on a conditioning vector to model the observation patterns over the same predictor model, HYPER-DFS generates subset-specific predictors on demand via a hypernetwork. To model the observation status, we use a Set Transformer representation [Lee et al., 2019], which maps discrete observation masks into a smooth, functional-aware geometry. Finally, to maintain the size of the hypernetwork small, HYPER-DFS uses a neural network to reduce the size of the primary network input, called the *compressor*, which is common practice in the hypernetworks literature [Chauhan et al., 2024].

4.1 Smooth Feature Set Encoding for HYPER-DFS

A key design question for HYPER-DFS is how to represent the knowledge status as a conditioning signal for the hypernetwork g_ϕ. Prior work uses binary masks. We argue this is a poor choice because binary masks treat all feature positions as interchangeable. This means that the encoding has no way to represent that some features are semantically related. So, two subsets that differ only in one semantically similar feature receive representations as distant as two subsets that differ in one entirely unrelated feature.

We instead propose encoding the observed feature subset using a Set Transformer encoder built from Induced Set Attention Blocks (ISAB) [Lee et al., 2019]. Let $\mathbf{e}_i^0, \mathbf{e}_i^1 \in \mathbb{R}^d$ denote two learned

embeddings for feature i, representing its absent and present states respectively, where d is the token-embedding dimension. Given the knowledge status $\mathbf{I}_S \in \{0,1\}^M$, we form the input matrix $\mathbf{T}(\mathbf{I}_S) \in \mathbb{R}^{M\times d}$ whose i-th row is $\mathbf{t}_i = I_{S,i}\,\mathbf{e}_i^1 + \big(1 - I_{S,i}\big)\,\mathbf{e}_i^0$, and apply the Set Transformer $f_\omega \colon \mathbb{R}^{M\times d} \to \mathbb{R}^{M\times d}$ with parameters ω, followed by sum-pooling along the token dimension and a neural output transformation $f_\rho \colon \mathbb{R}^d \to \mathbb{R}^{d'}$ with parameters ρ that maps the pooled representation to the conditioning vector of dimension d':

$$\mathbf{z}_{\omega,\rho}(\mathbf{I}_S) = f_\rho\left(\sum_{i=1}^{M}\big[f_\omega(\mathbf{T}(\mathbf{I}_S))\big]_i\right), \qquad \tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S) = \frac{\mathbf{z}_{\omega,\rho}(\mathbf{I}_S)}{\|\mathbf{z}_{\omega,\rho}(\mathbf{I}_S)\|_2}, \tag{5}$$

where $\big[f_\omega(\mathbf{T}(\mathbf{I}_S))\big]_i \in \mathbb{R}^d$ denotes the i-th output token of the encoder and $\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S) \in \mathbb{R}^{d'}$ is the conditioning vector passed to the hypernetwork. Both d and d' are hyperparameters of the architecture. The self-attention mechanism allows each feature's contribution to be modulated by the other features present, capturing inter-feature dependencies in the conditioning signal. Using distinct embeddings for absent and present states also prevents a systematic bias that would arise with presence-only embeddings: if the pooled representation depended only on $\sum_{i\in S}\mathbf{e}_i^1$, its norm would scale with $|S|$, biasing the normalisation statistics and gradient magnitudes towards the observed subset size. We discuss this in detail in Appendix A. The parameters (ω, ρ) are trained end-to-end with the hypernetwork and primary network.

4.2 Training the HYPER-DFS

Training the predictor. The predictor is trained using a variant of Eq. (4) with the encoding defined in Eq. (5) and a compressor $c_\eta \colon \mathbb{R}^M \to \mathbb{R}^{M'}$, where M' is a hyperparameter:

$$\arg\min_{\phi,\omega,\rho,\eta} \mathbb{E}_S\, \mathbb{E}_{x,y}\, \mathcal{L}(f_{g_\phi(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S))}(c_\eta(\tilde{\mathbf{x}}_S)),\, y). \tag{6}$$

However, this training objective has two problems. First, mixing many different knowledge statuses in the same batch can be problematic, especially at the beginning of training, as the gradient may cancel if the contributions from different S point in opposing directions [Yu et al., 2020]. Secondly, the compressor receives a backward signal that is itself modulated by the generated, mask-dependent primary network, which at initialisation behaves as a collection of random projection heads. This is the setting in which shared representations are known to be most prone to collapse [Chen and He, 2021]. We did not empirically observe this to be a problem in most cases, but it can be problematic in specific datasets.

To mitigate the issue, we introduce two complementary modifications. First, we introduce a pre-training phase in which each batch uses a fixed small number of knowledge status vectors per batch, so that the batch uses a single primary network. This removes possible within-batch cancelling S-specific gradient directions. Second, we apply a linear learning-rate warm-up over the first T_{warm} steps, which allows the hypernetwork and the compressor to co-adapt before the optimiser takes large steps and the hypernetwork commits to a particular mapping. The problem is described in more detail in Appendix C. The empirical study that shows the validity of the proposed solution is in Appendix D.1. This pre-training phase produces a warm-started predictor that is subsequently fine-tuned jointly with the selector, as described in the next paragraph.

Training the selector. The selector is trained similarly to other approaches in the literature [Chattopadhyay et al., 2023]. It consists of a small neural network q_ψ that takes as input the observed features and the knowledge status and produces a score for each candidate feature, then returns the feature with the highest score among the unobserved features. To train it, for each training sample $(\mathbf{x}, y)$, we draw a trajectory length $\tau \sim \text{Uniform}(\{1, \ldots, B_{\max}\})$, where $B_{\max}$ is the maximum acquisition budget the model is expected to handle at test time, and let the selector pick τ features one at a time. At each step t, the selector produces scores $\mathbf{s}_\psi(\tilde{\mathbf{x}}_{S_t}, \mathbf{I}_{S_t}) \in \mathbb{R}^M$, with scores of already-observed features set to $-\infty$ so they cannot be selected again. The chosen feature i_t is then used to update the knowledge status, $S_{t+1} = S_t \cup \{i_t\}$. Because $\arg\max$ is not differentiable, we replace it with the Gumbel-softmax with straight-through estimator [Jang et al., 2017, Maddison et al., 2017] during training, which lets gradients flow through each selection step. After τ steps, we apply the predictor to obtain $\hat{y} = f_{g_\phi(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_{S_\tau}))}(c_\eta(\tilde{\mathbf{x}}_{S_\tau}))$ and compute the cross-entropy loss $\mathcal{L}(\hat{y}, y)$, which is backpropagated through the full trajectory to update the selector and predictor parameters

jointly. At test time, the Gumbel-softmax is no longer needed, and the selector simply picks the highest-scoring unobserved feature at each step, recovering the deterministic q_ψ defined above.

5 Experimentation

5.1 Datasets and Methods

Datasets. We evaluate HYPER-DFS on a benchmark suite spanning synthetic, real-world tabular, and image domains. *Synthetic* datasets: the three binary tasks of Yoon et al. [2018] and **Cube** [Kompella et al., 2016, Zannone et al., 2019], an 8-class, 20-feature task, are included as standard benchmarks from the DFS literature. We additionally introduce two diagnostic benchmarks where subset-adaptive specialisation is necessary to obtain good performance: **Synergistic Pairs**, where individual features carry zero marginal information, and **Proxy Substitution**, where the Bayes-optimal predictor changes with the observed subset (see Appendix E.1 for details about these two new datasets). *Tabular:* nine datasets spanning medical, financial, genetic, and scientific domains: **Diabetes** [Strack et al., 2014], **Heart Disease** [Detrano et al., 1989], **Cirrhosis** [Fleming and Harrington, 2013], **Wine** [Aeberhard and Forina, 1992], **Yeast** [Nakai, 1991], **Bank Marketing** [Moro et al., 2014], **California housing** [Pace and Barry, 1997], **MiniBooNE** [Roe et al., 2005], **METABRIC** [Curtis et al., 2012]. Ranging from a few hundred to over 10^5 samples and 8 to 47 features. *Image:* **MNIST** [LeCun et al., 1998], **Fashion MNIST** [Xiao et al., 2017], **SVHN** [Netzer et al., 2011], **Imagenette** [Howard, 2019], a 10-class ImageNet subset, all evaluated under a patch-based sequential acquisition protocol, and **PCam** [Veeling et al., 2018], a histopathology benchmark in which metastatic tissue must be localised within a lymph node patch.

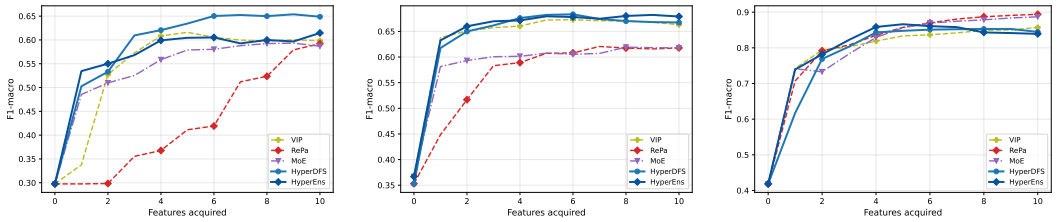
HYPER-DFS approaches. We evaluate a single-model and an ensembling variant. HYPER-DFS performs a single forward pass through the hypernetwork, producing one primary network per subset. **HyperEns** is a conventional ensemble of $K{=}5$ independently trained hypernetworks sharing the same selector.

Baselines. We compare against eight state-of-the-art DFS methods representing the main paradigms in the literature. **DIME** [Gadgil et al., 2024] and **VIP** [Chattopadhyay et al., 2023] are greedy acquisition methods: DIME maximises a learned estimate of CMI, whilst VIP selects features via a variational approximation to information gain. **CWCF** [Janisch et al., 2019] and **INVASE** [Yoon et al., 2018] are reinforcement learning approaches: CWCF trains a Q-learning agent for cost-sensitive sequential acquisition, whilst INVASE uses an actor-critic network for instance-wise variable selection. **SEFA** [Norcliffe et al., 2025] uses stochastic latent encodings to account for unobserved feature values during acquisition. **RePa** [Fumanal-Idocin et al., 2025] learns weight transforms to adjust the classifier to different feature subsets. **EDDI** [Ma et al., 2019] uses a variational autoencoder to impute missing values. For multi-model approaches, we include **AACO** [Valancius et al., 2024], which is a non-greedy method that maintains a dictionary of subset-specific classifiers and uses nearest-neighbour conditionals to guide sequential acquisition. We also construct three multi-model baselines based on VIP acquisition policy, using the same neural architecture as in the other baselines for the predictors: **Ensemble** trains an ensemble of $K = 5$ independent predictors and aggregates predictions by averaging; **Card.** partitions the parameter budget into 2, each trained on subsets within a specific cardinality range and used for routing at inference time. The cardinality range boundaries are chosen to distribute the cardinality range of the feature subsets evenly; **MoE** uses a learned routing function that takes as input the knowledge status and weights the decision of $K = 5$ models.

Evaluation Protocol. Experiments use a uniform feature cost setting, reporting average F1 score across all budget levels from 2 to 10 acquired features. All methods are evaluated using stratified 5-fold cross-validation and we report the mean and standard deviation across folds. To test feature subset generalisation, we evaluate the DFS methods on random feature subsets withheld entirely from training. Ablations and further results are detailed in Appendix D. Full implementation details, hyperparameter search ranges, and reproducibility information are provided in Appendix E.

Table 1: AUAC-F1 on synthetic benchmarks (mean over folds, %).

	Single-model							Multi-model				Hypernetworks	
Dataset	CWCF	DIME	EDDI	INVASE	RePa	SEFA	VIP	AACO	Card.	Ensemble	MoE	HyperEns	Hyper-DFS
Cube	29.81 ±2.88	31.97 ±0.79	18.20 ±2.04	22.99 ±1.79	**46.19** ±0.51	43.05 ±0.82	41.20 ±1.14	38.70 ±0.94	35.00 ±0.89	**40.40** ±1.76	39.15 ±1.93	**47.47** ±1.18	41.42 ±0.85
Sim1	69.60 ±3.02	67.47 ±1.07	65.15 ±3.02	67.76 ±1.49	75.97 ±0.71	74.99 ±0.96	**75.99** ±0.56	**74.80** ±0.83	57.77 ±0.83	72.05 ±3.66	72.70 ±2.57	**78.19** ±0.71	75.87 ±0.65
Sim2	56.84 ±4.07	55.67 ±2.72	48.68 ±2.17	53.76 ±2.43	67.71 ±0.17	65.99 ±0.88	**68.10** ±0.67	**64.80** ±0.78	53.81 ±4.17	62.20 ±3.59	60.54 ±1.77	**67.14** ±0.70	66.53 ±0.66
Sim3	57.43 ±3.45	56.91 ±3.08	38.97 ±0.01	51.67 ±1.75	**68.68** ±0.37	63.68 ±0.77	62.95 ±0.45	**63.82** ±0.66	50.27 ±4.31	55.58 ±2.84	60.01 ±3.69	**67.22** ±0.80	66.51 ±0.87
ProxySub	82.28 ±7.43	92.78 ±3.39	86.72 ±0.19	92.31 ±0.87	**96.77** ±0.28	96.28 ±0.25	96.61 ±0.36	**96.62** ±0.36	80.80 ±5.98	87.53 ±5.22	84.81 ±8.54	**96.86** ±0.26	96.73 ±0.25
SynPairs	52.43 ±2.68	53.28 ±2.26	48.48 ±2.06	49.69 ±1.61	51.53 ±4.83	63.29 ±0.70	**65.26** ±0.70	**61.76** ±1.92	51.80 ±2.60	60.50 ±3.61	60.66 ±3.05	**71.33** ±0.62	67.08 ±1.07
Avg Rank	11.67	11.00	14.33	12.83	**4.00**	6.00	4.67	**7.00**	13.17	9.67	9.50	**1.67**	4.17
Avg AUAC	58.07	59.68	51.03	56.36	67.81	67.88	**68.35**	**66.75**	54.91	63.04	62.98	**71.37**	69.02



(a) Diabetes (b) Metabric (c) Miniboone

Figure 1: F1-macro according to the number of acquired features for the top-performing methods on three representative tabular datasets. Full acquisition curves for all datasets are provided in Appendix D.2.

5.2 Empirical Results and Benchmarking

Tabular datasets. Results on synthetic benchmarks appear in Table 1 and on real-world tabular datasets in Table 2. HyperEns achieves the best average rank and the highest mean AUAC-F1 on both benchmarking suites, confirming that per-subset specialisation via hypernetworks combined with ensembling results in consistent gains across diverse task scenarios. Moreover, even without ensembling HYPER-DFS performs on average better than the strongest single-model baselines (RePa, VIP) and reaches the second-best average AUAC-F1. Among single-model methods, RePa and VIP are positioned as the strongest competitors across both suites, with VIP achieving the highest average AUAC-F1 and RePa the best ranking within that group. In the multi-model approaches, Ensemble and MoE perform comparably to VIP, and substantially outperform CardinalDFS, which confirms that static cardinality-based routing is not a good strategy overall.

Figure 1 presents acquisition curves for three datasets chosen to illustrate the range of observed behaviours. On Diabetes, HYPER-DFS achieves the largest gains, outperforming all other methods. On Metabric, both HYPER-DFS versions maintain a persistent advantage over all competitors, with the gap being largest at low budgets where per-subset specialisation is most valuable. On Miniboone, the hypernetwork methods dominate at low budgets but are overtaken by RePa and MoE beyond eight features. In this case, subset specialisation through the hypernetwork performs better with fewer features, suggesting that the weight sharing mechanism of hypernetworks can be more effective at lower cardinalities. In Appendix D, Table 8, we confirm this.

Table 3: AUAC-F1 on held-out feature subsets.

Dataset	VIP	MoE	Hyper-DFS
Avg Rank (Tab)	2.67	1.78	**1.56**
Avg AUAC (Tab)	46.55	63.11	**64.06**
Avg Rank (Img)	2.00	3.00	**1.00**
Avg AUAC (Img)	85.02	83.94	**87.57**

Image datasets. Table 4 reports AUAC-F1 on image patch benchmarks. For this comparison, we drop CWCF, EDDI and INVASE because of computational reasons. The HYPER-DFS variants achieved the highest score on four of five datasets, and the gap on SVHN to the best method (VIP) is of only 0.72

Table 2: AUAC-F1 on tabular benchmarks (mean ± std over 5-fold CV, %).

	Single-model							Multi-model				Hypernetworks	
Dataset	CWCF	DIME	EDDI	INVASE	RePa	SEFA	VIP	AACO	Card.	Ensemble	MoE	HyperEns	Hyper-DFS
Bank	54.31 ±5.07	63.20 ±5.57	63.20 ±1.28	70.12 ±1.13	**80.37** ±0.49	79.73 ±0.49	78.39 ±0.65	79.19 ±0.47	50.18 ±7.20	80.15 ±0.59	**80.29** ±0.37	**80.26** ±0.46	79.73 ±0.31
California	79.25 ±1.39	70.68 ±4.31	62.88 ±1.50	74.31 ±1.45	81.74 ±1.02	**82.25** ±0.25	79.86 ±0.61	**83.98** ±0.29	69.37 ±6.98	83.60 ±0.34	82.46 ±0.50	**82.92** ±0.46	81.93 ±0.76
Cirrhosis	44.72 ±3.04	46.41 ±4.66	39.11 ±3.01	43.25 ±2.50	49.92 ±2.33	**50.67** ±2.89	49.64 ±2.10	47.10 ±3.43	45.63 ±1.60	49.50 ±2.22	**50.12** ±2.46	**49.64** ±3.13	48.66 ±3.27
Diabetes	37.20 ±6.54	34.21 ±3.11	29.75 ±0.01	30.57 ±1.42	45.10 ±7.85	56.18 ±2.54	**59.18** ±1.80	56.13 ±2.05	50.72 ±8.44	53.52 ±4.33	**56.81** ±2.90	59.24 ±4.65	**62.81** ±1.99
Heart	72.19 ±7.97	73.45 ±6.10	68.00 ±5.99	71.98 ±5.03	81.55 ±3.93	76.12 ±4.17	**82.13** ±4.45	79.26 ±5.41	77.71 ±4.54	82.05 ±4.77	**82.81** ±4.35	**82.89** ±4.63	80.80 ±6.07
Metabric	36.68 ±0.04	46.82 ±6.47	45.11 ±6.23	45.22 ±2.52	59.73 ±4.98	63.20 ±1.09	**66.51** ±2.03	61.19 ±1.60	52.58 ±11.39	**62.74** ±1.58	60.78 ±2.89	**67.49** ±1.70	67.03 ±0.81
Miniboone	48.69 ±6.29	50.91 ±6.26	51.26 ±6.03	46.95 ±3.07	**85.72** ±0.62	71.57 ±4.60	83.15 ±0.12	85.05 ±0.22	82.09 ±0.94	**86.25** ±0.41	84.38 ±0.30	**84.13** ±1.10	83.52 ±0.53
Wine	86.29 ±5.76	86.14 ±4.37	75.98 ±3.11	80.88 ±9.11	94.95 ±2.77	**95.81** ±3.35	92.32 ±2.28	**95.45** ±2.21	93.29 ±1.73	94.67 ±1.14	95.28 ±2.20	**96.00** ±2.47	93.31 ±4.20
Yeast	34.86 ±5.69	35.17 ±3.96	20.42 ±4.62	26.99 ±3.18	**45.35** ±3.92	42.84 ±3.24	39.15 ±1.86	40.31 ±3.57	34.08 ±3.72	43.64 ±1.87	**45.86** ±1.46	41.95 ±1.91	**42.20** ±2.53
Avg Rank	17.33	16.78	19.00	18.00	**6.89**	7.56	9.78	9.33	15.33	6.00	**4.89**	**4.33**	7.11
Avg AUAC	54.91	56.33	50.63	54.47	69.38	68.71	**70.03**	69.74	61.74	70.68	**70.98**	**71.61**	71.11

Table 4: AUAC-F1 on image patch benchmarks (mean ± std over CV folds, %).

	Single-model				Multi-model				Hypernetworks	
Dataset	DIME	RePa	SEFA	VIP	AACO	Card.	Ensemble	MoE	HyperEns	Hyper-DFS
FashionMNIST	73.96 ±1.35	72.60 ±1.74	69.09 ±2.52	**80.80** ±0.50	**80.36** ±0.50	72.81 ±1.77	73.09 ±1.75	72.57 ±0.53	80.25 ±0.58	**81.19** ±0.25
Imagenette	94.05 ±0.61	93.90 ±0.30	94.14 ±0.47	**94.14** ±0.40	94.42 ±0.38	94.02 ±0.63	94.57 ±0.47	**94.57** ±0.35	**94.98** ±0.37	94.89 ±0.32
MNIST	84.37 ±1.34	83.77 ±1.30	77.10 ±2.48	**90.41** ±0.28	**90.04** ±0.23	83.66 ±1.62	84.52 ±2.24	81.56 ±2.92	90.44 ±0.23	**90.54** ±0.64
PCam	70.08 ±1.12	69.91 ±1.77	**71.13** ±0.77	68.58 ±3.03	68.98 ±0.50	**72.35** ±0.43	71.30 ±0.59	70.98 ±0.75	72.88 ±0.25	**73.39** ±0.71
SVHN	64.58 ±3.19	63.57 ±0.40	55.39 ±1.30	**69.99** ±0.98	**64.31** ±0.13	63.81 ±1.88	63.36 ±3.45	62.13 ±1.94	69.05 ±0.37	**69.27** ±1.49

points. On PCam, HYPER-DFS outperforms all baselines, including dedicated multi-model approaches by over one point. Notably, conventional multi-model methods do not consistently improve over strong single-model baselines on these datasets, and HyperEns likewise does not consistently outperform single-model HYPER-DFS. Among multi-model baselines, AACO is competitive on FashionMNIST and MNIST but does not generalise this advantage to the remaining datasets. Overall, these results indicate that the benefits of per-subset specialisation extend beyond tabular data to spatially structured inputs like images, and highlight the limitations of standard multi-model approaches.

Zero-shot subset generalisation. Table 3 reports zero-shot AUAC-F1 on held-out feature subsets across tabular and image benchmarks, comparing the strongest single-model baseline (VIP) and multi-model baseline (MoE) against HYPER-DFS. The results reveal a gap between single-model and multi-model approaches: VIP degrades substantially in the zero-shot setting on tabular data, while both MoE and HYPER-DFS surpass it by more than 15 points on average. Figures 2(a) and 2(b) illustrate two examples in tabular data in which VIP lags behind across all subset cardinalities, whereas MoE and HYPER-DFS remain competitive. On image datasets the relative ordering shifts: VIP recovers and outperforms MoE, yet HYPER-DFS consistently achieves the best performance. Figure 2(c) illustrates this pattern on Imagenette, where HYPER-DFS maintains a consistent advantage over both baselines across all subset cardinalities.

Conditioning representation. We compare the effects of using different task embedding vectors. For every pair of feature subsets (S, S'), we measure three distances: the cosine distance between their conditioning vectors, $d_z(S, S') = 1 - \cos(z_S, z_{S'})$; between the corresponding generated primary-network weights, $d_\theta(S, S') = 1 - \cos(\theta_S, \theta_{S'})$; and the raw Hamming distance $d_H(S, S')$. We additionally measure a functional distance $d_f(S, S')$, defined as the mean Jensen–Shannon divergence between the predictive distributions of the two generated networks over a common set of reference inputs. Table 6 shows that indeed ISAB embeddings show a stronger correlation between

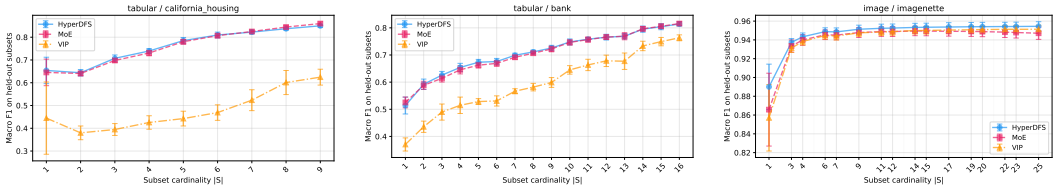


Figure 2: F1 score for unseen feature subsets in training, ordered according to the set cardinality. The three methods are shown: HYPER-DFS, MoE and VIP.

Table 5: Performance for tabular data using different encodings of the knowledge status for the hypernetwork input.

Variant	Bank	California	Cirrhosis	Diabetes	Heart	Metabric	Miniboone	Wine	Yeast	Overall	Rank
DeepSets	80.29 ±0.80	83.03 ±0.86	**49.68** ±3.27	57.68 ±2.56	80.95 ±5.40	65.79 ±2.25	84.22 ±1.50	93.21 ±2.86	41.38 ±2.98	70.69 ±17.02	2.44
RawMask	79.98 ±0.52	**83.15** ±1.13	49.05 ±2.85	58.44 ±2.93	80.80 ±3.80	66.40 ±1.46	85.11 ±0.70	94.54 ±3.61	41.89 ±2.63	71.04 ±17.13	2.22
Proposed	**80.31** ±0.45	82.65 ±1.13	49.57 ±3.03	**60.46** ±1.42	**81.39** ±3.68	**66.99** ±1.24	**85.74** ±0.32	**94.76** ±3.01	**42.20** ±4.34	**71.56** ±16.93	**1.33**

embedding and task distances using both Hamming and cosine distance. Figure 3 also shows the distances between models align much better with distances in the task encoding space when using the Set Transformer's encoder than the raw masks.

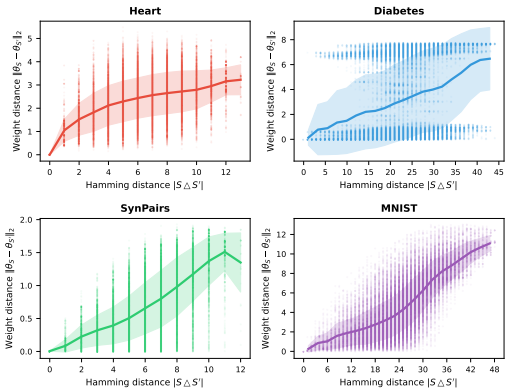
6 Related Work

The idea of using multiple classifiers in a DFS setting has been partially explored in AACO [Valancius et al., 2024], which constructs a dictionary of precomputed subset-specific classifiers for lower subset cardinalities. However, this method still uses hundreds of classifiers plus the shared amortised predictor for a higher number of features. An approach to subset adaptation was proposed in [Fumanal-Idocin et al., 2025], where a small reparametrisation of the predictor's final layers adjusts the model to each observed subset, following a similar principle to the hypernetwork parametrisation of [Ortiz et al., 2024]. However, although it is computationally efficient, the expressiveness of such reparametrisation is inherently constrained: the adapted model remains anchored to the base parameters and cannot represent classifiers that require substantially different decision boundaries for different subsets.

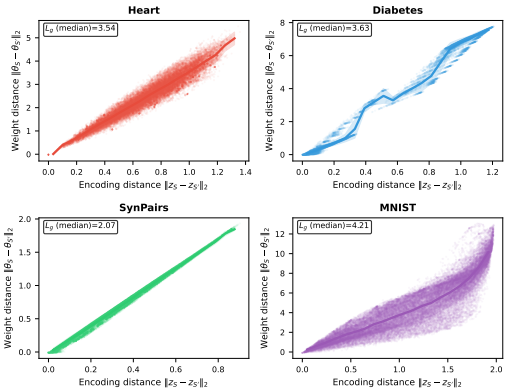
Pattern-by-pattern approaches, where a model is trained for each missingness pattern, i.e., optimising Eq. (3), show significantly better results than amortised solutions in some cases [Sell et al., 2024]. In Lobo et al. [2025], the authors also show that optimising Eq. (1) might never reach the optimal solution for logistic regression models. Less computationally intensive pattern-by-pattern approaches, such as threshold non-relevant feature interactions [Sell et al., 2024] or parameter-sharing schemes [Stempfle et al., 2023], have shown improved results in reducing the bias of amortised solutions while controlling the variance associated with pattern-by-pattern methods. However, these approaches are still limited in complex non-linear problems with a potentially exponential number of marginal distributions. Compared to our proposal, the hypernetwork captures shared structure across subsets, while still allowing enough flexibility to adapt to individual missingness patterns, providing a middle ground between amortised and expensive pattern-by-pattern approaches.

Table 6: Spearman correlations between distances in the conditioning space, predictor weights, and functional outputs.

Dataset (M)	$\rho(d_z, d_\theta)$	$\rho(d_z, d_f)$	$\rho(d_H, d_\theta)$	$\rho(d_H, d_f)$
Heart (13)	0.975	0.922	0.694	0.648
Synthetic pairs (12)	0.982	0.976	0.642	0.634
Diabetes (45)	0.983	0.984	0.539	0.537
MNIST (49)	0.996	0.997	0.572	0.552



(a) Hamming distance in the original space versus primary network weight distance.



(b) Hamming distance in the Set Transformer-based encoding space versus primary network weight distance.

Figure 3: Effects of Set Transformers encoding in the weight representation space.

In the broader hypernetwork literature, there have been proposals to use the conditioning vector as a task [Zhao et al., 2020] as a class specification [Przewięźlikowski et al., 2024, Chauhan et al., 2024] and even a support set of a dataset can be enough to generate weights for a model [Bonet et al., 2024]. Hypernetworks have also been used for context-conditional feature selection [Sristi et al., 2024]. These approaches have the advantage that the cost of generating many solutions is amortised through the hypernetwork [Beck et al., 2023b]. This amortisation property is particularly relevant for DFS: even with unlimited memory to store all 2^M possible models, the vast majority of subsets would receive insufficient gradient signal during training to learn effective parameters. However, prior work in multi-task learning has also shown that multiple tasks might have cancelling gradients or training instabilities [Shi et al., 2022], which existing literature on multi-task hypernetworks has addressed through pretrained components or by restricting the hypernetwork to low-rank adaptation [Zhao et al., 2020, Przewięźlikowski et al., 2024]. In HYPER-DFS, we mitigate gradient interference instead through a controlled mask distribution per mini-batch combined with a learning-rate warmup schedule, which avoids both component-wise training stages and capacity restrictions on the hypernetwork.

7 Conclusions and Limitations

In this paper we argued that per-subset specialisation is a structural limitation of existing dynamic feature selection, as no shared predictor can simultaneously be optimal across an exponential family of feature subsets. We also showed that existing mask-concatenation approaches are instances of embedding-based architectures and inherit the modularity penalty of Galanti and Wolf [2020]. We then proposed HYPER-DFS, a framework in which a Set Transformer encodes feature subsets into a continuous representation and a hypernetwork generates a dedicated primary network on demand. The continuous encoding produces a smooth conditioning space in which functionally similar observation masks result in functionally similar primary networks, something that the pure binary-mask formulation cannot do. We empirically support our results across synthetic, tabular, and image benchmarks, where HYPER-DFS competitive or better results than state-of-the-art DFS baselines, and multi-model baselines like ensembles and MoE.

Our contribution has limitations as well. The same smoothness that enables generalisation also limits diversity: hyperensembling over perturbed subset encodings produced no measurable gain. Training a hypernetwork is more expensive than training a single primary network of comparable capacity: across the tabular datasets, the complete Hyper-DFS model contains 0.68M–6.70M parameters, although the generated primary predictor itself contains only 4.9k–11.3k. A tighter theoretical account of how the per-subset gap distributes across subsets would help understand the performance ceiling of each method. The scope of our experimentation did not allow us to find statistical differences between methods, which a larger number of datasets would have provided.

8 Acknowledgement

This research and Javier Fumanal-Idocin were supported by EU Horizon Europe under the Marie Skłodowska-Curie COFUND grant No 101081327 YUFE4Postdocs. Raquel Fernandez-Peralta is funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V04- 00557. Dr Javier Andreu-Perez is supported in part by UKRI BBSRC project EyeWarn (ref: APP37953).

The authors acknowledge the use of the High Performance Computing Facility (Ceres) and its associated support services at the University of Essex in the completion of this work.

References

Stefan Aeberhard and M. Forina. Wine. UCI Machine Learning Repository, 1992. DOI: https://doi.org/10.24432/C5PC7J.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jacob Beck, Matthew Thomas Jackson, Risto Vuorio, and Shimon Whiteson. Hypernetworks in Meta-Reinforcement Learning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205, pages 1478–1487. PMLR, 2023a.

Jacob Beck, Risto Vuorio, Zheng Xiong, and Shimon Whiteson. Recurrent Hypernetworks are Surprisingly Strong in Meta-RL. In *Advances in Neural Information Processing Systems*, volume 36, pages 62121–62138, 2023b.

Chris M Bishop. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation*, 7(1):108–116, 1995.

David Bonet, Daniel Mas Montserrat, Xavier Giró-i Nieto, and Alexander G Ioannidis. Hyperfast: Instant classification for tabular data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11114–11123, 2024.

Oscar Chang, Lampros Flokas, and Hod Lipson. Principled Weight Initialization for Hypernetworks. In *International Conference on Learning Representations*, 2020.

Aditya Chattopadhyay, Kwan Ho Ryan Chan, Benjamin David Haeffele, Donald Geman, and Rene Vidal. Variational Information Pursuit for Interpretable Predictions. In *International Conference on Learning Representations*, 2023.

Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. A brief review of hypernetworks in deep learning. *Artificial Intelligence Review*, 57(9):235, 2024.

Xinlei Chen and Kaiming He. Exploring Simple Siamese Representation Learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

Ian Connick Covert, Wei Qiu, Mingyu Lu, Na Yoon Kim, Nathan J White, and Su-In Lee. Learning to maximize mutual information for dynamic feature selection. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 6424–6447. PMLR, 2023.

Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403): 346–352, 2012.

Robert C Detrano, András Jánosi, Walter Steinbrunn, Matthias Emil Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5):304–310, 1989.

Gabriel Erion, Joseph D Janizek, Carly Hudelson, Richard B Utarnachitt, Andrew M McCoy, Michael R Sayre, Nathan J White, and Su-In Lee. A cost-aware framework for the development of AI models for healthcare applications. *Nature Biomedical Engineering*, 6(12):1384–1398, 2022.

Thomas R Fleming and David P Harrington. *Counting processes and survival analysis*. John Wiley & Sons, 2013.

Javier Fumanal-Idocin, Raquel Fernandez-Peralta, and Javier Andreu-Perez. Model-agnostic dynamic feature selection with uncertainty quantification. *arXiv preprint arXiv:2502.01375*, 2025.

Soham Gadgil, Ian Covert, and Su-In Lee. Estimating Conditional Mutual Information for Dynamic Feature Selection. In *International Conference on Learning Representations*, 2024.

Tomer Galanti and Lior Wolf. On the Modularity of Hypernetworks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 249–256. PMLR, 2010.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, Large Minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

David Ha, Andrew Dai, and Quoc V Le. HyperNetworks. In *International Conference on Learning Representations*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Jeremy Howard. Imagenette. `https://github.com/fastai/imagenette`, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. Classification with costly features using deep reinforcement learning. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 33, pages 3959–3966, 2019.

Sergey Karayev, Mario J Fritz, and Trevor Darrell. Dynamic Feature Selection for Classification on a Budget. In *International Conference on Machine Learning Workshop on Prediction with Sequential Models*, 2013.

Varun Raj Kompella, Matthew Luciw, Marijn Frederik Stollenga, and Juergen Schmidhuber. Optimal Curiosity-Driven Modular Incremental Slow Feature Analysis. *Neural computation*, 28(8):1599–1662, 2016.

David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian Hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

Marine Le Morvan and Gaël Varoquaux. Imputation for prediction: beware of diminishing returns. In *International Conference on Learning Representations*, 2025.

Marine Le Morvan, Julie Josse, Thomas Moreau, Erwan Scornet, and Gaël Varoquaux. NeuMiss networks: differentiable programming for supervised learning with missing values. In *Advances in Neural Information Processing Systems*, volume 33, pages 5980–5990, 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A framework for Attention-based Permutation-Invariant Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3744–3753. PMLR, 2019.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse Gradient Descent for Multi-task Learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Wei Liu, Zhiying Deng, Zhongyu Niu, Jun Wang, Haozhao Wang, and Ruixuan Li. Exploring Practical Gaps in Using Cross Entropy to Implement Maximum Mutual Information Criterion for Rationalization. *Transactions of the Association for Computational Linguistics*, 13:577–594, 2025.

Angel D. Reyero Lobo, Alexis Ayme, Claire Boyer, and Erwan Scornet. A primer on linear classification with missing data. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258, pages 1225–1233. PMLR, 2025.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, José Miguel Hernández-Lobato, Sebastian Nowozin, and Cheng Zhang. EDDI: Efficient Dynamic Discovery of High-Value Information with Partial VAE. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 4234–4243. PMLR, 2019.

Chris Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*, 2017.

Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Christophe Muller, Erwan Scornet, and Julie Josse. When Pattern-by-Pattern Works: Theoretical and Empirical Insights for Logistic Models with Missing Values. *arXiv preprint arXiv:2507.13024*, 2025.

Kenta Nakai. Yeast. UCI Machine Learning Repository, 1991. DOI: https://doi.org/10.24432/C5KG68.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Alexander Luke Ian Norcliffe, Changhee Lee, Fergus Imrie, Mihaela Van Der Schaar, and Pietro Lio. Stochastic Encodings for Active Feature Acquisition. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 46784–46814. PMLR, 2025.

Jose Javier Gonzalez Ortiz, John Guttag, and Adrian Dalca. Magnitude Invariant Parametrizations Improve Hypernetwork Learning. In *International Conference on Learning Representations*, 2024.

R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33 (3):291–297, 1997.

Andrei Paleyes, Raoul-Gabriel Urma, and Neil D Lawrence. Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Computing Surveys*, 55(6):1–29, 2022.

Marcin Przewięźlikowski, Przemysław Przybysz, Jacek Tabor, Maciej Zięba, and Przemysław Spurek. HyperMAML: Few-shot adaptation of deep models with hypernetworks. *Neurocomputing*, 598: 128179, 2024.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *Proceedings of the 28th International Conference on Machine Learning*, pages 833–840, 2011.

Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted Decision Trees as an Alternative to Artificial Neural Networks for Particle Identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2-3):577–584, 2005.

Jürgen Schmidhuber. Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks. *Neural Computation*, 4(1):131–139, 1992.

Torben Sell, Thomas B Berrett, and Timothy I Cannings. Nonparametric classification with missing data. *The Annals of Statistics*, 52(3):1178–1200, 2024.

Yuge Shi, Jeffrey Seely, Philip Torr, Awni Hannun, Nicolas Usunier, Gabriel Synnaeve, et al. Gradient Matching for Domain Generalization. In *International Conference on Learning Representations*, 2022.

Ram Dyuthi Sristi, Ofir Lindenbaum, Shira Lifshitz, Maria Lavzin, Jackie Schiller, Gal Mishne, and Hadas Benisty. Contextual Feature Selection with Conditional Stochastic Gates. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

Lena Stempfle, Ashkan Panahi, and Fredrik D Johansson. Sharing Pattern Submodels for Prediction with Missing Values. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8): 9882–9890, 2023.

Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed Research International*, 2014(1):781670, 2014.

Michael Valancius, Maxwell Lennon, and Junier Oliva. Acquisition Conditioned Oracle for Nongreedy Active Feature Acquisition. In *Proceedings of the 41st International Conference on Machine Learning*, volume 267, pages 48957–48975. PMLR, 2024.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation Equivariant CNNs for Digital Pathology. In *Medical Image Computing and Computer-Assisted Intervention*, pages 210–218. Springer, 2018.

Johannes von Oswald, Christian Henning, Benjamin F Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jinsung Yoon, James Jordon, and Mihaela Van der Schaar. INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2018.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Sara Zannone, José Miguel Hernández-Lobato, Cheng Zhang, and Konstantina Palla. ODIN: Optimal Discovery of High-value INformation Using Model-based Deep Reinforcement Learning. In *International Conference on Machine Learning Real-world Sequential Decision Making Workshop*, 2019.

Dominic Zhao, Seijin Kobayashi, João Sacramento, and Johannes von Oswald. Meta-Learning via Hypernetworks. In *Neural Information Processing Systems Workshop on Meta-Learning*, 2020.

Appendix Overview

The appendix is organised into two parts:

Further Analysis of the primary networks weights in HYPER-DFS. These appendices cover the variance and gradient analysis of the primary network.

- Appendix A analyses the weight variance in the hypernetwork, and why using both absence and presence embeddings mitigates variance issues.
- Appendix B discusses loss enhancement to prevent representational collapse in the hypernetwork.
- Appendix C studies the problem of gradient dilution and its solutions.

Additional Empirical Results and Reproducibility. Includes additional experimental details.

- Appendix D provides the full results for the experiments discussed in the main body of the paper.
- Appendix E provides full implementation details, including dataset preprocessing, model architectures, hyperparameter ranges, and training configurations.

A Controlling Weight Variance in HYPER-DFS

In this appendix, we study how to control the weight variance through the hypernetwork training. For that, we propose two things: to control the norm of the hypernetwork input, and to use a scale regularisation loss. Both decisions are described in detail in the following.

A.1 The Subset-Size Bias Under Sum Aggregation in Set Transformer Encoding

Training hypernetworks is known to be harder than training standard networks of comparable size [Chang et al., 2020], because the gradient signal reaching ϕ is mediated by the primary network:

$$\nabla_\phi \mathcal{L} \;=\; J_\phi^\top \, \nabla_\theta \mathcal{L}, \qquad J_\phi \;=\; \frac{\partial g_\phi(\tilde{\mathbf{z}}_{\,omega,\rho}(\mathbf{I}_S))}{\partial \phi} \;\in\; \mathbb{R}^{|\theta|\times|\phi|}, \tag{7}$$

where J_ϕ is the Jacobian of the weight-generation function with respect to the hypernetwork parameters. If the element-wise variance of $\theta_S = g_\phi(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S))$ deviates substantially from the Xavier target ($\mathrm{Var}(W_{ij}) = 1/d_{\text{in}}$, where d_{in} is the input dimension of the corresponding layer), the primary network's pre-activations either vanish or saturate, suppressing $\nabla_\theta \mathcal{L}$, and the gradient reaching ϕ [Glorot and Bengio, 2010]. Existing work [Chang et al., 2020] addresses this by deriving conditions on g_ϕ's weight matrices so that the marginal distribution of the generated weights matches this target. However, their analysis treats the conditioning input distribution as fixed. This is problematic in a DFS setting, where the conditioning signal has systematically different norms depending on the cardinality of the knowledge status. Two complementary mechanisms can mitigate this problem: one is to normalise $\mathbf{z}_{\omega,\rho}(\mathbf{I}_S)$ (Eq. (5)), the other is to decouple its norm from $|S|$, which we achieve using absence and presence embeddings for each feature. In the following, we formalise how this norm and cardinality relationship affects the variance of the weights of the primary network and how our solutions are mitigating this problem.

Proposition 1 (Cardinality-dependent weight scale under different encoder designs)**.** *Let* $\mathbf{u}_{\omega,\rho}\colon \{0,1\}^M \to \mathbb{R}^{d'}$ *be a presence-only token-wise sum-pooled set encoder defined by*

$$\mathbf{u}_{\omega,\rho}(\mathbf{I}_S) \;=\; f_\rho\!\left(\sum_{i=1}^{M} (\mathbf{I}_S)_i \, f_\omega(\mathbf{e}_i^1)\right) \;=\; f_\rho\!\left(\sum_{i\in S} f_\omega(\mathbf{e}_i^1)\right),$$

and let $g_\phi\colon \mathbb{R}^{d'} \to \mathbb{R}^{|\theta|}$ *be a hypernetwork. We assume* f_ω*,* f_ρ*, and* g_ϕ *are linear at initialisation, in the spirit of variance-propagation analyses [Glorot and Bengio, 2010]. Assume the feature embedding pairs* $\{(\mathbf{e}_i^1, \mathbf{e}_i^0)\}_{i=1}^M$ *are independent across* i*, with each embedding zero-mean, per-coordinate variance* σ_e^2*, and finite fourth moments. Then:*

(i) ***Presence-only encoder, no normalisation.*** *Let* $\theta_S = g_\phi(\mathbf{u}_{\omega,\rho}(\mathbf{I}_S))$*. Then*

$$\mathbb{E}\,\|\theta_S\|_2^2 \;=\; |S| \cdot \kappa, \tag{8}$$

where $\kappa > 0$ *depends on the initialisation scales of* f_ω*,* f_ρ*,* g_ϕ*, and on* σ_e^2*, but does not itself depend on* $|S|$*. The squared norm of the generated weights grows linearly with the number of observed features.*

(ii) ***Presence-only encoder,*** L_2 ***normalisation.*** *Let* $\tilde{\mathbf{u}}_{\omega,\rho}(\mathbf{I}_S) = \mathbf{u}_{\omega,\rho}(\mathbf{I}_S)/\|\mathbf{u}_{\omega,\rho}(\mathbf{I}_S)\|_2$ *and* $\theta_S = g_\phi(\tilde{\mathbf{u}}_{\omega,\rho}(\mathbf{I}_S))$*. Then*

$$\mathbb{E}\,\|\theta_S\|_2^2 \;=\; \kappa', \tag{9}$$

for a constant $\kappa' > 0$ *independent of* $|S|$*.*

(iii) ***Encoder with absence embeddings.*** *Let* $\mathbf{z}_{\omega,\rho}\colon \{0,1\}^M \to \mathbb{R}^{d'}$ *be the encoder used in* HYPER-DFS *(Eq.* (5)*),*

$$\mathbf{z}_{\omega,\rho}(\mathbf{I}_S) \;=\; f_\rho\!\left(\sum_{i=1}^{M} f_\omega(\mathbf{t}_i)\right), \qquad \mathbf{t}_i = \mathbf{I}_{S,i}\,\mathbf{e}_i^1 + \big(1 - \mathbf{I}_{S,i}\big)\,\mathbf{e}_i^0,$$

and let $\theta_S = g_\phi(\mathbf{z}_{\omega,\rho}(\mathbf{I}_S))$*. Then*

$$\mathbb{E}\,\|\theta_S\|_2^2 \;=\; M \cdot \kappa'', \tag{10}$$

for a constant $\kappa'' > 0$ *independent of* $|S|$*.*

Proof. Since f_ω, f_ρ, and g_ϕ are linear at initialisation, θ_S is a linear function of the embeddings, and each linear map preserves the squared-norm scaling up to a multiplicative constant independent of $|S|$. It therefore suffices to track $\mathbb{E}\|\mathbf{u}_{\omega,\rho}(\mathbf{I}_S)\|_2^2$ in part (i), $\mathbb{E}\|\tilde{\mathbf{u}}_{\omega,\rho}(\mathbf{I}_S)\|_2^2$ in part (ii), and $\mathbb{E}\|\mathbf{z}_{\omega,\rho}(\mathbf{I}_S)\|_2^2$ in part (iii).

Part (i). Since f_ω is linear and the $\{\mathbf{e}_i^1\}_{i\in S}$ are independent and zero-mean, so are the summands $\{f_\omega(\mathbf{e}_i^1)\}_{i\in S}$. The cross terms in the expansion of the squared norm therefore vanish, giving

$$\mathbb{E}\left\|\sum_{i\in S} f_\omega(\mathbf{e}_i^1)\right\|_2^2 \;=\; \sum_{i\in S} \mathbb{E}\|f_\omega(\mathbf{e}_i^1)\|_2^2 \;=\; |S| \cdot c_1, \tag{11}$$

with $c_1 = \mathbb{E}\|f_\omega(\mathbf{e}_1^1)\|_2^2$. Applying the linear f_ρ and g_ϕ preserves the $|S|$-scaling up to a constant, giving $\mathbb{E}\|\theta_S\|_2^2 = |S| \cdot \kappa$.

Part (ii). By construction, $\|\tilde{\mathbf{u}}_{\omega,\rho}(\mathbf{I}_S)\|_2 = 1$, so $\mathbb{E}\|\tilde{\mathbf{u}}_{\omega,\rho}(\mathbf{I}_S)\|_2^2 = 1$ independently from $|S|$. Hence $\mathbb{E}\|\theta_S\|_2^2 = \kappa'$ for some constant κ' independent of $|S|$.

Part (iii). The token sum has M summands, with $\mathbf{t}_i = \mathbf{e}_i^1$ for $i \in S$ and $\mathbf{t}_i = \mathbf{e}_i^0$ for $i \in \bar{S}$. Independence and zero-mean of the embedding pairs across i give

$$\mathbb{E}\left\|\sum_{i=1}^{M} f_\omega(\mathbf{t}_i)\right\|_2^2 \;=\; \sum_{i\in S} \mathbb{E}\|f_\omega(\mathbf{e}_i^1)\|_2^2 + \sum_{i\in \bar{S}} \mathbb{E}\|f_\omega(\mathbf{e}_i^0)\|_2^2 \;=\; |S| \cdot c_1 + (M - |S|) \cdot c_0, \tag{12}$$

with $c_0 = \mathbb{E}\|f_\omega(\mathbf{e}_1^0)\|_2^2$. Since f_ω is linear and both embedding distributions share the same per-coordinate variance σ_e^2, we have $c_0 = c_1$, and the sum reduces to $M \cdot c_1$, independent of $|S|$. Applying the linear f_ρ and g_ϕ preserves this scaling, giving $\mathbb{E}\|\theta_S\|_2^2 = M \cdot \kappa''$. □

Remark. The cardinality independence in part (iii) depends on $c_0 = c_1$, which follows from our assumption that the absence and presence embeddings share the same per-coordinate variance. This holds at initialisation by construction, but as training progresses the embedding distributions can drift apart, and the dependence on $|S|$ may reappear. Part (ii), in contrast, gives a guarantee that does not rely on this matching condition. This is why HYPER-DFS combines absence embeddings with L_2 normalisation: the absence embeddings provide a uniform M-token representation, while the L_2 normalisation ensures that the scale of θ_S stays under control even if the embedding distributions diverge.

A.2 Output-Scale Regularisation

To encourage variance uniformity throughout training, we augment the objective with a penalty on the scale of the generated weights:

$$\mathcal{L}_{\text{scale}}(\phi, \omega, \rho) \;=\; \lambda \cdot \sum_{l=1}^{L} \mathbb{E}_S \left[\left(\frac{1}{n_l} \left\| g_\phi^{(l)}\big(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S)\big) \right\|_2^2 - \sigma_{l,\star}^2 \right)^2 \right], \tag{13}$$

where n_l is the number of parameters in layer l of the primary network, $g_\phi^{(l)}(\cdot) \in \mathbb{R}^{n_l}$ is the slice of the hypernetwork output that generates them, and $\sigma_{l,\star}^2$ is the Xavier target variance for that layer. L is the number of layers in the primary network, and the expectation over S is approximated by the mini-batch of subsets sampled alongside the training data. The coefficient $\lambda > 0$ is annealed to zero after an initial warm-up phase; its primary role is to stabilise the early training trajectory. $\mathcal{L}_{\text{scale}}$ is fully differentiable and adds negligible overhead to the main loss computation.

B Stabilizing the training of HYPER-DFS while preventing representation collapse

This appendix details the two auxiliary regularisers that complement the main cross-entropy objective in our training procedure: noise injection applied to the subset encoding, and a VICReg-style variance penalty that prevents either the encoder or the hypernetwork from degenerating into a constant map.

B.1 Stochasticity in HYPER-DFS training

Stochasticity can be introduced into hypernetworks in several ways. It is common practice to either perturb the conditioning input [Krueger et al., 2017] or generate the primary-network weights directly from noise. In HYPER-DFS, we inject noise into the hypernetwork's input, in $\mathbf{z}$-space:

$$\mathbf{z}^*_{\omega,\rho}(\mathbf{I}_S) = \mathbf{z}_{\omega,\rho}(\mathbf{I}_S) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0},\, \sigma^2 \mathbf{I}_{d'}), \qquad \tilde{\mathbf{z}}^*_{\omega,\rho}(\mathbf{I}_S) = \frac{\mathbf{z}^*_{\omega,\rho}(\mathbf{I}_S)}{\|\mathbf{z}^*_{\omega,\rho}(\mathbf{I}_S)\|_2}, \tag{14}$$

and pass $\tilde{\mathbf{z}}^*_{\omega,\rho}(\mathbf{I}_S)$ to the hypernetwork in place of $\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S)$ during training. This encourages the hypernetwork to vary smoothly as a function of $\mathbf{z}_{\omega,\rho}(\mathbf{I}_S)$, aligning representation-space proximity with functional similarity. It also penalises the Jacobian norm of g_ϕ with respect to its conditioning input [Bishop, 1995, Rifai et al., 2011], which discourages large weight changes in response to small encoding perturbations.

B.2 Preventing Representation Collapse

A known failure mode of hypernetworks is *representation collapse*: the hypernetwork may learn to produce nearly identical parameters regardless of the conditioning input, rendering the conditioning redundant. In our setting, collapse can occur at two levels: the encoder may map multiple, or even all, functionally distinct subsets to the same encoding, or the hypernetwork may map distinct encodings to the same weight configuration for the primary network.

We address both modes with a low-variance penalty applied to both the intermediate representations and the generated weights:

$$\mathcal{L}_{\text{collapse}} \;=\; -\frac{1}{d'} \sum_{j=1}^{d'} \operatorname{Var}\Big[\big(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S)\big)_j \Big] \;-\; \frac{1}{|\theta|} \sum_{k=1}^{|\theta|} \operatorname{Var}\big[\big(g_\phi(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S))\big)_k \big], \tag{15}$$

where $\operatorname{Var}[\cdot]$ denotes the empirical variance computed across the mini-batch.

Note that this loss does not involve predictions, so an uninformative feature may legitimately not change the model's output, but it should still shift $\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S)$ and, in turn, $g_\phi(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S))$. Both variances are computed over the same mini-batch already used for the cross-entropy step, so $\mathcal{L}_{\text{collapse}}$ adds no additional forward passes. This results in the following training

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{scale}}(t)\mathcal{L}_{\text{scale}} + \lambda_{\text{collapse}}\mathcal{L}_{\text{collapse}}, \tag{16}$$

where $\lambda_{\text{scale}}(t)$ is the annealed warm-up coefficient (Eq. (13)) and $\lambda_{\text{collapse}} = 0.01$ is fixed throughout training.

C Gradient Dilution in HYPER-DFS and the Pre-training Stabilisation Strategy

This appendix develops the training-stability analysis summarised in Section 4.2. Conditioning a hypernetwork on different feature subsets within a single mini-batch can induce *gradient dilution*: subset-specific contributions to the backward pass interfere destructively. The effect appears at the hypernetwork parameters ϕ and is amplified at the input compressor c_η. We motivate below the two remedies used during predictor training: a per-batch mask budget and a linear learning-rate warm-up.

Gradient dilution at ϕ and η. Consider a mini-batch of size B in which K distinct feature subsets $\{S_1, \dots, S_K\}$ appear, with $\mathcal{B}_k \subseteq \{1, \dots, B\}$ collecting the samples that share subset S_k. The gradient at the hypernetwork parameters decomposes as

$$\nabla_\phi \mathcal{L} \;=\; \frac{1}{B} \sum_{k=1}^{K} J_\phi\big(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_{S_k})\big)^\top \sum_{b \in \mathcal{B}_k} \nabla_\theta \mathcal{L}^{(b)} \,\big|_{\theta=\theta_{S_k}}, \tag{17}$$

where $\theta_{S_k} = g_\phi(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_{S_k}))$ are the weights generated for subset S_k, $J_\phi(\mathbf{z}) = \partial g_\phi(\mathbf{z})/\partial\phi \in \mathbb{R}^{|\theta|\times|\phi|}$ is the parameter-Jacobian of the weight-generation function, and $\mathcal{L}^{(b)}$ is the loss on the b-th sample. Samples sharing a subset accumulate constructively through a common Jacobian. However, g_ϕ at initialisation behaves as a near-random function, so the Jacobians for distinct k act as approximately uncorrelated linear maps. Their contributions therefore rarely share a coherent direction, reducing the effective signal in $\nabla_\phi \mathcal{L}$ [Yu et al., 2020].

The compressor is mask-independent in its forward pass, but its backward signal flows through the generated, mask-dependent primary network:

$$\nabla_\eta \mathcal{L} \;=\; \frac{1}{B} \sum_{b=1}^{B} J_\eta\big(\tilde{\mathbf{x}}^{(b)}_{S^{(b)}}\big)^\top J_{f,\theta}\big(\mathbf{h}^{(b)}\big)^\top \nabla_{\hat{y}^{(b)}_\theta} \mathcal{L}^{(b)} \,\big|_{\theta=\theta^{(b)}}, \tag{18}$$

where $S^{(b)}$ is the subset assigned to the b-th sample, $\theta^{(b)} = g_\phi(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_{S^{(b)}}))$ is the per-sample generated weight tensor, $\mathbf{h}^{(b)} = c_\eta(\tilde{\mathbf{x}}^{(b)}_{S^{(b)}})$ is the compressed representation, $\hat{y}^{(b)}_\theta = f_\theta(\mathbf{h}^{(b)})$ is the prediction at weights θ, $J_\eta(\mathbf{x}) = \partial c_\eta(\mathbf{x})/\partial\eta$ is the parameter-Jacobian of the compressor, and $J_{f,\theta}(\mathbf{h}) = \partial f_\theta(\mathbf{h})/\partial\mathbf{h}$ is the input-Jacobian of the primary network at weights θ. The bar fixes both $J_{f,\theta}$ and $\hat{y}^{(b)}_\theta$ at $\theta = \theta^{(b)}$. At initialisation, distinct subsets produce approximately independent weight configurations, and the corresponding input-Jacobians act as approximately uncorrelated linear maps. This is analogous to training a shared feature extractor behind randomly initialised, rapidly changing projection heads, a scenario prone to representation collapse [Yu et al., 2020, Chen and He, 2021].

A per-batch mask budget. Restricting each mini-batch to a single subset ($K = 1$, with $S^{(b)} = S$ for all b) gives $\theta^{(b)} = \theta_S$ for all b, and Eq. (18) simplifies to

$$\nabla_\eta \mathcal{L} \;=\; \frac{1}{B} \sum_{b=1}^{B} J_\eta\big(\tilde{\mathbf{x}}^{(b)}_{S}\big)^\top J_{f,\theta}\big(\mathbf{h}^{(b)}\big)^\top \nabla_{\hat{y}^{(b)}_\theta} \mathcal{L}^{(b)} \,\big|_{\theta=\theta_S}. \tag{19}$$

Every per-sample contribution is now routed through the same primary network, recovering the standard situation of a feature extractor trained jointly with a single downstream classifier; the same simplification applies to Eq. (17), where the outer sum collapses to the single term $k = 1$ and the bar fixes $\theta = \theta_S$. A strict budget of $K = 1$ removes the within-batch subset diversity needed by the variance penalty $\mathcal{L}_{\text{collapse}}$ (Eq. (15)) and forfeits the cooperation and generalisation effects of intra-batch diversity, so we adopt a small but non-trivial K in practice.

Linear learning-rate warm-up. Even with a small K, a transient instability persists. The scale regulariser $\mathcal{L}_{\text{scale}}$ (Eq. (13)) requires several optimiser steps to bring the empirical layerwise mean squared values $\|g^{(l)}_\phi(\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S))\|_2^2/n_l$ towards their Xavier targets $\sigma^2_{l,\star}$. During this transient, a large optimiser step can push θ_S into a saturated activation regime, suppressing $J_{f,\theta_S}(\mathbf{h})$ and the compressor gradient itself. Linearly warming up the global learning rate over the first T_{warm} steps [Goyal et al., 2017] keeps these early updates small, allowing g_ϕ to reach the regime in which $\mathcal{L}_{\text{scale}}$ takes effect before the optimiser commits to a particular mapping $\tilde{\mathbf{z}}_{\omega,\rho}(\mathbf{I}_S) \mapsto \theta_S$.

Table 7: Effect of changing the hypernetwork for a standard MLP in the performance of HYPER-DFS, measured using AUAC-F1.

Variant	Bank	California	Cirrhosis	Diabetes	Heart	Metabric	MiniBooNE	Wine	Yeast	Overall
Hyper-DFS (full)	**80.35**	82.65	**49.22**	**59.34**	82.09	**66.96**	**85.54**	93.67	**41.82**	**71.29**
	±0.48	±1.13	±3.38	±1.77	±3.86	±1.23	±0.25	±2.10	±2.25	±16.97
w/o hypernetwork	78.80	**83.14**	48.77	57.46	**82.10**	62.13	83.75	**94.59**	39.28	70.00
	±0.48	±0.55	±3.12	±3.70	±4.83	±7.34	±0.45	±3.94	±1.87	±18.02

Table 8: Improvement of the learned DFS policy compared to a random policy, measured using AUAC-F1.

	Single-model				Multi-model			Hypernetworks
Dataset	CWCF	RePa	SEFA	VIP	Card.	Ensemble	MoE	Hyper-DFS
Bank	0.78	11.86	11.85	**13.31**	2.59	11.85	**11.96**	12.31
California	2.46	6.81	6.52	**7.79**	-1.15	**7.17**	5.38	4.42
Cirrhosis	1.90	4.41	**6.53**	5.02	0.22	4.53	**4.78**	2.64
Diabetes	0.71	9.69	20.50	**21.85**	12.20	17.32	**19.76**	**22.48**
Heart	1.60	5.80	5.63	**6.30**	1.38	5.61	**5.95**	4.81
Metabric	2.14	**20.23**	14.39	17.85	14.06	**22.15**	16.43	18.98
Miniboone	3.52	20.08	21.75	**24.42**	16.49	**22.66**	21.49	16.84
Wine	6.35	7.11	6.82	**7.19**	4.30	7.69	**8.24**	7.61
Yeast	-0.71	**5.64**	4.10	1.87	-3.81	4.77	**6.70**	3.42
Avg Rank	7.22	4.00	3.89	**2.33**	7.11	3.11	**2.89**	4.11
Avg Δ	2.19	10.18	10.90	**11.73**	5.14	**11.53**	10.74	10.39

D Additional Experimental Results

In this appendix we cover additional experimental results. We report the results of our ablation studies, extended reports of the results reported in the main body of the paper, and we report the running times of the different methods tested.

D.1 Ablation Studies

To isolate the contribution of each design choice in HYPER-DFS, we conduct a series of four controlled ablations, varying one component at a time while holding all others at their default configuration.

Hypernetwork ablation. Table 7 reports the performance degradation when the hypernetwork is replaced by a single MLP with mask concatenation, holding the Set Transformer encoding fixed. HYPER-DFS improves AUAC-F1 on 6 of 9 datasets and the three exceptions lie within one standard deviation. The effect is most pronounced on Metabric, where the MLP baseline is 4.8 points worse on average and more variable across folds (std 7.34 vs 1.23), indicating that per-subset specialisation stabilises training in the high-dimensional regime. The largest gains concentrate on high-M / many-class datasets (Metabric, MiniBooNE, Diabetes, Yeast), which makes the hypernetwork especially interesting for hard DFS problems.

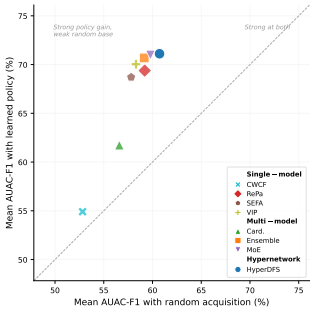


Figure 4: Mean AUAC-F1 averaged across nine tabular benchmarks under random feature acquisition (x-axis) versus learned acquisition policy (y-axis).

Acquisition policy. We ablate the DFS acquisition policy by replacing it with random sequential feature acquisition. The gain $\Delta\text{AUAC} = \text{AUAC}_{\text{policy}} - \text{AUAC}_{\text{random}}$ quantifies how much of the performance is attributable to the policy itself, as opposed to the model's general ability to handle arbitrary subsets. Results for tabular datasets are shown in Table 8. We can see that the magnitude of the improvement is, on average, around 10 points. Two methods are

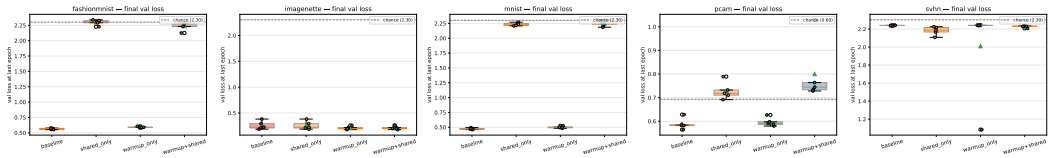


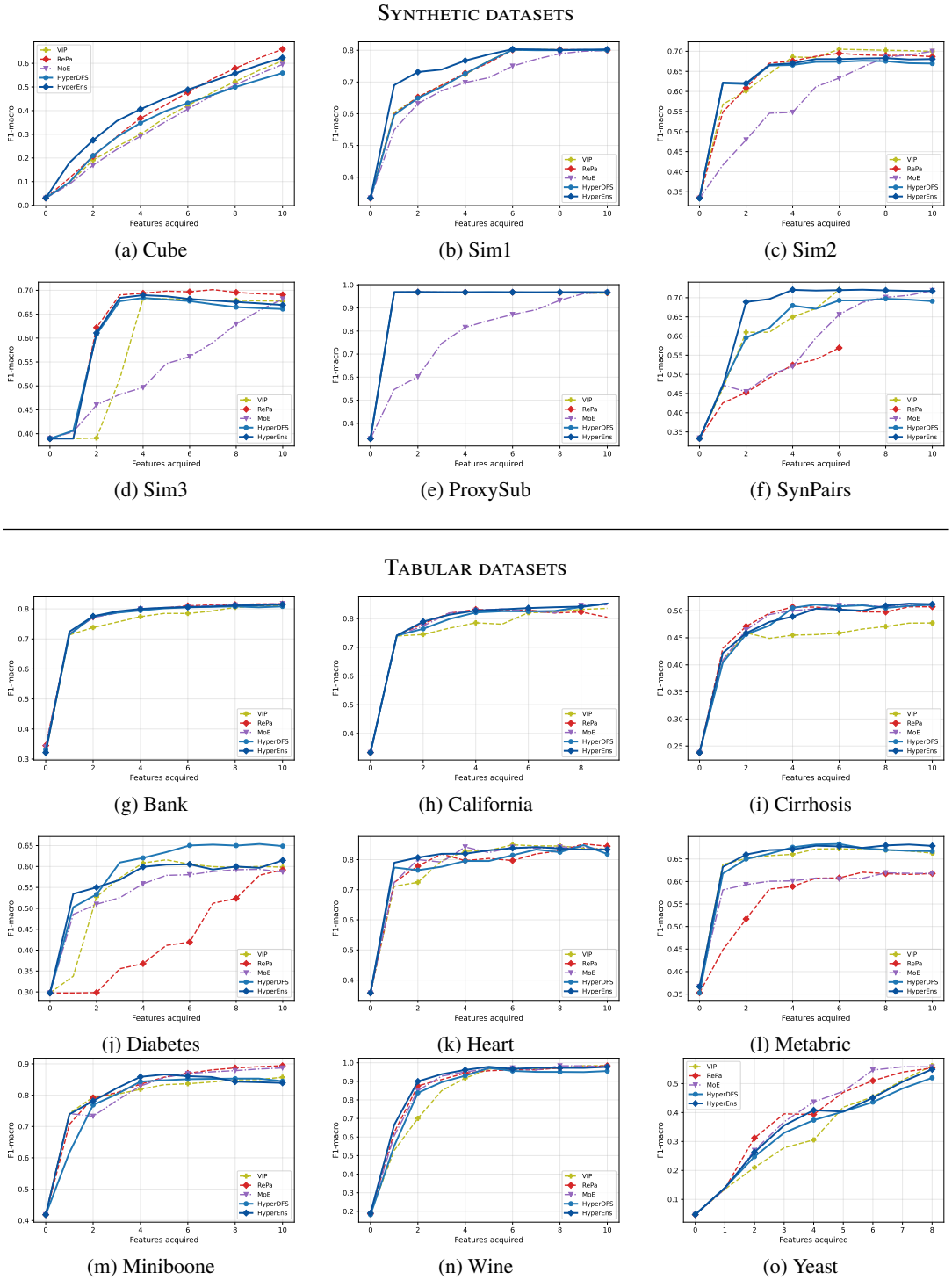
Figure 5: Effect in loss after 10 epochs of the mask restriction and LR warmup. Restricting the number of masks solved the problem of training collapse, but it also hurts performance on the first steps of the training.

Table 9: LR Warmup and shared-mask ablation: failed training rate and training improvement as percentage of the initial loss. Metrics reported after 10 epochs and averaged across 5 seeds.

Dataset	Config	Failed	Train improvement % (mean $\pm$ std)
fashionmnist	baseline	0/5	47.7 ± 0.6
	shared_only	0/5	55.4 ± 1.3
	warmup+shared	0/5	66.7 ± 1.3
	warmup_only	0/5	72.7 ± 0.2
imagenette	baseline	0/5	88.5 ± 1.5
	shared_only	0/5	88.5 ± 1.5
	warmup+shared	0/5	95.3 ± 0.3
	warmup_only	0/5	95.3 ± 0.3
mnist	baseline	0/5	58.4 ± 0.7
	shared_only	0/5	64.0 ± 1.7
	warmup+shared	0/5	74.7 ± 1.1
	warmup_only	0/5	78.5 ± 0.2
pcam	baseline	0/5	12.7 ± 2.0
	shared_only	0/5	11.4 ± 2.5
	warmup+shared	0/5	8.8 ± 0.7
	warmup_only	0/5	14.3 ± 0.9
svhn	baseline	5/5	0.2 ± 0.0
	shared_only	0/5	55.7 ± 1.5
	warmup+shared	0/5	50.7 ± 0.9
	warmup_only	4/5	10.8 ± 20.6

significantly below this value: CWCF and Cardinality-based routing,
which also underperformed several other methods. VIP policy improved performance the most over random acquisition. The relationship between both learned and random DFS policies is shown in Figure 4, where we can see that the performance of HYPER-DFS was superior using both random and learned policies, which also means that the HYPER-DFS predictor model is less dependent on the learned acquisition policy to achieve good results.

No mask restriction. The gradient issue of the compressor loss described in Section 4.2 was solved by restricting the number of different masks in the pretraining to one, and by setting a linear LR Warmup. For this ablation, we see the ratio of failed trainings for all the image datasets tested, shown in Table 9, after 10 epochs of training. We can see there that SVHN systematically fails to train for every seed tested without LR Warmup or mask restriction, while other datasets run fine. We can also see that LR Warmup solved the problem on one of five seeds, so restricting the number of masks is necessary at the beginning of training. However, we also report in Figure 5 that restricting the number of masks also hurts generalisation, suggesting that a compromise between setting the number of different masks to 1 and not controlling them might be the optimal solution to solve the training stability issues while retaining good performance in the early steps of training.



(a) Cube (b) Sim1 (c) Sim2

(d) Sim3 (e) ProxySub (f) SynPairs

(g) Bank (h) California (i) Cirrhosis

(j) Diabetes (k) Heart (l) Metabric

(m) Miniboone (n) Wine (o) Yeast

Figure 6: F1-macro acquisition curves for all synthetic (a–f) and tabular (g–l) datasets. Each curve reports the mean F1-macro across 5-fold CV as a function of the number of acquired features, for the top-performing methods.

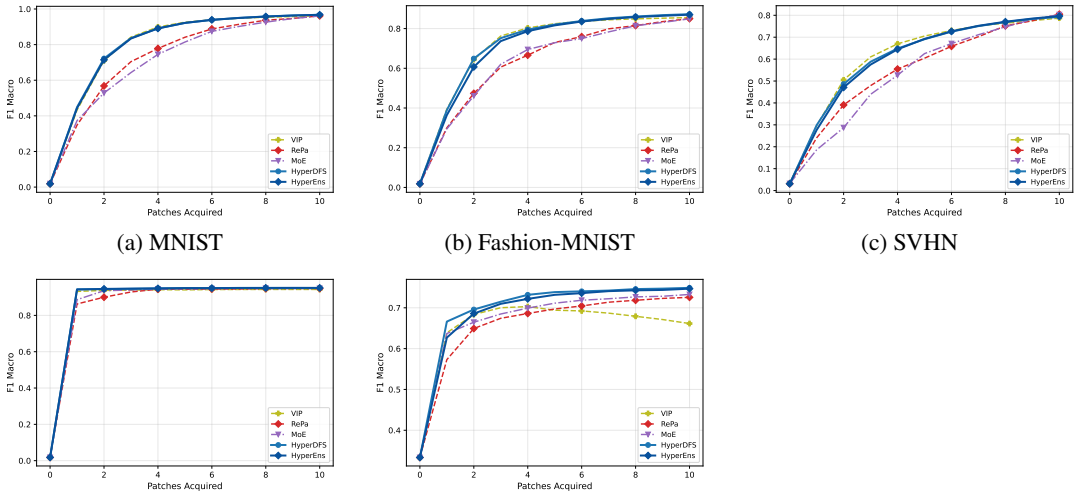
Table 10: Zero-shot AUAC-F1 on held-out feature subsets (mean ± std over CV folds, %).

Dataset	VIP	MoE	Hyper-DFS
Bank	59.54 ±0.63	70.51 ±0.58	**70.79** ±0.83
California	47.83 ±3.61	75.90 ±0.95	**76.10** ±1.18
Cirrhosis	24.51 ±0.64	**47.00** ±1.97	46.85 ±2.30
Diabetes	**49.84** ±1.08	45.67 ±1.83	49.51 ±1.48
Heart	44.94 ±3.45	**76.62** ±4.27	75.94 ±4.55
Metabric	59.17 ±0.96	57.32 ±2.15	**59.87** ±2.60
Miniboone	74.10 ±2.19	76.23 ±1.51	**81.08** ±1.42
Wine	34.46 ±5.23	**86.41** ±1.25	85.62 ±1.08
Yeast	24.57 ±1.55	**32.38** ±2.79	30.82 ±3.09
FashionMNIST	78.06 ±0.42	76.44 ±1.18	**81.22** ±0.59
Imagenette	94.28 ±0.38	94.26 ±0.57	**94.80** ±0.43
MNIST	82.71 ±0.51	81.12 ±0.98	**86.69** ±0.62
Avg Rank (Tabular)	2.67	1.78	**1.56**
Avg AUAC (Tabular)	46.55 ±15.70	63.11 ±17.18	**64.06** ±17.37
Avg Rank (Image)	2.00	3.00	**1.00**
Avg AUAC (Image)	85.02 ±6.82	83.94 ±7.54	**87.57** ±5.58

D.2 Extended Results

Figure 6 reports the full set of acquisition curves for synthetic and real-world tabular data omitted from the main text. We can see there consistent patterns across datasets. On the synthetic benchmarks, HyperEns achieves the steepest initial gains on Cube, Sim1, and SynPairs, confirming that per-subset specialisation is most valuable when few features have been acquired and the subset space is most heterogeneous. On ProxySub, all methods except MoE reach near-perfect performance with only two features, reflecting the low intrinsic difficulty of this task; the large gap for MoE suggests that learned routing struggles when the discriminative signal is concentrated in very few features. On Sim2 and Sim3, the hypernetwork variants lead at low budgets but are matched or slightly overtaken by RePa and VIP at higher budgets, consistent with the Miniboone pattern discussed in the main text: as the observed set grows, per-subset specialisation provides diminishing returns. Among the tabular datasets, Bank, California, and Wine exhibit rapid saturation where all top methods converge to similar performance beyond 4–6 features, leaving limited room for differentiation. Heart and SynPairs show the clearest advantage for HyperEns at low-to-mid budgets, while Cirrhosis and Yeast are the most challenging settings, with all methods clustered within narrow performance bands and no single approach consistently dominating. Notably, on Yeast, RePa and MoE achieve the best high-budget performance, suggesting that on tasks with many weakly informative features, the reparametrisation and routing strategies can be more effective than the hypernetwork when the full feature set is nearly observed. Across all datasets, a recurring theme is that the hypernetwork advantage is concentrated in the low-budget regime — precisely the regime of greatest practical interest in cost-sensitive acquisition.

IMAGE DATASETS

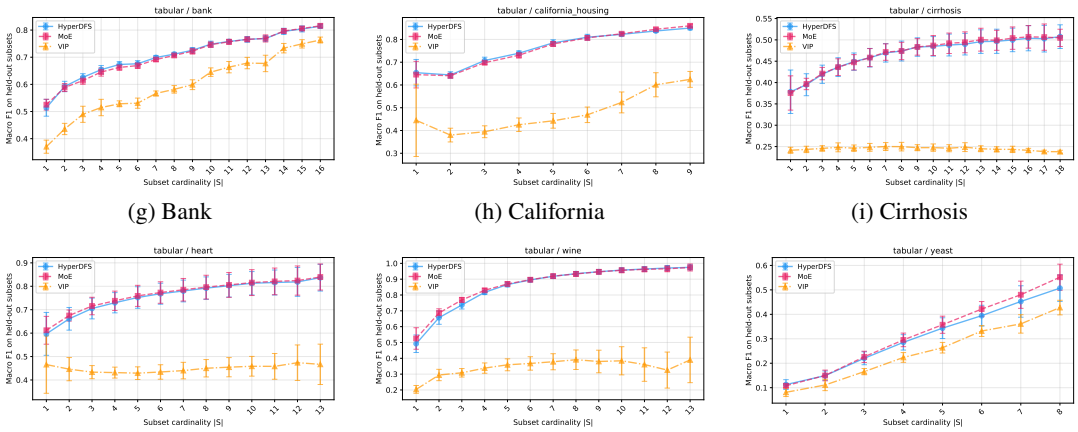


(a) MNIST (b) Fashion-MNIST (c) SVHN

(d) Imagenette (e) PCam

Figure 7: F1-macro acquisition curves for all image datasets for unseen feature subsets in training. Each curve reports the mean F1-macro across 5-fold CV as a function of the number of acquired features, for the top-performing methods.

TABULAR DATASETS



(g) Bank (h) California (i) Cirrhosis

(j) Heart (k) Wine (l) Yeast

Figure 8: F1-macro acquisition curves for all tabular datasets for unseen feature subsets in training. Each curve reports the mean F1-macro across 5-fold CV as a function of the number of acquired features, for the top-performing methods.

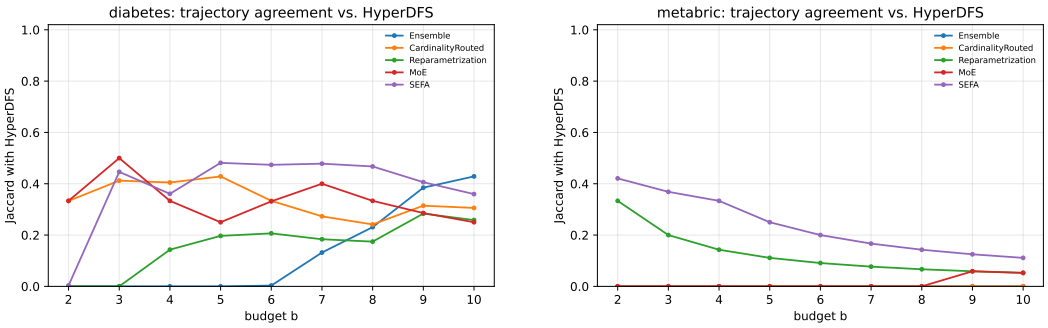


Figure 9: Evolution of Jaccard similarity of the features selected in each budget, comparing different DFS methods to HYPER-DFS.

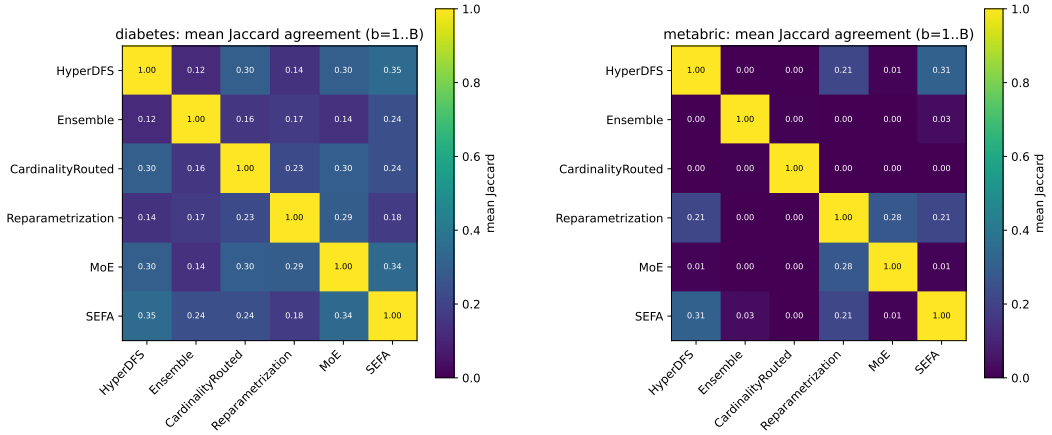
Table 10 reports the full per-dataset zero-shot AUAC-F1 results that underlie the aggregated report in the main text. VIP is the one clearly performing worse: on California, Cirrhosis, Heart, and Wine, VIP loses more than 22 absolute points to both MoE and the Hypernetwork, with the largest gap on Wine (-51.95 against MoE), whereas on Bank, Metabric, and Miniboone the gap shrinks to between 2 and 11 points. Diabetes is the only tabular dataset where VIP narrowly leads ($+0.33$ over HYPER-DFS, $+4.17$ over MoE), and Yeast is the one where all three methods cluster within a 7-point band on the lowest absolute scores in the benchmark. These cases suggest that single-model degradation under unseen subsets is most pronounced when the task admits distinct decision boundaries for different feature combinations, while on naturally hard datasets, the room for per-subset specialisation is small and the three approaches converge.

The two multi-model methods are close on tabular benchmarks: HYPER-DFS wins on four datasets (Bank, California, Metabric, Miniboone), MoE wins on four (Cirrhosis, Heart, Wine, Yeast), and the average rank (1.56 vs. 1.78) and average AUAC (64.06 vs. 63.11) place them within a fraction of a point of each other. The largest gap in either direction is on Miniboone ($+4.85$ for HYPER-DFS), which is the highest-dimensional tabular benchmark in the suite. We emphasise that this parity is achieved with a fundamentally different cost profile: MoE stores K distinct sets of expert parameters, whereas HYPER-DFS is a single model where all subset-specific classifiers are generated on demand from a shared hypernetwork. The image benchmarks show a more positive balance for VIP. However, HYPER-DFS still wins on all three datasets, with the largest gaps on MNIST ($+3.98$ over VIP) and FashionMNIST ($+3.16$ over VIP), while MoE underperforms VIP on every image benchmark. We can conclude that across both modalities, HYPER-DFS therefore matches or exceeds the strongest multi-model baseline while using the storage cost of a single model.

D.3 Sequential acquisition examples

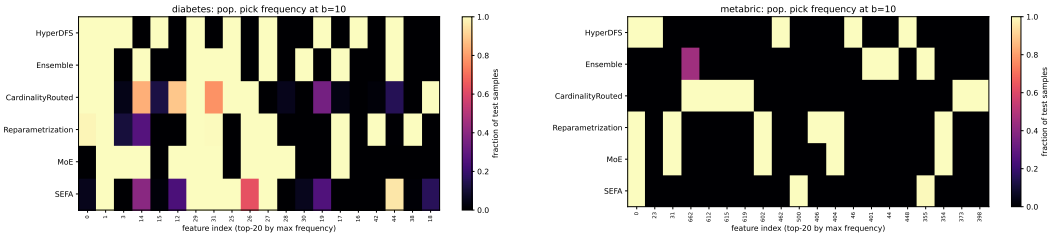
Here, we show an example of the level of agreement in feature selection of the different DFS schemes tested. Figure 9 shows the Jaccard agreement for different budget levels of different DFS methods with HYPER-DFS. We can see there that the selection similarities in Metabric, where the number of features is very large, are significantly smaller than in diabetes, which is somewhat expected. There is no clear relationship between agreement with HYPER-DFS (or with any other method whatsoever) and performance, which indicates that many possible selections can give similar performance. This can explain the poor performance of DIME in datasets with large amounts of features, where it needs to estimate the CMI of all of them in order to make good decisions. Other methods just need to focus on which features make bigger performance increments, which is an easier task.

A pairwise analysis of the feature similarities is also shown in Figure 10, where we can also see that in Metabric, sometimes the similarities are virtually 0 between feature selections. Although it is especially relevant to note that SEFA and Reparametrization (RePa), which are two other well-performing methods, have some significant overlap with HYPER-DFS as well as with each other. This suggests that there are some features that are key to reaching good decisions in some cases, no matter the model, even with such a large number of features.



(a) Diabetes (b) Metabric

Figure 10: Pairwise Jaccard similarity between feature subsets selected at different acquisition budgets for different DFS methods. Each cell (i, j) represents the overlap between subsets selected at budgets i and j, highlighting the stability of the acquisition policy across datasets.



(a) Diabetes (b) Metabric

Figure 11: Frequency of selection per sample of the top 20 most popular features selected overall by the corresponding methods for budget$=10$.

Figure 11 shows the top 20 features most queried by the selectors for these two datasets as well. Here we can see that some methods indeed rely on the same features very often, while we also note significant divergences in Metabric as well. It is quite shocking that for Diabetes, the most popular feature is never queried by SEFA or MoE, which is another indication that models are learning different ways to get relevant information from different features as well, which might also be carrying redundant information.

D.4 Training times

Figure 12 reports wall-clock training times as a function of dataset size for HYPER-DFS and the single-model baselines using the same computational resources. On small datasets (fewer than $\sim$500 samples), all methods are fast, with most completing within a minute. As dataset size grows, costs diverge considerably. SEFA is by far the most expensive method at medium and large scales, exceeding two hours on *Diabetes*. DIME also becomes prohibitively slow at medium scale, exceeding 45 minutes on *Bank* and *California*. HYPER-DFS is below the average across all dataset sizes, and scales similarly to CWCF and Reparametrization at large n. On MiniBooNE (n=130,064), HYPER-DFS requires approximately the same training time as CWCF and Reparametrization ($\sim$90–105 minutes), and is roughly 2.3$\times$ faster than VIP.

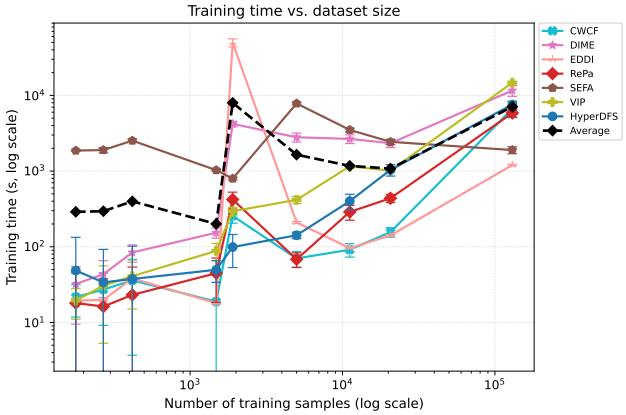


Figure 12: Running times for the different DFS algorithms tested in tabular datasets.

Table 11: Tabular benchmark datasets. N: samples, M: features, C: classes.

Dataset	Domain	N	M	C
Wine [Aeberhard and Forina, 1992]	Scientific	178	13	3
Heart Disease [Detrano et al., 1989]	Medical	270	13	2
Cirrhosis [Fleming and Harrington, 2013]	Medical	418	18	3
Yeast [Nakai, 1991]	Biology	1,484	8	10
METABRIC [Curtis et al., 2012]	Genomics	1,904	689	2
Bank Marketing [Moro et al., 2014]	Financial	11,162	16	2
California Housing [Pace and Barry, 1997]	Real estate	20,640	9	2
Diabetes [Strack et al., 2014]	Medical	101766	45	3
MiniBooNE [Roe et al., 2005]	Physics	130,064	50	2

E Reproducibility

E.1 Datasets Preprocessing Details

Synthetic datasets. Table 12 summarises the six synthetic benchmarks. All datasets are generated with $N = 10{,}000$ samples. **Sim1–3** and **Cube** follow the original protocols of Yoon et al. [2018] and Kompella et al. [2016] respectively, and are included for comparability with the DFS literature. These benchmarks favour instance-wise feature selectors: once the relevant feature set is identified, the Bayes-optimal predictor is essentially the same function regardless of which subset is observed.

Synergistic Pairs ($M = 12$) consists of six independent feature pairs, each drawn i.i.d. from $\mathcal{N}(0, 1)$. Three pairs (x_1, x_2), (x_3, x_4), (x_5, x_6) are signal pairs; the remaining three are pure noise. The label is generated as

$$p(y = 1 \mid \mathbf{x}) = \sigma(x_1x_2 + x_3x_4 + x_5x_6)\,, \tag{20}$$

where σ is the logistic function. Because $\mathbb{E}[x_ix_j \mid x_i] = x_i \cdot \mathbb{E}[x_j] = 0$ for independent features, every individual feature has zero mutual information with y. Signal is only accessible when both members of a pair are observed simultaneously, which requires the predictor to adapt its weights to the observed subset.

Proxy Substitution ($M = 10$) generates a shared latent variable $z \sim \mathcal{N}(0, 1)$ observed through five heteroscedastic proxies $x_i = z + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ with $\boldsymbol{\sigma} = (0.1,\ 0.5,\ 1.0,\ 2.5,\ 5.0)$, and five independent noise features $x_6, \ldots, x_{10} \sim \mathcal{N}(0, 1)$. Labels are $y = \mathbf{1}[z > 0]$. The Bayes-optimal linear predictor for any observed subset S is the Gauss-Markov precision-weighted estimator

$$\hat{z}(S) = \frac{\sum_{i \in S_p} x_i/\sigma_i^2}{1 + \sum_{i \in S_p} 1/\sigma_i^2}, \qquad p(y = 1 \mid \mathbf{x}_S) = \Phi\Big(\hat{z}(S) \cdot \sqrt{1 + \textstyle\sum_{i \in S_p} \sigma_i^{-2}}\Big)\,, \tag{21}$$

Table 12: Synthetic benchmark datasets. M: features, C: classes, M^*: number of features relevant to the Bayes-optimal predictor. All datasets use $N = 10{,}000$ generated samples.

Dataset	Reference	M	C	M^*	Signal structure
Sim1	Yoon et al. [2018]	11	2	6	Linear combination of fixed feature set
Sim2	Yoon et al. [2018]	11	2	6	Nonlinear interactions within fixed feature set
Sim3	Yoon et al. [2018]	11	2	3–5	Instance-wise gating; relevant set depends on x_{10}
Cube	Kompella et al. [2016]	20	8	3	Class-specific discriminative feature triplets
Syn. Pairs	Ours	12	2	6	Product interactions; zero marginal signal per feature
Proxy Sub.	Ours	10	2	5	Heteroscedastic proxies; precision-weighted Bayes optimum

Table 13: Image benchmark datasets under patch-based sequential acquisition. M: number of patches (acquisition steps), d: patch feature dimension, C: classes. † subsampled from the original size.

Dataset	Reference	N_{train}	N_{test}	M	d	C
MNIST	LeCun et al. [1998]	60,000	10,000	49	16	10
Fashion MNIST	Xiao et al. [2017]	60,000	10,000	49	16	10
SVHN	Netzer et al. [2011]	73,257	26,032	64	48	10
Imagenette	Howard [2019]	9,469	3,925	25	512	10
PCam†	Veeling et al. [2018]	50,000	10,000	64	144	2

where $S_p = S \cap \{1, \ldots, 5\}$ and Φ is the standard normal cumulative distribution function. The optimal weight for each proxy depends on which other proxies are available, changing substantially with the observed subset and making fixed-weight single models systematically suboptimal.

Tabular datasets. Table 11 summarises the nine real-world benchmarks. All datasets undergo the same preprocessing pipeline, applied independently within each cross-validation fold to prevent data leakage: numeric features are imputed at the training-fold median and standardised to zero mean and unit variance. Categorical features are imputed with the training-fold mode. For California Housing, the median house value target is binarised at its global median. For METABRIC, we dropped survival months and death from cancer fields to prevent target leakage, and also the patient id row.

Image datasets. Table 13 summarises the five image benchmarks. All datasets are evaluated under the same protocol: each image is divided into a spatial grid of non-overlapping patches, and the budget b controls how many patches are revealed at test time. The acquisition order is determined by the model's policy and all remaining patches are masked (zeroed) before prediction.

MNIST and Fashion MNIST: each 28×28 grayscale image is divided into a 7×7 grid of 4×4 pixel patches, yielding $M = 49$ patches with a raw pixel representation of dimension $d = 16$. Pixel values are normalised to $[0, 1]$.

SVHN [Netzer et al., 2011] consists of 32×32 RGB street-view digit images. Each image is divided into an 8×8 grid of 4×4 pixel patches, yielding $M = 64$ patches of dimension $d = 48$ (raw RGB pixels, normalised to $[0, 1]$). For this dataset, in HYPER-DFS, we use a fix the number of knowledge statuses per batch in the predictor training phase to 3.

Imagenette is a 10-class subset of ImageNet comprising higher-resolution natural images. Raw pixels are not used directly; instead, images are resized to 160px and centre-cropped to 128×128 before being passed through a frozen ResNet-18 backbone pretrained on ImageNet [He et al., 2016]. The spatial feature map from the final residual block is pooled to a 5×5 grid and reshaped to $M = 25$ patch tokens of dimension $d = 512$. The backbone weights are fixed throughout all experiments; only the DFS model heads are trained. Note that the frozen backbone might have been trained on some of the test images of Imagenette, so this comparison should be seen as a controlled comparison between different DFS methods and not as an absolute performance study.

PCam [Veeling et al., 2018] is a binary histopathology dataset of 96×96 RGB lymph node patches, labelled by the presence of metastatic tissue in their centre 32×32 region. We divide the image into an 8×8 grid of 12×12 pixel patches, yielding $M = 64$ patches of dimension $d = 144$. The label in this dataset is determined by the central region, which the acquisition policy must learn to prioritise.

Table 14: HYPER-DFS training hyperparameters.

Hyperparameter	Value
Optimiser	Adam
Learning rate	10^{-2}
Weight decay	10^{-4}
Gradient clip norm	5.0
LR schedule	Cosine ($\eta_{\min}$=0.01η)
Max epochs	200
Early stopping patience	30
Validation fraction	10% with 5-fold evaluation
Noise σ for training	0.20
λ_{scale}	0.1
Scale warm-up epochs	50
$\lambda_{\text{collapse}}$	0.01
Primary mid layer size / Number of layers	64 / 2
Hypernetwork mid layer size /Number of layers	128 /2
Compressor output dim (only on image datasets)	16

Due to the scale of the dataset, we use a stratified subsample of 50,000 images and evaluate via k-fold cross-validation (no fixed test split).

E.2 Model training details

About the uniform sampling. Eq. (1) and Eq. (6) use a uniform distribution over S. Sampling S uniformly over $2^{[M]}$ via independent Bernoulli$(1/2)$ on each variable concentrates the cardinality $|S|$ around $M/2$, leaving small and large subsets severely under-represented. To obtain balanced coverage across cardinalities, we instead sample the cardinality first and then a set of that cardinality: $k \sim \mathcal{U}(\{1, \dots, M\})$, $S \sim \mathcal{U}\Big(\binom{[M]}{k}\Big)$.

HYPER-DFS. Table 14 lists the fixed hyperparameter configuration used in all HYPER-DFS experiments. The compressor is only used for image datasets to reduce the input size of the hypernetwork. We use a linear LR warmup that ramps the learning rate from $\eta/100$ to η over the first 5 epochs, after which a cosine schedule decays it back to $\eta/100$ over the remaining epochs.

Multi-Model Baselines **Ensemble** averages the models resulting from different seed initialisations ($K = 5$). **CardinalityRouted** trains two specialist MLPs with hidden layers $(64, 32)$, one for subsets of cardinality $|S| \leq \lfloor M/2 \rfloor$ and one for $|S| > \lfloor M/2 \rfloor$. Each specialist is trained only on its assigned cardinality range; at inference, the appropriate specialist is selected by hard routing. **MoE** uses $K = 5$ expert MLPs (hidden dim 64) with a router using a routing embedding dimension of 64. A load-balancing auxiliary loss (λ_{balance}=0.01) encourages uniform expert utilisation during training.

Literature Single-Model Baselines For all literature baselines, we use the original published implementations when available. Hyperparameters are set to the author-recommended values from the respective papers.

Computational resources Experiments were carried out on a local workstation (AMD Ryzen 5 5600X, 6 cores / 12 threads, paired with a single NVIDIA RTX 5050 8GB) running Ubuntu and a GPU server with compute nodes with a NVIDIA GTX 1080 Ti. The full synthetic protocol and the tabular benchmark sweep (7 datasets x all baselines x multiple seeds and feature budgets) together amount to on the order of a few hundred single-GPU hours. Individual jobs typically complete in minutes to an hour.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction claim that our paper analyses the limitations of existing dynamic feature selection works (and related fields) and that our proposal can mitigate some of their problems. Section 3 presents these problems in detail, plus our reasoning of *why* our proposal should be able to overcome those limitations. The experiments also back up our claims with superior results to the other stablished methods.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mention the limitations of our existing framework in the final paragraph of the conclusions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: A theorem and its proof is present in Appendix A.1.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The appendix includes a reproducibility section where we discuss all the methods and hyperparameters used.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available with the submission and will be publicly available upon publication as well.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: This is explained in the Experiments section of the main body of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined, or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report standard deviation of all metrics reported as results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This is also discussed in the reproducibility section in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work has no possible relevant negative impact that we are aware of.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This research has no particular negative or societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our research has no relevant risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All baselines and datasets used in the comparison are properly cited. Any relevant work discussed here is properly credited as well.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No assets are part of this submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve human subjects in any way.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects in any way.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We did not use LLMs to develop any of the core parts of this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.