# Which Examples Preserve Decisions? Verified Inverse Design for Model Selection

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Model selection recurs across deployment slices and robustness tests, while training and evaluation data can be expensive to acquire, retain, or repeatedly process. We study decision-preserving acquisition (DPA): given a fixed model-selection protocol and an observed multi-scenario winner map, identify compact training data support that reproduces the same winners with prescribed margins after retraining. Unlike coreset or data-evaluation objectives, DPA targets the stability of the model-selection decision itself. In an idealized convex ERM regime, DPA admits a Karush–Kuhn–Tucker (KKT) feasibility view, while the associated cost-minimization problem is combinatorial; these structural facts motivate local proposal models combined with direct verification. We propose VERDICT (**VER**ified **D**ecision **I**nverse **C**onsistency **T**raining), a propose-and-verify method for DPA. VERDICT derives a local decision-response operator, uses trust-region surrogates over working sets to propose lower-cost designs, and accepts a candidate only after rerunning the fixed protocol verifies the target margins under the stated decision convention. We evaluate VERDICT in settings where verification is feasible, including decision-preserving subset selection under a train-and-verify setting and downstream subset selection for evaluation.

## 1 Introduction

Model selection is a standard step in machine-learning pipelines: practitioners compare candidate model classes or hyperparameter configurations and choose a winner under a validation protocol (Feurer et al., 2015). In many applications, this protocol goes beyond a single average score: candidates are compared across *scenarios*, by which we mean protocol-specified evaluation contexts such as domains (Sagawa et al., 2019; Koh et al., 2021), data slices (Zhang et al., 2022; Tae & Whang, 2021), or noisy/robustness conditions (Birgé et al., 2013; Raschka, 2018; Ma et al., 2021). The observed selection outcome can be thought of as a scenario-wise winner map, which records the candidate selected in each context.

In such settings, practitioners may want to know which parts of the available data are actually needed to support the observed selection outcome, and whether the declared protocol omits scenarios on which that outcome would change. We therefore study an inverse, protocol-relative question: with the candidate set, training routine, validation scenarios, and decision convention fixed, what compact training or evaluation evidence is sufficient to reproduce the observed scenario-wise winners within some prescribed margins?

Existing subset selection, coreset, and data evaluation methods typically target predictive performance, full-data approximation, or the effect of individual examples on a scalar quantity (Mirzasoleiman et al., 2020; Ghorbani & Zou, 2019; Koh & Liang, 2017). These are useful goals, but they do not directly address our target: choosing low-cost training evidence so that a fixed model-selection protocol reproduces the same scenario-wise winners after retraining. We call this problem *decision-preserving acquisition* (DPA). DPA searches for compact or lower-cost acquisition designs that preserve the observed winner map with prescribed margins. This leads naturally to a decision-preserving view of data acquisition. The forward protocol maps an acquisition design to trained candidate models and then to scenario-wise winners; our goal is to search for a low-cost design that reproduces an observed winner map with prescribed margins. We call such a winner map **realizable** under a budget if some admissible acquisition design reproduces it after retraining

under the fixed protocol. This viewpoint separates two questions: when can a winner map be preserved by an admissible design, and how can we search for such designs in practice? In a convex empirical risk minimization regime, realizability can be expressed as feasibility of a lower-level Karush–Kuhn–Tucker (KKT) system. The corresponding cost-minimization problem remains combinatorial even in simple convex settings, motivating local search procedures whose outputs are checked by the original protocol.

With these observations, we propose VERDICT (**VER**ified **D**ecision **I**nverse **C**onsistency **T**raining), a propose-and-verify method for decision-preserving acquisition. VERDICT uses implicit differentiation to build a local decision-response operator, solves trust-region surrogate problems over working sets to propose lower-cost designs, and accepts a proposal only after retraining verifies that the original scenario-wise winners and prescribed margins are preserved.

**Contributions.** We make the following contributions:

- **Decision-preserving acquisition.** We formulate data acquisition around preserving the outcome of a fixed model-selection protocol. Given an observed multi-scenario winner map, DPA seeks low-cost training evidence that reproduces the same winners with prescribed margins after retraining.
- **Verified local search.** We derive a local decision-response operator that predicts how acquisition changes affect decision margins, and use it to propose lower-cost candidate designs. Candidates are accepted only after rerunning the fixed training/evaluation protocol verifies the required margins.
- **Structural context.** In an idealized convex ERM regime, decision preservation can be expressed through a KKT feasibility system, and the associated cost-minimization problem is NP-hard. These results motivate us to devise a propose-and-verify search rather than exact global optimization for VERDICT.
- **Empirical evaluation and downstream applications.** Our empirical focus is on settings where the fixed protocol can be rerun and decision preservation can be verified directly: fixed-feature convex ERM for training-subset acquisition and fixed-model evaluation-subset selection. Across tabular, text, and vision datasets, VERDICT finds compact verified subsets while random, stratified, influence-function-based, and diversity-based baselines often fail to preserve the same decision map at matched sizes. We also study component ablations and demonstrate a scenario-discovery application: held-out failures identify omitted decision-critical scenarios, and after adding them back, VERDICT preserves the expanded scenario set in cases where all same-size baselines fail.

## 2 Related work

### 2.1 Subset selection, coresets, and budgeted training-data acquisition

A large literature studies subset selection for training efficiency or performance preservation. Coreset-style methods select a subset (or reweight samples) so that training on the reduced set closely matches training on the full dataset, in empirical risk minimization and deep learning pipelines (Mirzasoleiman et al., 2020). Related ideas appear in core-set formulations for active learning under labeling costs (Sener & Savarese, 2018), and in data evaluation methods that score data by estimated contribution to validation objectives (Ghorbani & Zou, 2019). Recent work studies budgeted data pruning and selection in large-scale regimes, including settings where data curation changes scaling behavior (Sorscher et al., 2022). In large language model fine-tuning, several methods propose task-specific data selection procedures under limited budgets (Wang et al., 2025). These lines of work usually target predictive performance or training efficiency. They do not encode a winner-take-all rule defined by comparing multiple models across scenarios, nor do they impose margin constraints intended to preserve an observed selection map under a stated decision convention.

### 2.2 Training-data attribution and influence estimation

A complementary literature asks which training points influence model behavior or a validation outcome. Influence functions estimate the effect of upweighting or removing a training point on a downstream metric (Koh & Liang, 2017). Gradient-trajectory methods such as TracIn approximate influence using gradient dot products along the training path (Pruthi et al., 2020), and TRAK develops a scalable attribution technique for large models (Park et al., 2023). Recent work examines influence estimation in large language models

and other high-dimensional settings, including rescaled and robust variants intended to improve stability (Rubinstein & Hopkins, 2025; Tu et al., 2025), and evaluations that report failure modes of standard influence estimates on large models (Li et al., 2025). Other attribution approaches use only the final trained model to estimate data importance (Wei et al., 2025), and some introduce alternative approximations such as zeroth-order influence estimators (Kokhlikyan et al., 2025). Outlier-focused analyses propose signals for identifying detrimental or mislabeled samples (Chhabra et al., 2025). Extensions to other generative model families yield scalable influence-based data attribution methods (Mlodozeniec et al., 2025). Our decision-response operator is related to influence functions in that it maps data perturbations to changes in a downstream quantity. The differences are the outcome and the goal: we map acquisition perturbations to shifts in decision margins across a quasi-active model set, and use this local model to propose constrained acquisition updates for an inverse feasibility problem.

### 2.3 Inverse optimization, bilevel structure, and decision-focused learning

Inverse optimization studies how to choose inputs so that observed decisions are optimal for a forward optimization problem (Heuberger, 2004). When the forward problem is convex, KKT conditions yield reformulations as MPECs (Luo et al., 1996) or other bilevel structures. Surveys provide overviews of bilevel optimization techniques and challenges (Colson et al., 2007). Recent work develops scalable inverse optimization methods in kernelized or large-scale settings (Long et al., 2024). Decision-focused learning incorporates the downstream decision problem into the training objective, often by differentiating through a proxy of the combinatorial optimizer or using a relaxation (Donti et al., 2017; Wilder et al., 2019). A survey summarizes foundations and benchmarks in decision-focused learning (Mandi et al., 2024). Our setting is closer to an inverse feasibility problem for a discrete decision rule induced by scenario-wise model comparisons. We address this with a verified bilevel template: a continuous relaxation proposes candidate acquisitions, and acceptance requires retraining-based checks when verification is feasible.

## 3 Problem Setup

We formalize decision-preserving acquisition under a model-selection protocol that fixes a set of candidate models, a training routine, validation scenarios, validation functionals, and a decision convention. The design variable is the acquired training evidence. Starting from a baseline design that induces an observed winner in each scenario, our goal is to find a lower-cost design that, after retraining the candidates under the same protocol, preserves those winners with specified margins. All notational uses are introduced here and Appendix C.1.

We use the term **scenario** to denote a protocol-defined evaluation context indexed by $t \in \mathcal{T}$, under which candidate models are compared via validation functionals $F_{m,t}$. Scenarios may correspond to domains, fixed data slices, robustness conditions, or other evaluation contexts specified by the protocol. For model $m \in \mathcal{M}$ and scenario $t$, $F_{m,t}$ denotes the validation loss used to compare candidates in that context, with lower values preferred. A **design** $\alpha$ specifies the acquired training evidence used by the pipeline to train each candidate model. We write $F_{m,t}(\alpha)$ for the validation loss obtained by training model $m$ under design $\alpha$ and then evaluating it in scenario $t$. Given a baseline design and a fixed decision convention, the **winner** in scenario $t$, denoted $D_{\mathrm{val}}(t) \in \mathcal{M}$, is the candidate selected by that convention.

To preserve this decision with **stability margin** $\gamma_t \geq 0$, the original winner must remain ahead of every competing model under the same protocol:

$$F_{m,t}(\alpha) - F_{D_{\mathrm{val}}(t),t}(\alpha) \geq \gamma_t \quad \forall m \in \mathcal{M} \setminus \{D_{\mathrm{val}}(t)\}.$$

The decision-preserving acquisition problem is to search for a low-cost design $\alpha$ that satisfies these margin constraints after the pipeline is rerun. Thus feasibility is defined by the forward protocol itself: candidate models are retrained under $\alpha$, evaluated on the fixed scenarios, and checked against the original winner map.

**Candidate Pool, Models, and Scenarios.** We fix a pool of training examples $\mathcal{P} = \{z_i\}_{i=1}^N \subset \mathcal{Z}$, a set of models or hyperparameter configurations $\mathcal{M} = \{1, \ldots, M\}$, and a set of validation scenarios $\mathcal{T} = \{1, \ldots, T\}$.

**Design Variables and Costs.** A design specifies which training points are acquired and at which quality level. Let

$$\mathcal{Q} = \{q^{(1)}, \dots, q^{(K)}\}$$

be a finite set of available quality levels, where a quality level may encode annotation fidelity, preprocessing, or measurement reliability. We encode a design by binary allocation variables

$$\alpha = (\alpha_{i,k})_{i\in[N],\, k\in[K]} \in \{0,1\}^{N\times K},$$

where $\alpha_{i,k} = 1$ means that training point $z_i$ is acquired at quality $q^{(k)}$.We enforce at most one quality level per point:

$$\sum_{k=1}^{K} \alpha_{i,k} \le 1 \quad \forall i.$$

Acquiring point $z_i$ at quality $q^{(k)}$ incurs cost $c_{i,k} := c_k + c_i^{\text{fix}}$, where $c_k \ge 0$ is the quality-dependent cost and $c_i^{\text{fix}} \ge 0$ is a point-specific overhead. The total acquisition cost is

$$C(\alpha) = \sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_{i,k} c_{i,k}.$$

We also require a minimum acquisition size $n_{\min} \ge 1$,

$$\sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_{i,k} \ge n_{\min},$$

which rules out degenerate designs that acquire too little training evidence for the protocol to be meaningful. Under the one-quality-per-point constraint, the acquisition cardinality is

$$|\alpha| := \sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_{i,k}.$$

Given a budget $B > 0$, we write $\mathcal{S}_B(n_{\min})$ for the feasible set of designs satisfying the one-quality-per-point constraint, the minimum acquisition size, and the budget constraint $C(\alpha) \le B$.

**The Forward Pipeline.** Given a design $\alpha$, the pipeline acquires training data according to $\alpha$, trains each candidate model, and selects a winner by comparing validation functionals across scenarios. Let $w_m(\alpha)$ be the parameters returned by the training protocol. When training is stochastic, $w_m(\alpha)$ is defined under the randomness convention used for selection and verification, which is specified in Appendix C.3. Each scenario $t$ specifies a validation functional $F_{m,t}(w)$ evaluated at trained parameters, and we write

$$F_{m,t}(\alpha) := F_{m,t}\big(w_m(\alpha)\big).$$

**Inverse Design Interpretation.** The forward pipeline maps an acquisition design $\alpha$ to trained models, validation losses, and ultimately a scenario-wise winner map. This gives a decision-preserving selection problem: given observed winners under a baseline design, search for a lower-cost design whose forward execution reproduces those winners with the prescribed margins.

**Relaxation and Structural Constraints.** The acquisition design is combinatorial, since $\alpha_{i,k} \in \{0,1\}$ indicates whether point $z_i$ is acquired at quality $q^{(k)}$. For surrogate optimization, we allow relaxed variables $\alpha_{i,k} \in [0,1]$ while keeping the one-quality-per-point constraint $\sum_k \alpha_{i,k} \le 1$. In the convex ERM setting, relaxed $\alpha$ acts as a training weight; in general pipeline settings, relaxed candidates are rounded before retraining and verification. The minimum acquisition constraint $\sum_{i,k} \alpha_{i,k} \ge n_{\min}$ rules out degenerate designs that acquire too little training evidence for the model-selection protocol to be meaningful.

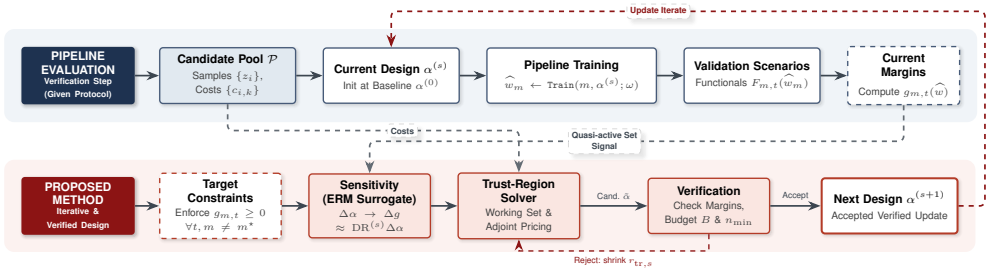



Figure 1: **Overview of decision-preserving acquisition. Left:** given a design $\alpha$, a protocol trains models and evaluates validation functionals across scenarios to form margin constraints. **Right:** we propose a lower-cost candidate $\tilde{\alpha}$ using a local decision-response model, then accept it only after retraining verifies all margins under the decision convention that defines the observed winners.

### 3.1 Decision map, margins, and verification

Fix a baseline design $\alpha^{(0)}$ and a decision convention, either a fixed-seed convention or an aggregated convention over a finite seed set, with deterministic tie-breaking. For each scenario $t \in \mathcal{T}$, let $D_{\mathrm{val}}(t) \in \mathcal{M}$ denote the observed winner under this convention. To preserve decisions with stability margins $\gamma_t \geq 0$, we enforce for every competitor $m \neq D_{\mathrm{val}}(t)$ the constraint

$$g_{m,t}(\alpha) := F_{m,t}(\alpha) - F_{D_{\mathrm{val}}(t),t}(\alpha) - \gamma_t \geq 0.$$

Under the aggregated convention, we define

$$\bar{g}_{m,t}(\alpha) := \bar{F}_{m,t}(\alpha) - \bar{F}_{D_{\mathrm{val}}(t),t}(\alpha) - \gamma_t,$$

where $\bar{F}_{m,t}(\alpha)$ aggregates $F_{m,t}$ over the seed set as specified in Appendix C.3. Appendix C provides the operational semantics of the relaxation, the full randomness convention used for verification, and the margin selection protocol.

### 3.2 Inverse Objective and Verification

We minimize acquisition cost subject to preserving the selection map with margins and operational constraints, under the same convention used to define $D_{\mathrm{val}}$:

$$\min_{\alpha \in \mathcal{S}_B(n_{\min})} \quad C(\alpha) \quad \text{s.t.} \quad g_{m,t}(\alpha) \geq 0 \;\; \forall t, \; \forall m \neq D_{\mathrm{val}}(t),$$

with $g_{m,t}$ replaced by $\bar{g}_{m,t}$ under the aggregated convention. When relaxed surrogates are used, we round to $\alpha^b$ as in Definition C.1, and acceptance and verification are performed on $\alpha^b$ under the decision convention in Definition C.4.

## 4 Decision Response and Verified Preservation

This section formalizes the decision-preservation problem and derives the local response model used by VERDICT. We first show that, in convex ERM, decision preservation can be written as a feasibility problem through the lower-level KKT conditions of the trained models. This provides a formal reference point for the inverse view, but does not make the associated cost-minimization problem easy: even simple convex instances remain combinatorial. We then derive local response operators that approximate how acquisition changes affect decision margins. In general training pipelines, these operators are used only to propose candidate updates; accepted designs are validated by rerunning the fixed protocol and checking the prescribed margin constraints directly.

For $\boldsymbol{w} = (w_m)_{m \in \mathcal{M}} \in \prod_{m \in \mathcal{M}} \mathcal{W}$, define

$$g_{m,t}(\boldsymbol{w}) := F_{m,t}(w_m) - F_{D_{\mathrm{val}}(t),t}(w_{D_{\mathrm{val}}(t)}) - \gamma_t.$$

For a design $\boldsymbol{\alpha}$, write $g_{m,t}(\boldsymbol{\alpha}) := g_{m,t}(\boldsymbol{w}^\star(\boldsymbol{\alpha}))$ in the convex empirical risk minimization world and $g_{m,t}(\boldsymbol{\alpha};\omega) := g_{m,t}(w_{\text{pipe}}(\boldsymbol{\alpha};\omega))$ in the pipeline world. Under the aggregated convention, $\bar{g}_{m,t}(\boldsymbol{\alpha})$ is defined by the same decision convention in Definition C.4. This notation lets us express decision preservation either as constraints on trained parameters or, via the forward map, as constraints on designs.

### 4.1 Inverse Feasibility via KKT Conditions in Convex ERM

In the convex ERM regime, the forward map $\boldsymbol{\alpha} \mapsto \boldsymbol{w}^\star(\boldsymbol{\alpha})$ is defined by solving a strongly convex objective for each model. This allows us to replace retraining by an optimality system and to phrase inverse realizability as feasibility of a single set of constraints. In this regime, each candidate model $m$ is trained by minimizing an acquisition-weighted ERM objective over $w \in \mathcal{W}$, where $R_m$ is a regularizer and $\ell_m$ is the per-example loss at quality level $q^{(k)}$. For each candidate model $m$, let $L_m(w;\alpha)$ denote the training objective optimized under acquisition design $\alpha$, such that

$$L_m(w;\alpha) = R_m(w) + \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_{i,k}\, \ell_m(w; z_i, q^{(k)}),$$

**Definition 4.1** (Normal Cones)**.** For a closed convex set $C \subset \mathbb{R}^d$, $N_C(x)$ denotes the convex-analytic normal cone at $x$.

**Assumption 4.2** (Convex Forward Regime)**.** For every $m \in \mathcal{M}$ and every $\boldsymbol{\alpha} \in \mathcal{S}_B^{\text{rel}}(n_{\min})$, the map $w \mapsto L_m(w;\boldsymbol{\alpha})$ is differentiable and $\mu$-strongly convex on $\mathcal{W}$, for some $\mu > 0$ independent of $(m, \boldsymbol{\alpha})$.

Under Assumption 4.2, each forward solution satisfies the variational inclusion

$$0 \in \nabla_w L_m(w;\boldsymbol{\alpha}) + N_{\mathcal{W}}(w). \tag{1}$$

**Definition 4.3** (Budgeted Inverse KKT System)**.** Fix $B > 0$. The inverse system consists of variables $(\boldsymbol{\alpha}, \boldsymbol{w})$ satisfying

$$\boldsymbol{\alpha} \in \mathcal{S}_B^{\text{rel}}(n_{\min}), \tag{2}$$

$$0 \in \nabla_w L_m(w_m;\boldsymbol{\alpha}) + N_{\mathcal{W}}(w_m), \quad \forall m \in \mathcal{M}, \tag{3}$$

$$g_{m,t}(\boldsymbol{w}) \geq 0, \quad \forall t \in \mathcal{T},\ \forall m \neq D_{\text{val}}(t). \tag{4}$$

Integral designs correspond to $\boldsymbol{\alpha} \in \mathcal{S}_B(n_{\min}) \subset \mathcal{S}_B^{\text{rel}}(n_{\min})$.

**Proposition 4.4** (Equivalence of Realizability and Inverse Feasibility)**.** *Suppose Assumption 4.2 holds. The observed decision map is realizable under budget $B$ in convex ERM if and only if the inverse system in Definition 4.3 is feasible. The proof for Proposition 4.4 is in Appendix D.*

### 4.2 Local Response and Decision-Response Operators

The idealized cost-minimization form of DPA is combinatorial even in the simple convex setting of Theorem D.1, we build a local proposal model around a verified reference design that preserves the decision map. The goal is to approximate how trained parameters, and therefore margins, change under small perturbations of $\boldsymbol{\alpha}$, yielding operators used in a first-order surrogate.

**Assumption 4.5** (Reference Design Regularity)**.** Assume there exists a (possibly relaxed) design $\boldsymbol{\alpha}^\sharp$ such that $g_{m,t}(\boldsymbol{\alpha}^\sharp) \geq 0$ for all $t$ and all $m \neq D_{\text{val}}(t)$. Assume $w_m^\sharp := w_m^\star(\boldsymbol{\alpha}^\sharp) \in \text{int}(\mathcal{W})$ and that the Hessian of the training objective with respect to parameters at reference point $H_m^\sharp := \nabla_w^2 L_m(w_m^\sharp;\boldsymbol{\alpha}^\sharp)$ is positive definite for all $m$.

**Assumption 4.6** (Local Smoothness)**.** Assume $w \mapsto \ell_m(w; z_i, q^{(k)})$ and $w \mapsto R_m(w)$ are twice differentiable near $w_m^\sharp$, and the validation functionals used in the quasi-active set are differentiable near $w_m^\sharp$.

**Lemma 4.7** (Implicit Differentiation of Parameters)**.** *Assume Assumptions 4.5 and 4.6. For any perturbation $\Delta\boldsymbol{\alpha}$,*

$$Dw_m^\star(\boldsymbol{\alpha}^\sharp)[\Delta\boldsymbol{\alpha}] = -(H_m^\sharp)^{-1}\left(\tfrac{1}{N}\sum_{i,k} \Delta\boldsymbol{\alpha}_{i,k} \times \nabla_w \ell_m(w_m^\sharp; z_i, q^{(k)})\right).$$

**Definition 4.8** (Response Operator)**.** Define $\mathsf{Resp}^\sharp : \mathbb{R}^{N\times K} \to \mathbb{R}^{Md}$ by

$$\mathsf{Resp}^\sharp(\Delta\boldsymbol{\alpha}) := -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\Delta\boldsymbol{\alpha}_{i,k}\, r^\sharp_{i,k},$$

where $r^\sharp_{i,k}$ stacks the model-wise directions $(H^\sharp_m)^{-1}\nabla_w \ell_m(w^\sharp_m; z_i, q^{(k)})$ into $\mathbb{R}^{Md}$. Then $\boldsymbol{w}^\star(\boldsymbol{\alpha}^\sharp + \Delta\boldsymbol{\alpha}) \approx \boldsymbol{w}^\star(\boldsymbol{\alpha}^\sharp) + \mathsf{Resp}^\sharp(\Delta\boldsymbol{\alpha})$ for small $\Delta\boldsymbol{\alpha}$.

**Definition 4.9** (Quasi-Active Sets and Decision-Response)**.** Fix a threshold $\tau \geq 0$ and define the quasi-active index set

$$J^\tau := \Big\{(m,t) : m \neq D_{\mathrm{val}}(t),\ t \in \mathcal{T},\ g_{m,t}(\boldsymbol{\alpha}^\sharp) \leq \tau\Big\}.$$

Let $g_{J^\tau}$ stack the margins over $(m,t) \in J^\tau$. Define the decision-response operator

$$\mathsf{DR}^{\sharp,\tau}(\Delta\boldsymbol{\alpha}) := Dg_{J^\tau}\big(\boldsymbol{w}^\star(\boldsymbol{\alpha}^\sharp)\big)\,\mathsf{Resp}^\sharp(\Delta\boldsymbol{\alpha}).$$

### 4.3 Local Convex Surrogate and Candidate Expansion

The operators above give a local linear model for the quasi-active margins around $\boldsymbol{\alpha}^\sharp$. We solve a convex surrogate with a trust region, then use dual information to expand a working set.

Fix $\boldsymbol{\alpha}^\sharp$ and $J^\tau$. Let $s^{\sharp,\tau} := g_{J^\tau}(\boldsymbol{w}^\star(\boldsymbol{\alpha}^\sharp))$. We treat the budget either as a hard constraint via $\mathcal{S}^{\mathrm{rel}}_B(n_{\min})$ or through the cost objective. Let $\mathcal{S}$ denote the chosen feasible set (i.e., $\mathcal{S}^{\mathrm{rel}}(n_{\min})$, or $\mathcal{S}^{\mathrm{rel}}_B(n_{\min})$ if the budget is hard). A first-order convex surrogate is

$$\begin{aligned} \min_{\boldsymbol{\alpha},\,\xi}\quad & C(\boldsymbol{\alpha}) + \lambda_{\mathrm{lin}}\|\xi\|_1 \\ \text{s.t.}\quad & \boldsymbol{\alpha} \in \mathcal{S}, \\ & \mathsf{DR}^{\sharp,\tau}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^\sharp) \geq -s^{\sharp,\tau} - \xi, \\ & \xi \geq 0, \quad \big\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\sharp\big\|_F \leq r_{\mathrm{tr}}. \end{aligned} \tag{5}$$

The slack $\xi$ accounts for linearization error and incomplete working sets; candidates are accepted or rejected by full verification.

**Proposition 4.10** (Adjoint Scoring for Expansion)**.** *Let $u \geq 0$ be dual multipliers associated with the linearized margin constraints in equation 5. Define scores by*

$$s_{i,k} := -\big((\mathsf{DR}^{\sharp,\tau})^* u\big)_{i,k}.$$

*Pairs with large magnitude scores are predicted to have large first-order impact on quasi-active margins and can be added to a working set for the next surrogate solve.*

### 4.4 Verified Trust-Region SCA

In the general training setting, the convex surrogate acts as a proposal mechanism. We certify candidates by rerunning the pipeline and verifying margin constraints under the chosen convention, ensuring accepted designs satisfy the requirements.

**Proposition 4.11** (Verification of Accepted Designs)**.** *Fix a decision convention and a training pipeline. Any algorithm accepting a candidate design only after retraining and verifying margin constraints yields decision-feasible designs. If budget and minimum-acquisition constraints are enforced at acceptance (including on rounded designs), the output satisfies these constraints.*

*Remark* 4.12 (Relation to Influence Functions)*.* The adjoint scores in Proposition 4.10 rank examples by their predicted impact on decision margins. We use this ranking for budgeted data selection when repeated verification is impractical.

**Algorithm 1** Verified Trust-Region SCA

**Require:** Baseline $\boldsymbol{\alpha}^{(0)}$ (valid per convention); $r_{\mathrm{tr},0} > 0$; $\tau_{\min} \geq 0$, $\beta > 0$; factors $\beta^{\pm}$.

**for** $s = 0, 1, 2, \ldots$ **do**
  **(1) Quasi-active set.** Set $\tau_s \leftarrow \max\{\tau_{\min}, \beta r_{\mathrm{tr},s}\}$. Construct $J^{(s)}$ via proxy or ERM surrogate (Definition 4.9) union with pipeline active-set.
  **(2) Local operators.** Compute $\mathsf{DR}^{(s)}$ and adjoints.
  **(3) Surrogate solve.** Obtain relaxed $\boldsymbol{\alpha}^e$ by solving equation 5 with $r_{\mathrm{tr},s}$ and $\boldsymbol{\alpha}^\sharp = \boldsymbol{\alpha}^{(s)}$.
  **(4) Rounding.** Round $\boldsymbol{\alpha}^e \to \boldsymbol{\alpha}^b$. If the hard constraint $|\boldsymbol{\alpha}^b| < n_{\min}$ holds, apply completion. **Reject** step if operational constraints fail.
  **(5) Verification.** Retrain and check:
    (a) **Decisions:** $g_{m,t}(\boldsymbol{\alpha}^b; \omega_0) \geq 0$ (fixed-seed) or $\bar{g}_{m,t}(\boldsymbol{\alpha}^b) \geq 0$ (aggregated).
    (b) **Operational:** $C(\boldsymbol{\alpha}^b) \leq B$ (if hard budget) and $|\boldsymbol{\alpha}^b| \geq n_{\min}$.
  **if** all checks pass **then**
    $\boldsymbol{\alpha}^{(s+1)} \leftarrow \boldsymbol{\alpha}^b$, $r_{\mathrm{tr},s+1} \leftarrow \beta^+ r_{\mathrm{tr},s}$ **(Accept)**
  **else**
    $r_{\mathrm{tr},s} \leftarrow \beta^- r_{\mathrm{tr},s}$, re-solve Step 3 **(Reject)**
  **end if**
  **stop** if $r_{\mathrm{tr},s} <$ tolerance or no valid improvement found.
**end for**

*Remark* 4.13 (The Ideal Cost Objective is Combinatorial). Proposition 4.4 gives a feasibility characterization in the convex ERM case, but it does not imply that the lowest-cost decision-preserving design can be computed efficiently. The ideal cost-minimization version of decision preservation is combinatorial even in a simple convex setting. This is shown in Appendix D.2.

## 5 Computational Experiments

This section evaluates decision-preserving acquisition across three modalities: tabular, image, and text classification. Our main experiment studies training-subset acquisition under a fixed convex ERM protocol: given a full-data model-selection decision, we ask how much training evidence is needed so that retraining candidate models on a subset preserves the same scenario-wise winners with prescribed margins. We then evaluate two downstream uses of the same decision-preservation framework: decision-critical scenario discovery and low-cost evaluation sets for model selection. We evaluate all methods on nine fixed-feature classification datasets, namely Credit, Bank, Adult from the UCI Machine Learning Repository (Dua & Graff, 2017), OxfordPets (Parkhi et al., 2012), STL-10 (Coates et al., 2011), CIFAR10 and CIFAR100 (Krizhevsky et al., 2009), 20Newsgroups (Lang, 1995), and AG News (Zhang et al., 2015), all of which are treated as supervised classification problems under a fixed-feature convex ERM protocol. All experiments are run on Python 3.11.13 with five seeds on an Intel Xeon Gold 6230R (104 cores), 188 GiB RAM, and an NVIDIA RTX A6000 (48 GiB VRAM). We report mean and standard deviation over randomized methods.

In all experiments, we instantiate the single-quality specialization of the general formulation: $K = 1$, $c_{i,1} = 1$, and $\alpha_{i,1} \equiv \alpha_i$. Therefore cost equals retained subset size. The experiments test decision-preserving subset selection under fixed-quality evidence; evaluating tradeoffs between different acquisition quality levels is left for future work. For each scenario $t$, the full protocol selects a winner $D_{\mathrm{val}}(t)$ according to validation loss. A candidate acquisition design is feasible if, after executing the same forward protocol under that design, the original winner remains ahead of every competitor by the required margin:

$$F_{m,t}(\alpha) - F_{D_{\mathrm{val}}(t),t}(\alpha) \geq \gamma_t, \qquad \forall t,\ m \neq D_{\mathrm{val}}(t),$$

where lower validation loss is better and $\gamma_t$ is set as a fraction of the full-data winner gap. Unless otherwise stated, we use $\gamma_t = \eta\, \mathrm{gap}_t$ with $\eta = 0.5$.

### 5.1 Direct Training Acquisition

This experiment directly instantiates the inverse-design formulation in the convex ERM setting. The raw input modality determines the feature map $\phi$, but after feature construction all methods operate on the same fixed design matrix. The full training pool defines the reference protocol: each candidate model is trained on the full pool, evaluated on validation scenarios, and compared by scenario-wise validation loss. This produces the observed winner map $D_{\text{val}}(t)$ and full-data winner gaps.

**Fixed-feature convex ERM protocol and scenarios.** All direct-acquisition experiments use fixed features. Each example $z_i$ is mapped once to $x_i = \phi(z_i)$ before acquisition: frozen image representations for vision, one-hot categorical plus standardized numerical features for tabular data, and TF-IDF/SVD features fit once on the training pool for text. Acquisition therefore changes only the weighted convex ERM objective, not the representation. For every acquired subset, we retrain all candidate models under the same objective and verify the original protocol-level margin constraints on fixed validation scenarios.

A scenario is a validation slice on which candidate models are compared. Vision and text scenarios are class-wise or class-group slices, while tabular scenarios are protocol-defined feature slices such as demographic groups, service categories, risk groups, and quantile buckets. Scenario construction details are given in Appendix F.

We compare VERDICT with six subset-selection baselines: uniform random sampling, class-stratified sampling, HighLoss, GradNorm, K-center, and DecisionIF. HighLoss and GradNorm select examples with the largest reference-model training loss and gradient norm, respectively. K-center performs greedy farthest-first selection in the frozen feature space, using a class-balanced heuristic for large datasets. DecisionIF is a one-shot decision-influence baseline that estimates, at the full-data solution, the first-order effect of removing each training example on quasi-active decision margins and retains examples whose removal is predicted to be most harmful. Meanwhile, VERDICT starts from the full feasible design and iteratively proposes deletions using a local decision-response model. At each iteration, it forms a quasi-active constraint set, solves a greedy trust-region deletion surrogate, retrains all candidate models on the proposed subset, and accepts the proposal only if the original winner map, margins, and operational constraints are verified. We run VERDICT for 30 proposal iterations and report ablations of $\eta$ and VERDICT components in Appendices G.2 and G.3.

**Accepted and rejected proposals.** Table 1 reports the smallest retained training fraction at which each method passes the same retrain-and-check decision-preservation test. VERDICT, HighLoss, GradNorm and DecisionIF are deterministic and have no seed variance. VERDICT achieves the best verified fraction on 8 of 9 datasets, with especially large gains in vision: STL10 and CIFAR100 require only 7.3% and 2.0% of the training set respectively, while HighLoss, GradNorm and coverage-based baselines require substantially larger fractions or fail within the evaluated budget grid. This indicates that preserving a model-selection decision can require a much smaller and more targeted support than preserving generic training-set coverage. The only exception is Adult, where stratified sampling outperforms VERDICT. This suggests a regime in which the winner map is largely preserved by coarse scenario balance rather than by the decision-response signal alone. VERDICT is most useful when feasible subsets are rare under generic sampling and depend on decision-relevant examples, while simpler sampling can be competitive when the protocol is highly redundant or balance-driven.

**Verified search behavior.** Table 1 does not reveal whether feasibility is rare, whether the local surrogate is well calibrated, or whether generic sampling already succeeds at comparable budgets. Table 2 separates these regimes by examining random feasibility at VERDICT's final budget and the behavior of the verified local search. The accepted-path columns show that reductions in data are not obtained through only infinitesimal edits: several datasets admit large verified deletions in a single accepted proposal, followed by additional verified reductions. This supports the intended use of the response model as a large-step proposal mechanism for the inverse design problem.

The Random@final diagnostic asks whether VERDICT's final retained fraction is also feasible for unstructured random subsets. For most datasets, this rate is zero, indicating that VERDICT's final subsets are not typical random feasible subsets at the same size. Bank is the main exception: random subsets sometimes satisfy the

Table 1: Smallest retraining-verified retained fraction needed for exact decision feasibility; lower is better. Randomized methods show mean ± standard deviation over 5 seeds. Entries $> 0.950$ indicate that the method did not become feasible within the evaluated budget grid.

| Task | Dataset | $n$ | Methods | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Random | Stratified | HighLoss | GradNorm | KCenter | DecisionIF | VERDICT |
| Tabular | Credit | 30000 | $0.867 \pm 0.029$ | $0.933 \pm 0.029$ | $> 0.950$ | $> 0.950$ | $0.960 \pm 0.042$ | $> 0.950$ | **0.427** |
| | Bank | 45211 | $0.520 \pm 0.045$ | $0.540 \pm 0.042$ | $> 0.950$ | $> 0.950$ | $0.740 \pm 0.192$ | $> 0.950$ | **0.413** |
| | Adult | 48842 | $0.520 \pm 0.329$ | $\mathbf{0.360 \pm 0.055}$ | 0.900 | 0.900 | $0.970 \pm 0.045$ | $> 0.950$ | 0.619 |
| Vision | OxfordPets | 7349 | $> 0.950$ | $> 0.950$ | $> 0.950$ | 0.950 | $> 0.950$ | 0.350 | **0.221** |
| | STL10 | 13000 | $0.850 \pm 0.100$ | $0.883 \pm 0.104$ | $> 0.950$ | 0.950 | $0.990 \pm 0.022$ | 0.350 | **0.073** |
| | CIFAR10 | 50000 | $0.917 \pm 0.104$ | $0.833 \pm 0.029$ | 0.950 | 0.950 | $0.860 \pm 0.108$ | 0.950 | **0.144** |
| | CIFAR100 | 50000 | $0.550 \pm 0.132$ | $0.417 \pm 0.076$ | $> 0.950$ | $> 0.950$ | $0.500 \pm 0.122$ | 0.300 | **0.020** |
| Text | 20Newsgroups | 18846 | $0.817 \pm 0.076$ | $0.800 \pm 0.000$ | $> 0.950$ | 0.800 | $0.800 \pm 0.035$ | $> 0.950$ | **0.458** |
| | AG News | 127600 | $0.950 \pm 0.000$ | $0.950 \pm 0.000$ | $> 0.950$ | $> 0.950$ | $0.950 \pm 0.000$ | $> 0.950$ | **0.544** |

constraints at VERDICT's final retained fraction, suggesting that the corresponding winner map has more redundancy under the chosen scenarios. Even there, VERDICT still attains the smallest retained fraction in Table 1, so random feasibility at the VERDICT budget suggests that the decision-preservation constraints are relatively redundant or insensitive to which exact examples are selected at that retained fraction. The rejected-proposal columns show why retrain-and-check verification is necessary. In every dataset, the surrogate eventually proposes a candidate with positive predicted margin that becomes infeasible after retraining, sometimes with winner flips. The effect is especially pronounced on Adult, Bank, STL10, and CIFAR10, where surrogate-feasible proposals can become clearly infeasible after retraining. The violations/winner-flips column further separates two boundary effects: some rejected proposals only break the prescribed margin, while others change the selected winner in at least one scenario.

Table 2: Verified search diagnostics for direct train-and-verify DPA.

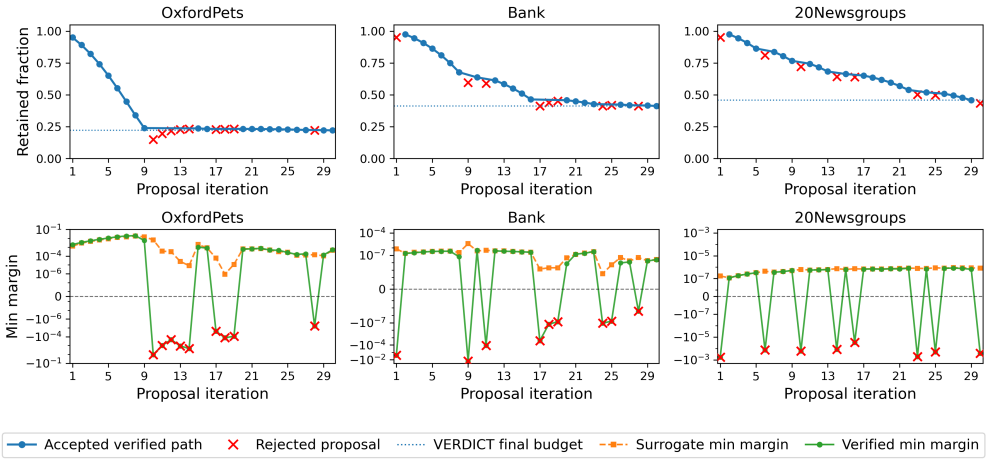
| Dataset | Final frac. | Random@ VERDICT final frac. | Accept | Reject | Largest accepted deletion | Final verified margin | First rejected surrogate→ verified | Violations/ winner flips |
|---|---|---|---|---|---|---|---|---|
| Credit | 0.427 | 0/50 | 23 | 7 | 1077 | $7.27\times10^{-5}$ | $3.80\times10^{-5} \rightarrow -1.59\times10^{-4}$ | 3/3 |
| Bank | 0.413 | 19/50 | 21 | 9 | 2582 | $2.50\times10^{-8}$ | $7.36\times10^{-7} \rightarrow -2.40\times10^{-3}$ | 5/4 |
| Adult | 0.619 | 0/50 | 22 | 8 | 1193 | $2.54\times10^{-6}$ | $4.30\times10^{-6} \rightarrow -3.02\times10^{-4}$ | 2/2 |
| OxfordPets | 0.221 | 0/50 | 21 | 9 | 313 | $4.96\times10^{-4}$ | $6.72\times10^{-3} \rightarrow -1.04\times10^{-2}$ | 1/1 |
| STL10 | 0.073 | 0/50 | 22 | 8 | 426 | $6.70\times10^{-4}$ | $7.60\times10^{-3} \rightarrow -5.57\times10^{-2}$ | 5/1 |
| CIFAR10 | 0.144 | 0/50 | 23 | 7 | 2783 | $3.43\times10^{-3}$ | $9.60\times10^{-4} \rightarrow -2.31\times10^{-4}$ | 1/0 |
| CIFAR100 | 0.020 | 0/50 | 22 | 8 | 3302 | $1.44\times10^{-4}$ | $4.33\times10^{-3} \rightarrow -5.42\times10^{-4}$ | 1/0 |
| 20Newsgroups | 0.458 | 0/50 | 22 | 8 | 501 | $6.58\times10^{-7}$ | $1.71\times10^{-7} \rightarrow -5.79\times10^{-4}$ | 1/1 |
| AG News | 0.544 | 0/50 | 19 | 11 | 7421 | $2.09\times10^{-6}$ | $2.36\times10^{-7} \rightarrow -1.99\times10^{-4}$ | 1/1 |

**Verified local search trajectories.** Figure 2 and Figure 3 in Appendix G.1 show the verified search process under the 30-proposal protocol. The key diagnostic is the discrepancy between the surrogate-predicted margin and the margin obtained after retraining across proposals. Across datasets, accepted proposals usually have positive surrogate and retraining-verified margins, indicating that the local response model gives useful directions for reducing the retained fraction. At the same time, rejected proposals expose a consistent failure mode: a proposal may appear margin-preserving under the surrogate while violating the decision margin after retraining. In three representative datasets, the regimes differ: OxfordPets compresses rapidly and then saturates near a verified boundary; Bank benefits from the longer horizon, recovering from early and mid-search surrogate failures; and 20Newsgroups progresses more slowly near the feasibility boundary.

### 5.2 Downstream Applications

**Decision-critical scenario discovery.** DPA can be used to audit whether a declared protocol omits decision-critical scenarios. Since the guarantee is protocol-relative, omitted scenarios are not guaranteed to be preserved. We therefore run VERDICT on a set of scenarios that excludes a scenario family, then evaluate the resulting subset on the held-out family. A held-out failure indicates that the omitted scenario contains

Figure 2: Verified local search trajectories for VERDICT. **Top row**: retained subset fraction across proposals. **Bottom row**: surrogate-predicted minimum decision margin for the corresponding proposals. The margin axis uses a symmetric logarithmic scale to show both small positive margins and negative verification failures.




constraints that should be added to the declared protocol. After adding the omitted scenarios back, we report the additional retained evidence that VERDICT requires to preserve the expanded protocol. Table 3 summarizes representative examples across tabular, text, and vision data; full details and experimental results on leave-one-scenario audits and per-seed baseline results are presented in Appendix H.1.

Table 3: Decision-critical scenario discovery across three modalities. $\rho_{\text{search}}$ is the retained fraction selected using only the search scenario set; $\rho_{\text{expanded}}$ is the retained fraction after adding the omitted family back. $\Delta_{\min}^{\text{heldout}}$ is the minimum held-out margin before expansion, while $\Delta_{\min}^{\text{expanded}}$ is the final minimum margin after expansion. Negative held-out margins indicate violations.

| Task | Dataset | Omitted family | Held-out audit | | | Expanded protocol | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Flips | Viol. | $\Delta_{\min}^{\text{heldout}}$ | $\rho_{\text{search}}$ | $\rho_{\text{expanded}}$ | $\Delta_{\min}^{\text{expanded}}$ | Best baseline feasible |
| Tabular | Adult | age slices | 2 | 5 | $-2.15\times10^{-3}$ | 0.378 | 0.619 | $+2.54\times10^{-6}$ | 0/20 |
| Text | AG News | world news | 1 | 5 | $-8.18\times10^{-2}$ | 0.070 | 0.544 | $+2.09\times10^{-6}$ | 0/20 |
| Vision | STL10 | failure splits | 3 | 9 | $-5.73\times10^{-2}$ | 0.061 | 0.073 | $+6.70\times10^{-4}$ | 0/20 |

**Low-cost evaluation subsets for model selection.** A second use of DPA is to construct compact evaluation subsets for repeated model selection. In this setting, candidate models are trained as usual, but instead of evaluating them on the full validation set, VERDICT selects a small validation subset whose losses preserve the same scenario-wise winners and margins as the full validation protocol. This gives practitioners a cheaper evaluation set for comparing many candidate models, hyperparameters, or checkpoints while retaining an explicit decision-preservation certificate. In this downstream application, acquisition reduces the validation evidence used to evaluate an already trained candidate set. We demonstrate this in Appendix H.2.

## 6 Discussion and Limitations

**Interpreting decision-preserving designs** In many model-selection workflows, the downstream action is a often comparison result: which hyperparameters win on a slice, which model remains best under a robustness condition, or which candidate should be selected before a final full-data training or deployment step. In these settings, the relevant evidence maintains the pairwise margins defining the selection outcome. This distinction helps explain the direct-acquisition results. Coverage- or difficulty-based rules often require much larger retained fractions or fail within the evaluated grid' VERDICT instead targets near-active winner-versus-competitor comparisons, enabling smaller decision-preserving subsets on several datasets. The exceptions are also informative: on Adult and Bank, generic sampling can be competitive, suggesting that

when the winner map is driven by coarse balance or redundant scenarios, decision-aware search has less room to improve. VERDICT is most useful when feasibility at the same retained fraction is rare under generic subsets and preservation depends on examples that support the active comparisons.

**Candidate pools and scenarios determine the preservation problem** The retained fraction in DPA must be interpreted relative to the model-selection protocol that produced the winner map. Well-separated candidate models can yield a stable winner map that many subsets preserve, whereas near-tied candidates make preservation harder because small acquisition changes may change the ordering. Likewise, coarse or redundant scenarios may require only broad balance, while fine-grained scenarios impose many local comparisons simultaneously. This helps explain Tables 1 and 2: small retained fractions may indicate a compact set of margin-critical examples, whereas larger fractions may reflect close comparisons or broader coverage needs. Thus, the random-feasibility diagnostic can separate these regimes. If random subsets at VERDICT's final size are often feasible, the winner map is relatively insensitive to the selected examples; otherwise, preservation depends on active-comparison structure and targeted search becomes important. Thus, DPA makes the candidate pool and scenario design measurable parts of the preservation problem.

**Compression paths and preservation boundaries** VERDICT's search path can be viewed as moving toward the boundary of the designs that preserve the original winner map. Early proposals often remove many examples while keeping the verified margins positive, because the subset only needs to maintain the comparisons that determine the selection outcome. Near the boundary, however, the local response model may still identify plausible deletions even though retraining reveals a margin violation or winner flip. The final design shows how much the protocol can be compressed by verified local search, while rejected proposals show where further compression begins to change the decision. The trajectories illustrate different regimes: rapid compression followed by saturation on OxfordPets, CIFAR100, STL10, and AG News; gradual boundary-seeking on Bank, Credit and CIFAR10, and slower progress with repeated failures on 20Newsgroups and Adult. Combined with the Random@final diagnostic in Table 2, these patterns help distinguish redundant decision protocols from those whose preservation depends on more specific active comparisons.

**Role of the convex ERM analysis.** The convex ERM setting provides a clean reference case for the inverse view of DPA. In that regime, preservation can be expressed through the optimality conditions of the trained candidates, and the associated cost-minimization problem remains combinatorial. We use this analysis to motivate a local proposal model combined with direct verification, rather than to suggest that the surrogate alone characterizes general training pipelines. In the pipeline setting, the response model guides candidate updates, while the original train-and-evaluate protocol determines which designs are retained.

**Scope and limitations.** DPA is protocol-relative: its guarantees apply only to the specified candidate pool, scenarios, margins, costs, and decision convention. Our fixed-feature convex ERM setting isolates decision preservation from representation learning and stochastic training effects, while allowing accepted designs to be certified by explicit retraining or reevaluation. VERDICT trades acquisition or evaluation savings for search cost, and as a local search method can fail when the response model mispredicts margins, rounding breaks feasibility, or the winner map depends on broad coverage rather than near-active comparisons. Its certificates should not be interpreted as robustness, fairness, or deployment guarantees outside the declared scenario set. Extending DPA to end-to-end nonconvex models, highly stochastic pipelines, multi-quality acquisition, and expensive-verification settings remains future work.

## 7 Conclusion

We introduced decision-preserving acquisition, an inverse formulation that asks which acquired evidence is sufficient for a fixed model-selection protocol to reproduce an observed winner map with prescribed margins. VERDICT implements a verified local-search procedure that proposes lower-cost designs and accepts them only after rerunning the fixed protocol. Empirically, the direct train-and-verify experiments show that decision-preserving designs can be much smaller than subsets found by generic coverage-, difficulty-, or influence-based heuristics. The results suggest that model-selection outcomes can have compact but protocol-dependent evidence support, which can be interpreted when verification is feasible.

## References


Lucien Birgé et al. Robust tests for model selection. *From probability to statistics and back: high-dimensional models and processes–A Festschrift in honor of Jon A. Wellner*, pp. 47–64, 2013.

Anshuman Chhabra, Bo Li, Jian Chen, Prasant Mohapatra, and Hongfu Liu. Outlier gradient analysis: Efficiently identifying detrimental training samples for deep learning models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=v77ZMzbsBA.

Kenneth L Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):1–30, 2010.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.

Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems*, 30, 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Clemens Heuberger. Inverse combinatorial optimization: A survey on problems, methods, and results. *Journal of combinatorial optimization*, 8(3):329–361, 2004.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133–142, 2002.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021.

Narine Kokhlikyan, Kamalika Chaudhuri, and Saeed Mahloujifar. Z0-inf: Zeroth order approximation for data influence, 2025. URL https://arxiv.org/abs/2510.11832.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ken Lang. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, pp. 331–339. Elsevier, 1995.

Zhe Li, Wei Zhao, Yige Li, and Jun Sun. Do influence functions work on large language models? In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 14367–14382, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.775. URL https://aclanthology.org/2025.findings-emnlp.775/.

Youyuan Long, Tolga Ok, Pedro Zattoni Scroccaro, and Peyman Mohajerin Esfahani. Scalable kernel inverse optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Mktgayam7U.

Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

Wei Ma, Mike Papadakis, Anestis Tsakmalis, Maxime Cordy, and Yves Le Traon. Test selection for deep learning systems. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 30(2):1–22, 2021.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

Bruno Kacper Mlodozeniec, Runa Eschenhagen, Juhan Bae, Alexander Immer, David Krueger, and Richard E. Turner. Influence functions for scalable data attribution in diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=esYrEndGsr.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020.

Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

Ittai Rubinstein and Samuel B. Hopkins. Rescaled influence functions: Accurate data attribution in high dimension. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=0T9lN3m3pc.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Gayathri Saranathan, Cong Xu, Mahammad Parwez Alam, Tarun Kumar, Martin Foltin, Soon Yee Wong, and Suparna Bhattacharya. Sublime: Subset selection via rank correlation prediction for data-efficient llm evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30572–30593, 2025.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35: 19523–19536, 2022.

Ki Hyun Tae and Steven Euijong Whang. Slice tuner: A selective data acquisition framework for accurate and fair machine learning models. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 1771–1783, 2021.

Zhuozhuo Tu, Cheng Chen, and Yuxuan Du. RRInf: Efficient influence function estimation via ridge regression for large language models and text-to-image diffusion models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18494–18507, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.933. URL https://aclanthology.org/2025.emnlp-main.933/.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601, 2024.

Shaobo Wang, Xiangqi Jin, Ziming Wang, Jize Wang, Jiajun Zhang, Kaixin Li, Zichen Wen, Zhong Li, Conghui He, Xuming Hu, and Linfeng Zhang. Data whisperer: Efficient data selection for task-specific LLM fine-tuning via few-shot in-context learning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23287–23305, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1135. URL https://aclanthology.org/2025.acl-long.1135/.

Dennis Wei, Inkit Padhi, Soumya Ghosh, Amit Dhurandhar, Karthikeyan Natesan Ramamurthy, and Maria Chang. Final-model-only data attribution with a unifying view of gradient-based methods. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=rccgEdFTlH.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1658–1665, 2019.

Peiwen Yuan, Yueqi Zhang, Shaoxiong Feng, Yiwei Li, Xinglin Wang, Jiayi Shi, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. Beyond one-size-fits-all: Tailored benchmarks for efficient evaluation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15591–15615, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.759. URL https://aclanthology.org/2025.acl-long.759/.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Xiaoyu Zhang, Jorge Piazentin Ono, Huan Song, Liang Gou, Kwan-Liu Ma, and Liu Ren. Sliceteller: A data slice-driven approach for machine learning model validation. *IEEE transactions on visualization and computer graphics*, 29(1):842–852, 2022.

# A Reproducibility Statement

We provide the information needed to reproduce the experiments in the paper, including the hardware and software environment, dataset splits, preprocessing steps, hyperparameters, implementation choices for VERDICT and the baselines, and the evaluation protocol used across settings. We report results over multiple random seeds where applicable using a consistent aggregation procedure, and we describe the metrics and verification checks used to generate each table and figure. We will release the full code package upon paper acceptance, which includes scripts for preprocessing, training and evaluation, configuration files for our method and all baselines, and instructions for data preparation when external resources are required.

# B Impact Statement

This work develops an auditing-oriented tool: it can identify low-cost training evidence sufficient to reproduce an observed model-selection map under explicit margins and a specified decision convention. This can support transparency in data curation and quantify how much evidence is needed to support a deployed selection decision. However, the same capability can be misused to rationalize predetermined outcomes by cherry-picking scenarios, margins, candidate pools, or cost models to favor a desired selection. We mitigate this risk by requiring an explicit, documented decision convention (including aggregation and tie-breaking) and verifying feasibility under that convention when retraining is feasible; recommending that scenario sets, pools, and margin protocols be pre-specified and reported; and encouraging sensitivity analyses and the use of independent holdout scenarios not used in the inverse-design loop.

# C Problem Setup Details: Relaxations, Randomness, and Margins

## C.1 Notation and Protocol Conventions

Tables 4–8 summarize the main notation used throughout the paper. We use calligraphic letters for finite sets and roman or italic letters for their cardinalities when clarification is helpful. All decision-preservation statements are relative to the fixed candidate set, scenarios, margins, costs, and decision convention specified by the protocol.

Table 4: Notation for protocol objects.

| Symbol | Meaning |
|---|---|
| $\mathcal{P} = \{z_i\}_{i=1}^{N}$ | Candidate pool of training examples |
| $z_i$ | Training example indexed by $i$ |
| $N$ | Number of examples in the candidate pool |
| $\mathcal{M}$ | Finite set of candidate models or hyperparameter configurations |
| $M = \lvert\mathcal{M}\rvert$ | Number of candidate models |
| $\mathcal{T}$ | Finite set of validation scenarios |
| $T = \lvert\mathcal{T}\rvert$ | Number of validation scenarios |
| $t \in \mathcal{T}$ | Scenario index, e.g., domain, slice, class group, or robustness condition |
| $m \in \mathcal{M}$ | Candidate model or configuration index |
| $F_{m,t}$ | Validation functional for model $m$ on scenario $t$; lower is better |
| $F_{m,t}(\alpha)$ | Validation loss after training model $m$ under acquisition design $\alpha$ |
| $D_{\mathrm{val}}(t)$ | Winner selected by the baseline protocol on scenario $t$ |
| $\omega$ | Randomness used by the training and/or evaluation pipeline |
| $\omega_0$ | Fixed seed used under the fixed-seed decision convention |
| $\overline{F}_{m,t}(\alpha)$ | Aggregated validation loss over a specified finite seed set |

## C.2 Operational Semantics of Relaxations

The inverse-design problem is combinatorial, so we use a convex relaxation to search over designs. Since the pipeline consumes discrete acquisitions, we specify how a relaxed $\alpha$ is interpreted during search and how

Table 5: Notation for acquisition designs and costs.

| Symbol | Meaning |
|---|---|
| $\mathcal{Q} = \{q^{(1)}, \ldots, q^{(K)}\}$ | Available quality levels for acquired evidence |
| $K$ | Number of quality levels |
| $\alpha \in \{0,1\}^{N \times K}$ | Integral acquisition design |
| $\alpha_{i,k}$ | Indicator that example $z_i$ is acquired at quality level $q^{(k)}$ |
| $\alpha_{i,k} \in [0,1]$ | Relaxed acquisition weight used in surrogate optimization |
| $\widehat{\alpha}$ | Rounded integral design evaluated by the verification protocol |
| $\lvert\alpha\rvert$ | Acquisition cardinality, $\sum_{i,k} \alpha_{i,k}$ |
| $c_{i,k}$ | Cost of acquiring example $z_i$ at quality level $q^{(k)}$ |
| $c_k$ | Quality-dependent acquisition cost |
| $c_i^{\text{fix}}$ | Point-specific acquisition overhead |
| $C(\alpha)$ | Total acquisition cost, $\sum_{i,k} \alpha_{i,k} c_{i,k}$ |
| $B$ | Acquisition budget |
| $n_{\min}$ | Minimum acquisition size required for a valid protocol run |
| $\mathcal{S}_B(n_{\min})$ | Feasible set of integral designs satisfying budget and size constraints |
| $\mathcal{S}_B^{\text{rel}}(n_{\min})$ | Relaxed feasible set used by surrogate problems |

Table 6: Notation for training and decision preservation.

| Symbol | Meaning |
|---|---|
| $w_m(\alpha)$ | Parameters returned by the training pipeline for model $m$ under design $\alpha$ |
| $w_m^\star(\alpha)$ | Exact optimizer for model $m$ in the convex ERM regime |
| $w_{\text{pipe}}(\alpha; \omega)$ | Pipeline output under design $\alpha$ and randomness $\omega$ |
| $\gamma_t$ | Required stability margin for scenario $t$ |
| $g_{m,t}(\alpha)$ | Decision margin against competitor $m$ on scenario $t$ |
| $\overline{g}_{m,t}(\alpha)$ | Aggregated decision margin under the aggregated seed convention |
| $\text{gap}_t$ | Full-data winner gap on scenario $t$ |
| $\eta$ | Fraction used to set margins, e.g., $\gamma_t = \eta\, \text{gap}_t$ |

Table 7: Notation for convex ERM analysis.

| Symbol | Meaning |
|---|---|
| $\ell_m(w; z_i, q^{(k)})$ | Per-example training loss for model $m$ at quality level $q^{(k)}$ |
| $R_m(w)$ | Regularizer for model $m$ |
| $L_m(w; \alpha)$ | Acquisition-weighted ERM objective for model $m$ |
| $\mathcal{W}$ | Parameter domain |
| $N_{\mathcal{W}}(w)$ | Normal cone of $\mathcal{W}$ at $w$ |
| $\mu$ | Strong convexity parameter |
| $H_m^\sharp$ | Hessian $\nabla_w^2 L_m(w_m^\sharp; \alpha^\sharp)$ at a reference design |
| $\alpha^\sharp$ | Reference design used to form the local response approximation |
| $w_m^\sharp$ | Trained parameters at the reference design $\alpha^\sharp$ |

Table 8: Notation for VERDICT local search.

| Symbol | Meaning |
|---|---|
| $\Delta\alpha$ | Perturbation to the acquisition design |
| $\mathrm{Resp}^\sharp$ | Local parameter-response operator mapping $\Delta\alpha$ to predicted parameter change |
| $\mathrm{DR}^{\sharp,\tau}$ | Decision-response operator mapping $\Delta\alpha$ to predicted margin change |
| $\tau$ | Quasi-active margin threshold |
| $\mathcal{J}^\tau$ | Quasi-active set of near-binding winner-versus-competitor constraints |
| $s^{\sharp,\tau}$ | Vector of current quasi-active margins at $\alpha^\sharp$ |
| $\xi$ | Slack variable in the local surrogate problem |
| $\lambda_{\mathrm{lin}}$ | Penalty weight on surrogate slack |
| $r_{\mathrm{tr}}$ | Trust-region radius |
| $r_{\mathrm{tr},s}$ | Trust-region radius at VERDICT iteration $s$ |
| $\beta^+, \beta^-$ | Trust-region expansion and shrinkage factors |
| $u$ | Dual multipliers for linearized margin constraints |
| $s_{i,k}$ | Adjoint score used to rank acquisition variables for working-set expansion |
| $\alpha^{(s)}$ | Verified design at VERDICT iteration $s$ |
| $\widetilde{\alpha}$ | Relaxed candidate proposed by the surrogate solver |
| $\widehat{\alpha}$ | Rounded candidate passed to retrain-and-check verification |

constraints are enforced at verification time. We also consider the convex relaxation

$$\begin{aligned} \alpha &\in [0,1]^{N\times K}, \\ \sum_{k=1}^{K} \alpha_{i,k} &\le 1 \quad \forall i, \\ \sum_{i,k} \alpha_{i,k} &\ge n_{\min}, \\ C(\alpha) &\le B, \end{aligned}$$

and denote the relaxed feasible set by $\mathcal{S}_B^{\mathrm{rel}}(n_{\min})$.

**Definition C.1** (Pipeline-world semantics for relaxed designs)**.** In convex ERM, $\alpha_{i,k} \in [0,1]$ acts as a weight in the training objective. In the pipeline world, we use two interpretations for a relaxed design $\alpha \in \mathcal{S}_B^{\mathrm{rel}}(n_{\min})$:

1. **Weighted training.** The pipeline consumes per-example weights and scales per-example gradients by $\alpha_{i,k}$.
2. **Stochastic acquisition.** $\alpha_{i,k}$ is an inclusion probability for pair $(i,k)$, implemented with dependent rounding to enforce per-$i$ constraints; the pipeline trains on the realized binary design.

If the implementation supports only integral designs, we solve relaxed surrogate subproblems to propose $\alpha$ and then round to a binary design $\alpha^b \in \mathcal{S}_B(n_{\min})$. All verification checks are performed on $\alpha^b$, and hard constraints such as $C(\alpha) \le B$ are enforced on $\alpha^b$ when rounding is used.

*Remark* C.2 (Cost semantics under relaxation). Under stochastic acquisition, $C(\alpha) = \sum_{i,k} \alpha_{i,k} c_{i,k}$ is the expected acquisition cost. Under weighted training, $\alpha$ is used to generate a candidate, while operational cost and hard budget checks are enforced on the rounded design $\alpha^b$.

### C.3 Randomness Conventions and Verification

When training is stochastic, preserving a selection requires fixing how randomness is handled. We formalize training with an explicit randomness input and verify candidates under the same convention used to define the winners.

**Definition C.3** (Training pipeline protocol)**.** For a seed/protocol specification $\omega$, define

$$\widehat{w}_m(\alpha;\omega) := \texttt{Train}(m,\alpha;\omega),$$

$$w_{\text{pipe}}(\alpha;\omega) := (\widehat{w}_m(\alpha;\omega))_{m\in\mathcal{M}}.$$

We observe a winner $D_{\text{val}}(t) \in \mathcal{M}$ for each scenario $t$ under a fixed decision convention, including deterministic tie-breaking.

**Definition C.4** (Observed decision map and randomness convention)**.** Fix a baseline design $\alpha^{(0)}$. The observed selection map $D_{\text{val}} : \mathcal{T} \to \mathcal{M}$ is defined under one of two conventions:

1. **Fixed-seed convention.** There exists a seed $\omega_0$ such that

$$D_{\text{val}}(t) \in \arg\min_{m\in\mathcal{M}} F_{m,t}\big(\widehat{w}_m(\alpha^{(0)};\omega_0)\big),$$

   with a deterministic tie-break rule if needed.

2. **Aggregated convention.** Fix a finite seed set $\Omega$ and an aggregator Agg (such as the mean or median). Define

$$\bar{F}_{m,t}(\alpha) := \text{Agg}_{\omega\in\Omega} F_{m,t}\big(\widehat{w}_m(\alpha;\omega)\big),$$

   and set

$$D_{\text{val}}(t) \in \arg\min_{m\in\mathcal{M}} \bar{F}_{m,t}(\alpha^{(0)}),$$

   with a deterministic tie-break rule if needed.

All verification uses the same convention (including aggregation and tie-breaking) that defines $D_{\text{val}}$.

**Margin Constraints.** Margins convert winner preservation into a stability requirement by enforcing that the winner stays ahead of competitors by at least $\gamma_t$. Fix margin requirements $\gamma_t \geq 0$. For any design $\alpha$ and competitor $m \neq D_{\text{val}}(t)$, define the margin constraint

$$g_{m,t}(\alpha) := F_{m,t}(\alpha) - F_{D_{\text{val}}(t),t}(\alpha) - \gamma_t.$$

Under the aggregated convention, define

$$\bar{g}_{m,t}(\alpha) := \bar{F}_{m,t}(\alpha) - \bar{F}_{D_{\text{val}}(t),t}(\alpha) - \gamma_t.$$

We call $\alpha$ *decision-feasible* if it satisfies the corresponding margin system under the convention defining $D_{\text{val}}$.

### C.4 Margin Selection Protocol

Margins should be chosen so the inverse problem is feasible. Unless otherwise stated, we set margins using the following protocol. The parameter $\eta$ scales the baseline gaps into target margins.

*Remark* C.5 (Margin protocol from a baseline). Fix $\eta \in (0,1)$ and a baseline design $\alpha^{(0)}$ that produced $D_{\mathrm{val}}$ under the chosen convention.

- **Fixed-seed convention:** define

$$\mathrm{gap}_t(\alpha^{(0)}) := \min_{m \neq D_{\mathrm{val}}(t)} \Big( F_{m,t}\big(\widehat{w}_m(\alpha^{(0)};\omega_0)\big) - F_{D_{\mathrm{val}}(t),t}\big(\widehat{w}_{D_{\mathrm{val}}(t)}(\alpha^{(0)};\omega_0)\big) \Big),$$

  and set $\gamma_t := \eta \max\{\mathrm{gap}_t(\alpha^{(0)}), 0\}$.

- **Aggregated convention:** define

$$\mathrm{gap}_t(\alpha^{(0)}) := \min_{m \neq D_{\mathrm{val}}(t)} \Big( \bar{F}_{m,t}(\alpha^{(0)}) - \bar{F}_{D_{\mathrm{val}}(t),t}(\alpha^{(0)}) \Big),$$

  and set $\gamma_t := \eta \max\{\mathrm{gap}_t(\alpha^{(0)}), 0\}$.

In experiments, we fix $\eta$ (equivalently, $\gamma_t$) as part of the decision-convention specification.

# D Proofs

## D.1 Proof of Proposition 4.4 (Realizability ⇔ inverse KKT feasibility)

In the convex ERM regime, each model solves

$$w_m^\star(\boldsymbol{\alpha}) \in \arg\min_{w \in \mathcal{W}} L_m(w;\boldsymbol{\alpha}),$$

and margins are evaluated at $\boldsymbol{w}^\star(\boldsymbol{\alpha}) = (w_m^\star(\boldsymbol{\alpha}))_{m \in \mathcal{M}}$. We show that the observed selection map is realizable under budget $B$ if and only if the inverse system in Definition 4.3 is feasible.

**(⇒) Realizability implies feasibility.** Assume realizability under budget $B$: there exists $\boldsymbol{\alpha} \in \mathcal{S}_B^{\mathrm{rel}}(n_{\min})$ such that

$$g_{m,t}(\boldsymbol{w}^\star(\boldsymbol{\alpha})) \geq 0 \qquad \forall t \in \mathcal{T},\ \forall m \neq D_{\mathrm{val}}(t).$$

Let $\boldsymbol{w} := \boldsymbol{w}^\star(\boldsymbol{\alpha})$. By optimality of each $w_m^\star(\boldsymbol{\alpha})$ and the KKT/variational characterization for convex constraints,

$$0 \in \nabla_w L_m(w_m^\star(\boldsymbol{\alpha});\boldsymbol{\alpha}) + N_{\mathcal{W}}(w_m^\star(\boldsymbol{\alpha})), \qquad \forall m \in \mathcal{M}.$$

Thus $(\boldsymbol{\alpha}, \boldsymbol{w})$ satisfies equation 2–equation 4, hence the inverse system is feasible.

**(⇐) Feasibility implies realizability.** Conversely, assume the inverse system is feasible: there exist $(\boldsymbol{\alpha}, \boldsymbol{w})$ such that

$$\boldsymbol{\alpha} \in \mathcal{S}_B^{\mathrm{rel}}(n_{\min}), \qquad 0 \in \nabla_w L_m(w_m;\boldsymbol{\alpha}) + N_{\mathcal{W}}(w_m)\ \ \forall m, \qquad g_{m,t}(\boldsymbol{w}) \geq 0\ \ \forall (m,t).$$

For closed convex $\mathcal{W}$ and differentiable $L_m(\cdot;\boldsymbol{\alpha})$, the inclusion $0 \in \nabla_w L_m(w_m;\boldsymbol{\alpha}) + N_{\mathcal{W}}(w_m)$ is necessary and sufficient for $w_m$ to be a minimizer of $L_m(\cdot;\boldsymbol{\alpha})$ over $\mathcal{W}$. Under Assumption 4.2, the minimizer is unique, so $w_m = w_m^\star(\boldsymbol{\alpha})$ for all $m$ and $\boldsymbol{w} = \boldsymbol{w}^\star(\boldsymbol{\alpha})$. Substituting into the margin constraints yields

$$g_{m,t}(\boldsymbol{w}^\star(\boldsymbol{\alpha})) \geq 0 \qquad \forall t \in \mathcal{T},\ \forall m \neq D_{\mathrm{val}}(t),$$

so the decision map is realizable under budget $B$.

□

### D.2 Proof of Theorem D.1 (NP-Hardness of the Idealized DPA Objective)

**Theorem D.1** (NP-Hardness of the Idealized DPA Objective)**.** *Computing the lowest cost of a decision-preserving design is NP-hard, even under the following restrictions:*

1. *a single quality level ($K = 1$) with per-point costs $c_i > 0$,*
2. *two models ($M = 2$) with $\mathcal{W} = \mathbb{R}^p$ and strongly convex quadratic training objectives,*
3. *linear validation functionals and $T = p$ scenarios indexed by coordinates, with a fixed winner $D_{\text{val}}(t) \equiv 1$ for all $t \in \{1, \dots, p\}$.*

We give a polynomial-time reduction from Minimum-Cost Set Cover. An instance consists of a universe $U = \{1, \dots, p\}$, subsets $\{S_i\}_{i=1}^N$ with $S_i \subseteq U$, and costs $\{c_i\}_{i=1}^N$ with $c_i > 0$. The goal is to select a minimum-cost subcollection whose union covers $U$.

**Preprocessing (ensuring feasibility).** If $\bigcup_{i=1}^N S_i \neq U$, add one extra set $S_{N+1} = U$ with cost $c_{N+1} := 1 + \sum_{i=1}^N c_i$. This makes the instance feasible and does not change the optimum among the original sets unless coverage was impossible. Relabel so the collection still has size $N$ and covers $U$.

**Constructing the inverse-design instance.** We build an instance of decision-preserving acquisition with:

- One quality level ($K = 1$), so the design is $\alpha \in \{0, 1\}^N$ (equivalently, $\boldsymbol{\alpha} \in \{0, 1\}^{N \times 1}$).
- Two models ($M = 2$) and parameter domain $\mathcal{W} = \mathbb{R}^p$.
- $T = p$ scenarios indexed by coordinates $t \in \{1, \dots, p\}$, and a fixed observed winner $D_{\text{val}}(t) \equiv 1$ for all $t$.

**Costs.** Set the per-point costs to match set-cover costs: $c_{i,1} := c_i$, hence

$$C(\alpha) = \sum_{i=1}^N c_i \alpha_i.$$

Set $n_{\min} = 1$.

**Training objectives (strongly convex quadratics).** Fix $\mu > 0$ and define ERM objectives on $\mathbb{R}^p$:

$$L_1(w; \alpha) := \frac{\mu}{2}\|w\|_2^2, \qquad L_2(w; \alpha) := \frac{\mu}{2}\|w\|_2^2 - \frac{1}{N}\sum_{i=1}^N \alpha_i \, v_i^\top w,$$

where $v_i \in \{0, 1\}^p$ encodes membership:

$$(v_i)_t := \mathbb{1}\{t \in S_i\}.$$

Both objectives are $\mu$-strongly convex quadratics.

**Closed-form minimizers.** Model 1 has the unique minimizer

$$w_1^\star(\alpha) = 0.$$

For model 2,

$$\nabla_w L_2(w; \alpha) = \mu w - \frac{1}{N}\sum_{i=1}^N \alpha_i v_i,$$

so the unique minimizer is

$$w_2^\star(\alpha) = \frac{1}{\mu N}\sum_{i=1}^N \alpha_i v_i.$$

In particular, for each coordinate $t$,

$$(w_2^\star(\alpha))_t = \frac{1}{\mu N}\sum_{i=1}^{N}\alpha_i(v_i)_t = \frac{1}{\mu N}\sum_{i:\, t\in S_i}\alpha_i.$$

**Validation (linear) and margins.** For each scenario $t \in \{1,\dots,p\}$ define

$$F_{1,t}(w) := 0, \qquad F_{2,t}(w) := w_t.$$

Set

$$\gamma_t := \frac{1}{\mu N} \quad \text{for all } t.$$

Since $D_{\text{val}}(t) \equiv 1$, the margin constraint for competitor $m = 2$ is

$$g_{2,t}(\boldsymbol{w}^\star(\alpha)) = F_{2,t}(w_2^\star(\alpha)) - F_{1,t}(w_1^\star(\alpha)) - \gamma_t = (w_2^\star(\alpha))_t - \frac{1}{\mu N} \geq 0,$$

which is equivalent to

$$\sum_{i:\, t\in S_i}\alpha_i \geq 1.$$

**Feasibility $\iff$ set cover.** Thus, all margin constraints hold if and only if every element $t \in U$ belongs to at least one selected set $\{S_i : \alpha_i = 1\}$, which is exactly the set cover condition.

**Objective correspondence.** Minimizing $C(\alpha) = \sum_i c_i\alpha_i$ over feasible designs matches the minimum-cost set cover objective.

**Polynomial size and time.** The reduction constructs $\{v_i\}_{i=1}^N$, two quadratic objectives, and $p$ linear validation functionals. Its size is polynomial in the set-cover input size, and the mapping is computable in polynomial time.

**Conclusion.** Computing $C^\star(D_{\text{val}})$ is NP-hard. Equivalently, deciding whether $C^\star(D_{\text{val}}) \leq L$ for a given threshold $L$ is NP-hard. This holds even with $K = 1$, $M = 2$, strongly convex quadratic training, and linear validation.

□

## E Additional details for Algorithm 1

This appendix records options referenced by Algorithm 1 and Section 4. All correctness statements concern accepted designs and follow from the acceptance checks.

### E.1 Violation functionals for decision feasibility

**Definition E.1** (Pipeline violation functional)**.** Let $g_{m,t}(\boldsymbol{w}^\star(\alpha))$ denote the (pipeline) margin for competitor $m$ under scenario $t$, so decision feasibility means $g_{m,t}(\boldsymbol{w}^\star(\alpha)) \geq 0$ for all relevant $(m,t)$. Define the positive-part operator $[x]_+ := \max\{x, 0\}$.

**Fixed-seed violation.** Under the fixed-seed convention, define

$$V_{\text{pipe}}^{\text{fix}}(\alpha) := \sum_{t\in\mathcal{T}}\sum_{m\neq D_{\text{val}}(t)}\big[-g_{m,t}(\boldsymbol{w}^\star(\alpha))\big]_+.$$

**Aggregated violation.** Under the aggregated convention, let $\bar{g}_{m,t}(\alpha)$ denote the aggregated margin (Definition C.4) and define

$$V_{\text{pipe}}^{\text{agg}}(\alpha) := \sum_{t\in\mathcal{T}}\sum_{m\neq D_{\text{val}}(t)}\big[-\bar{g}_{m,t}(\alpha)\big]_+.$$

In either case, we write $V_{\text{pipe}}(\alpha)$ when the convention is clear from context.

### E.2 Proxy margins under the aggregated convention

When the decision convention aggregates over randomness (Definition C.4), the verified constraints depend on $\bar{g}_{m,t}(\alpha)$ and can be expensive to evaluate. For quasi-active set construction and local models, one may use a proxy $\tilde{g}^{(s)}_{m,t}$ that is differentiable and cheaper than $\bar{g}_{m,t}$. Examples include using a smaller seed batch, replacing the aggregator by a smooth approximation, or using an ERM-based surrogate that correlates with pipeline scores. Only the acceptance check uses $\bar{g}_{m,t}$.

### E.3 Optional merit function and sufficient decrease

If one wants a scalar progress test in addition to feasibility, define a pipeline violation $V_{\text{pipe}}(\alpha)$ as in Definition E.1 (fixed-seed or aggregated). Given a penalty weight $\lambda_{\text{pen}} \geq 0$, define

$$\mathcal{M}(\alpha) := C(\alpha) + \lambda_{\text{pen}} V_{\text{pipe}}(\alpha).$$

A sufficient decrease rule accepts $\alpha^b$ only if it is verified decision-feasible and

$$\mathcal{M}(\alpha^b) \leq \mathcal{M}(\alpha^{(s)}) - \kappa\, r_{\text{tr},s},$$

where $r_{\text{tr},s}$ is the trust-region radius at iteration $s$ (Algorithm 1), and $\kappa \geq 0$ is a chosen constant.

## F Scenario Construction Details

For each dataset, we define validation scenarios before acquisition. Table 9 summarizes how the validation scenarios are instantiated in each modality. For tabular datasets, we use named slices derived from semantically meaningful covariates in the original data. Adult uses sex, age, and education slices; Credit uses sex, age, credit-limit, and repayment-history slices; and Bank uses housing-loan status, age, and previous-contact slices. These scenarios test whether the selected training evidence preserves the same model-selection conclusion across interpretable subpopulations rather than only on the aggregate validation set. For vision datasets, scenarios are constructed from validation labels because the cached representation contains frozen image features rather than additional metadata. We use classwise scenarios for CIFAR10 and STL10, and fixed class groups for larger-label datasets such as CIFAR100 and Oxford Pets so that each scenario has adequate validation support. For text datasets, scenarios correspond to topic labels: AG News uses its four topic classes, while 20 Newsgroups uses one scenario per newsgroup class in the main experiment. Across all datasets, the scenarios are fixed before acquisition and are never optimized; acquisition changes only the training subset used to retrain the candidate convex ERM models.

Table 9: Scenario construction for the convex ERM acquisition experiment. Each scenario is a fixed validation subset; acquisition changes only the training evidence.

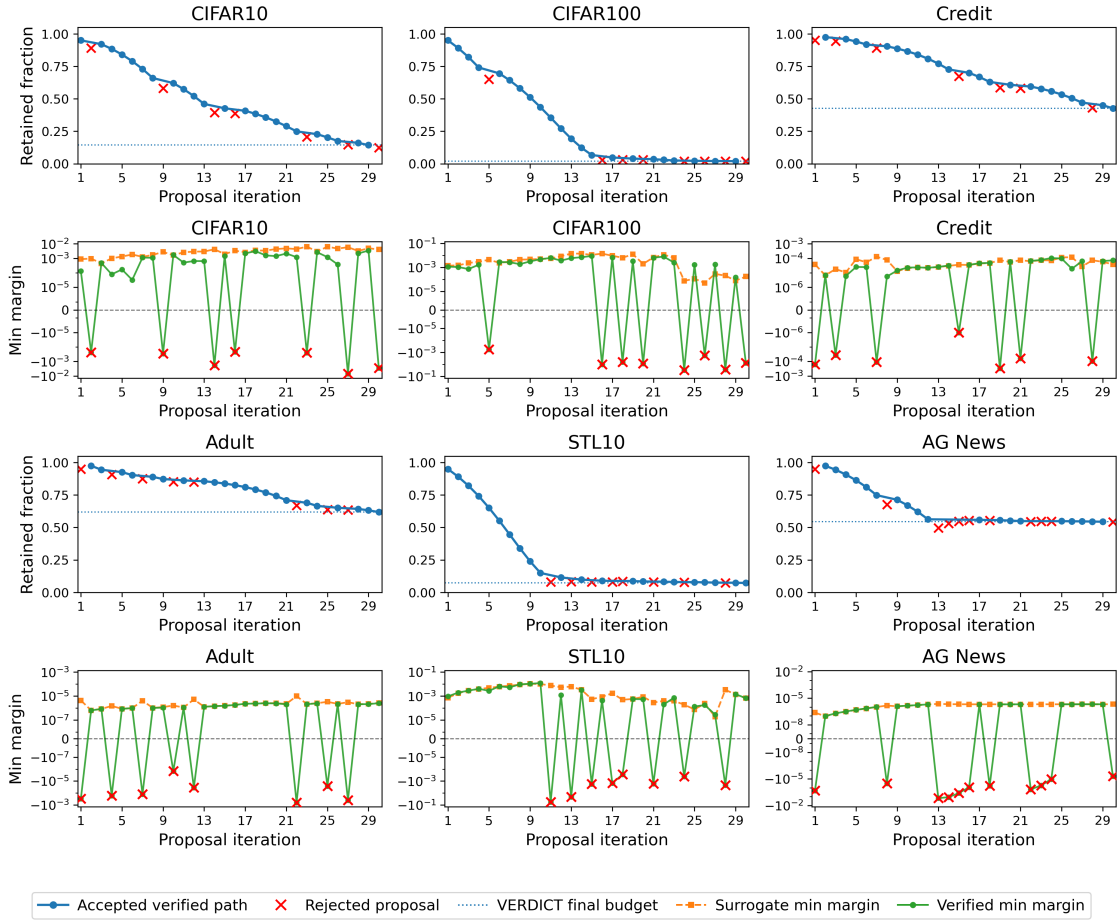
| Task | Frozen representation | Dataset | Scenarios | $T$ |
|---|---|---|---|---|
| Tabular | One-hot + standardized numeric | Adult | Sex, age, education | 6 |
| | | Credit | Sex, age, credit, repayment | 6 |
| | | Bank | Housing, age, previous contact | 5 |
| Vision | ResNet50 | CIFAR10 | Classwise | 10 |
| | | CIFAR100 | Fixed class groups | 10 |
| | | STL10 | Classwise | 10 |
| | | Oxford Pets | Breed/class groups | 10 |
| Text | TF–IDF + SVD | AG News | Topic classes | 4 |
| | | 20 Newsgroups | Newsgroup classes | 6 |

# G Verified Search in Direct Training Acquisition

## G.1 Verified local search trajectories

**Additional verified search trajectories.** Figure 3 reports VERDICT trajectories for six additional datasets. The same pattern observed in the main text persists across modalities: the surrogate-predicted margins are usually positive and often track the verified margins along accepted updates, but rejected proposals can have positive surrogate margins while their retraining-verified margins become negative. This confirms that the local decision-response model is useful for generating candidate deletions, but does not itself certify decision preservation. The trajectories also show that VERDICT's gains are not due to a single search behavior. CIFAR10 and Credit exhibit gradual verified compression with intermittent rejected proposals, indicating that the algorithm can continue reducing the subset after local surrogate failures. CIFAR100 and STL10 show much sharper compression, reaching very small verified fractions. In contrast, Adult and AG News show flatter trajectories after moderate compression, suggesting regimes where additional deletions are frequently rejected or where the decision map is already partly governed by coarse balance. Overall, the appendix trajectories support the same conclusion as the main figure: VERDICT converts an imperfect local response model into a safe acquisition procedure by accepting only retraining-verified proposals.

Figure 3: Verified local search trajectories for VERDICT for the remaining 6 datasets.




## G.2 Ablation of Margin Sensitivity

We sweep the margin multiplier $\eta$ for VERDICT, varying the constraint from winner preservation only ($\eta = 0$) to preservation of a larger fraction of the full-data winner gap. Figure 4 reports the smallest retraining-verified

retained fraction found under each setting. Several datasets, such as Adult, Bank, and Credit, show the expected increase in retained fraction as $\eta$ grows. Others, such as AG News and STL10, are relatively flat, suggesting that the selected subsets preserve the decision map with slack across multiple margin levels. Some datasets exhibit non-monotone found costs. This non-monotonicity is a property of the finite-budget search: increasing $\eta$ makes the decision-preservation constraints stricter and can only remove feasible subsets. However, VERDICT does not enumerate all feasible subsets; it follows a local proposal trajectory. Changing $\eta$ changes which constraints are active, which deletions are proposed, and which candidates pass verification, so the retained fraction found by the search need not vary monotonically.

Figure 4: Sensitivity of VERDICT to the preservation margin with $\eta \in \{0, 0.25, 0.5, 0.75\}$

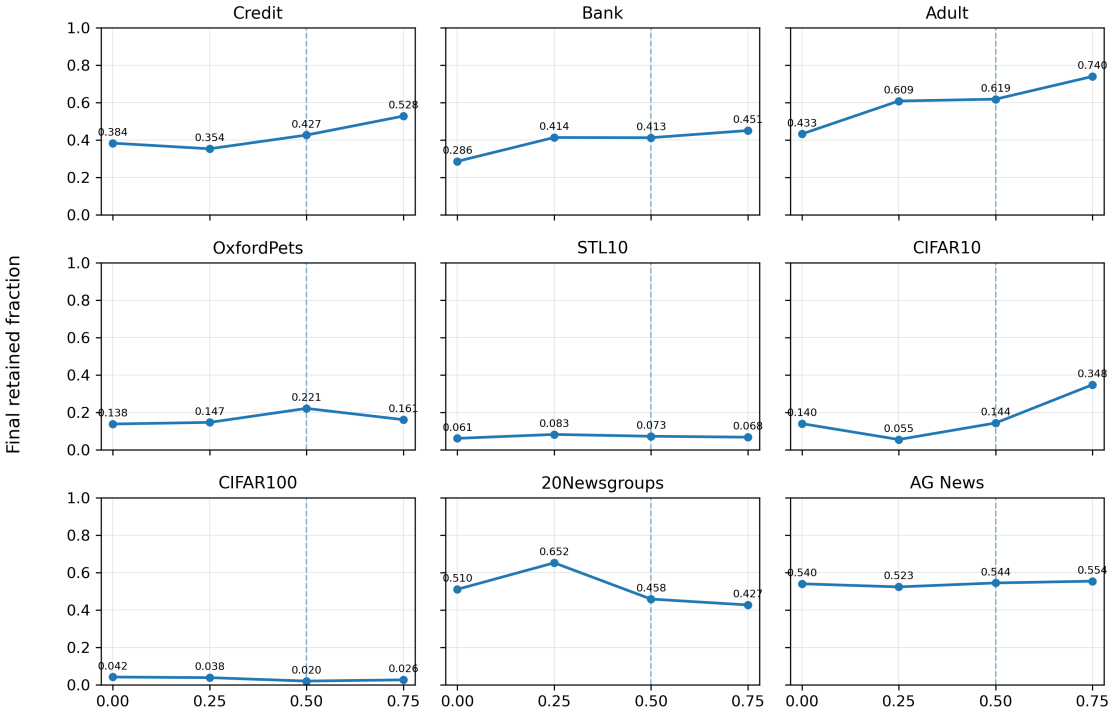



Table 10: Margin-sensitivity diagnostics for VERDICT. A/R reports accepted/rejected proposals; margin is the final verified minimum margin. All returned subsets are retraining-verified; small positive margins indicate that the search often stops near the feasibility boundary.

| Dataset | $\eta = 0$ | | $\eta = 0.25$ | | $\eta = 0.5$ | | $\eta = 0.75$ | |
|---|---|---|---|---|---|---|---|---|
| | A/R | Margin | A/R | Margin | A/R | Margin | A/R | Margin |
| 20Newsgroups | 22/8 | $1.10\times10^{-6}$ | 22/8 | $7.44\times10^{-7}$ | 22/8 | $6.58\times10^{-7}$ | 22/8 | $8.65\times10^{-8}$ |
| Adult | 21/9 | $8.11\times10^{-6}$ | 22/8 | $4.66\times10^{-6}$ | 22/8 | $2.54\times10^{-6}$ | 22/8 | $4.61\times10^{-6}$ |
| AG News | 20/10 | $2.21\times10^{-6}$ | 19/11 | $2.25\times10^{-6}$ | 19/11 | $2.09\times10^{-6}$ | 17/12 | $2.01\times10^{-6}$ |
| Bank | 21/9 | $9.72\times10^{-7}$ | 21/9 | $1.99\times10^{-7}$ | 21/9 | $2.50\times10^{-8}$ | 20/10 | $2.50\times10^{-7}$ |
| CIFAR10 | 24/6 | $6.19\times10^{-3}$ | 23/7 | $6.07\times10^{-4}$ | 23/7 | $3.43\times10^{-3}$ | 23/7 | $6.48\times10^{-4}$ |
| CIFAR100 | 24/6 | $1.00\times10^{-2}$ | 22/8 | $9.41\times10^{-5}$ | 22/8 | $1.44\times10^{-4}$ | 23/7 | $1.54\times10^{-3}$ |
| Credit | 23/7 | $7.83\times10^{-5}$ | 23/7 | $4.14\times10^{-5}$ | 23/7 | $7.27\times10^{-5}$ | 23/7 | $8.44\times10^{-6}$ |
| OxfordPets | 21/9 | $1.57\times10^{-4}$ | 21/9 | $2.18\times10^{-4}$ | 21/9 | $4.96\times10^{-4}$ | 16/11 | $1.12\times10^{-4}$ |
| STL10 | 20/10 | $3.56\times10^{-4}$ | 21/9 | $3.93\times10^{-5}$ | 22/8 | $6.70\times10^{-4}$ | 18/12 | $3.32\times10^{-4}$ |

### G.3 Ablation of VERDICT components

We further ablate the main components of VERDICT in the direct subset-acquisition setting on six datasets, two for each task modality. These variants are designed to isolate the roles of the three main ingredients of the method: verification by the original forward protocol, response-guided proposal construction, and local control of the surrogate approximation. **VERDICT-NoVerify** removes the retrain-and-check acceptance rule and follows the local surrogate path without using verified forward-execution feedback during search. This shows that surrogate feasibility alone is not sufficient for decision preservation. **VERDICT-RandomWS** keeps the same surrogate, trust-region, and verification machinery, but replaces response-guided working-set construction with uniformly random working sets of comparable size; this tests whether the decision-response signal is useful for identifying candidate examples to modify. **VERDICT-FixedTR** keeps the decision-response model and verification step, but disables adaptive trust-region updates, using a fixed radius throughout the search; this isolates the effect of adapting the local step size when managing approximation error. **VERDICT-AllConstraints** replaces the quasi-active constraint set with the full set of scenario-model margin constraints in the local surrogate, while retaining the VERDICT proposal and verification loop.

Default VERDICT achieves the best retained fraction on three of the six datasets, with especially strong reductions on Credit, STL10, and CIFAR100. The remaining cases are informative rather than contradictory: they show that, once verification is fixed, the particular local proposal model can matter in a problem-dependent way. On Adult, random working sets obtain a smaller verified subset, suggesting that response-guided working-set construction is not always the best exploration strategy. On the two text datasets, AllConstraints is strongest, indicating that the quasi-active constraint set may be too narrow when useful information is distributed across a broader set of scenario-model comparisons. Thus, Table 11 does not suggest a single universally optimal local surrogate; rather, it shows that the DPA objective can be approached by several verified proposal mechanisms, whose effectiveness depends on the structure of the scenario margins.

Table 11: Component ablation for direct decision-preserving subset acquisition. Each variant modifies one part of the VERDICT search procedure while keeping the same target winner map, margin constraints, and retrain-and-check evaluation protocol.

| Task | Dataset | n | VERDICT | VERDICT NoVerify | VERDICT RandomWS | VERDICT FixedTR | VERDICT AllConstraints |
|---|---|---|---|---|---|---|---|
| Tabular | Credit | 30000 | **0.427** | 0.947 | 0.767 ± 0.019 | 0.513 | 0.857 |
| | Adult | 48842 | 0.619 | 0.950 | **0.431 ± 0.085** | 0.664 | 0.777 |
| Vision | STL10 | 13000 | **0.073** | 0.080 | 0.197 ± 0.130 | 0.217 | 0.101 |
| | CIFAR100 | 50000 | **0.020** | 0.650 | 0.310 ± 0.111 | 0.215 | 0.033 |
| Text | 20Newsgroups | 18846 | 0.458 | 0.950 | 0.735 ± 0.010 | 0.464 | **0.441** |
| | AG News | 127600 | 0.544 | 0.950 | 0.897 ± 0.044 | 0.359 | **0.133** |

This interpretation is reinforced by the diagnostic results in Table 12. Across datasets, the accepted VERDICT outputs have positive but often small final verified margins, suggesting that the search tends to move close to the decision-preserving feasibility boundary rather than stopping at overly conservative subsets. This behavior is consistent with the trajectory in Figure 2 and Figure 3: the local surrogate enables aggressive deletion, while verification prevents the search from crossing into infeasible designs. The importance of this verification step is most apparent in NoVerify, which has negative final verified margins on all datasets, showing that following the surrogate trajectory alone does not reliably preserve the target decisions. Similarly, for default VERDICT, every rejected proposal is surrogate-feasible, yet the first such proposal has negative verified margin after retraining and produces margin violations, and sometimes winner flips. Thus, rejection corrects genuine failures of the local approximation rather than merely adding conservative bookkeeping. The remaining variants of VERDICT show a more dataset-dependent pattern: random working sets and all-constraint surrogates can be competitive in some cases, but their behavior is less uniform across tasks. Taken together, these results separate two aspects of the method: verification is essential for reliable decision preservation, while the best local surrogate configuration can depend on the geometry of the scenario margins and the informativeness of the response signal.

Table 12: Verification diagnostics across acquisition methods. All margins are reported after final verification, and the first rejected margin reports the surrogate margin followed by the verified margin after retraining.

| Dataset | Method | Accept / Reject | Final verified margin | First rejected margin surrogate → verified | Viol. / Flip |
|---|---|---|---|---|---|
| Credit | **VERDICT** | 23/7 | $7.27\times10^{-5}$ | $3.80\times10^{-5} \rightarrow -1.59\times10^{-4}$ | 3/3 |
| | NoVerify | 2/0 | $-8.92\times10^{-4}$ | — | — |
| | RandomWS | 21.3/8.7 | $9.83\times10^{-7}$ | $5.40\times10^{-6} \rightarrow -1.13\times10^{-5}$ | 2.3/0.7 |
| | FixedTR | 13/17 | $1.18\times10^{-5}$ | $3.80\times10^{-5} \rightarrow -1.59\times10^{-4}$ | 3/3 |
| | AllConstraints | 21/9 | $1.38\times10^{-7}$ | $1.15\times10^{-6} \rightarrow -1.20\times10^{-4}$ | 3/2 |
| Adult | **VERDICT** | 22/8 | $2.54\times10^{-6}$ | $4.30\times10^{-6} \rightarrow -3.02\times10^{-4}$ | 2/2 |
| | NoVerify | 1/0 | $-3.02\times10^{-4}$ | — | — |
| | RandomWS | 22.3/7.7 | $4.74\times10^{-6}$ | $2.83\times10^{-7} \rightarrow -2.78\times10^{-6}$ | 1/0.3 |
| | FixedTR | 8/22 | $4.13\times10^{-7}$ | $4.30\times10^{-6} \rightarrow -3.02\times10^{-4}$ | 2/2 |
| | AllConstraints | 22/8 | $2.88\times10^{-6}$ | $1.62\times10^{-6} \rightarrow -6.56\times10^{-6}$ | 1/0 |
| STL10 | **VERDICT** | 22/8 | $6.70\times10^{-4}$ | $7.60\times10^{-3} \rightarrow -5.57\times10^{-2}$ | 5/1 |
| | NoVerify | 11/0 | $-5.57\times10^{-2}$ | — | — |
| | RandomWS | 21.5/8.5 | $1.13\times10^{-4}$ | $1.11\times10^{-4} \rightarrow -1.98\times10^{-4}$ | 1/0.2 |
| | FixedTR | 30/0 | $1.32\times10^{-2}$ | — | — |
| | AllConstraints | 20/10 | $1.55\times10^{-4}$ | $3.06\times10^{-5} \rightarrow -1.82\times10^{-4}$ | 1/1 |
| CIFAR100 | **VERDICT** | 22/8 | $1.44\times10^{-4}$ | $4.33\times10^{-3} \rightarrow -5.42\times10^{-4}$ | 1/0 |
| | NoVerify | 5/0 | $-5.42\times10^{-4}$ | — | — |
| | RandomWS | 22/8 | $1.39\times10^{-4}$ | $1.18\times10^{-5} \rightarrow -1.31\times10^{-4}$ | 1/0 |
| | FixedTR | 30/0 | $1.45\times10^{-2}$ | — | — |
| | AllConstraints | 22/8 | $7.20 \times 10^{-4}$ | $2.15\times10^{-5} \rightarrow -2.01\times10^{-3}$ | 1/1 |
| 20Newsgroups | **VERDICT** | 22/8 | $6.58\times10^{-7}$ | $1.71\times10^{-7} \rightarrow -5.79\times10^{-4}$ | 1/1 |
| | NoVerify | 1/0 | $-5.79\times10^{-4}$ | — | — |
| | RandomWS | 13/9.3 | $3.26\times10^{-9}$ | $4.82\times10^{-8} \rightarrow -8.25\times10^{-5}$ | 1.3/0.3 |
| | FixedTR | 15/15 | $3.82\times10^{-7}$ | $1.71\times10^{-7} \rightarrow -5.79\times10^{-4}$ | 1/1 |
| | AllConstraints | 19/11 | $1.33\times10^{-9}$ | $1.11\times10^{-7} \rightarrow -2.37\times10^{-6}$ | 1/0 |
| AG News | **VERDICT** | 19/11 | $2.09\times10^{-6}$ | $2.36\times10^{-7} \rightarrow -1.99\times10^{-4}$ | 1/1 |
| | NoVerify | 1/0 | $-1.99\times10^{-4}$ | — | — |
| | RandomWS | 10.7/9 | $1.52\times10^{-8}$ | $1.95\times10^{-8} \rightarrow -9.42\times10^{-5}$ | 1/0 |
| | FixedTR | 20/10 | $8.62\times10^{-6}$ | $2.36\times10^{-7} \rightarrow -1.99\times10^{-4}$ | 1/1 |
| | AllConstraints | 24/6 | $1.19\times10^{-5}$ | $1.77\times10^{-7} \rightarrow -5.74\times10^{-6}$ | 1/0 |

# H Downstream Applications

## H.1 Decision-critical scenario discovery

**Setup and audit procedure.** VERDICT verifies decision preservation with respect to the declared scenario set. This protocol-relative nature suggests a natural auditing use case: if a practitioner is unsure whether a scenario family should be included in the declared scenario set, we can temporarily omit it, run VERDICT on the remaining scenarios, and test whether the selected subset also preserves the omitted family. Importantly, this audit does not change the candidate models, training routine, validation functionals, or decision convention. It varies only the scenario specification used during subset selection.

For each dataset, we construct the audit by holding out either semantically meaningful scenario families or automatically generated scenario groups, while keeping the candidate models, training routine, validation losses, and decision convention fixed. For tabular datasets, the omitted scenario family is specified by interpretable feature slices: age slices for Adult and Bank. For text datasets, we perform leave-one-family audits: AG News holds out one label class at a time, and 20Newsgroups holds out one topic family at a time. For vision datasets, scenarios are automatically constructed as class groups from validation labels. We

evaluate five random held-out class-group splits, each using seven class groups as $T_{\text{search}}$ and three as $T_{\text{heldout}}$, then use the strongest failing split for the expanded-scenario experiment.

Let $T_{\text{search}}$ denote the scenarios declared to VERDICT during subset selection, and let $T_{\text{heldout}}$ denote omitted scenarios used only for audit. The audit has three stages. First, VERDICT selects a subset using only $T_{\text{search}}$, and the selected subset is verified to preserve the original winners and margins on those search scenarios. Second, we retrain the candidate models on the selected subset and evaluate the omitted scenarios $T_{\text{heldout}}$. A held-out winner flip or margin violation indicates that the omitted family contains decision-critical constraints not implied by the declared search scenarios. Third, we form an expanded scenario set $T_{\text{expanded}} = T_{\text{search}} \cup T_{\text{heldout}}$, rerun VERDICT on this expanded scenario specification, and measure the additional retained evidence required to preserve the enlarged decision map.

We report metrics corresponding to the three stages of the audit. The held-out audit columns measure whether the subset selected from the search scenario set transfers to omitted scenarios: winner flips count scenarios where the selected winner changes, margin violations count failed winner-versus-competitor constraints, and $\Delta_{\min}^{\text{heldout}}$ records the worst held-out margin. The retained fractions $\rho_{\text{search}}$ and $\rho_{\text{expanded}}$ measure the evidence cost before and after adding the omitted scenarios back, while $\Delta_{\min}^{\text{expanded}}$ verifies whether the expanded scenario set is preserved after rerunning VERDICT. Finally, baseline feasible counts summarize how often random, stratified, KCenter, or DecisionIF subsets at VERDICT's expanded-scenario retained size also satisfy all expanded-scenario constraints. Together, these metrics distinguish three effects: whether an omission is decision-critical, how costly it is to cover, and whether success is specific to VERDICT's selected evidence rather than retained size alone.

**Experimental results.** Table 13 shows that held-out failures can diagnose incomplete scenario specifications. In five out of six datasets, VERDICT preserves the search scenario set, but the selected subset fails on omitted scenarios through winner flips or negative held-out margins. These failures indicate that the omitted scenarios impose decision constraints not implied by the search scenarios alone. Adding the omitted scenarios back makes the diagnostic actionable. VERDICT then finds feasible subsets with nonnegative margins on the expanded scenario set, and the increase from $\rho_{\text{search}}$ to $\rho_{\text{expanded}}$ measures the evidence cost of covering the omitted scenarios. This cost varies substantially: Adult and AG News require large increases, OxfordPets and 20Newsgroups require moderate increases, and STL10 requires only a small increase, suggesting that decision-critical scenarios may differ in how much additional evidence they require.

Table 13: Scenario-specification audits across three modalities. $\rho_{\text{search}}$ is the retained fraction selected using only the search scenario set; $\rho_{\text{expanded}}$ is the retained fraction after adding the omitted family back. $\Delta_{\min}^{\text{heldout}}$ is the minimum held-out decision margin, so negative values indicate margin violations.

| Task | Dataset | Omitted family | Held-out audit | | | Expanded protocol | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Flips | Viol. | $\Delta_{\min}^{\text{heldout}}$ | $\rho_{\text{search}}$ | $\rho_{\text{expanded}}$ | $\Delta_{\min}^{\text{expanded}}$ | Best baseline feasible |
| Tabular | Adult | age slices | 2 | 5 | $-2.15\times10^{-3}$ | 0.378 | 0.619 | $2.54\times10^{-6}$ | 0/20 |
| | Bank | age slices | 0 | 0 | $5.63\times10^{-5}$ | 0.397 | 0.413 | $2.50\times10^{-8}$ | 6/20 |
| | Credit | age slices | | | | | | | |
| Vision | OxfordPets | failure splits | 1 | 2 | $-9.51\times10^{-2}$ | 0.116 | 0.221 | $4.96\times10^{-4}$ | 0/20 |
| | STL10 | failure splits | 3 | 9 | $-5.73\times10^{-2}$ | 0.061 | 0.073 | $6.70\times10^{-4}$ | 0/20 |
| | STL10 | failure splits | 3 | 9 | $-5.73\times10^{-2}$ | 0.061 | 0.073 | $6.70\times10^{-4}$ | 0/20 |
| | STL10 | failure splits | 3 | 9 | $-5.73\times10^{-2}$ | 0.061 | 0.073 | $6.70\times10^{-4}$ | 0/20 |
| Text | AG News | world news | 1 | 5 | $-8.18\times10^{-2}$ | 0.070 | 0.544 | $2.09\times10^{-6}$ | 0/20 |
| | 20Newsgroups | computing | 1 | 1 | $-2.62\times10^{-3}$ | 0.411 | 0.458 | $6.58\times10^{-7}$ | 0/20 |

We compare random, stratified sampling, KCenter, and DecisionIF subsets at the same retained size selected by VERDICT for $T_{\text{expanded}}$. Table 14 shows that these gains are not explained merely by retaining more data. For the five positive discovery cases, random, stratified sampling, K-center, and DecisionIF subsets at VERDICT's expanded-scenario retained size all fail to preserve the expanded scenario set. This indicates that the selected evidence, not only the retained fraction, is important for decision preservation. Bank provides an important contrast. Its omitted age slices do not produce a strong scenario-specification gap, and several same-size baselines as VERDICT are feasible in several seeds. This shows that the audit is not biased toward

Table 14: Baseline comparison across datasets. Feasible rate reports the fraction of feasible runs. Avg winner agreement, Avg violation rate, Avg min margin, and Avg winner flips are averaged across runs.

| Dataset | Method | Feasible rate | Avg winner agreement | Avg violation rate | Avg min margin | Avg winner flips |
|---|---|---|---|---|---|---|
| Adult | VERDICT | **1/1** | **1.00** | **0.000** | $\mathbf{2.54\times10^{-6}}$ | **0.0** |
| | Random | 0/5 | 0.60 | 0.079 | $-2.19\times10^{-4}$ | 2.4 |
| | Stratified | 0/5 | 0.567 | 0.113 | $-2.82\times10^{-4}$ | 2.6 |
| | KCenter | 0/5 | 0.633 | 0.088 | $-3.71\times10^{-4}$ | 2.2 |
| | DecisionIF | 0/5 | 0.333 | 0.167 | $-2.55\times10^{-3}$ | 4.0 |
| Bank | VERDICT | **1/1** | **1.00** | **0.000** | $\mathbf{2.50\times10^{-8}}$ | **0.0** |
| | Random | 4/5 | **1.00** | 0.005 | $-8.11\times10^{-7}$ | **0.0** |
| | Stratified | 2/5 | 0.88 | 0.050 | $-2.24\times10^{-5}$ | 0.6 |
| | KCenter | 0/5 | 0.48 | 0.235 | $-1.64\times10^{-4}$ | 2.6 |
| | DecisionIF | 0/5 | 0.00 | 0.575 | $-7.12\times10^{-2}$ | 5.0 |
| AG News | VERDICT | **1/1** | **1.00** | **0.000** | $\mathbf{2.09\times10^{-6}}$ | **0.0** |
| | Random | 0/5 | 0.65 | 0.194 | $-2.35\times10^{-3}$ | 1.4 |
| | Stratified | 0/5 | 0.75 | 0.156 | $-2.34\times10^{-3}$ | 1.0 |
| | KCenter | 0/5 | 0.75 | 0.156 | $-2.54\times10^{-3}$ | 1.0 |
| | DecisionIF | 0/5 | 0.75 | 0.031 | $-3.94\times10^{-3}$ | 1.0 |
| 20Newsgroups | VERDICT | **1/1** | **1.00** | **0.000** | $\mathbf{6.58\times10^{-7}}$ | **0.0** |
| | Random | 0/5 | 0.233 | 0.221 | $-1.93\times10^{-3}$ | 4.6 |
| | Stratified | 0/5 | 0.333 | 0.208 | $-1.75\times10^{-3}$ | 4.0 |
| | KCenter | 0/5 | 0.500 | 0.188 | $-1.13\times10^{-3}$ | 3.0 |
| | DecisionIF | 0/5 | 0.167 | 0.229 | $-6.01\times10^{-2}$ | 5.0 |
| OxfordPets | VERDICT | **1/1** | **1.00** | **0.000** | $\mathbf{4.96\times10^{-4}}$ | **0.0** |
| | Random | 0/5 | 0.80 | 0.0375 | $-4.87\times10^{-2}$ | 2.0 |
| | Stratified | 0/5 | 0.80 | 0.0300 | $-4.21\times10^{-2}$ | 2.0 |
| | KCenter | 0/5 | 0.80 | 0.0275 | $-5.27\times10^{-2}$ | 2.0 |
| | DecisionIF | 0/5 | 0.70 | 0.0375 | $-5.34\times10^{-2}$ | 3.0 |
| STL10 | VERDICT | **1/1** | **1.00** | **0.000** | $\mathbf{6.70\times10^{-4}}$ | **0.0** |
| | Random | 0/5 | 0.62 | 0.200 | $-7.88\times10^{-2}$ | 3.8 |
| | Stratified | 0/5 | 0.64 | 0.185 | $-6.42\times10^{-2}$ | 3.6 |
| | KCenter | 0/5 | 0.82 | 0.100 | $-8.54\times10^{-2}$ | 1.8 |
| | DecisionIF | 0/5 | 0.80 | 0.075 | $-2.85\times10^{-2}$ | 2.0 |

always declaring omitted scenarios decision-critical. Instead, VERDICT distinguishes cases where omitted scenarios introduce new decision constraints from cases where the declared search scenarios already transfer.

### H.2 Evaluation-Subset Protocol for Low-Cost Model Selection

Model selection can be expensive even after candidate models have been trained, because each candidate must be evaluated on a potentially large validation set. In this downstream application, the acquisition variable is not the training set but the evaluation subset used to compare a fixed set of trained candidate models. The goal is to select a small subset of validation examples whose aggregate losses induce the same model-selection conclusion as the full validation set.

This setting instantiates the decision-preserving view with a different forward protocol. The candidate models are fixed, the full validation set defines the reference decision, and a subset is feasible if evaluating the candidates only on that subset preserves the relevant comparison structure. Depending on the intended use case, preservation may mean preserving the top-ranked model, preserving all pairwise orderings above a margin, or preserving the full model ranking. We therefore report both ranking metrics and decision metrics.

For a validation subset $S$, let $F_m(S)$ denote the average loss of candidate model $m$ on $S$, and let $m^\star$ be the winner under the full validation set. A low-cost evaluation subset should satisfy

$$F_m(S) - F_{m^\star}(S) \geq \gamma_m \quad \text{for all } m \neq m^\star,$$

where $\gamma_m$ is a prescribed margin derived from the full-validation gap. This is the evaluation-subset analogue of the DPA constraint used in the training-acquisition experiments. In addition to ranking correlations, we therefore evaluate whether the selected subset preserves the model-selection decision itself.

We first define the low-cost model-selection protocol. For each dataset and backbone, we train a fixed pool of candidate classifiers and compute their losses on the full validation set. The full validation set determines the reference ranking and reference winner. Each subset-selection method then chooses a small budgeted subset of validation examples. We evaluate the candidate models on this subset and compare the induced model ranking and winner against the full-validation reference.

We report the following metrics:

- **Winner agreement**: whether the subset selects the same top model as the full validation set.
- **Selection regret**: the full-validation loss difference between the model selected by the subset and the true full-validation winner.
- **Pairwise decision agreement**: fraction of candidate pairs whose ordering is preserved by the subset.
- **Margin violation rate**: the fraction of required winner-vs-competitor margins violated on the subset.
- **Minimum decision margin**: the smallest subset margin between the reference winner and any competitor.
- **Kendall $\tau$ and Spearman $\rho$**: rank-correlation metrics measuring preservation of the full model ranking.
- **Budget fraction and runtime**: the evaluation cost of the subset relative to the full validation set.

This metric suite distinguishes two goals. Ranking correlations measure whether the subset approximately preserves the full ordering of all candidate models. Decision metrics measure the downstream model-selection consequence: whether a practitioner using the subset would deploy the same model as they would using the full validation set. These metrics make the downstream application operational. A subset is useful for low-cost model selection if it either recovers the exact full-validation winner or selects a model with low full-validation regret while preserving most candidate comparisons. Thus, the reported metrics should be interpreted as model-selection fidelity metrics, not merely as generic ranking-correlation statistics.

**Backbone Robustness for Evaluation Subsets** We study evaluation subset selection in vision datasets to preserve model comparisons under a fixed protocol, in the spirit of data efficient evaluation and tailored benchmark design Vivek et al. (2024); Saranathan et al. (2025); Yuan et al. (2025). For vision datasets, we repeat the ranking preservation protocol across five frozen vision backbones: ResNet50, ResNet18, ConvNeXt-Tiny, Swin-T, and ViT-B/16. We cache frozen features for the train and validation splits, then train 20 linear classifiers with varying random seeds and hyperparameters on top of these features to generate the validation score matrix. We run subset selection at various budgets and measure the Kendall $\tau$ correlation between the subset ranking and the full dataset ranking.

Experiments cover subsets with sizes of 0.5%, 1%, 2%, and 5%. We also evaluate 10% and 20% when estimating minimum budget thresholds. We compare Random sampling, Stratified sampling, Difficulty-based selection based on hard example mining (Shrivastava et al., 2016), Disagreement-based selection, and k center clustering on cached embeddings following Sener & Savarese (2018) to VERDICT. We also include score-space baselines: loss-space k center and k medoids (as in tailored benchmark selection Yuan et al. (2025)), a Frank–Wolfe mean coreset motivated by Frank–Wolfe coreset constructions Clarkson (2010), and a pairwise hinge baseline inspired by pairwise ranking objectives Joachims (2002).

**Detailed Kendall $\tau$ breakdown across budgets (ResNet50)** Since the main text uses ResNet50 as the default backbone, Table 15 provides the full Kendall $\tau$ correlation breakdown across budgets for ResNet50 (mean$_{\pm std}$ over 5 seeds), complementing the multi-backbone summary metrics reported below. We analyze robustness using three metrics: fidelity at low budgets, stability against ranking inversions, and sample efficiency. Table 16 reports the macro-averaged Kendall $\tau$ at a 0.5% budget. VERDICT maintains high

Table 15: **Detailed Kendall $\tau$ correlation breakdown across budgets (ResNet50).** Each cell reports $\text{mean}_{\pm std}$ over 5 seeds.

| Budget | Method | CIFAR10 | CIFAR100 | Oxford Pets | STL10 |
|---|---|---|---|---|---|
| 0.5% | Difficulty | $0.074_{\pm 0.000}$ | $-0.400_{\pm 0.000}$ | $-0.032_{\pm 0.000}$ | $0.737_{\pm 0.000}$ |
| | Disagreement | $-0.032_{\pm 0.000}$ | $-0.442_{\pm 0.000}$ | $-0.032_{\pm 0.000}$ | $0.747_{\pm 0.000}$ |
| | K center | $0.535_{\pm 0.159}$ | $0.655_{\pm 0.144}$ | $0.213_{\pm 0.089}$ | $-0.158_{\pm 0.653}$ |
| | Random | $0.554_{\pm 0.131}$ | $0.741_{\pm 0.063}$ | $0.139_{\pm 0.159}$ | $-0.221_{\pm 0.658}$ |
| | Stratified | $0.579_{\pm 0.182}$ | $0.693_{\pm 0.138}$ | $0.232_{\pm 0.220}$ | $-0.629_{\pm 0.020}$ |
| | Loss k-center | $0.204_{\pm 0.009}$ | $0.145_{\pm 0.038}$ | $0.636_{\pm 0.009}$ | $0.937_{\pm 0.000}$ |
| | Loss k-medoids | $0.255_{\pm 0.005}$ | $0.122_{\pm 0.019}$ | $0.674_{\pm 0.000}$ | $0.958_{\pm 0.000}$ |
| | FW mean | $0.825_{\pm 0.042}$ | $0.937_{\pm 0.021}$ | $0.105_{\pm 0.000}$ | $0.074_{\pm 0.000}$ |
| | Pairwise hinge | $-0.118_{\pm 0.020}$ | $0.076_{\pm 0.033}$ | $-0.074_{\pm 0.000}$ | $-0.074_{\pm 0.000}$ |
| | **VERDICT** | $\mathbf{0.899}_{\pm 0.047}$ | $\mathbf{0.935}_{\pm 0.046}$ | $\mathbf{0.556}_{\pm 0.005}$ | $\mathbf{0.789}_{\pm 0.000}$ |
| 1.0% | Difficulty | $0.084_{\pm 0.000}$ | $-0.379_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.695_{\pm 0.000}$ |
| | Disagreement | $0.095_{\pm 0.000}$ | $-0.347_{\pm 0.000}$ | $-0.021_{\pm 0.000}$ | $0.726_{\pm 0.000}$ |
| | K center | $0.545_{\pm 0.038}$ | $0.655_{\pm 0.117}$ | $0.122_{\pm 0.060}$ | $0.267_{\pm 0.510}$ |
| | Random | $0.659_{\pm 0.094}$ | $0.718_{\pm 0.065}$ | $0.162_{\pm 0.183}$ | $-0.339_{\pm 0.475}$ |
| | Stratified | $0.707_{\pm 0.111}$ | $0.699_{\pm 0.070}$ | $0.392_{\pm 0.191}$ | $0.011_{\pm 0.658}$ |
| | Loss k-center | $0.219_{\pm 0.014}$ | $0.145_{\pm 0.017}$ | $0.674_{\pm 0.000}$ | $0.941_{\pm 0.014}$ |
| | Loss k-medoids | $0.255_{\pm 0.005}$ | $0.152_{\pm 0.046}$ | $0.794_{\pm 0.006}$ | $0.907_{\pm 0.005}$ |
| | FW mean | $0.865_{\pm 0.014}$ | $0.956_{\pm 0.009}$ | $0.232_{\pm 0.000}$ | $0.305_{\pm 0.000}$ |
| | Pairwise hinge | $-0.145_{\pm 0.009}$ | $0.036_{\pm 0.012}$ | $-0.074_{\pm 0.000}$ | $-0.074_{\pm 0.000}$ |
| | **VERDICT** | $\mathbf{0.920}_{\pm 0.028}$ | $\mathbf{0.968}_{\pm 0.025}$ | $\mathbf{0.895}_{\pm 0.062}$ | $\mathbf{0.798}_{\pm 0.014}$ |
| 2.0% | Difficulty | $0.074_{\pm 0.000}$ | $-0.326_{\pm 0.000}$ | $-0.011_{\pm 0.000}$ | $0.726_{\pm 0.000}$ |
| | Disagreement | $0.179_{\pm 0.000}$ | $-0.358_{\pm 0.000}$ | $-0.063_{\pm 0.000}$ | $0.726_{\pm 0.000}$ |
| | K center | $0.663_{\pm 0.049}$ | $0.705_{\pm 0.106}$ | $0.116_{\pm 0.041}$ | $0.337_{\pm 0.284}$ |
| | Random | $0.672_{\pm 0.144}$ | $0.834_{\pm 0.044}$ | $0.244_{\pm 0.239}$ | $-0.352_{\pm 0.611}$ |
| | Stratified | $0.735_{\pm 0.085}$ | $0.897_{\pm 0.019}$ | $0.339_{\pm 0.335}$ | $-0.253_{\pm 0.577}$ |
| | Loss k-center | $0.276_{\pm 0.012}$ | $0.175_{\pm 0.009}$ | $0.709_{\pm 0.027}$ | $0.968_{\pm 0.000}$ |
| | Loss k-medoids | $0.248_{\pm 0.021}$ | $0.164_{\pm 0.025}$ | $0.709_{\pm 0.009}$ | $0.989_{\pm 0.000}$ |
| | FW mean | $0.888_{\pm 0.012}$ | $0.983_{\pm 0.009}$ | $0.242_{\pm 0.000}$ | $0.516_{\pm 0.000}$ |
| | Pairwise hinge | $-0.154_{\pm 0.006}$ | $0.585_{\pm 0.083}$ | $-0.074_{\pm 0.000}$ | $-0.074_{\pm 0.000}$ |
| | **VERDICT** | $\mathbf{0.935}_{\pm 0.038}$ | $\mathbf{0.987}_{\pm 0.012}$ | $\mathbf{0.958}_{\pm 0.024}$ | $\mathbf{0.798}_{\pm 0.014}$ |
| 5.0% | Difficulty | $0.105_{\pm 0.000}$ | $-0.284_{\pm 0.000}$ | $0.074_{\pm 0.000}$ | $0.758_{\pm 0.000}$ |
| | Disagreement | $0.221_{\pm 0.000}$ | $-0.095_{\pm 0.000}$ | $0.053_{\pm 0.000}$ | $0.758_{\pm 0.000}$ |
| | K center | $0.682_{\pm 0.067}$ | $0.821_{\pm 0.116}$ | $0.141_{\pm 0.041}$ | $0.661_{\pm 0.067}$ |
| | Random | $0.800_{\pm 0.105}$ | $0.893_{\pm 0.040}$ | $0.371_{\pm 0.343}$ | $0.246_{\pm 0.413}$ |
| | Stratified | $0.773_{\pm 0.123}$ | $0.886_{\pm 0.047}$ | $0.371_{\pm 0.288}$ | $0.126_{\pm 0.500}$ |
| | Loss k-center | $0.307_{\pm 0.012}$ | $0.194_{\pm 0.016}$ | $0.798_{\pm 0.014}$ | $1.000_{\pm 0.000}$ |
| | Loss k-medoids | $0.297_{\pm 0.005}$ | $0.185_{\pm 0.006}$ | $0.794_{\pm 0.016}$ | $1.000_{\pm 0.000}$ |
| | FW mean | $0.937_{\pm 0.020}$ | $0.992_{\pm 0.005}$ | $0.579_{\pm 0.000}$ | $0.811_{\pm 0.000}$ |
| | Pairwise hinge | $-0.158_{\pm 0.000}$ | $0.813_{\pm 0.033}$ | $-0.368_{\pm 0.000}$ | $-0.074_{\pm 0.000}$ |
| | **VERDICT** | $\mathbf{0.981}_{\pm 0.022}$ | $\mathbf{0.996}_{\pm 0.009}$ | $\mathbf{0.958}_{\pm 0.024}$ | $\mathbf{0.859}_{\pm 0.071}$ |

rank preservation ($\tau \approx 0.95 - 0.98$) regardless of whether the backbone is a Convolutional Neural Network or a Vision Transformer. Baselines fluctuate depending on the feature space geometry and degrade under ConvNeXt or Swin-T features. Averages can mask failures on specific datasets. Table 17 counts the datasets

Table 16: Macro-averaged Kendall $\tau$ at 0.5% budget across datasets. VERDICT remains consistent across architectures.

| Backbone | VERDICT | Random | Stratified | Difficulty | Disagreement | k-center |
|---|---|---|---|---|---|---|
| ConvNeXt-T | **0.984**±0.017 | 0.637±0.278 | 0.736±0.129 | -0.355±0.429 | -0.162±0.555 | 0.484±0.410 |
| ResNet18 | **0.962**±0.041 | 0.518±0.335 | 0.537±0.405 | 0.687±0.154 | 0.647±0.143 | 0.381±0.355 |
| ResNet50 | **0.952**±0.053 | 0.333±0.512 | 0.209±0.547 | 0.189±0.653 | 0.155±0.669 | 0.178±0.570 |
| Swin-T | **0.976**±0.032 | 0.211±0.562 | 0.343±0.492 | 0.568±0.494 | 0.540±0.582 | 0.189±0.606 |
| ViT-B/16 | **0.971**±0.036 | 0.255±0.402 | 0.345±0.434 | 0.818±0.112 | 0.752±0.238 | 0.171±0.522 |

where methods produce a ranking inversion (Kendall $\tau < 0$) at the 0.5% budget. A negative correlation implies the subset is misleading and performs worse than random guessing. VERDICT yields zero inversions across all backbones. Difficulty and Disagreement methods exhibit instability in this low budget regime and produce inversions in most datasets with ConvNeXt features. Table 18 summarizes the minimum budget

Table 17: Number of datasets with Ranking Inversion (Kendall $\tau < 0$) at 0.5% budget. Lower is better.

| Backbone | VERDICT | Random | Stratified | Difficulty | Disagreement | k-center |
|---|---|---|---|---|---|---|
| ConvNeXt-T | **0** | 1 | 0 | 6 | 5 | 1 |
| ResNet18 | **0** | 1 | 2 | 0 | 0 | 2 |
| ResNet50 | **0** | 2 | 2 | 2 | 2 | 3 |
| Swin-T | **0** | 3 | 1 | 1 | 1 | 3 |
| ViT-B/16 | **0** | 1 | 2 | 0 | 0 | 3 |

required for VERDICT to reach a correlation of 0.9. The 0.5% budget is sufficient in 32 out of 35 cases. We identify a harder regime: CIFAR100 with ResNet50 (2%). The method resolves the ranking using a small fraction of the data even in these cases.

Table 18: Minimum budget required for VERDICT to reach Kendall $\tau \geq 0.9$.

| Dataset | ConvNeXt Tiny | ResNet18 | ResNet50 | Swin T | ViT B/16 |
|---|---|---|---|---|---|
| CIFAR10 | 0.5% | 0.5% | 0.5% | 0.5% | 0.5% |
| CIFAR100 | 0.5% | 0.5% | 2% | 0.5% | 0.5% |
| OxfordPets | 0.5% | 0.5% | 0.5% | 0.5% | 0.5% |
| STL10 | 0.5% | 0.5% | 0.5% | 0.5% | 0.5% |

Detailed per-backbone tables reporting dataset metrics, budget curves, and baseline comparisons are provided in Tables 19 through 23.

Table 19: Results for **ConvNeXt-Tiny**.

**(1) Kendall $\tau$ @ 0.5% per dataset**

| Method | CIFAR10 | CIFAR100 | OxfordPets | STL10 |
|---|---|---|---|---|
| VERDICT | 0.96±0.01 | 1.00 | 1.00±0.00 | 0.96 |
| Random | 0.74±0.06 | 0.70±0.05 | 0.70±0.10 | -0.04±0.43 |
| Stratified | 0.71±0.08 | 0.83±0.06 | 0.56±0.46 | 0.54±0.09 |
| Difficulty | -0.60 | -0.62 | -0.40 | -0.40 |
| Disagreement | -0.64 | -0.54 | -0.56 | -0.40 |
| k-center | 0.76±0.05 | 0.87±0.01 | 0.62±0.08 | 0.42±0.06 |

**(2) Macro Kendall $\tau$ vs Budget**

| Budget | VERDICT | Random | Stratified | Difficulty | Disagreement | k-center |
|---|---|---|---|---|---|---|
| 0.5% | 0.984±0.017 | 0.637±0.278 | 0.736±0.129 | -0.355±0.429 | -0.162±0.555 | 0.484±0.410 |
| 1% | 0.989±0.015 | 0.709±0.163 | 0.639±0.327 | -0.353±0.426 | 0.087±0.577 | 0.513±0.302 |
| 2% | 0.989±0.012 | 0.759±0.135 | 0.700±0.185 | -0.292±0.483 | 0.368±0.496 | 0.555±0.405 |

**(3) Min Budget** ($\tau \geq 0.9$)

| Dataset | VERDICT | Baseline min budget | Baseline at min budget | Base. $\tau$@0.5% | Base. $\tau$@Max |
|---|---|---|---|---|---|
| CIFAR10 | 0.5% | 10% | Disagreement | -0.64 | 0.97 |
| CIFAR100 | 0.5% | 10% | Disagreement | -0.54 | 0.94 |
| OxfordPets | 0.5% | 20% | Stratified | 0.56 | 0.92 |
| STL10 | 0.5% | 20% | Difficulty | -0.40 | 0.93 |

Table 20: Results for **ResNet18**.

**(1) Kendall $\tau$ @ 0.5% per dataset**

| Method | CIFAR10 | CIFAR100 | OxfordPets | STL10 |
|---|---|---|---|---|
| VERDICT | 1.00±0.01 | 0.99±0.01 | 0.92 | 0.98 |
| Random | 0.75±0.10 | 0.81±0.07 | -0.04±0.27 | 0.10±0.35 |
| Stratified | 0.82±0.03 | 0.86±0.03 | -0.09±0.40 | -0.06±0.11 |
| Difficulty | 0.77 | 0.86 | 0.71 | 0.84 |
| Disagreement | 0.80 | 0.53 | 0.73 | 0.79 |
| k-center | 0.67±0.06 | 0.70±0.09 | -0.13±0.10 | -0.13 |

**(2) Macro Kendall $\tau$ vs Budget**

| Budget | VERDICT | Random | Stratified | Difficulty | Disagreement | k-center |
|---|---|---|---|---|---|---|
| 0.5% | 0.962±0.041 | 0.518±0.335 | 0.537±0.405 | 0.687±0.154 | 0.647±0.143 | 0.381±0.355 |
| 1% | 0.975±0.039 | 0.555±0.365 | 0.651±0.297 | 0.753±0.163 | 0.672±0.148 | 0.506±0.326 |
| 2% | 0.978±0.034 | 0.617±0.346 | 0.689±0.285 | 0.767±0.163 | 0.729±0.119 | 0.545±0.336 |

**(3) Min Budget** ($\tau \geq 0.9$)

| Dataset | VERDICT | Baseline min budget | Baseline at min budget | Base. $\tau$@0.5% | Base. $\tau$@Max |
|---|---|---|---|---|---|
| CIFAR10 | 0.5% | 5% | Random | 0.75 | 0.95 |
| CIFAR100 | 0.5% | 2% | Stratified | 0.86 | 0.98 |
| OxfordPets | 0.5% | 1% | Difficulty | 0.71 | 1.00 |
| STL10 | 0.5% | 1% | Difficulty | 0.84 | 1.00 |

Table 21: Results for **ResNet50**.

**(1) Kendall $\tau$ @ 0.5% per dataset**

| Method | CIFAR10 | CIFAR100 | OxfordPets | STL10 |
|---|---|---|---|---|
| VERDICT | 0.90±0.06 | 0.87±0.09 | 0.91 | 1.00 |
| Random | 0.34±0.09 | 0.56±0.12 | -0.49±0.25 | -0.36±0.68 |
| Stratified | 0.29±0.31 | 0.50±0.19 | -0.29±0.38 | -0.83±0.02 |
| Difficulty | 0.22 | 0.03 | 0.71 | 0.94 |
| Disagreement | 0.14 | 0.06 | 0.62 | 0.98 |
| k-center | 0.57±0.14 | 0.46±0.18 | -0.23±0.19 | -0.26±0.61 |

**(2) Macro Kendall $\tau$ vs Budget**

| Budget | VERDICT | Random | Stratified | Difficulty | Disagreement | k-center |
|---|---|---|---|---|---|---|
| 0.5% | 0.952±0.053 | 0.333±0.512 | 0.209±0.547 | 0.189±0.653 | 0.155±0.669 | 0.178±0.570 |
| 1% | 0.958±0.046 | 0.390±0.456 | 0.462±0.308 | 0.202±0.648 | 0.186±0.651 | 0.197±0.599 |
| 2% | 0.964±0.039 | 0.442±0.546 | 0.541±0.428 | 0.226±0.655 | 0.229±0.629 | 0.215±0.603 |

**(3) Min Budget** ($\tau \geq 0.9$)

| Dataset | VERDICT | Baseline min budget | Baseline at min budget | Base. $\tau$@0.5% | Base. $\tau$@Max |
|---|---|---|---|---|---|
| CIFAR10 | 0.5% | > 20% | - | - | - |
| CIFAR100 | 2% | 20% | Stratified | 0.50 | 0.95 |
| OxfordPets | 0.5% | > 20% | - | - | - |
| STL10 | 0.5% | 0.5% | Difficulty | 0.94 | 1.00 |

Table 22: Results for **Swin-T**.

**(1) Kendall $\tau$ @ 0.5% per dataset**

| Method | CIFAR10 | CIFAR100 | OxfordPets | STL10 |
|---|---|---|---|---|
| VERDICT | 0.96±0.04 | 0.98±0.02 | 0.91 | 0.99 |
| Random | 0.34±0.29 | 0.71±0.10 | -0.54±0.10 | -0.38±0.56 |
| Stratified | 0.23±0.16 | 0.64±0.15 | 0.20±0.48 | -0.71±0.18 |
| Difficulty | 0.73 | 0.71 | 0.70 | 0.91 |
| Disagreement | 0.74 | 0.81 | 0.77 | 0.91 |
| k-center | 0.60±0.06 | 0.69±0.05 | -0.26±0.26 | 0.67 |

**(2) Macro Kendall $\tau$ vs Budget**

| Budget | VERDICT | Random | Stratified | Difficulty | Disagreement | k-center |
|---|---|---|---|---|---|---|
| 0.5% | 0.976±0.032 | 0.211±0.562 | 0.343±0.492 | 0.568±0.494 | 0.540±0.582 | 0.189±0.606 |
| 1% | 0.982±0.020 | 0.309±0.545 | 0.356±0.492 | 0.588±0.572 | 0.535±0.596 | 0.376±0.527 |
| 2% | 0.983±0.020 | 0.439±0.422 | 0.522±0.460 | 0.612±0.548 | 0.592±0.540 | 0.421±0.507 |

**(3) Min Budget** ($\tau \geq 0.9$)

| Dataset | VERDICT | Baseline min budget | Baseline at min budget | Base. $\tau$@0.5% | Base. $\tau$@Max |
|---|---|---|---|---|---|
| CIFAR10 | 0.5% | 20% | Difficulty | 0.73 | 0.94 |
| CIFAR100 | 0.5% | > 20% | - | - | - |
| OxfordPets | 0.5% | 10% | Difficulty | 0.70 | 0.98 |
| STL10 | 0.5% | 0.5% | Difficulty | 0.91 | 1.00 |

Table 23: Results for **ViT-B/16**.

**(1) Kendall $\tau$ @ 0.5% per dataset**

| Method | CIFAR10 | CIFAR100 | OxfordPets | STL10 |
|---|---|---|---|---|
| VERDICT | 0.99±0.00 | 0.98±0.01 | 0.93±0.01 | 0.99 |
| Random | 0.42±0.36 | 0.65±0.18 | -0.35±0.57 | 0.03±0.40 |
| Stratified | 0.41±0.39 | 0.66±0.09 | -0.03±0.40 | -0.46±0.13 |
| Difficulty | 0.94 | 0.90 | 0.85 | 0.83 |
| Disagreement | 0.86 | 0.86 | 0.85 | 0.87 |
| k-center | 0.62±0.38 | 0.80±0.04 | -0.17±0.68 | -0.70±0.07 |

**(2) Macro Kendall $\tau$ vs Budget**

| Budget | VERDICT | Random | Stratified | Difficulty | Disagreement | k-center |
|---|---|---|---|---|---|---|
| 0.5% | 0.971±0.036 | 0.255±0.402 | 0.345±0.434 | 0.818±0.112 | 0.752±0.238 | 0.171±0.522 |
| 1% | 0.977±0.026 | 0.300±0.501 | 0.403±0.521 | 0.845±0.121 | 0.788±0.229 | 0.326±0.533 |
| 2% | 0.983±0.022 | 0.437±0.403 | 0.494±0.509 | 0.868±0.103 | 0.857±0.102 | 0.528±0.332 |

**(3) Min Budget ($\tau \geq 0.9$)**

| Dataset | VERDICT | Baseline min budget | Baseline at min budget | Base. $\tau$@0.5% | Base. $\tau$@Max |
|---|---|---|---|---|---|
| CIFAR10 | 0.5% | 0.5% | Difficulty | 0.94 | 1.00 |
| CIFAR100 | 0.5% | 1% | Difficulty | 0.90 | 0.93 |
| OxfordPets | 0.5% | 2% | Difficulty | 0.85 | 1.00 |
| STL10 | 0.5% | 1% | Difficulty | 0.83 | 1.00 |

# I Runtime and scalability

## I.1 Runtime components and measurement protocol

For each dataset, we record the following runtime components for the direct train-and-verify search:

- **End-to-end search time** (`select_time_sec`): total wall-clock time for the full VERDICT search path under the fixed proposal budget.
- **Proposal-construction time** (`score_time_sec`): time spent computing the decision-response scores, forming the working-set proposal, and constructing the proposed subset at each search iteration.
- **Incumbent retrain-and-verify time** (`current_train_verify_time_sec`): time spent retraining and checking the currently accepted design at each iteration.
- **Proposed-design retrain-and-verify time** (`candidate_retrain_verify_time_sec`): time spent retraining on the newly proposed design and verifying the decision-margin constraints under the fixed protocol.
- **Per-iteration averages**: proposal-construction and retrain-and-verify totals divided by the number of search iterations. These averages summarize the cost of one VERDICT proposal-and-check step.

All timings are single-run wall-clock seconds. In the direct-acquisition experiments, VERDICT uses a fixed 30-iteration search budget, so the per-iteration quantities are obtained by dividing the corresponding totals by 30 iterations.

## I.2 Direct training acquisition runtime breakdown

The runtime breakdown shows that the direct train-and-verify search is generally dominated by proposal construction rather than by retraining-based verification. On most datasets, the decision-response scoring and subset-proposal step accounts for the largest share of wall-clock time, especially on CIFAR10 and CIFAR100. By contrast, retraining and verification of the incumbent and proposed designs remains comparatively modest. VERDICT is fast on tabular datasets, while text and vision datasets require more time due to larger feature spaces, more scenarios, or larger candidate/margin computations. The main outlier is CIFAR100, where the end-to-end runtime is 2127.96s and almost all of the cost comes from proposal construction rather than verification. These results indicate that the verification step, although essential for correctness, is not the primary computational bottleneck in most settings; scaling VERDICT further would mainly require accelerating the decision-response scoring and working-set proposal computation.

Table 24: Runtime breakdown across datasets on direct training acquisition over 30 proposal iterations.

| Dataset | End-to-end search | Proposal construction | Incumbent retrain-verify | Proposed-design retrain+verify | Proposal construction per iter. | Retrain-verify per iter. |
|---|---|---|---|---|---|---|
| Credit | 4.22 | 3.01 | 0.62 | 0.58 | 0.10 | 0.04 |
| Bank | 2.38 | 1.65 | 0.36 | 0.35 | 0.06 | 0.02 |
| Adult | 8.44 | 6.23 | 1.11 | 1.08 | 0.21 | 0.07 |
| OxfordPets | 138.94 | 86.88 | 26.23 | 25.51 | 2.90 | 1.72 |
| STL10 | 90.14 | 29.96 | 29.80 | 30.01 | 1.00 | 1.99 |
| CIFAR10 | 467.31 | 386.65 | 41.71 | 38.29 | 12.89 | 2.67 |
| CIFAR100 | 2127.96 | 2046.56 | 41.54 | 39.47 | 68.22 | 2.70 |
| 20Newsgroups | 129.64 | 104.07 | 13.10 | 12.43 | 3.47 | 0.85 |
| AG News | 285.11 | 173.42 | 57.19 | 54.40 | 5.78 | 3.72 |

### I.3 Evaluation runtime breakdown on vision datasets

For each dataset, budget, method, and seed, we record:

- **Verification and metrics time** (`verify_metrics_time_sec`): time to compute model means on the selected subset and compute Kendall $\tau$, Spearman $\rho$, winner agreement, pairwise agreement, margin violation rate, and the subset minimum margin magnitude.
- **End-to-end time within the script** (`end2end_time_sec`): selection time plus verification and metrics time.

Table 25 reports a breakdown for CIFAR10, CIFAR100, STL10, and Oxford Pets at budgets near 5% with three seeds. Across all datasets and methods, the verification and metrics stage takes about 1 to 3 milliseconds, while selection dominates `end2end_time_sec`. The table also reports ranking and decision preservation metrics under the same cached score setting, enabling direct comparison of methods at fixed budgets.

Table 25: Runtime breakdown (mean$_{\pm\, std}$ over seeds).

| Dataset | Budget | Method | Select Time (s) | Verify Time (s) | End-to-End Time (s) |
|---|---|---|---|---|---|
| CIFAR10 | 0.05 | VERDICT | $14.489_{\pm 0.052}$ | $0.001_{\pm 0.000}$ | $14.490_{\pm 0.052}$ |
| CIFAR10 | 0.05 | kcenter | $37.454_{\pm 0.757}$ | $0.002_{\pm 0.000}$ | $37.456_{\pm 0.757}$ |
| CIFAR10 | 0.05 | random | $0.000_{\pm 0.000}$ | $0.001_{\pm 0.000}$ | $0.001_{\pm 0.000}$ |
| CIFAR100 | 0.05 | VERDICT | $14.750_{\pm 0.356}$ | $0.001_{\pm 0.000}$ | $14.751_{\pm 0.356}$ |
| CIFAR100 | 0.05 | kcenter | $37.981_{\pm 0.957}$ | $0.003_{\pm 0.001}$ | $37.983_{\pm 0.956}$ |
| CIFAR100 | 0.05 | random | $0.000_{\pm 0.000}$ | $0.001_{\pm 0.000}$ | $0.001_{\pm 0.000}$ |
| OxfordPets | 0.05 | VERDICT | $0.405_{\pm 0.038}$ | $0.001_{\pm 0.000}$ | $0.406_{\pm 0.038}$ |
| OxfordPets | 0.05 | kcenter | $0.262_{\pm 0.007}$ | $0.002_{\pm 0.000}$ | $0.264_{\pm 0.007}$ |
| OxfordPets | 0.05 | random | $0.000_{\pm 0.000}$ | $0.001_{\pm 0.000}$ | $0.001_{\pm 0.000}$ |
| STL10 | 0.05 | VERDICT | $0.786_{\pm 0.034}$ | $0.001_{\pm 0.000}$ | $0.787_{\pm 0.034}$ |
| STL10 | 0.05 | kcenter | $0.534_{\pm 0.005}$ | $0.002_{\pm 0.000}$ | $0.535_{\pm 0.006}$ |
| STL10 | 0.05 | random | $0.000_{\pm 0.000}$ | $0.001_{\pm 0.000}$ | $0.001_{\pm 0.000}$ |

### I.4 Scaling with number of examples $N$

Table 26 evaluates scaling on CIFAR100 by subsampling the evaluation set to fractions $\{0.10, 0.25, 0.50, 1.00\}$ while keeping the cached model set fixed. We report budgets 1% and 5% and three seeds. For VERDICT, selection time increases with the number of examples, while verification and metrics time remains near 0.001 seconds across fractions.

### I.5 Scaling with number of models $M$

Table 27 evaluates scaling on CIFAR100 by truncating the cached score matrix to $M \in \{5, 10, 20\}$ models while keeping the evaluation set fixed. We report budgets 1% and 5% and three seeds. For VERDICT, selection time increases with $M$ in this range, while verification and metrics time remains near 0.001 seconds across values of $M$.

The results in Tables 25–27 measure the cost of selection and verification after scores are cached. They do not include the one time cost of computing cached scores and embeddings. When reporting deployment costs, this precomputation cost should be accounted for separately, since it depends on the model architecture, hardware, and batching strategy used for evaluation.

Table 26: Scaling in $N$ runtime summary (mean$_{\pm\, std}$ over 3 seeds and 3 runs).

| Dataset | Frac. | Budget | Method | End-to-End Time (s) | Select Time (s) | Verify Time (s) |
|---|---|---|---|---|---|---|
| CIFAR100 | 0.10 | 0.01 | VERDICT | $0.23_{\pm 0.07}$ | $0.23_{\pm 0.07}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.10 | 0.01 | kcenter | $0.13_{\pm 0.00}$ | $0.13_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.10 | 0.01 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.10 | 0.05 | VERDICT | $0.73_{\pm 0.04}$ | $0.73_{\pm 0.04}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.10 | 0.05 | kcenter | $0.42_{\pm 0.00}$ | $0.42_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.10 | 0.05 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.25 | 0.01 | VERDICT | $0.98_{\pm 0.04}$ | $0.98_{\pm 0.04}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.25 | 0.01 | kcenter | $0.58_{\pm 0.11}$ | $0.58_{\pm 0.10}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.25 | 0.01 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.25 | 0.05 | VERDICT | $4.76_{\pm 0.03}$ | $4.76_{\pm 0.03}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.25 | 0.05 | kcenter | $2.32_{\pm 0.00}$ | $2.32_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.25 | 0.05 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.50 | 0.01 | VERDICT | $1.49_{\pm 0.05}$ | $1.49_{\pm 0.05}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.50 | 0.01 | kcenter | $1.74_{\pm 0.19}$ | $1.74_{\pm 0.19}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.50 | 0.01 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.50 | 0.05 | VERDICT | $7.34_{\pm 0.08}$ | $7.34_{\pm 0.08}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.50 | 0.05 | kcenter | $8.23_{\pm 0.25}$ | $8.23_{\pm 0.25}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 0.50 | 0.05 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 1.00 | 0.01 | VERDICT | $3.00_{\pm 0.04}$ | $3.00_{\pm 0.04}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 1.00 | 0.01 | kcenter | $6.74_{\pm 0.15}$ | $6.74_{\pm 0.15}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 1.00 | 0.01 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 1.00 | 0.05 | VERDICT | $14.89_{\pm 0.24}$ | $14.89_{\pm 0.25}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 1.00 | 0.05 | kcenter | $37.67_{\pm 0.93}$ | $37.67_{\pm 0.93}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 1.00 | 0.05 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |

Table 27: Scaling in $M$ runtime summary (mean$_{\pm\, std}$ over 3 seeds and 3 runs).

| Dataset | Max Models | Budget | Method | End-to-End Time (s) | Select Time (s) | Verify Time (s) |
|---|---|---|---|---|---|---|
| CIFAR100 | 5 | 0.01 | VERDICT | $2.28_{\pm 0.06}$ | $2.28_{\pm 0.06}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 5 | 0.01 | kcenter | $7.29_{\pm 0.13}$ | $7.29_{\pm 0.13}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 5 | 0.01 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 5 | 0.05 | VERDICT | $11.75_{\pm 0.54}$ | $11.75_{\pm 0.54}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 5 | 0.05 | kcenter | $36.82_{\pm 0.95}$ | $36.81_{\pm 0.95}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 5 | 0.05 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 10 | 0.01 | VERDICT | $2.39_{\pm 0.03}$ | $2.39_{\pm 0.03}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 10 | 0.01 | kcenter | $7.19_{\pm 0.11}$ | $7.19_{\pm 0.11}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 10 | 0.01 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 10 | 0.05 | VERDICT | $11.86_{\pm 0.13}$ | $11.86_{\pm 0.13}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 10 | 0.05 | kcenter | $37.17_{\pm 0.55}$ | $37.17_{\pm 0.55}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 10 | 0.05 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 20 | 0.01 | VERDICT | $2.97_{\pm 0.05}$ | $2.97_{\pm 0.05}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 20 | 0.01 | kcenter | $7.26_{\pm 0.09}$ | $7.26_{\pm 0.09}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 20 | 0.01 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 20 | 0.05 | VERDICT | $14.83_{\pm 0.47}$ | $14.83_{\pm 0.47}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 20 | 0.05 | kcenter | $36.02_{\pm 0.83}$ | $36.02_{\pm 0.83}$ | $0.00_{\pm 0.00}$ |
| CIFAR100 | 20 | 0.05 | random | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ | $0.00_{\pm 0.00}$ |