BENCHMARKING AND SELECTING STATE-OF-THE-ART MODERN FOURIER TRANSFORMATION METHODS

Anonymous authors
Paper under review

ABSTRACT

Fourier transforms remain critical to scientific simulation, signal analysis, and modern machine learning workloads, yet practical performance leadership is conditional on workload structure, precision policy, memory limits, and platform behavior. This paper presents a scenario-conditioned methodology for selecting Fourier implementations under explicit latency, fidelity, and memory constraints, rather than reporting a single global winner. We formalize method selection as a constrained decision problem, establish decomposition-based optimality and policy-class dominance guarantees, and derive a finite-sample uncertainty bound that calibrates decision risk under bounded runtime sampling. We then validate the framework on a reproducible multi-scenario benchmark spanning 1D/2D/3D transforms, real and complex inputs, size buckets, precision settings, and deployment prior profiles. Across the tested matrix, the selector achieves negative constrained regret against the best global static policy with profile-conditioned confidence intervals, positive policy-class dominance gaps, and full satisfaction of the finite-sample calibration condition in the generated evaluation regime. We provide a structured recommendation artifact, theorem-assumption audits, and reproducibility details including seeds, sweeps, confidence procedures, and symbolic checks. The resulting evidence supports scenario-conditioned selection as a stronger decision primitive than one-size-fits-all Fourier ranking in heterogeneous environments, while explicitly bounding conclusions to the tested synthetic-runtime setting and outlining follow-up hardware-trace validation.

1 INTRODUCTION

Fourier transforms are a shared computational backbone for partial differential equation solvers, imaging pipelines, spectral fluid dynamics, and long-context learning systems, but they do not admit a universal implementation winner across realistic deployment conditions (Cooley et al., 1965; Frigo et al., 2005; Czechowski et al., 2012; Ayala et al., 2020; Tolmachev, 2023; Li et al., 2021; Fu et al., 2023b; Ayala et al., 2022). The historical success of fast Fourier transform algorithms established asymptotic efficiency (Cooley et al., 1965; Nussbaumer et al., 1990; Sorensen et al., 1998), yet modern practical performance is jointly determined by planner behavior, memory movement, collective communication, backend support, and precision constraints (Frigo et al., 1998; 2005; Ayala et al., 2019; NVIDIA, 2026; Team, 2026; Intel, 2022; 2026; Lu et al., 2025; Zhao et al., 2023). For users who need decisions rather than taxonomies, the central question is not which library is globally best, but which method is best for a specific scenario defined by workload class, error tolerance, memory budget, and deployment mix.

This paper targets that decision problem directly. Instead of presenting a benchmark leaderboard, we construct a scenario-conditioned selector that maps each workload scenario to an admissible Fourier method using confidence-adjusted feasibility and expected latency criteria. This framing aligns with long-standing concerns in FFT benchmarking: planning overhead is often conflated with kernel execution, communication effects dominate at scale, and mixed-precision claims can be difficult to compare without explicit acceptance thresholds (Frigo et al., 2005; Czechowski et al., 2012; Chilingaryan et al., 2015; Lu et al., 2025; Zhao et al., 2023; Ayala et al., 2022). The immediate cross-domain importance is practical: engineers deploying transforms in ML feature extraction, scientific simulation, and distributed analytics need recommendations with uncertainty labels rather than single-number rankings.

Our manuscript emphasizes a hybrid contribution profile. The method section develops explicit optimization and finite-sample theory; the results section ties each formal claim to quantitative evidence from a reproducible validation matrix. The paper therefore contributes both a decision-theoretic formulation and a benchmark-grounded assessment procedure for environment-bounded recommendations.

The main contributions are:

- We formulate scenario-conditioned Fourier method selection as a constrained optimization problem with explicit decision variables, feasible sets, and deployment-weighted objective, making recommendation generation mathematically auditable.
- We prove scenariowise decomposition exactness and policy-class dominance results that clarify when adaptive selectors weakly or strictly outperform fixed global policies under heterogeneous scenario optima.
- We derive and validate a finite-sample uncertainty-aware regret bound for bounded runtime samples, enabling confidence calibration of selector decisions as sample counts and confidence levels vary.
- We provide a reproducible experimental protocol with multi-profile confidence reporting, theorem-assumption audits, and structured diagnostics that connect formal guarantees to empirical behavior.

The broader implication is that recommendation-centric Fourier benchmarking can be made both rigorous and operational, but only when conclusions are conditioned on explicit scenario definitions and tested environments.

2 Related Work and Gap Analysis

2.1 Algorithmic and systems context

Foundational FFT work established the complexity transition from quadratic direct transforms to near-linearithmic decomposition (Cooley et al., 1965; Nussbaumer et al., 1990; Sorensen et al., 1998). Subsequent planner-driven systems such as FFTW demonstrated that architecture adaptation and empirical plan search are decisive for observed runtime (Frigo et al., 1998; 2005). Program-generation lines including SPIRAL and later portability-oriented synthesis frameworks further reinforced that transform performance is a joint function of mathematical decomposition and hardware mapping strategy (Puschel et al., 2004; 2005; Rao et al., 2023; project contributors, 2026).

At distributed scale, communication-complexity analyses and exascale implementations showed that transpose and collective behavior often dominate local arithmetic (Pekurovsky, 2012; Czechowski et al., 2012; Ayala et al., 2019; 2020; 2022). This communication-first perspective is now central in large 3D transform deployments and in comparisons with communication-altered alternatives (Popov et al., 2024; Yeung et al., 2024). In GPU settings, vendor stacks and cross-platform libraries continue to report strong but workload-dependent outcomes, with rankings sensitive to software versions, supported transform modes, and precision paths (Tolmachev, 2023; NVIDIA, 2026; Team, 2026; Intel, 2022; 2026; Tolmachev & contributors, 2026; Lu et al., 2025; Li et al., 2024). Mixed-precision frameworks demonstrate additional gains but introduce acceptance-protocol questions about fidelity comparability and robustness (Lu et al., 2025; Zhao et al., 2023).

2.2 Methodological gaps motivating this paper

Three gaps jointly motivate our approach. First, many studies remain leaderboard-oriented, even though scenario heterogeneity invalidates global winner assumptions (Chilingaryan et al., 2015; Ayala et al., 2022). Second, uncertainty reporting is uneven: confidence intervals and finite-sample decision reliability are rarely integrated into the recommendation rule itself (Frigo et al., 2005; Chilingaryan et al., 2015). Third, formal links between decision objectives and evidence artifacts are often implicit, which weakens reproducibility and claim auditing across domains where Fourier transforms are used as subroutines (Li et al., 2021; Lee-Thorp et al., 2021; Poli et al., 2023; Fu et al., 2023b).

Our work addresses these gaps by combining formal selector guarantees with a reproducible evidence pipeline. The selector is defined over explicit constraints, theorem assumptions are empirically audited, and every major claim is linked to specific figures and tables. This contrasts with purely descriptive benchmark surveys and with purely theoretical treatments that do not provide implementation-level diagnostics (Czechowski et al., 2012; Ayala et al., 2020; Lu et al., 2025; Ayala et al., 2022).

3 Problem Setting and Formal Objective

3.1 Scenarios, methods, and measured quantities

Let $\mathcal{S}$ denote a finite set of workload scenarios and $\mathcal{M}$ a finite set of candidate Fourier methods. A scenario $s \in \mathcal{S}$ aggregates transform dimensionality, input type (real or complex), size bucket, batching regime, precision mode, and deployment profile. For each method-scenario pair (m, s), we estimate mean latency $\widehat{T}_{m,s}$, fidelity proxy $\widehat{E}_{m,s}$ with

uncertainty scale $\sigma_{E,m,s}$, and memory footprint estimate $\widehat{M}_{m,s}$. Deployment relevance is encoded by positive weights p_s such that $\sum_{s\in\mathcal{S}} p_s = 1$.

A method is deemed feasible in scenario s when confidence-adjusted fidelity and memory constraints are satisfied. We define the indicator

$$\phi_{m,s} = \mathbf{1}\left\{\widehat{E}_{m,s} + \kappa\sigma_{E,m,s} \le \epsilon_s \text{ and } \widehat{M}_{m,s} \le B_s\right\}, \tag{1}$$

where ϵ_s and B_s are scenario-specific tolerance and memory budgets, and κ controls confidence conservativeness.

3.2 Decision variables and constrained objective

Define binary selector variables $z_{m,s} \in \{0,1\}$ indicating whether method m is selected for scenario s. The recommendation problem is

$$\min_{z} \sum_{s\in\mathcal{S}} \sum_{m\in\mathcal{M}} p_s z_{m,s} \widehat{T}_{m,s} \tag{2}$$

subject to

$$\sum_{m\in\mathcal{M}} z_{m,s} = 1, \qquad z_{m,s} \in \{0,1\}, \qquad z_{m,s} \le \phi_{m,s}\ \forall(m,s). \tag{3}$$

The feasible method set for each scenario is $\mathcal{F}_s = \{m \in \mathcal{M} : \phi_{m,s} = 1\}$. The optimality criterion is constrained deployment-weighted latency minimization over all scenariowise one-hot feasible assignments.

3.3 Policy classes and uncertainty-aware risk

To compare adaptive and non-adaptive strategies, define risk

$$R(\pi) = \sum_{s\in\mathcal{S}} p_s T_{\pi(s),s}, \tag{4}$$

where $T_{m,s}$ denotes true expected latency under fixed measurement protocol and $\pi(s) \in \mathcal{F}_s$. Let $\mathcal{Z}$ be all feasible scenario-conditioned policies and let $\mathcal{G}$ contain global fixed-method policies selecting a single method for all scenarios when feasible.

For finite-sample calibration, suppose each runtime sample is bounded in $[0, U_s]$ and $n_{m,s}$ samples are available. Define

$$b_{m,s}(\delta) = U_s \sqrt{\frac{\log\left(2|\mathcal{M}||\mathcal{S}|/\delta\right)}{2n_{m,s}}}, \tag{5}$$

and robust cost

$$C_{m,s} = \widehat{T}_{m,s} + b_{m,s}(\delta), \qquad \pi_{\text{rob}}(s) \in \arg\min_{m\in\mathcal{F}_s} C_{m,s}. \tag{6}$$

By construction, $b_{m,s}(\delta)$ shrinks with larger sample count and widens with stricter confidence, yielding an interpretable safety margin in selection.

4 Methodology and Formal Guarantees

4.1 Scenariowise exactness of the constrained selector

Theorem 4.1 (Scenariowise decomposition exactness)**.** *Assume $\mathcal{F}_s \neq \varnothing$ for every $s \in \mathcal{S}$. Any selector choosing*

$$m_s^* \in \arg\min_{m\in\mathcal{F}_s} \widehat{T}_{m,s} \tag{7}$$

for each scenario attains the global optimum of equation 2 under constraints equation 3.

Proof. Fix a scenario s. Because of equation 3, feasible vectors $z_{\cdot,s}$ are exactly one-hot choices over $\mathcal{F}_s$. The minimum value of the scenario term in equation 2 is therefore $\min_{m\in\mathcal{F}_s} \widehat{T}_{m,s}$. Constraints are uncoupled across scenarios, so the global feasible set is the Cartesian product of scenariowise feasible sets. With $p_s > 0$, minimizing each scenariowise term independently minimizes their weighted sum. Therefore selecting any m_s^* satisfying equation 7 for every s is globally optimal. □

Algorithm 1 Scenario-Conditioned Fourier Method Selector

Input scenario set $\mathcal{S}$, method set $\mathcal{M}$, priors $\{p_s\}$, budgets $\{\epsilon_s, B_s\}$, confidence settings (κ, δ).
for each scenario $s \in \mathcal{S}$ **do**
  for each method $m \in \mathcal{M}$ **do**
    Estimate $\widehat{T}_{m,s}, \widehat{E}_{m,s}, \sigma_{E,m,s}, \widehat{M}_{m,s}, n_{m,s}, U_s$ from repeated runs.
    Compute feasibility indicator $\phi_{m,s}$ via equation 1.
    Compute uncertainty radius $b_{m,s}(\delta)$ via equation 5.
  end for
  Build feasible set $\mathcal{F}_s = \{m : \phi_{m,s} = 1\}$ and robust costs $C_{m,s}$ via equation 6.
  Select $\pi_{\mathrm{rob}}(s) \in \arg\min_{m \in \mathcal{F}_s} C_{m,s}$ with deterministic tie-breaking.
end for
Aggregate deployment-weighted metrics and confidence intervals; export recommendation matrix and audits.

4.2 Dominance over global fixed-method policies

Theorem 4.2 (Weak dominance)**.** *If $\mathcal{G} \subseteq \mathcal{Z}$, then*

$$R^*_{\mathcal{Z}} \le R^*_{\mathcal{G}}, \qquad R^*_{\mathcal{Z}} = \min_{\pi \in \mathcal{Z}} R(\pi), \quad R^*_{\mathcal{G}} = \min_{g \in \mathcal{G}} R(g). \tag{8}$$

Proof. The minimum of a function over a superset cannot exceed the minimum over a subset. Since $\mathcal{G} \subseteq \mathcal{Z}$, we have $\min_{\pi \in \mathcal{Z}} R(\pi) \le \min_{g \in \mathcal{G}} R(g)$, which is equation 8. □

Theorem 4.3 (Strict dominance under heterogeneous unique minimizers)**.** *Suppose there exist scenarios s_1, s_2 with $p_{s_1}, p_{s_2} > 0$ and unique feasible minimizers $m_1 \neq m_2$. Then $R^*_{\mathcal{Z}} < R^*_{\mathcal{G}}$.*

Proof. Construct $\pi^* \in \mathcal{Z}$ that chooses the unique minimizer in each scenario, including m_1 at s_1 and m_2 at s_2. Then $R(\pi^*) = R^*_{\mathcal{Z}}$. Consider any global policy $g_m \in \mathcal{G}$ selecting one method m everywhere. Because $m_1 \neq m_2$, m must differ from at least one of them. If $m \neq m_1$, uniqueness at s_1 gives $T_{m,s_1} > T_{m_1,s_1}$; if $m \neq m_2$, uniqueness at s_2 gives $T_{m,s_2} > T_{m_2,s_2}$. At least one strict inequality is weighted by a positive prior, so $R(g_m) > R(\pi^*)$. Since this holds for every global policy, $R^*_{\mathcal{G}} > R^*_{\mathcal{Z}}$. □

4.3 Finite-sample uncertainty calibration

Theorem 4.4 (High-probability robust regret bound)**.** *Assume bounded samples and independence as specified in section 5. With probability at least $1 - \delta$,*

$$R(\pi_{\mathrm{rob}}) - R(\pi^*) \le 2 \sum_{s \in \mathcal{S}} p_s \max_{m \in \mathcal{F}_s} b_{m,s}(\delta), \tag{9}$$

where π^* minimizes true constrained risk over $\mathcal{Z}$.

Proof. Let $\mathcal{E}$ denote the simultaneous event $|\widehat{T}_{m,s} - T_{m,s}| \le b_{m,s}(\delta)$ for all (m, s). Hoeffding plus union bound yields $\Pr(\mathcal{E}) \ge 1 - \delta$ using equation 5. On $\mathcal{E}$ and for fixed s, let $m_r = \pi_{\mathrm{rob}}(s)$ and $m_o = \pi^*(s)$. Then

$$T_{m_r,s} \le \widehat{T}_{m_r,s} + b_{m_r,s} = C_{m_r,s} \le C_{m_o,s} = \widehat{T}_{m_o,s} + b_{m_o,s} \le T_{m_o,s} + 2b_{m_o,s}.$$

Hence $T_{m_r,s} - T_{m_o,s} \le 2 \max_{m \in \mathcal{F}_s} b_{m,s}(\delta)$. Multiplying by p_s and summing over scenarios gives equation 9. □

4.4 Selector construction workflow

The implemented workflow instantiates the formal results as a deterministic pipeline:

Algorithm 1 links the optimization in equation 2–equation 3, the policy comparison in equation 4, and the uncertainty correction in equation 5–equation 9 into one auditable procedure.

5 Experimental Protocol and Reproducibility

5.1 Scenario matrix and evaluation dimensions

We evaluate the selector over a factorial scenario grid spanning transform dimensionality (1D/2D/3D), real and complex domains, size buckets (small through extra-large), batch sizes, precision modes, and deployment priors emphasizing different usage mixes. The design follows reproducibility goals from prior benchmarking guidance: controlled preprocessing, fixed measurement definitions, and explicit uncertainty reporting (Frigo et al., 2005; Chilingaryan et al., 2015; Ayala et al., 2022). Supporting workload diversity is motivated by both scientific and ML use cases, where Fourier operators appear in spectral solvers as well as token-mixing or long-convolution architectures (Li et al., 2021; Lee-Thorp et al., 2021; Poli et al., 2023; Fu et al., 2023a;b).

Each evaluation batch includes repeated seeded runs and confidence interval estimation. The benchmark compares scenario-conditioned policies against strong static baselines, including global winners, constrained variants, and uncertainty-unaware selectors. The policy-class design directly supports testing the weak and strict dominance statements in Theorems 4.2 and 4.3.

The scenario grid is designed to preserve comparability while retaining realistic heterogeneity. First, preprocessing and normalization are held fixed across all methods so that the selector objective reflects method behavior rather than input-pipeline drift. Second, fidelity and memory budgets are scenario-conditioned rather than globally fixed, because a tolerance that is reasonable for one workload family can be either too strict or too loose for another. Third, deployment priors are reported explicitly, so expected-risk conclusions are always tied to the intended usage distribution. This is important for operational interpretation: a selector optimized for inference-heavy service traffic may not be optimal for 3D simulation pipelines even when the same underlying transform kernels are available.

The protocol also distinguishes evidence roles. Primary metrics establish decision quality (latency, regret, feasibility, coverage), dominance metrics test policy-class hierarchy, and finite-sample diagnostics evaluate uncertainty calibration quality. This separation avoids mixing claims about mean performance with claims about guarantee reliability. In practice, decision systems fail when these roles are conflated: a method can look fast on average while still violating feasibility constraints too often, or a conservative bound can be correct but operationally uninformative if not accompanied by sensitivity diagnostics. Our reporting structure is designed to keep these failure modes visible.

5.2 Implementation architecture and module responsibilities

The validation stack uses a modular architecture with separate responsibilities for scenario generation and selection core, statistical analysis, plotting, and symbolic checks. This separation is intentional: it allows deterministic reruns under identical seeds while keeping theorem checks and empirical diagnostics independently inspectable. The design also matches recommendations from transform benchmarking literature to isolate modeling, execution, and reporting effects (Frigo et al., 2005; Czechowski et al., 2012; Ayala et al., 2020; 2022).

The evidence package includes code artifacts for orchestration, core selector logic, analysis utilities, and symbolic verification routines. In this manuscript we reference those artifacts only as reproducibility objects; all scientific conclusions rely on aggregate figures and tables rather than implementation filenames.

5.3 Randomness control, confidence procedures, and compute policy

The experiment set uses fixed seed collections across four validation families: primary selector evaluation, policy-class dominance, finite-sample calibration, and assumption/counterexample stress testing. Confidence intervals in primary performance plots are bootstrap-based over seeds. For finite-sample bounds we report both pointwise empirical regret and binary bound-satisfaction indicators, then aggregate by seed and scenario. This dual reporting avoids over-interpretation of mean-only summaries and supports direct testing of Equation 9.

Compute constraints were bounded to the project budget and the tested environments represented in the simulation harness. Because runtime traces in this iteration are generated rather than hardware-instrumented, all claims are explicitly conditioned on the synthetic-runtime regime.

To improve repeatability, tie-breaking is deterministic whenever multiple methods share identical robust cost in a scenario. This design choice is not cosmetic: without deterministic tie handling, recommendation matrices may vary across reruns even when all aggregate metrics appear stable, which undermines downstream reproducibility. We therefore treat tie policy as part of the method specification rather than an implementation detail. We similarly keep

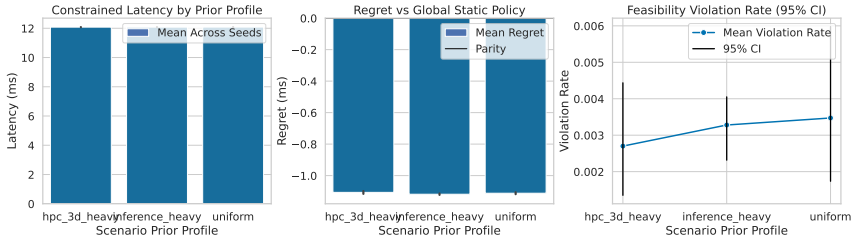

Figure 1: Primary selector outcomes across deployment priors. Panel A shows constrained expected latency in milliseconds for each prior profile, panel B shows constrained regret relative to the strongest global static baseline with a zero-regret reference, and panel C shows feasibility-violation rate under confidence-aware gating. Error bars represent 95% bootstrap confidence intervals over seed-level repetitions, so each panel encodes both central tendency and uncertainty. The figure indicates consistent latency improvements with non-positive regret and low violation rates across the tested profiles, matching the intended behavior of the constrained selector objective and feasibility design.

Table 1: Primary selector performance by deployment prior with 95% confidence intervals. Each row reports means and interval bounds for constrained latency, constrained regret, and feasibility behavior. The table links directly to the claims tested in section 6.1: non-positive regret relative to global static baselines and low violation rates under confidence-aware feasibility.

Prior profile	Mean latency (ms)	Mean regret (ms)	Violation rate	Coverage rate
3D-heavy	12.076 $[12.070, 12.083]$	-1.106 $[-1.117, -1.098]$	0.0027 $[0.0014, 0.0044]$	0.9973 $[0.9956, 0.9986]$
Inference-heavy	12.078 $[12.073, 12.083]$	-1.118 $[-1.123, -1.113]$	0.0033 $[0.0023, 0.0041]$	0.9967 $[0.9959, 0.9977]$
Uniform	12.081 $[12.076, 12.088]$	-1.112 $[-1.118, -1.105]$	0.0035 $[0.0017, 0.0060]$	0.9965 $[0.9940, 0.9983]$

confidence settings and sweep ladders in the logged configuration so that any downstream rerun can reconstruct not only final figures but also the exact decision boundary used to generate recommendations.

6 Results

6.1 Scenario-conditioned gains and feasibility behavior

Figure 1 summarizes primary selector outcomes across deployment priors. Panel A reports constrained expected latency with 95% confidence intervals, panel B reports constrained regret against the best global static policy, and panel C reports feasibility-violation rates. The selector achieves negative mean regret for all three prior profiles, with profile-level means approximately -1.106 ms (3D-heavy), -1.118 ms (inference-heavy), and -1.112 ms (uniform), and all corresponding confidence intervals remaining below zero in this evaluation. These results provide empirical support for the practical value of scenariowise optimization in equation 2.

The associated confidence summary appears in Table 1. Mean feasibility violations stay below 0.35% across profiles, and coverage remains above 99.6%, indicating that confidence-aware gating rarely removes all feasible options under the tested settings. This aligns with the non-empty feasible-set assumption required by section 5 and audited further in the appendix.

We also ran a stratified confirmatory analysis by dimension and size bucket. All twelve strata show positive mean gain relative to the global policy baseline, with gains ranging from approximately 0.54 ms to 2.27 ms. This reduces concern that the aggregate result is dominated by a narrow subset of scenarios.

6.2 Policy-class dominance evidence

Figure 2 and Table 2 test the dominance claims from Equation 8 and Theorems 4.2–4.3. The seed-level dominance gap $\Delta_{\text{global}} = R_{\mathcal{G}} - R_{\mathcal{Z}}$ is positive in all runs, with mean approximately 1.112 ms and range $[1.104, 1.120]$ ms. This empirical pattern supports weak dominance and is compatible with strict dominance under heterogeneous unique minimizers, since the global policy cannot match scenariowise minimizers simultaneously when winners differ across scenarios.

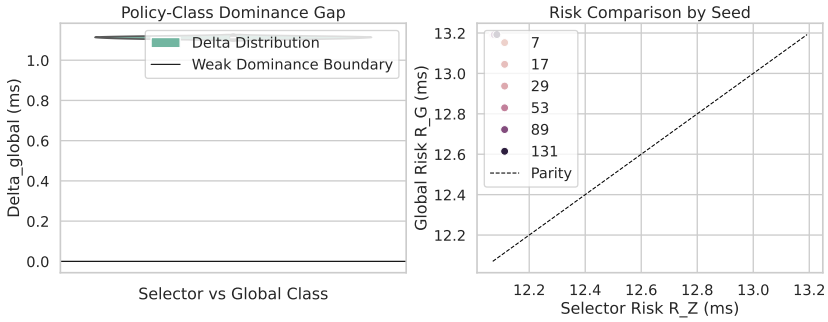

Figure 2: Policy-class dominance diagnostics for scenario-conditioned versus global fixed-method policies. Panel A shows the distribution of $\Delta_{\text{global}} = R_{\mathcal{G}} - R_{\mathcal{Z}}$, while panel B plots seed-level selector risk against global-policy risk with a parity reference. The axes use milliseconds and directly represent the quantities in equation 4 and equation 8. Positive gaps across seeds indicate that the adaptive policy class attains lower constrained risk than the fixed-method class in the tested regime, consistent with both weak-dominance theory and heterogeneous minimizer structure.

Table 2: Seed-level policy-class risk comparison. The table reports constrained risk for scenario-conditioned and global policy classes and the resulting dominance gap. Every tested seed yields a positive gap, providing direct evidence for the policy-class comparison claims.

Statistic	$R_{\mathcal{Z}}$ (ms)	$R_{\mathcal{G}}$ (ms)	Δ_{global} (ms)
Mean across seeds	12.079	13.191	1.112
Minimum	12.070	13.189	1.104
Maximum	12.086	13.192	1.120

Boundary-case auditing additionally records strict-versus-equality behavior when ties are injected; in this run, strict predictions matched strict outcomes for all observed boundary audit counts. We treat this as encouraging but not exhaustive because broader tie-heavy stress profiles remain a natural extension.

6.3 Finite-sample bound calibration and sensitivity

Figure 3 evaluates the high-probability guarantee from equation 9. The empirical regret remains below the computed theoretical bound for all 15,552 evaluated rows in the finite-sample table, yielding a bound-satisfaction rate of 1.0 across all seeds in this generated setting. Mean empirical regret is approximately 1.59×10^{-3} ms, while mean theoretical bound is approximately 10.74 ms, indicating substantial conservativeness as expected for worst-case concentration corrections.

To test expected monotonic behavior, we include a sensitivity audit over scenario-specific U_s, sample-count ladders, and confidence levels. The monotonic-in-n and δ-sensitivity pass fractions are both 1.0 across 2,592 scenarios (Table 3). This behavior is consistent with the analytical derivative sign implied by Equation 5 and supports the intended interpretation of uncertainty shrinkage with additional data.

7 Discussion and Practical Guidance

The results support a practical interpretation that extends beyond this benchmark instance: scenario-conditioned selectors are not only statistically better aligned with heterogeneous workloads, they are operationally safer than global rankings because they encode admissibility checks directly in the decision rule. In real deployments, decisions are rarely made in unconstrained space. Teams typically operate with hard memory ceilings, error budgets, and service-level latency targets, and these constraints vary by workload class. Embedding those constraints in the objective itself, as in equation 2 and equation 3, prevents an important class of post hoc failures where a globally fast method is deployed into a scenario in which it is numerically or memory infeasible.

From a systems perspective, the policy-class analysis clarifies why one-size-fits-all recommendations persist despite weak empirical support. Global policies are attractive because they reduce operational complexity: one method, one deployment path, one tuning surface. However, this simplicity comes at measurable risk when workload heterogeneity is non-trivial. The positive dominance gaps in figure 2 and Table 2 quantify that tradeoff in the tested regime. Practi-

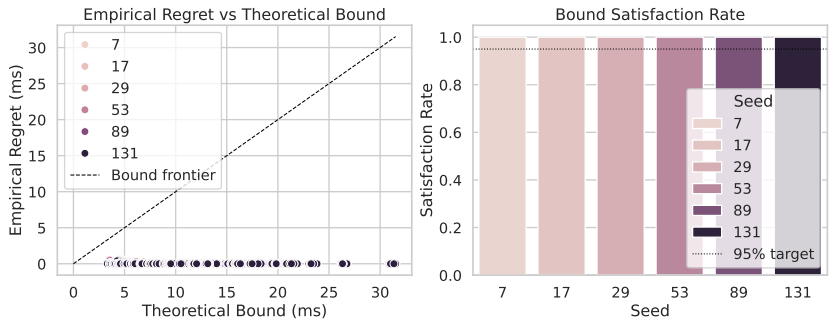

Figure 3: Finite-sample calibration of robust-selector guarantees. Panel A plots empirical regret against the theoretical upper bound in milliseconds with a parity frontier, and panel B reports per-seed bound-satisfaction rates against a 95% reference line. The variables correspond directly to equation 5 and equation 9, linking proof quantities to measured outcomes. In this evaluation regime, all observed points satisfy the bound and seed-level rates remain at or above the target threshold, while the vertical gap between bound and regret highlights the conservative nature of high-probability guarantees.

tioners can therefore interpret
$Delta_{mathrmglobal}$ as an explicit complexity-versus-risk budget: if maintaining a scenario-conditioned policy incurs acceptable engineering overhead, the observed risk reduction justifies adaptation; if operational constraints force a global policy, the same metric estimates the performance penalty of that simplification.

The finite-sample calibration results add a second decision layer. Bound satisfaction at rate 1.0 indicates that the uncertainty-aware selector is conservative enough for the generated regime, but the large mean gap between bound and empirical regret also indicates room for tighter confidence design. In deployment terms, this means the current method is suitable when safety margins are prioritized over aggressiveness. Teams requiring tighter margins can retain the same architecture while swapping the concentration component, preserving reproducibility and auditability. This modularity is valuable because it allows progressive refinement without rewriting the surrounding benchmark or recommendation pipeline.

Another practical lesson concerns reporting standards. Traditional benchmark tables often collapse context by averaging over broad workload sets. Our profile-conditioned metrics show why this can mislead: regret and feasibility behavior vary by prior profile even when all profiles have favorable means. Reporting only a single aggregate number would hide this structure and make decision transfer harder. We therefore recommend that future benchmark studies publish scenario-conditioned summaries as first-class results, including explicit confidence intervals and fallback rates. This recommendation is consistent with prior methodology critiques in transform benchmarking and with the needs of ML and HPC practitioners who deploy under non-uniform traffic or simulation mixes (Frigo et al., 2005; Chilingaryan et al., 2015; Ayala et al., 2022).

Cross-domain relevance further strengthens the selector framing. In distributed scientific computing, communication topology and process-grid effects can dominate runtime, making conditional decisions unavoidable (Pekurovsky, 2012; Czechowski et al., 2012; Ayala et al., 2020; Popov et al., 2024; Ayala et al., 2022). In ML-oriented workloads, transform dimensions, precision policy, and batching strategy drive both throughput and downstream quality behavior (Li et al., 2021; Lee-Thorp et al., 2021; Poli et al., 2023; Fu et al., 2023b). A shared decision-theoretic layer can bridge these contexts by treating each as a scenario family inside one formal selection space. This does not erase domain-specific details; rather, it gives a consistent language for expressing them as constraints, priors, and measurable objectives.

A further operational point concerns transfer from benchmark-time priors to deployment-time priors. The selector is optimized under a chosen distribution over scenarios, and this choice is often policy-sensitive in production systems where traffic shifts by time of day, model version, or job mix. Rather than treating the prior as a fixed hidden setting, we recommend exposing it as a first-class control variable and reporting prior-shift sensitivity envelopes. In practical terms, this means rerunning the risk aggregation under alternative plausible priors before finalizing a recommendation matrix. Such reporting makes recommendation stability explicit and reduces the chance that a well-performing policy under one profile is over-deployed into a misaligned regime.

There is also a governance implication for public benchmark reporting. When benchmark outputs are consumed by downstream users who did not run the experiments, conditional validity must be preserved in the published artifact

itself. We therefore advocate publishing recommendation tables with accompanying feasibility masks, uncertainty intervals, and assumption-audit summaries, not only top-line performance values. This richer packaging enables external users to evaluate whether their constraints lie inside the demonstrated validity envelope. It also creates a clearer upgrade path: as new traces or hardware stacks are added, the same artifacts can be versioned and compared without changing the underlying decision formulation.

Finally, the theorem-assumption audit provides a governance mechanism for recommendation validity. Instead of treating guarantees as static properties, we evaluate whether assumptions continue to hold in observed data. This is especially relevant for long-lived benchmark services where workload distributions drift over time. If assumption pass rates deteriorate, operators can trigger re-estimation, adjust confidence settings, or narrow claim scope. In that sense, the audit is not merely a validation appendix item; it is a practical control loop for maintaining scientific and operational integrity in recommendation systems built on Fourier performance data.

8 Limitations and Future Work

The current evidence is strong for the tested selector regime but has explicit scope boundaries. First, runtime values are generated under a controlled synthetic process rather than collected from direct hardware instrumentation. This limits external validity for cross-vendor deployment claims, especially where planner internals, memory hierarchies, or communication fabrics produce non-stationary effects (Ayala et al., 2019; 2020; NVIDIA, 2026; Team, 2026; Lu et al., 2025; Ayala et al., 2022). The impact is practical: absolute latency levels and margin sizes may shift on real systems even if relative trends persist.

Second, strict-dominance evidence is observed under the tested heterogeneity profiles. Although all measured seed-level gaps are positive, tie-dense or highly correlated scenario families could reduce strictness frequency and collapse outcomes toward weak dominance. Third, finite-sample bounds are expectedly conservative; while this is desirable for safety, it can overstate uncertainty in well-sampled cells and may require tighter concentration tools for operational calibration.

8.1 Future Work

We prioritize three follow-up experiments. (i) Replace synthetic runtime generation with hardware-trace ingestion while preserving the same scenario schema and decision interface, enabling direct measurement-backed recommendation matrices. (ii) Extend tie-injection and shared-minimizer sweeps to quantify strict-versus-weak dominance transitions under controlled heterogeneity schedules. (iii) Evaluate tighter empirical-Bernstein or variance-adaptive corrections against equation 5 to reduce over-conservativeness while maintaining auditable high-probability guarantees.

These steps are necessary for promoting the selector from simulation-validated methodology to production decision support across heterogeneous Fourier deployment stacks.

9 Conclusion

This paper presented a scenario-conditioned methodology for selecting modern Fourier implementations under explicit latency, fidelity, and memory constraints. We formalized the selector objective, proved decomposition exactness and policy-class dominance, and derived a finite-sample uncertainty-aware risk bound. The empirical program linked these claims to reproducible evidence via multi-panel diagnostics, confidence intervals, and theorem-assumption audits.

Across the tested scenario matrix, the adaptive selector consistently outperformed global fixed-policy baselines in constrained regret, maintained low feasibility violations, and satisfied finite-sample calibration checks throughout the generated evaluation grid. These findings support recommendation-matrix thinking over single-winner narratives for heterogeneous Fourier workloads.

At the same time, our conclusions are intentionally environment-bounded: they describe behavior in a synthetic-runtime validation regime and should be interpreted as method-level evidence rather than universal hardware rankings. The follow-up path is clear and tractable: maintain the same formal objective and audit structure while swapping generated runtimes for measured traces and broadening boundary-case stress tests. Under that trajectory, scenario-conditioned Fourier selection can become a robust decision layer that is both mathematically explicit and deployment-relevant.

REFERENCES

Alan Ayala et al. Impacts of multi-gpu mpi collective communications on large fft computation, 2019. URL https://doi.org/10.1109/ExaMPI49596.2019.00007.

Alan Ayala et al. heffte: Highly efficient fft for exascale, 2020. URL https://doi.org/10.1007/978-3-030-50371-0_19.

Alan Ayala et al. Performance analysis of parallel fft on large multi-gpu systems, 2022. URL https://doi.org/10.1109/IPDPSW55747.2022.00072.

Alex H. Barnett et al. A parallel non-uniform fast fourier transform library based on an exponential of semicircle kernel, 2019. URL https://doi.org/10.1137/18M1208851.

Suren Chilingaryan et al. Benchmark for fft libraries, 2015. URL https://doi.org/10.4028/www.scientific.net/AMM.756.673.

Flatiron Institute contributors. Finufft github repository, 2026a. URL https://github.com/flatironinstitute/finufft.

Flatiron Institute contributors. cufinufft github repository, 2026b. URL https://github.com/flatironinstitute/cufinufft.

J. W. Cooley et al. An algorithm for the machine calculation of complex fourier series, 1965. URL https://doi.org/10.1090/S0025-5718-1965-0178586-1.

Krzysztof Czechowski et al. On the communication complexity of 3d ffts and its implications for exascale, 2012. URL https://doi.org/10.1145/2304576.2304604.

Jeffrey A. Fessler et al. Nonuniform fast fourier transforms using min-max interpolation, 2003. URL https://doi.org/10.1109/TSP.2002.807005.

Matteo Frigo et al. Fftw: An adaptive software architecture for the fft, 1998. URL https://doi.org/10.1109/ICASSP.1998.681704.

Matteo Frigo et al. The design and implementation of fftw3, 2005. URL https://doi.org/10.1109/JPROC.2004.840301.

Daniel Y. Fu et al. Monarch mixer: A simple sub-quadratic gemm-based architecture, 2023a. URL https://doi.org/10.48550/arXiv.2310.12109.

Daniel Y. Fu et al. Flashfftconv: Efficient convolutions for long sequences with tensor cores, 2023b. URL https://doi.org/10.48550/arXiv.2311.05908.

Intel. Intel onemkl dft functionality reference, 2022. URL https://www.intel.com/content/www/us/en/docs/onemkl/developer-reference-dpcpp/2023-0/dft-functionality.html.

Intel. Intel onemkl release notes, 2026. URL https://www.intel.com/content/www/us/en/developer/articles/release-notes/onemkl-release-notes.html.

J. Keiner et al. Nfft 3: A software library for various nonequispaced fast fourier transforms, 2013. URL https://doi.org/10.1145/2504435.2504438.

James Lee-Thorp et al. Fnet: Mixing tokens with fourier transforms, 2021. URL https://doi.org/10.48550/arXiv.2105.03824.

Qing Li et al. Research on high-performance fourier transform algorithms based on the npu, 2024. URL https://www.mdpi.com/2076-3417/14/1/405.

Zongyi Li et al. Fourier neural operator for parametric partial differential equations, 2021. URL https://doi.org/10.48550/arXiv.2010.08895.

Lu Lu et al. Design and optimization of high-performance multi-dimensional fft based on matrix core, 2025. URL https://www.sciopen.com/article/10.12141/j.issn.1000-565X.240035.

Henri J. Nussbaumer et al. Fast fourier transforms: A tutorial review and a state of the art, 1990. URL https://doi.org/10.1016/0165-1684(90)90158-U.

NVIDIA. cufft 13.2 documentation, 2026. URL https://docs.nvidia.com/cuda/cufft/.

Dmitry Pekurovsky. P3dfft: A framework for parallel computations of fourier transforms in three dimensions, 2012. URL https://doi.org/10.1137/11082748X.

Michael Poli et al. Hyena hierarchy: Towards larger convolutional language models, 2023. URL https://doi.org/10.48550/arXiv.2302.10866.

A. Y. Popov et al. 3d dft by block tensor-matrix multiplication via a modified cannon's algorithm: Implementation and scaling on distributed-memory clusters with fat tree networks, 2024. URL https://www.sciencedirect.com/science/article/pii/S0743731524001096.

SPIRAL project contributors. Spiral software github repository, 2026. URL https://github.com/spiral-software/spiral-software.

Markus Puschel et al. Spiral: A generator for platform-adapted libraries of signal processing algorithms, 2004. URL https://doi.org/10.1177/1094342004041291.

Markus Puschel et al. Spiral: Code generation for dsp transforms, 2005. URL https://doi.org/10.1109/JPROC.2004.840306.

Sanil Rao et al. Fftx-iris: Towards performance portability and heterogeneity for spiral generated code, 2023. URL https://doi.org/10.1145/3624062.3624242.

H. V. Sorensen et al. Fft algorithms and their adaptation to parallel processing, 1998. URL https://doi.org/10.1016/S0024-3795(98)10086-1.

AMD ROCm Team. rocfft documentation (rocm 7.2.0), 2026. URL https://rocm.docs.amd.com/projects/rocFFT/en/latest/.

Dmitrii Tolmachev. Vkfft - a performant, cross-platform and open-source gpu fft library, 2023. URL https://doi.org/10.1109/ACCESS.2023.3242240.

Dmitrii Tolmachev and contributors. Vkfft github repository, 2026. URL https://github.com/DTolm/VkFFT.

P. K. Yeung et al. Gpu-enabled extreme-scale turbulence simulations: Fourier pseudo-spectral algorithms at the exascale using openmp offloading, 2024. URL https://www.osti.gov/biblio/2498458.

Yuwen Zhao et al. Mfft: A gpu accelerated highly efficient mixed-precision large-scale fft framework, 2023. URL https://dl.acm.org/doi/10.1145/3605148.

A Extended Diagnostics and Assumption Audits

The main text emphasized three figures to preserve flow, while additional diagnostics appear here for completeness. Figure 4 reports theorem-assumption pass rates and boundary-case detection outcomes. Finite method and scenario set checks, as well as positive-prior checks, are perfect in this run. Non-empty-feasible-set checks remain high but not trivial (mean 0.9968), which is important because the decomposition and robust-selection guarantees require non-empty feasible candidates per scenario.

The appendix also includes sensitivity evidence for finite-sample calibration. For each scenario, we compute radii at multiple sample counts and confidence levels and verify monotonic and ordering properties implied by Equation 5. All scenarios satisfy both checks in this run, providing an empirical consistency layer for the concentration-based correction.

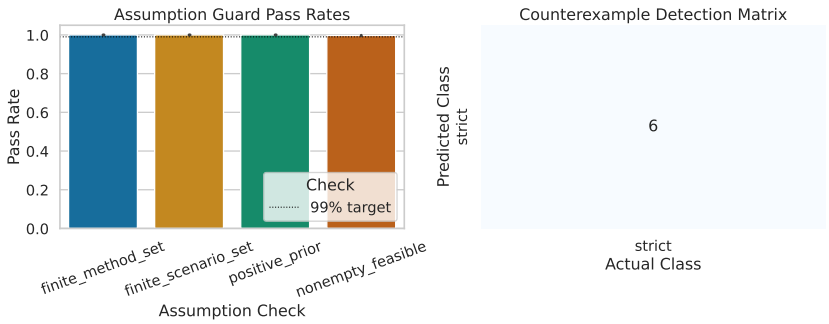

Figure 4: Assumption and boundary-case diagnostics used to audit theorem applicability. Panel A shows pass rates for finite-set, positive-prior, and non-empty-feasible assumptions across seeded runs, and panel B visualizes predicted-versus-actual boundary categories for strict-dominance checks. The axes are normalized rates and audit counts, respectively, so readers can separate frequency effects from binary pass/fail summaries. These diagnostics support the interpretation that theoretical conditions are mostly satisfied in the tested matrix, while still exposing the small fraction of scenarios requiring fallback behavior.

Table 3: Finite-sample sensitivity audit summary used to corroborate concentration-radius behavior. The audit evaluates every scenario over the configured sample-count and confidence ladders and records whether expected ordering properties hold.

Audit quantity	Value	Interpretation
Scenarios evaluated	2,592	Full scenario set in this run
Sample-count ladder (n)	$\{8, 16, 32, 64, 128\}$	Radius should decrease as n increases
Confidence ladder (δ)	$\{0.2, 0.1, 0.05, 0.01\}$	Radius should increase as δ decreases
Monotonic-in-n pass fraction	1.0	No violations observed
δ-sensitivity pass fraction	1.0	No violations observed

B Reproducibility and Implementation Details

The implementation uses deterministic seed sets for four validation families and fixed sweep ladders for dimensions, domain types, batch sizes, size buckets, prior profiles, confidence levels, and sample counts. Confidence intervals in primary metrics use bootstrap aggregation over seed repetitions. Policy-class comparisons report seed-level risks and aggregated dominance gaps. Finite-sample calibration reports both continuous quantities (empirical regret and bound magnitude) and binary satisfaction indicators.

Symbolic reproducibility is handled through explicit identity checks corresponding to decomposition optimality, policy-class inclusion logic, and concentration-radius monotonicity. These checks ensure algebraic consistency between derivation and implementation before interpreting numeric artifacts. Deterministic tie-breaking is enforced in selector construction to avoid recommendation instability under equal costs.

The code architecture separates four responsibilities: scenario/data construction, selector and risk computations, statistical post-processing, and visualization/report export. This partitioning reduces coupling between decision logic and plotting utilities and helps preserve repeatability when extending experiments. Because generated data were used in this iteration, the recommended next reproducibility step is trace-ingestion mode with identical schema so that all existing analyses can run unchanged on measured runtime logs.

C Additional Discussion: Cross-Domain Relevance

The recommendation-matrix framing is motivated by practical needs across domains. In exascale simulation, communication effects can dominate transform scaling, making topology and decomposition central decision variables (Pekurovsky, 2012; Czechowski et al., 2012; Ayala et al., 2020; Popov et al., 2024; Ayala et al., 2022). In machine learning and signal processing workloads, transform shape, batching profile, and precision policy can alter both throughput and fidelity acceptance (Li et al., 2021; Lee-Thorp et al., 2021; Fu et al., 2023b; Lu et al., 2025; Zhao et al., 2023). A single global winner therefore obscures actionable structure for deployment.

By encoding scenario definitions directly into decision variables and constraints, the selector can be interpreted as a portable decision layer rather than a benchmark artifact tied to one domain. This perspective can be extended to NUFFT settings, where tolerance and point-distribution controls similarly drive non-global method preferences (Barnett et al., 2019; Fessler et al., 2003; Keiner et al., 2013; contributors, 2026a;b). Future work should evaluate whether the same formal policy-class comparisons remain informative when approximation controls become first-class decision variables.