LEARNING-AUGMENTED RENT-OR-BUY WITH MULTIPLE PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study online algorithms with multiple predictions for two classical rent-or-buy problems: the Bahncard problem and dynamic TCP acknowledgment. In the Bahncard problem, a generalization of ski rental, trips of varying costs arrive online, and the algorithm may purchase a fixed-price card that discounts travel for a fixed duration; the objective is to minimize the total cost of cards and trips. In dynamic TCP acknowledgment, requests arrive online, and the algorithm may send a fixed-cost acknowledgment for all outstanding requests at any time; the objective is to minimize the total acknowledgment cost and request latency. Both problems are well-studied in both the classical online setting and the learning-augmented setting with a single prediction. But, unlike ski rental, they have not been studied when the algorithm has to choose between more than one prediction. In this paper, we address this gap by establishing tight consistency bounds for these problems for both deterministic and randomized algorithms with two predictions. For Bahncard with any discount factor $\beta \in [0, 1]$, we show that the optimal deterministic consistency is

$$R^*_{\mathrm{BC}}(\beta) = \beta + \frac{1-\beta}{2}\left(\sqrt{\beta^2 + 2\beta + 5} + 1 - \beta\right),$$

which interpolates between the golden ratio at $\beta = 0$ and 1 at $\beta = 1$. In contrast, we show that the optimal deterministic consistency for dynamic TCP acknowledgment is $\sqrt{2}$, revealing that the two problems behave differently. We obtain similar tight bounds for randomized algorithms for both problems, and further establish consistency–robustness tradeoffs when predictions may be inaccurate. Finally, we complement our theoretical results with numerical simulations on synthetically generated instances.

1 INTRODUCTION

Learning-augmented algorithms, also called *algorithms with predictions*, have emerged as a prominent area of research at the intersection of machine learning and algorithm design (Mitzenmacher & Vassilvitskii, 2020; 2022). The central idea is to use machine-learned predictions to improve the quality of online decision-making, bypassing worst-case bounds in classical online algorithms. An important research direction in this area is the use of *multiple predictions* (obtained, e.g., from different ML models and/or human experts) by an online algorithm, with the goal of matching the performance of the best prediction without knowing its identity. This line of work was initiated by Gollapudi & Panigrahi (2019) for the ski rental problem, and has since been applied to many problems including set cover, facility location, scheduling, load balancing, and matching. (Almanza et al., 2021; Anand et al., 2022; Dinitz et al., 2022; Antoniadis et al., 2023; Lattanzi et al., 2026). In spite of this progress, little is known about utilizing multiple predictions in the original rent-or-buy setting it was introduced for, beyond the ski rental problem. In this paper, we address this gap by studying the canonical rent-or-buy problems, Bahncard and dynamic TCP acknowledgment, with multiple predictions.

Rent-or-buy is one of the most elementary forms of online decision-making. A decision maker can repeatedly pay a short-term cost or incur a larger fixed cost that reduces subsequent costs. Buying is preferable when demand persists, while renting is preferable when it ends soon; the difficulty is that

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

the duration and intensity of future demand are unknown. The ski-rental problem isolates this tradeoff in its simplest form. Its optimal deterministic and randomized competitive ratios, 2 and $e/(e-1)$, are foundational examples in competitive analysis (Karlin et al., 1994; Borodin & El-Yaniv, 1998).

In practice, rent-or-buy decisions are often temporal and recurrent. In the Bahncard problem, trips of heterogeneous prices arrive over time and a card bought at time s discounts only the trips in a fixed interval $[s, s+D)$ (Fleischer, 2001). In dynamic TCP acknowledgment, requests arrive over time, one acknowledgment clears the current backlog, and every outstanding request accumulates delay until the next acknowledgment (Dooly et al., 1998). The classical theory gives these problems closely related guarantees. Fleischer proved the optimal deterministic Bahncard ratio $2-\beta$, where β is the fraction of a trip fare paid under a valid card Fleischer (2001). Dynamic TCP acknowledgment has optimal deterministic ratio 2 Dooly et al. (1998). Karlin, Kenyon, and Randall obtained the optimal randomized ratios $e/(e-1+\beta)$ for Bahncard and $e/(e-1)$ for TCP acknowledgment Karlin et al. (2003).

In recent years, both the Bahncard and dynamic TCP acknowledgment problems have been extensively studied in the learning-augmented setting with a *single prediction* (Bamas et al., 2020; Im et al., 2023; Drygala et al., 2023; Zhao et al., 2024; Zhu et al., 2026). In this paper, we consider the Bahncard and dynamic TCP acknowledgment problems with multiple predictions. Our first goal is to obtain a *consistent* algorithm with two predictions. In other words, assuming one of the predictions is correct, how close can we get to an optimal solution? We show that this is already a non-trivial question, and our intuition from ski rental does not directly translate to these problems. Nevertheless, we give *tight* consistency bounds for both deterministic and randomized algorithms for both Bahncard and dynamic TCP acknowledgment problems. Next, we consider the *robustness-consistency* tradeoff with two predictions: without worsening the consistency bounds, can we make the algorithms robust to the possibility of neither prediction being correct? We explore this question both from an algorithm design standpoint and also from a lower bounds perspective that isolate the necessary tradeoffs between robustness and consistency parameters for both problems. Finally, we perform numerical simulations to empirically evaluate our algorithms on synthetic data, thereby complementing our analytical bounds.

1.1 Our contributions

Our main contributions are as follows.

- **Tight consistency bounds for Bahncard.** Recall that $\beta \in [0,1]$ is the discount factor in the Bahncard problem, i.e., the fraction of cost of a trip that has to be paid when holding a card. We give algorithms and matching lower bounds to establish tight consistency ratios of

$$R^*_{\mathrm{BC}}(\beta) := \beta + \frac{1-\beta}{2}\left(\sqrt{\beta^2+2\beta+5}+1-\beta\right).$$

  for deterministic algorithms and $4/(3+\beta)$ for randomized algorithms for the Bahncard problem with two predictions. At $\beta = 0$, these ratios become $\varphi = (1+\sqrt{5})/2$ and $4/3$ respectively, matching the special case of of the ski rental problem (Gollapudi & Panigrahi, 2019). At $\beta = 1$, purchasing a Bahncard provides no benefit, and both ratios equal one.
- **Tight consistency bounds for dynamic TCP acknowledgment.** We give algorithms and matching lower bounds to establish tight consistency ratios of $\sqrt{2}$ for deterministic algorithms and $(1+\sqrt{2})/2$ for randomized algorithms for the dynamic TCP acknowledgment problem with two predictions. Interestingly, these bounds are different from those for the Bahncard problem (and also from the ski rental problem, which is a special case of the latter), thereby establishing a qualitative distinction between these problems.
- **Robustness–consistency tradeoffs for Bahncard.** For deterministic algorithms, we preserve the optimal consistency $R^*_{\mathrm{BC}}(\beta)$ while obtaining robustness $1+(1-\beta)/z_\beta$, where z_β is defined as the positive solution of the equation

$$(1+\beta)z^2 + (1-\beta)z - 1 = 0,$$

  This robustness is tight among algorithms with finite robustness and the same consistency. In particular, at $\beta = 0$ we obtain the tight pair (φ, φ^2), while at $\beta = 1$ both guarantees equal

one. For randomized algorithms, attaining the optimal consistency $4/(3+\beta)$ rules out any finite strict robustness when $\beta < 1$. Nevertheless, we obtain the additive guarantee

$$\mathbb{E}[\mathrm{ALG}] \leq \frac{e}{e-1+\beta}\,\mathrm{OPT} + 1,$$

whose leading coefficient is the optimal competitive ratio without predictions.

- **Robustness–consistency tradeoffs for dynamic TCP acknowledgment.** We give a deterministic algorithm that retains consistency $\sqrt{2}$ and has robustness at most φ^2; any such guarantee is subject to the classical lower bound of 2. We also give a randomized algorithm that retains the optimal consistency $(1+\sqrt{2})/2$ and satisfies

$$\mathbb{E}[\mathrm{ALG}] \leq \frac{e}{e-1}\,\mathrm{OPT} + 1.$$

  Since every instance of dynamic TCP acknowledgment has $\mathrm{OPT} \geq 1$, this implies strict robustness $1 + e/(e-1)$, while the classical robustness lower bound is $e/(e-1)$.
- **Empirical evaluation.** We complement our theoretical results with experiments evaluating the proposed algorithms across different problem instances. We generate problem instances using the experimental testbeds in Zhao et al. (2024) for the Bahncard problem and Im et al. (2023) for dynamic TCP acknowledgment. We compare our algorithms against three types of baselines: the best deterministic and randomized algorithms without prediction, for both problems. Our algorithms either significantly outperform the baselines, or are comparable to the best baseline.

Organization. Section 3 defines the two problems as well as the notions of consistency and robustness. We then give the deterministic consistency results as well as the robustness-consistency tradeoffs for the Bahncard problem in Section 4 and for the dynamic TCP acknowledgment problem in Section 5. The detailed proofs as well as bounds for randomized algorithms are deferred to the appendix. We conclude with experimental results in Section 6.

2 Related work

We have already described prior work on rent-or-buy problems in the classical online setting. So, we summarize existing literature in learning-augmented models for these problems here.

Learning-augmented and multiple-prediction algorithms. The consistency–robustness framework was developed in work on caching and online optimization with machine-learned advice Lykouris & Vassilvitskii (2018); Purohit et al. (2018). Gollapudi & Panigrahi (2019) initiated the multiple prediction model for ski rental and characterized the deterministic consistency bounds through generalized Fibonacci constants. More general portfolio and predictor-combination frameworks were developed in several follow up works across problem domains (Almanza et al., 2021; Anand et al., 2022; Dinitz et al., 2022; Antoniadis et al., 2023; Lattanzi et al., 2026). Our paper focuses on two classical rent-or-buy problems, Bahncard and dynamic TCP acknowledgment, that have not been previously considered in the multiple prediction setting to the best of our knowledge.

Predictions for Bahncard. Bamas, Maggiori, and Svensson's studied both the Bahncard and the dynamic TCP acknowledgment problems in a primal-dual framework that accepts a predicted solution as its input (Bamas et al., 2020). Drygala, Nagarajan, and Svensson study predictions carrying query costs and quantify the information needed to construct a near-optimal Bahncard schedule (Drygala et al., 2023). Zhao et al. (2024) develop PFSUM, which combines observed past demand with a forecast of aggregate trip cost in a future validity window and provides consistency-robustness guarantees. Finally, Zhu et al. (2026) study the Bahncard problem with a sample of the trip sequence given as the offline prediction. Our model instead supplies two complete, conflicting trip sequences and studies the cost of identifying the correct one together with protection when neither is correct.

Predictions for dynamic TCP acknowledgment. Im, Moseley, Xu, and Zhang study dynamic TCP acknowledgment with one predicted request sequence (Im et al., 2023). They introduce a partition-based error measure and obtain deterministic and randomized algorithms with consistency

arbitrarily close to one and robustness arbitrarily close to the optimal classical ratios. In contrast, our setting has two predictions. We determine the exact consistency constants under the promise that one is correct and then hold those constants fixed while studying robustness after both predictions fail.

3 Model and preliminaries

We work in continuous time. Inputs are finite and may contain several events at the same time. For either problem, let I denote the realized instance and let P_1, P_2 be two complete predicted instances supplied offline (i.e., at time 0). An online algorithm observes the events of I chronologically and makes irrevocable decisions without observing future events. We write $\mathrm{ALG}(I; P_1, P_2)$ for its cost and $\mathrm{OPT}(I)$ for the minimum offline cost; problem-specific subscripts are used when necessary. For randomized algorithms, expectations are taken over the algorithm's internal randomness; the adversary is oblivious to the realized random bits.

For a deterministic algorithm $\mathcal{A}$, define *consistency* and *robustness* as follows:

$$\mathrm{cons}(\mathcal{A}) := \sup_{P_1,P_2} \sup_{I\in\{P_1,P_2\}} \frac{\mathcal{A}(I;P_1,P_2)}{\mathrm{OPT}(I)}, \tag{1}$$

$$\mathrm{rob}(\mathcal{A}) := \sup_{P_1,P_2} \sup_{I} \frac{\mathcal{A}(I;P_1,P_2)}{\mathrm{OPT}(I)}. \tag{2}$$

For randomized algorithms, replace each numerator by its expectation. Consistency is the competitive ratio under the promise that one prediction is exact, while robustness permits an arbitrary instance.

3.1 Ski rental and discounted ski rental

We normalize the buying cost to one and the rental rate to one. This is without loss of generality by scaling time and costs appropriately. In discounted ski rental, an algorithm rents at unit rate until it buys; after buying, it pays rate $\beta \in [0, 1]$ until the unknown horizon L. An offline optimum either never buys or buys immediately, and hence $\mathrm{OPT}_{\mathrm{SR},\beta}(L) = \min\{L, 1 + \beta L\}$. For $\beta < 1$, the break-even horizon is $1/(1-\beta)$; for $\beta = 1$, buying is useless. Discounted ski rental is a subclass of Bahncard instances whose card duration exceeds the relevant time horizon. Thus, lower bounds for discounted ski rental transfer to the Bahncard problem.

3.2 The Bahncard problem

A Bahncard instance I is a finite sequence of trips. Trip j occurs at time $\tau_j \in \mathbb{R}_{\geq 0}$ and has regular cost $c_j \geq 0$. A card costs one[1] and remains valid for a fixed duration $D > 0$. If purchased at time s, it is valid on the half-open interval $[s, s + D)$. A trip outside every valid card costs c_j, while a covered trip costs βc_j for a fixed $\beta \in [0, 1]$. For a card schedule S, let $N(S)$ be its number of cards and define

$$\mathrm{cost}_\beta(S; I) := N(S) + \sum_{j \text{ uncovered by } S} c_j + \sum_{j \text{ covered by } S} \beta c_j.$$

We write $\mathrm{OPT}_\beta(I) := \min_S \mathrm{cost}_\beta(S; I)$.

Cards may be purchased at arbitrary real times. This convention does not improve the optimum: a useful card purchased when no trip occurs can be delayed until the next trip it covers, preserving all useful coverage. We may therefore assume that cards are purchased at trip times. Upon observing a trip, an online algorithm sees its time and price and may buy a card before paying for it.

Residual-demand decomposition. Let $T(I) := \sum_{j\in I} c_j$ and let $\widehat{I}$ have the same trip times as I but residual costs $\widehat{c}_j = (1-\beta)c_j$. For every card schedule S,

$$\mathrm{cost}_\beta(S; I) = N(S) + \sum_{j \text{ uncovered}} c_j + \sum_{j \text{ covered}} \beta c_j = \beta T(I) + \mathrm{cost}_0(S; \widehat{I}). \tag{3}$$

[1] Unit cost is without loss of generality by appropriate cost scaling.

The first term is independent of S, so

$$\mathrm{OPT}_\beta(I) = \beta T(I) + \mathrm{OPT}_0(\widehat{I}). \tag{4}$$

This identity reduces every scheduling choice to a full-subsidy Bahncard instance while retaining the unavoidable baseline cost $\beta T(I)$. It is used throughout our consistency and robustness analyses.

3.3 Dynamic TCP acknowledgment

A dynamic TCP acknowledgment instance I is a finite sequence of request release times $0 \le r_1 \le r_2 \le \cdots \le r_n$. At any time, the algorithm may send an acknowledgment, which serves every currently outstanding request. If request j is first acknowledged at time $a(j) \ge r_j$, it incurs latency $a(j) - r_j$. For a finite acknowledgment schedule S, normalized cost is

$$\mathrm{cost}_{\mathrm{TCP}}(S; I) := |S| + \sum_{j=1}^{n} \big(a(j) - r_j\big).$$

Every request must eventually be acknowledged. Therefore, the minimum feasible cost $\mathrm{OPT}_{\mathrm{TCP}}(I) \ge 1$ for any (on-empty) instance I.

A convenient rewriting of latency is in terms of the number of outstanding requests. If $b_S(u)$ is the number of requests outstanding under S at time u, then

$$\sum_{j=1}^{n} (a(j) - r_j) = \int_0^\infty b_S(u)\, du.$$

4 Deterministic Algorithms for Bahncard

In this section, we give deterministic algorithms that achieves tight consistency bounds for any $\beta \in [0, 1]$ for the Bahncard problem with two predictions. Later, we extend our algorithms to also attain robustness guarantees. All missing proofs are in the appendix.

Let t be the first time at which the two predicted trip sequences differ. Before t, the algorithm sees the same trips under either prediction; the observed trips at t identify the realized prediction. Note that by standard convention, trips at any time (and therefore also at t in particular) are observed before a purchase decision at that time.

It will be convenient to define states $e \in [0, D)$ at time t to encode the algorithm's choices before time t. State 0 means that no card purchased strictly before t remains valid after t. State $e > 0$ means that the latest expiration among cards purchased strictly before t is $t + e$.

Now, define $P(e)$ to be the minimum cost of serving all trips before t and ending in state e. For prediction $i \in \{1, 2\}$, define $F_i(e)$ to be the minimum cost of serving trips at and after t, given free initial coverage on $[t, t + e)$. Let $C_i(e) := P(e) + F_i(e)$ be the total cost if prediction i realizes and the algorithm has state e at time t.

We also denote $O_i^0 := \mathrm{OPT}_0(\widehat{P}_i)$. (Recall the residual instances $\widehat{P}_1$ and $\widehat{P}_2$ from (3).) Replacing the prefix of a solution by a minimum-cost prefix with the same boundary state cannot hurt its continuation. Hence $O_i^0 = \min_{e \in [0, D)} C_i(e)$. Fix a minimizing state e_i for each prediction $C_i(e_i) = O_i^0$.

We will choose between e_1 and e_2. The following two inequalities show that the cost of choosing the wrong one can be controlled solely through the difference between their optimal prefix costs.

Lemma 1 (Prefix inequality). *For $0 \le e \le e' < D$, we have $0 \le P(e') - P(e) \le 1$.*

Lemma 2 (Suffix inequality). *For each $i \in \{1, 2\}$ and $0 \le e \le e' < D$, we have $0 \le F_i(e) - F_i(e') \le 1$.*

4.1 Optimal Consistency for all Discount Factors

We are now ready to establish tight consistency bounds for every discount factor $\beta \in [0, 1]$.

Recall that e_1, e_2 are the minimizing states for the two predicted requests schedules. If $e_1 = e_2$, the common state is optimal for both predictions. Otherwise, assume without loss of generality that $e_1 < e_2$, relabeling if necessary. Let

$$p := P(e_1), \qquad d := P(e_2) - P(e_1). \tag{5}$$

By Lemma 1, $d \in [0, 1]$. Moreover, $e_2 > 0$, so every prefix realizing e_2 contains a card and hence

$$p + d = P(e_2) \geq 1. \tag{6}$$

The threshold algorithm for Bahncard. For $\beta = 1$, the algorithm does not buy any card. For $\beta < 1$, it uses the following threshold rule:

$$e^\star = \begin{cases} e_2, & d \leq \lambda_\beta\, p, \\ e_1, & d > \lambda_\beta\, p, \end{cases} \tag{7}$$

for some parameter λ_β that we will fix later. It executes a minimum-cost solution on the common prefix between P_1, P_2 for the interval $[0, t)$ while realizing state $e^\star$. This means if $e^\star > 0$, then the algorithm buys a card at time $t + e^\star - D$ and minimizes the cost of the solution on the common prefix for the interval $[0, t + e^\star - D)$; otherwise, if $e^\star = 0$, then the algorithm minimizes the cost of the solution on the common prefix for the interval $[0, t)$. At time t, once the remaining request schedule is revealed, the algorithm optimizes the solution for the remaining schedule subject to its current state. This amounts to optimizing from time $t + e^\star - D$ onward in the first case, and from time t onward in the second case.

Intuitively, the threshold algorithm compares the additional cost d of the more expensive state with the scale p of the cheaper prefix, and chooses the more expensive state (which also covers requests farther into the future) if their ratio is below the threshold λ_β. The value of λ_β is set by balancing the competitive ratio obtained by the algorithm for the two choices. We give this calculation below.

Recall the residual-demand decomposition from (3), (4) and write $q := 1 - \beta$. For $\beta < 1$, let z_β be the positive solution of

$$(1 + \beta)z^2 + (1 - \beta)z - 1 = 0,$$

which gives

$$z_\beta = \frac{\sqrt{\beta^2 + 2\beta + 5} - (1 - \beta)}{2(1 + \beta)} = \frac{2}{\sqrt{\beta^2 + 2\beta + 5} + 1 - \beta}. \tag{8}$$

Define

$$\lambda_\beta := \frac{1 - z_\beta}{z_\beta} = \frac{z_\beta}{1 + \beta z_\beta}, \qquad R^*(\beta) := 1 + q\lambda_\beta. \tag{9}$$

The equality between the two expressions for λ_β follows directly from (8). Equivalently,

$$R^*(\beta) = \beta + \frac{1 - \beta}{2}\left(\sqrt{\beta^2 + 2\beta + 5} + 1 - \beta\right). \tag{10}$$

At $\beta = 0$, we have $z_0 = \lambda_0 = 1/\varphi$ and $R^*(0) = \varphi$, where φ is the golden ratio.

Theorem 3. *For every $\beta \in [0, 1]$, the threshold algorithm is $R^*(\beta)$-consistent.*

Proof Sketch. The case $\beta = 1$ is immediate since $R^*(1) = 1$ and the algorithm is optimal since it does not buy any case in this case. By Lemmas 1 and 2, choosing e_2 when P_1 occurs costs at most d more than optimum. The optimum pays at least p/q by (4), so $d \leq \lambda_\beta\, p$ bounds the ratio by $1 + q\,\lambda_\beta$.

If e_1 is chosen but P_2 occurs, the excess cost (denote it b) is at most $1 - d$. When $p < 1$, no prefix card can be present in state e_1, so $e_1 = 0$. The optimum for P_2 is at least $(1 + \beta b)/q$; paying all residual fare from state 0 must cost at least $1 + b$. Also $p + d \geq 1$ and $d > \lambda_\beta\, p$, where $1 - d < z_\beta$. As a consequence, the excess ratio is at most

$$\frac{qb}{1 + \beta b} \leq \frac{qz_\beta}{1 + \beta z_\beta} = q\lambda_\beta.$$

When $p \geq 1$, the optimum is at least $(1 + d)/q$, while $d > \lambda_\beta\, p \geq \lambda_\beta \geq \sqrt{2} - 1$. Thus the excess ratio is at most $q(1 - d)/(1 + d) \leq q\lambda_\beta$.

The details of this proof appear in Section A of the appendix. □

Matching lower bound and robustness-consistency tradeoff. We give a discounted ski rental instance (which is a special case of Bahncard) that establishes a matching lower bound (Theorem 6) and therefore optimality of Theorem 3 for every value of $\beta \in [0, 1]$ in Section A of the appendix.

Furthermore, when neither prediction is exact, we give a modified version of the threshold algoirthm in Section B of the appendix that retains the consistency bound in Theorem 3, and is $1 + (1 - \beta)/z_\beta$-robust. We also show that this consistency-robustness tradeoff is optimal across all deterministic algorithms with optimal consistency (Theorem 8).

5 Deterministic Algorithms for Dynamic TCP Acknowledgment

We now consider the dynamic TCP acknowledgment problem. As in the Bahncard problem, our first goal is to establish a tight consistency bound for deterministic algorithms. As earlier, denote by t be the first divergence time of the two predicted request sequences. More precisely, the arrivals agree before t, and the observed arrivals at t identify the realized prediction before any acknowledgment decision at that time.

5.1 Boundary states

Let R^- be the set of requests released before t; by definition of t, these are common between the two predictions and will be realized in the actual input. The effect at t of every acknowledgment sent before t is completely described by the set of outstanding requests. Because one acknowledgment clears the entire current backlog, every possible outstanding set is a suffix of R^- in release-time order. We call such a suffix a *boundary state* and denote it by B. The states have the following reverse inclusion property: a larger set B represents an older last acknowledgment and therefore more outstanding requests. Different acknowledgment times producing the same outstanding set are identified as the same state.

Let $P(B)$ be the minimum cost incurred before t by a schedule ending in state B. This includes all acknowledgment costs before t and all latency accumulated before t, including the latency of requests still outstanding at t. For $i \in \{1, 2\}$, let $F_i(B)$ be the minimum additional cost from time t onward on I_i, starting with backlog B. If an Acknowledgment is sent at time t, then it belongs to $F_i(B)$ and not in $P(B)$. Adopting our notation from Bahncard, we denote

$$C_i(B) := P(B) + F_i(B), \qquad O_i := \mathrm{OPT}(I_i) = \min_B C_i(B),$$

and let $B_i := \arg\min_B C_i(B)$.

A key distinction between the Bahncard and TCP acknowledgment problems is that the prefix cost $P(B)$ may not be monotone in B. This is because it includes the latency of the outstanding requests till time t. Consequently, we only have one-sided bounds the boundary condition that we state below.

Lemma 4 (TCP boundary inequalities)**.** *If $B \supsetneq B'$, i.e., B contains more outstanding requests than B', then*

$$P(B') \le P(B) + 1, \tag{11}$$

$$0 \le F_i(B) - F_i(B') \le 1 \qquad \text{for } i \in \{1, 2\}. \tag{12}$$

Moreover, B is nonempty, every prefix realizing B' as well as every continuation starting from B contains at least one acknowledgment. Thus,

$$P(B') \ge 1, \qquad F_i(B) \ge 1. \tag{13}$$

5.2 An algorithm with optimal consistency

Recall that B_1, B_2 respectively denote the optimal backlog state at time t for the two predicted sequences P_1, P_2 respectively. If $B_1 = B_2$, the algorithm executes an optimal solution on the common prefix that reaches this common state at time t. After time t, the algorithm performs an optimal continuation for the revealed sequence. Clearly, the algorithm is optimal on either prediction.

So, suppose $B_1 \neq B_2$, and assume without loss of generality $B_1 \supsetneq B_2$, relabeling if necessary. Thus, B_1 has more outstanding requests and B_2 has a more recent acknowledgment at time t. Denote the difference in prefix costs $d := P(B_2) - P(B_1) \le 1$, where the last inequality is by (11).

The threshold algorithm for dynamic TCP acknowledgment. Our algorithm uses a threshold $d^* := 2 - \sqrt{2}$. to choose between the states B_1, B_2 as follows:

$$B^* = \begin{cases} B_2, & d \leq d^*, \\ B_1, & d > d^*. \end{cases} \tag{14}$$

Before t, it executes a minimum-cost solution ending at state B^* denoting the set of outstanding requests. At time t, the actual request sequence is identified and the algorithm executes an optimal continuation from state B^*, which incurs costs $F_i(B^*)$.

Theorem 5. *The threshold algorithm for dynamic TCP acknowledgment is $\sqrt{2}$-consistent.*

Sketch. If B_2 is chosen but I_1 occurs, the excess is at most d, while $O_1 \geq 2 - d$ by Lemma 4. Thus the ratio is at most $2/(2-d) \leq \sqrt{2}$ when $d \leq d^*$. If B_1 is chosen but I_2 occurs, the excess is at most $1 - d$ and $O_2 \geq 1$, and hence the ratio at most $2 - d < \sqrt{2}$ when $d > d^*$. Equating $2/(2-d)$ and $2 - d$ yields $d^* = 2 - \sqrt{2}$.

The details of this proof appear in Section E of the appendix. □

Matching lower bound and robustness-consistency tradeoff. We construct a two-request sequence to give a matching lower bound (Theorem 14) on consistency of deterministic algorithms for dynamic TCP acknowledgment in Section E of the appendix. This shows that the threshold algorithms achieves optimal consistency for deterministic algorithms.

In Section F of the appendix, we modify the threshold algorithm to yield Theorem 16 that preserves the optimal $\sqrt{2}$-consistency and simultaneously achieves φ^2-robustness, where φ is the golden ratio.

6 Experiments

In this section, we give an experimental evaluation of the algorithms that we proposed in this paper; further experiments and details of the setup are in Section J of the appendix.

- **Bahncard:** We use the experimental setup of Zhao et al. (2024) with the following parameters: validity $D = 500$, card cost $C = 5000$, and number of days $N = 2000$. There are two types of request sequences: *Commuter* which are periodic requests, and *Occasional* which are non-periodic requests (see Section J for details). We compare our algorithms with the classical deterministic and randomized algorithms (that do not use predictions) from Fleischer (2001) and Karlin et al. (2003) respectively, and also with the single-prediction algorithm PFSUM in Zhao et al. (2024).
- **Dynamic TCP Acknowledgment Problem (DTAP):** We use the experimental setup of Im et al. (2023) with the following parameters: latency cost $1/d = 1/500$ per time-step and number of packet arrivals $T = 1000$. We compare our algorithms with the classical deterministic and randomized algorithms (that do not use predictions) from Dooly et al. (1998) and Karlin et al. (2003) respectively, and also with the single-prediction algorithm ALA in Im et al. (2023).

(For single-prediction algorithms, we average the two predictions given to our algorithms to generate a unique prediction.) We perform two types of experiments. The first set of experiments retain the correct instance as a prediction and add a second noisy prediction (1-EXACT + 1-NOISY); our goal is to compare our deterministic and randomized consistent (but not robust) algorithms with the baselines. In the second set of experiments, we create two independent noisy predictions (2-NOISY) and test our robust algorithms against the baselines. All noisy predictions are generated by injecting random noise starting at 40% of the overall time horizon. This ensures that the predictions are indistinguishable initially but then start deviating at an intermediate stage.

Due to space constraints, in this section, we only give our experimental results for the 1-EXACT + 1-NOISY experiments for the Commuters request sequence in Bahncard and for DTAP. All other experimental results including the 2-NOISY regime appear in Section J of the appendix. Our experimental results in this section are presented in Figure 1 (Bahncard) and Figure 2 (DTAP). (In figures, shadow bands represent 95th percentile confidence intervals.) We highlight our main findings:

- For Bahncard (with $\beta = 0.6$), our Deterministic and Randomized algorithms are almost optimal (1.0000 to 1.0009), with Randomized slightly below Deterministic at large noise level p. These outperform PFSUM (around 1.06) and classical baselines (around 1.08 and 1.12) by an order of magnitude (in suboptimal cost).
- For DTAP, our Deterministic and Randomized algorithms are almost optimal (1.0000 to 1.0017) across all three arrival distributions (Pareto, Poisson, Iterated Poisson). They outperform ALA which degrades to 1.2 for Iterated Poisson arrivals, and the classical deterministic and randomized algorithms which incur even higher ratios. The difference in suboptimal cost between our algorithms and the baselines is several orders of magnitude.

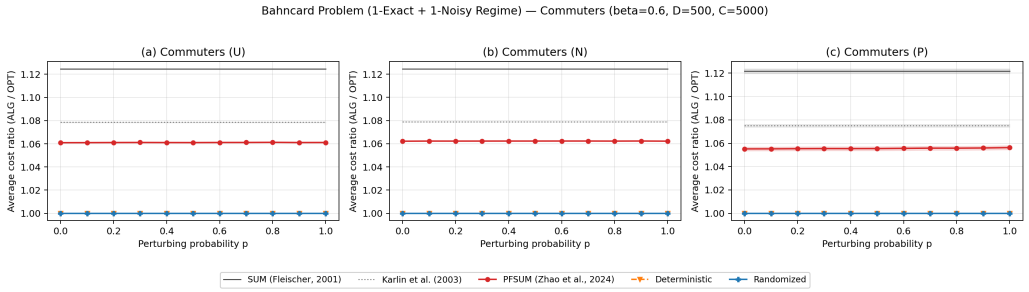


Figure 1: Bahncard [1-EXACT + 1-NOISY Case, $\beta = 0.6$]: Competitive ratio vs noise level p of the noisy prediction comparing **SUM** (Fleischer, 2001), **Karlin** (Karlin et al., 2003), **PFSUM** (Zhao et al., 2024), and our **Deterministic** and **Randomized** algorithms for the Commuter request pattern using Uniform, Normal, and Pareto distributions for generating trips. The lines for **Deterministic** are hidden under those of **Randomized** since they both obtain essentially optimal solutions throughout the range.

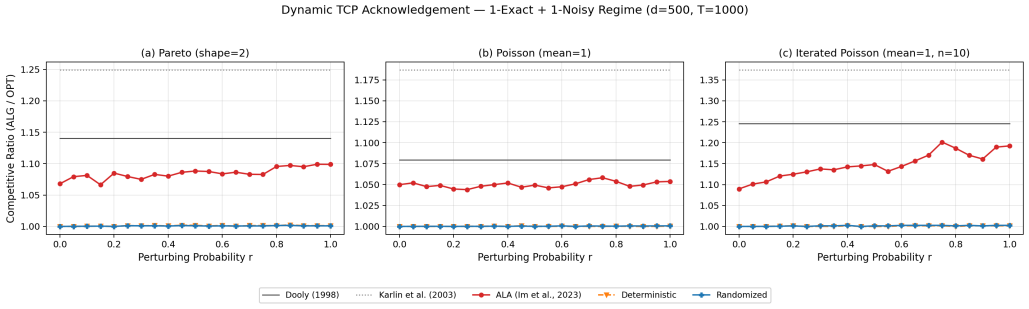


Figure 2: DTAP [1-EXACT + 1-NOISY Case]: Competitive ratio vs replacement rate $r \in [0, 1]$ comparing **Dooly** Dooly et al. (1998), **Karlin** Karlin et al. (2003), **ALA** Im et al. (2023), and our **Deterministic** and **Randomized** algorithms using Pareto, Poisson, and Iterated Poisson distributions for generating packets. The lines for **Deterministic** are hidden under those of **Randomized** since they both obtain essentially optimal solutions throughout the range.

7 Closing Remarks

In this paper, we studied rent-or-buy problems with multiple predictions. In particular, we showed that choosing between two predictions is already a non-trivial task in both the Bahncard and the dynamic TCP acknowledgment problems, and give optimal consistency bounds and robust-consistency tradeoffs across deterministic and randomized algorithms for both problems.

Perhaps the most interesting open question is to consider $k > 2$ predictions for rent-or-buy problems. For the ski rental problem, Gollapudi & Panigrahi (2019) derived tight deterministic bounds for all values of k using limiting behavior of k-acci sequences. Can we obtain the same bounds for the Bahncard problem where the bounds for $k = 2$ match those for ski rental? We have already seen that the DTAP bounds deviate from ski rental bounds for $k = 2$; yet as $k \to \infty$, both problems converge to 2. This suggests studying larger values of k for the DTAP problem might reveal interesting structure that is distinct from the ski rental and Bahncard problems.

AI USE STATEMENT

Generative AI tools were used to assist in this research. Some initial results were obtained via AI tools, and various bounds were sharpened manually to tight ones. AI was also used to write code and generate figures in experimental evaluations. A combination of AI and manual editing was used in writing the text in the final manuscript. The author(s) take full responsibility for all contents.

REPRODUCIBILITY STATEMENT

All algorithmic guarantees in this paper are supported by complete theoretical proofs. We give the full experimental setup and instance constructions needed to reproduce the empirical results. The code used for experiments has been uploaded as supplementary material.

REFERENCES

Matteo Almanza, Flavio Chierichetti, Silvio Lattanzi, Alessandro Panconesi, and Giuseppe Re. Online facility location with multiple advice. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 4661–4673, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/250473494b245120a7eaf8b2e6b1f17c-Abstract.html.

Keerti Anand, Rong Ge, Amit Kumar, and Debmalya Panigrahi. Online algorithms with multiple predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 582–598, 2022. URL https://proceedings.mlr.press/v162/anand22a.html.

Antonios Antoniadis, Christian Coester, Marek Eliáš, Adam Polak, and Bertrand Simon. Mixing predictions for online metric algorithms. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 969–983, 2023. URL https://proceedings.mlr.press/v202/antoniadis23b.html.

Étienne Bamas, Andreas Maggiori, and Ola Svensson. The primal–dual method for learning augmented algorithms. In *Advances in Neural Information Processing Systems 33*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e834cb114d33f729dbc9c7fb0c6bb607-Abstract.html.

Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Algorithms with prediction portfolios. In *Advances in Neural Information Processing Systems 35*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/7f9220f90cc85b0da693643add6618e6-Abstract-Conference.html.

Daniel R. Dooly, Sally A. Goldman, and Stephen D. Scott. TCP dynamic acknowledgment delay: Theory and practice. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 389–398, 1998. doi: 10.1145/276698.276715. URL https://doi.org/10.1145/276698.276715.

Marina Drygala, Sai Ganesh Nagarajan, and Ola Svensson. Online algorithms with costly predictions. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 8078–8101, 2023. URL https://proceedings.mlr.press/v206/drygala23a.html.

Rudolf Fleischer. On the bahncard problem. *Theoretical Computer Science*, 268(1):161–174, 2001. doi: 10.1016/S0304-3975(00)00266-8. URL https://doi.org/10.1016/S0304-3975(00)00266-8.

Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2319–2327, 2019. URL https://proceedings.mlr.press/v97/gollapudi19a.html.

Sungjin Im, Benjamin Moseley, Chenyang Xu, and Ruilong Zhang. Online dynamic acknowledgement with learned predictions, 2023. URL https://arxiv.org/abs/2305.18227.

Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994.

Anna R. Karlin, Claire Kenyon, and Dana Randall. Dynamic TCP acknowledgment and other stories about $e/(e-1)$. *Algorithmica*, 36(3):209–224, 2003. doi: 10.1007/s00453-003-1013-x. URL https://doi.org/10.1007/s00453-003-1013-x.

Silvio Lattanzi, Debmalya Panigrahi, and Ola Svensson. Online rounding and learning-augmented algorithms for facility location. In *International Conference on Learning Representations (ICLR)*, 2026.

Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3302–3311, 2018. URL https://proceedings.mlr.press/v80/lykouris18a.html.

Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. In Tim Roughgarden (ed.), *Beyond the Worst-Case Analysis of Algorithms*, pp. 646–662. Cambridge University Press, 2020.

Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Commun. ACM*, 65(7): 33–35, 2022.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In *Advances in Neural Information Processing Systems 31*, pp. 9684–9693, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/73a427badebe0e32caa2e1fc7530b7f3-Abstract.html.

Hailiang Zhao, Xueyan Tang, Peng Chen, and Shuiguang Deng. Learning-augmented algorithms for the bahncard problem. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/cd96cb9a239c37b39dbf34f3f5a4c56f-Abstract-Conference.html.

Davidson Zhu, Sreenivas Gollapudi, and Debmalya Panigrahi. Learning-augmented rent-or-buy with a sample. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

A BAHNCARD: DEFERRED DETERMINISTIC CONSISTENCY PROOFS

A.1 BOUNDARY-STATE INEQUALITIES

Proof of Lemma 1. Take a prefix schedule realizing state e'. Replace every card that remains valid beyond $t+e$ by one card purchased at $t-D+e$. Every removed card was purchased no earlier than $t-D+e$, so, on the prefix ending at t, its coverage is contained in that of the replacement. At least one card is removed, and the replacement costs no more than the removed cards. The new schedule realizes state e, proving $P(e) \le P(e')$.

Conversely, add to a minimum-cost prefix schedule for e a card purchased at $t-D+e'$. This realizes e' at additional cost one, so $P(e') \le P(e)+1$. □

Proof of Lemma 2. More initial coverage cannot increase the optimal continuation cost, so $F_i(e') \le F_i(e)$. In the other direction, start in state e, purchase a card at $t+e$, and follow an optimal continuation for state e'. Since $e'-e<D$, the added card covers $[t+e, t+e')$ and extends beyond it. This is a feasible continuation of cost at most $F_i(e')+1$. □

A.2 Full proof of the deterministic upper bound

Proof. The case $\beta = 1$ is immediate: never buying a card is optimal. Fix $\beta < 1$, so $q > 0$. For realization i, let T_i be its total undiscounted trip cost and let

$$\overline{O}_i := \mathrm{OPT}_\beta(P_i) = \beta T_i + O_i^0.$$

Because paying for every residual trip without a card is feasible, $O_i^0 \le qT_i$. Consequently,

$$\overline{O}_i = \beta T_i + O_i^0 \ge \frac{O_i^0}{q} \ge \frac{P(e_i)}{q}. \tag{15}$$

If the algorithm chooses the state belonging to the realized prediction, it is optimal. It remains to analyze the two possible mistakes.

Suppose first that $d \le \lambda_\beta p$, so the algorithm chooses the more expensive state e_2, but prediction 1 is realized. The suffix inequality gives

$$C_1(e_2) - O_1^0 = d + F_1(e_2) - F_1(e_1) \le d.$$

The additive term βT_1 is independent of the schedule, so the same additive excess holds for the discounted objective. If $p > 0$, then (15) and the threshold condition imply

$$\frac{\mathrm{ALG}_\beta(P_1)}{\overline{O}_1} \le 1 + \frac{d}{\overline{O}_1} \le 1 + q\frac{d}{p} \le 1 + q\lambda_\beta = R^*(\beta).$$

If $p = 0$, the threshold condition forces $d = 0$, and the algorithm incurs no excess.

Now suppose that $d > \lambda_\beta p$, so the algorithm chooses the cheaper state e_1, but prediction 2 is realized. Set

$$b := C_2(e_1) - O_2^0.$$

By the prefix and suffix inequalities,

$$0 \le b \le 1 - d. \tag{16}$$

We consider separately whether $p < 1$ or $p \ge 1$.

If $p < 1$, a minimum-cost prefix realizing e_1 cannot contain a card, since one card alone costs one. Hence $e_1 = 0$, and p is exactly the total residual demand in the common prefix. Let $M_2 := qT_2$ be the total residual demand of prediction 2. Paying for all residual trips is a feasible continuation from state 0, and therefore

$$M_2 \ge C_2(e_1) = O_2^0 + b \ge 1 + b,$$

where the last inequality uses $O_2^0 \ge P(e_2) = p + d \ge 1$. It follows that

$$\overline{O}_2 = O_2^0 + \frac{\beta}{q} M_2 \ge 1 + \frac{\beta}{q}(1 + b) = \frac{1 + \beta b}{q}. \tag{17}$$

Furthermore, (6) gives $p \ge 1 - d$. Thus

$$d > \lambda_\beta p \ge \lambda_\beta (1 - d) \quad \Longrightarrow \quad 1 - d < \frac{1}{1 + \lambda_\beta} = z_\beta.$$

Since $x \mapsto x/(1 + \beta x)$ is increasing for $x \ge 0$, we obtain from (16) and (17) that

$$\frac{\mathrm{ALG}_\beta(P_2)}{\overline{O}_2} \le 1 + \frac{qb}{1 + \beta b} \le 1 + \frac{q(1 - d)}{1 + \beta(1 - d)} \le 1 + \frac{qz_\beta}{1 + \beta z_\beta} = 1 + q\lambda_\beta = R^*(\beta).$$

It remains to consider $p \ge 1$. By (15), (5),

$$\overline{O}_2 \ge \frac{p + d}{q} \ge \frac{1 + d}{q},$$

and hence

$$\frac{\mathrm{ALG}_\beta(P_2)}{\overline{O}_2} \le 1 + q\frac{1 - d}{1 + d}. \tag{18}$$

The polynomial $f_\beta(z) := (1+\beta)z^2 + (1-\beta)z - 1$ is strictly increasing for $z \geq 0$, and

$$f_\beta(1/\sqrt{2}) = (1-\beta)\left(\frac{1}{\sqrt{2}} - \frac{1}{2}\right) \geq 0.$$

Therefore $z_\beta \leq 1/\sqrt{2}$ and $\lambda_\beta = (1-z_\beta)/z_\beta \geq \sqrt{2}-1$. The threshold condition now gives $d > \lambda_\beta p \geq \lambda_\beta$, and thus

$$\frac{1-d}{1+d} \leq \frac{1-\lambda_\beta}{1+\lambda_\beta} \leq \lambda_\beta;$$

the last inequality is equivalent to $\lambda_\beta^2 + 2\lambda_\beta - 1 \geq 0$. Substitution into (18) proves the claimed bound. □

A.3 Matching deterministic lower bound

The upper bound is already tight on the discounted ski-rental subclass of Bahncard. Take the card duration to exceed the entire instance horizon and view trip cost as demand arriving at unit rate. Define two predicted horizons

$$\theta^* := \frac{z_\beta}{1-\beta}, \qquad L^* := \frac{1+z_\beta}{1-\beta} = \frac{1}{1-\beta} + \theta^*. \tag{19}$$

Theorem 6 (Lower bound)**.** *For every $\beta \in [0,1)$, no deterministic Bahncard algorithm with two exact predictions has competitive ratio smaller than $R^*(\beta)$, even when one card covers the entire instance. For $\beta = 1$, the trivial lower bound of one is tight.*

Proof. Fix $\beta < 1$. Give the algorithm the two predictions with horizons θ^* and L^* from (19). The predictions are indistinguishable before time θ^*.

If the algorithm buys a card strictly before θ^*, realize the short prediction. The algorithm costs at least $1 + \beta\theta^*$, whereas the offline optimum rents throughout and costs θ^*. Its competitive ratio is therefore at least

$$\begin{aligned}\frac{1+\beta\theta^*}{\theta^*} &= \beta + \frac{1-\beta}{z_\beta} \\ &= 1 + (1-\beta)\frac{1-z_\beta}{z_\beta} = R^*(\beta).\end{aligned}$$

Otherwise, realize the long prediction. Once the horizon is identified at θ^*, the best the algorithm can do is either continue renting or buy immediately, and hence its cost is at least

$$\min\{L^*,\, 1 + \beta L^* + (1-\beta)\theta^*\} = L^*.$$

The equality follows from $L^* = 1/(1-\beta) + \theta^*$. The offline optimum buys immediately and costs $1 + \beta L^*$. Consequently,

$$\begin{aligned}\frac{\mathrm{ALG}_\beta(L^*)}{\mathrm{OPT}_\beta(L^*)} &\geq \frac{L^*}{1+\beta L^*} = \frac{1+z_\beta}{1+\beta z_\beta} \\ &= 1 + (1-\beta)\frac{z_\beta}{1+\beta z_\beta} = R^*(\beta),\end{aligned}$$

where the final equality uses (9). The same argument can be implemented with finite Bahncard instances by replacing continuous demand with trips of cost δ and taking $\delta \to 0$. □

Combining Theorems 3 and 6, the exact deterministic competitive ratio with two exact predictions is

$$R_{2,\beta} = \beta + \frac{1-\beta}{2}\left(\sqrt{\beta^2 + 2\beta + 5} + 1 - \beta\right), \qquad \beta \in [0,1].$$

It decreases continuously from $R_{2,0} = \varphi$ to $R_{2,1} = 1$.

B BAHNCARD: DETERMINISTIC CONSISTENCY–ROBUSTNESS TRADEOFF

We now allow both predictions to be wrong. We ask whether the optimal two-prediction consistency guarantees proved in the main text can be retained while also obtaining a worst-case guarantee on arbitrary actual instances. The answer differs sharply for deterministic and randomized algorithms.

As before, the card price is one, its validity duration is D, and a covered trip costs a β fraction of its full fare. Put

$$q := 1 - \beta.$$

For $\beta < 1$, recall that $z_\beta \in (0, 1)$ is the positive root of

$$(1 + \beta)z^2 + (1 - \beta)z - 1 = 0, \tag{20}$$

and that

$$\lambda_\beta = \frac{1 - z_\beta}{z_\beta} = \frac{z_\beta}{1 + \beta z_\beta}, \qquad R^*(\beta) = 1 + q\lambda_\beta. \tag{21}$$

The equivalent balancing identity

$$\beta + \frac{q}{z_\beta} = 1 + \frac{qz_\beta}{1 + \beta z_\beta} = R^*(\beta) \tag{22}$$

follows from (20).

For an algorithm $\mathcal{A}$, define strict consistency and robustness by

$$\mathrm{cons}(\mathcal{A}) := \sup_{P_1,P_2} \sup_{I \in \{P_1,P_2\}} \frac{\mathbb{E}[\mathcal{A}(I; P_1, P_2)]}{\mathrm{OPT}_\beta(I)},$$
$$\mathrm{rob}(\mathcal{A}) := \sup_{P_1,P_2} \sup_{I} \frac{\mathbb{E}[\mathcal{A}(I; P_1, P_2)]}{\mathrm{OPT}_\beta(I)}.$$

For deterministic algorithms the expectation is omitted. Unlike dynamic TCP acknowledgment, a nonempty Bahncard instance can have arbitrarily small optimum cost. Consequently, an additive unit cost cannot in general be absorbed into a multiplicative ratio. This distinction is essential below.

B.1 DETERMINISTIC ALGORITHMS

We first describe a certified implementation of the deterministic boundary-state algorithm. It is convenient to work with the residual fare qc_j of trip j; the term $\beta \sum_j c_j$ is independent of the card schedule.

Certified purchases. Run the boundary-state rule of Section 4.1, but regard a proposed card purchase as pending until it is z_β-*certified*. A pending purchase is certified when the residual fare of the actually observed trips that would otherwise remain uncovered reaches z_β. If the predicted window ends before this happens, cancel the pending purchase. When a card is purchased, suppress any proposed purchase made while that card is valid. At the first divergence, recompute the continuation using the surviving prediction. If both predictions become nonviable, discard them and continue the same z_β-certified rule using only observed trips. Thus every card that is actually purchased has a disjoint certificate consisting of z_β units of previously uncovered residual fare.

The next lemma is the standard delayed-purchase charging lemma, stated in the form needed here. Its first inequality is the parameterized version of the classical Bahncard analysis; the second is its boundary-state refinement.

Lemma 7 (Certified boundary charging)**.** *For every $z \in (0, 1]$, the z-certified boundary implementation has the following properties.*

1. *On an arbitrary actual instance, independently of the predictions,*

$$\mathrm{ALG}_\beta(I) \le \left(1 + \frac{q}{z}\right) \mathrm{OPT}_\beta(I). \tag{23}$$

2. *If one of the two predictions is exact and the boundary threshold is chosen so that* $(1-z)/z = z/(1+\beta z)$*, then*

$$\mathrm{ALG}_\beta(I) \le \max\left\{\beta + \frac{q}{z}, 1 + \frac{qz}{1+\beta z}\right\} \mathrm{OPT}_\beta(I). \tag{24}$$

Proof. We give the charging argument because certification is also what prevents a wrong prediction from causing an unjustified card purchase.

First set $\beta = 0$ and measure demand in residual fare. Partition time at the expirations of cards bought by the certified algorithm. A purchased card is preceded by a certificate of z units of uncovered demand in a window of length at most D; certificates of distinct cards are disjoint. Overlay an offline card schedule. If an offline card intersects a certificate, charge the online card and the certificate to that offline card. Otherwise the entire certificate is paid uncovered by the offline schedule. The usual left-to-right uncrossing ensures that every offline card and every unit of offline uncovered demand receives at most one such block. A block charged to one unit of offline card cost has cost at most $1 + 1/z$ times its charge, and a block charged to uncovered demand satisfies the same bound. Therefore the residual problem is $(1 + 1/z)$-competitive.

For general β, write $T(I) = \sum_j c_j$ and use the residual decomposition

$$\mathrm{cost}_\beta(S; I) = \beta T(I) + \mathrm{cost}_0(S; \widehat{I}), \qquad \mathrm{OPT}_\beta(I) = \beta T(I) + \mathrm{OPT}_0(\widehat{I}),$$

where the fares of $\widehat{I}$ are qc_j. Since paying all residual fare without a card is feasible, $\mathrm{OPT}_0(\widehat{I}) \le qT(I)$, and hence $\beta T(I) \ge (\beta/q)\mathrm{OPT}_0(\widehat{I})$ when $q > 0$. Applying the residual bound and maximizing over the ratio of the baseline to the residual optimum gives

$$\begin{aligned}\frac{\mathrm{ALG}_\beta(I)}{\mathrm{OPT}_\beta(I)} &\le \frac{\beta T(I) + (1+1/z)\mathrm{OPT}_0(\widehat{I})}{\beta T(I) + \mathrm{OPT}_0(\widehat{I})} \\ &\le \frac{\beta/q + 1 + 1/z}{\beta/q + 1} = 1 + \frac{q}{z}.\end{aligned}$$

This proves (23).

Suppose now that one prediction is exact. Perform the same left-to-right uncrossing, but retain the unique boundary block that crosses the first divergence. There are two possible boundary mistakes. If the algorithm commits to the longer-coverage state and the shorter state is optimal, its certificate supplies z units of residual fare; charging the unit card cost and the unavoidable baseline on that certificate gives the first factor $\beta + q/z$. If it commits to the shorter-coverage state and the longer state is optimal, the lost residual coverage is at most z, while the optimum block contains one card and at least $\beta z/q$ units of baseline cost. The excess ratio is therefore at most

$$1 + \frac{qz}{1+\beta z}.$$

All completed blocks telescope exactly as in the prefix and suffix inequalities of Section 4.1; finite card duration creates no additional term because a delayed card expires later than the proposed card. Taking the larger of the two boundary charges proves (24). □

Choosing $z = z_\beta$ balances the two consistency losses.

Theorem 8 (Tight deterministic consistency–robustness pair)**.** *For every* $\beta \in [0, 1)$*, the* z_β*-certified boundary algorithm satisfies*

$$\mathrm{cons}(\mathcal{A}) \le R^*(\beta), \tag{25}$$

$$\mathrm{rob}(\mathcal{A}) \le 1 + \frac{1-\beta}{z_\beta}. \tag{26}$$

Moreover, among deterministic algorithms with finite robustness and consistency at most $R^*(\beta)$*, the robustness bound in* (26) *is best possible. For* $\beta = 1$*, never buying a card is both* 1*-consistent and* 1*-robust.*

Proof. The upper bounds follow from Lemma 7 and the balancing identity (22).

For the lower bound, it suffices to use the discounted ski-rental subclass, in which one card covers the entire horizon and demand arrives at unit rate. Fix $\beta < 1$ and abbreviate $z = z_\beta$ and $R = R^*(\beta)$. Give the algorithm the two predicted horizons

$$S := \frac{z}{q}, \qquad L_\varepsilon := \frac{1 + z + \varepsilon}{q},$$

where $\varepsilon > 0$ will tend to zero. If the algorithm buys at time zero, an actual horizon tending to zero makes its robustness infinite. We may therefore assume that a finite-robustness algorithm does not buy at time zero.

If it buys at a time $\tau \in (0, S)$, realize the exact short prediction. Its cost is at least

$$1 + \beta S + q\tau,$$

whereas the optimum rents and costs S. By (22), the ratio at $\tau = 0$ already equals R; therefore every $\tau > 0$ violates consistency R. Hence the algorithm cannot buy before S.

On the exact long prediction, renting throughout has ratio strictly larger than R for every $\varepsilon > 0$. The algorithm must therefore buy at some time $\tau_\varepsilon < L_\varepsilon$. Consistency implies

$$1 + \beta L_\varepsilon + q\tau_\varepsilon \le R(1 + \beta L_\varepsilon),$$

and thus

$$\tau_\varepsilon \le \frac{R - 1}{q}(1 + \beta L_\varepsilon) = \lambda_\beta(1 + \beta L_\varepsilon) = S + O(\varepsilon).$$

Now realize an unpredicted horizon immediately after the algorithm's first purchase. For sufficiently small ε, this horizon is below the break-even point $1/q$, so the offline optimum rents. Letting first the gap after the purchase and then ε tend to zero gives ratio at least

$$1 + \frac{1}{S} = 1 + \frac{q}{z}.$$

This matches (26). □

At full discount, $z_0 = 1/\varphi$, so the tight pair is

$$\mathrm{cons} = \varphi, \qquad \mathrm{rob} = 1 + \varphi = \varphi^2.$$

Both quantities converge to one as β goes to 1.

C BAHNCARD: RANDOMIZED CONSISTENCY BOUNDS

We now determine the exact randomized competitive ratio for two predictions, one of which is exact, for every discount factor $\beta \in [0, 1]$. The adversary is oblivious: it may choose which prediction is realized after seeing the algorithm and its distribution, but not after seeing the algorithm's realized random bits. The optimal ratio is

$$R^{\mathrm{rand}}_{2,\beta} = 1 + \frac{1 - \beta}{3 + \beta}. \tag{27}$$

Thus the ratio decreases continuously from $4/3$ at $\beta = 0$ to 1 at $\beta = 1$. The endpoint $4/3$ agrees with the tight two-expert ski-rental bound of Gollapudi & Panigrahi (2019).

C.1 RANDOMIZING BETWEEN BOUNDARY STATES

Fix $\beta < 1$ and write $q := 1 - \beta$. Apply the boundary-state construction of Section 4 to the two residual predictions, whose trip costs are multiplied by q. Thus,

$$C_i(e) = P(e) + F_i(e), \qquad O^0_i := \mathrm{OPT}_0(\widehat{P}_i) = \min_e C_i(e),$$

and e_i is a minimizing state for prediction i. Let T_i denote the total undiscounted trip cost of P_i and put

$$\overline{O}_i := \mathrm{OPT}_\beta(P_i) = \beta T_i + O^0_i. \tag{28}$$

If $e_1 = e_2$, the algorithm deterministically selects this common state and is optimal under either realization. Otherwise, relabel the predictions so that $e_1 < e_2$, and define

$$p := P(e_1), \qquad d := P(e_2) - P(e_1). \tag{29}$$

The prefix inequality gives $d \in [0, 1]$. Since $e_2 > 0$, every prefix realizing e_2 contains a card, and therefore

$$p + d = P(e_2) \geq 1. \tag{30}$$

Choosing e_2 can be suboptimal only when P_1 is realized; choosing e_1 can be suboptimal only when P_2 is realized. Define upper bounds $A = A(p, d)$ and $B = B(p, d)$ on the corresponding excess competitive ratios as follows. If $0 < p < 1$, set

$$A := \frac{qd}{p}, \qquad B := \frac{q(1-d)}{1 + \beta(1-d)}. \tag{31}$$

If $p \geq 1$, set

$$A := \frac{qd}{p}, \qquad B := \frac{q(1-d)}{p + d}. \tag{32}$$

The exceptional case $p = 0$ forces $d = 1$ by (30); in this case the algorithm chooses e_1 deterministically.

For $p > 0$, the algorithm chooses the more expensive, longer-coverage state e_2 with probability

$$\alpha(p, d) := \frac{B}{A + B}, \tag{33}$$

and chooses e_1 with probability $1 - \alpha = A/(A + B)$. It executes a minimum-cost residual prefix attaining the sampled state. At the first divergence time, the realized prediction is identified, and the algorithm executes an optimal residual continuation from its sampled state. The same card schedule is used on the original discounted instance.

Lemma 9 (The two boundary losses)**.** *If e_2 is selected and P_1 is realized, then*

$$\frac{\mathrm{cost}_\beta(e_2; P_1)}{\overline{O}_1} \leq 1 + A. \tag{34}$$

If e_1 is selected and P_2 is realized, then

$$\frac{\mathrm{cost}_\beta(e_1; P_2)}{\overline{O}_2} \leq 1 + B. \tag{35}$$

Here A and B are defined by (31) *and* (32)*.*

Proof. Because paying every residual trip without a card is feasible, $O_i^0 \leq qT_i$. The residual-demand decomposition therefore gives

$$\overline{O}_i = \beta T_i + O_i^0 \geq \frac{O_i^0}{q} \geq \frac{P(e_i)}{q}. \tag{36}$$

Suppose first that e_2 is selected and P_1 is realized. By the suffix inequality,

$$\begin{aligned} C_1(e_2) - O_1^0 &= P(e_2) - P(e_1) + F_1(e_2) - F_1(e_1) \\ &\leq d. \end{aligned}$$

The schedule-independent baseline βT_1 is the same for both schedules. Thus (36), with $P(e_1) = p$, gives

$$\frac{\mathrm{cost}_\beta(e_2; P_1)}{\overline{O}_1} \leq 1 + \frac{d}{\overline{O}_1} \leq 1 + \frac{qd}{p} = 1 + A.$$

Now suppose that e_1 is selected and P_2 is realized, and set

$$b := C_2(e_1) - O_2^0.$$

The prefix and suffix inequalities imply

$$0 \le b \le 1 - d. \tag{37}$$

If $p \ge 1$, then (36) yields

$$\overline{O}_2 \ge \frac{P(e_2)}{q} = \frac{p+d}{q},$$

and hence

$$\frac{\mathrm{cost}_\beta(e_1; P_2)}{\overline{O}_2} \le 1 + \frac{qb}{p+d} \le 1 + \frac{q(1-d)}{p+d} = 1 + B.$$

It remains to consider $p < 1$. A minimum-cost prefix attaining e_1 cannot contain a card, since a card alone costs one. Hence $e_1 = 0$, and p equals the total residual demand in the common prefix. Let $M_2 := qT_2$ be the total residual demand in prediction P_2. Paying all residual demand without a card is a feasible schedule attaining state $e_1 = 0$, so

$$M_2 \ge C_2(e_1) = O_2^0 + b.$$

Moreover, $O_2^0 \ge P(e_2) = p + d \ge 1$, and consequently $M_2 \ge 1 + b$. It follows that

$$\begin{aligned}\overline{O}_2 &= O_2^0 + \frac{\beta}{q} M_2 \\ &\ge 1 + \frac{\beta}{q}(1+b) = \frac{1+\beta b}{q}.\end{aligned}$$

Therefore

$$\frac{\mathrm{cost}_\beta(e_1; P_2)}{\overline{O}_2} \le 1 + \frac{qb}{1+\beta b} \le 1 + \frac{q(1-d)}{1+\beta(1-d)} = 1 + B,$$

where the second inequality uses (37) and the monotonicity of $x/(1+\beta x)$. □

Theorem 10 (Randomized upper bound)**.** *For every $\beta \in [0,1]$, the randomized boundary-state algorithm is $4/(3+\beta)$-competitive in expectation against an oblivious adversary.*

Proof. The case $\beta = 1$ is immediate: never purchasing a card is optimal. Fix $\beta < 1$. If $p = 0$, then $d = 1$, and (37) implies that choosing e_1 is optimal under both realizations. We may therefore assume $p > 0$.

If P_1 is realized, the algorithm is optimal when it chooses e_1, and Lemma 9 gives

$$\frac{\mathbb{E}[\mathrm{ALG}(P_1)]}{\overline{O}_1} \le 1 + \alpha A. \tag{38}$$

Similarly, if P_2 is realized,

$$\frac{\mathbb{E}[\mathrm{ALG}(P_2)]}{\overline{O}_2} \le 1 + (1-\alpha) B. \tag{39}$$

The choice (33) balances these two bounds:

$$\alpha A = (1-\alpha) B = \frac{AB}{A+B}. \tag{40}$$

We maximize this balanced loss over all feasible p, d. First suppose $0 < p < 1$. By (30), $p \ge 1 - d$. Since $AB/(A+B)$ is nondecreasing in each of A and B, we may replace p by $1-d$ in the upper bound for A. Writing $z := 1 - d$, we obtain

$$\begin{aligned}\frac{AB}{A+B} &\le q \, \frac{\frac{1-z}{z}\frac{z}{1+\beta z}}{\frac{1-z}{z} + \frac{z}{1+\beta z}} \\ &= \frac{qz(1-z)}{1 - qz + qz^2}.\end{aligned} \tag{41}$$

Let $u := z(1-z)$. Then $u \le 1/4$ and the last expression is

$$\frac{qu}{1-qu},$$

which is increasing in u. Its maximum is attained at $z = 1/2$ and equals

$$\frac{q}{4-q} = \frac{1-\beta}{3+\beta}. \tag{42}$$

Now suppose $p \geq 1$. Equations (32) imply

$$A \leq qd, \qquad B \leq q\frac{1-d}{1+d}.$$

Consequently,

$$\frac{AB}{A+B} \leq q\frac{d(1-d)}{1+d^2} \leq \frac{q}{4} \leq \frac{q}{4-q}. \tag{43}$$

The middle inequality is equivalent to $5d^2 - 4d + 1 \geq 0$, which holds for every real d. Combining (38), (39), (42), and (43), we get

$$\frac{\mathbb{E}[\mathrm{ALG}(P_i)]}{\mathrm{OPT}_\beta(P_i)} \leq 1 + \frac{1-\beta}{3+\beta} = \frac{4}{3+\beta}, \qquad i \in \{1,2\}.$$

□

C.2 Matching lower bound

The upper bound is tight on the discounted ski-rental subclass. Take the card duration to exceed the entire instance horizon and view trip cost as demand arriving at unit rate. This continuous description can be replaced by trips of cost δ spaced δ time units apart and then taking $\delta \to 0$.

Theorem 11 (Randomized lower bound)**.** *For every $\beta \in [0,1)$, no randomized Bahncard algorithm with two predictions, one of which is exact, has expected competitive ratio smaller than $4/(3+\beta)$ against an oblivious adversary. For $\beta = 1$, the lower bound one is tight.*

Proof. Fix $\beta < 1$, write $q = 1 - \beta$, and give the algorithm the two predicted horizons

$$\theta := \frac{1}{2q}, \qquad L := \frac{3}{2q} = \theta + \frac{1}{q}. \tag{44}$$

The predictions are indistinguishable strictly before θ. Since $\theta < 1/q < L$, their offline optimum costs are

$$\mathrm{OPT}_\beta(\theta) = \theta, \qquad \mathrm{OPT}_\beta(L) = 1 + \beta L. \tag{45}$$

Consider an arbitrary randomized algorithm and let

$$r := \Pr[\text{the algorithm purchases a card strictly before } \theta].$$

This probability depends only on the algorithm's distribution and the common history. If the short prediction is realized, then on the event of an early purchase the algorithm pays at least $1 + \beta\theta$, while otherwise it pays at least θ. Hence

$$\frac{\mathbb{E}[\mathrm{ALG}(\theta)]}{\mathrm{OPT}_\beta(\theta)} \geq 1 + r\left(\frac{1+\beta\theta}{\theta} - 1\right) = 1 + rq. \tag{46}$$

Suppose instead that the long prediction is realized. Conditional on no card being purchased before θ, the algorithm has already paid θ in uncovered demand when the prediction is identified. Its least possible total cost is

$$\min\{L,\, 1 + \beta L + q\theta\} = L,$$

where the equality follows from $L = \theta + 1/q$. Conditional on an early purchase, its cost is at least the offline optimum. Therefore

$$\begin{aligned}\frac{\mathbb{E}[\mathrm{ALG}(L)]}{\mathrm{OPT}_\beta(L)} &\geq r + (1-r)\frac{L}{1+\beta L} \\ &= 1 + (1-r)\frac{q}{2+\beta}.\end{aligned} \tag{47}$$

The oblivious adversary chooses whichever of the two fixed predictions gives the larger expected ratio. For every $r \in [0,1]$, the maximum of (46) and (47) is at least its minimum over r. The minimum occurs when

$$rq = (1-r)\frac{q}{2+\beta}, \qquad \text{that is,} \qquad r = \frac{1}{3+\beta}.$$

The common excess ratio is

$$\frac{q}{3+\beta} = \frac{1-\beta}{3+\beta}.$$

Thus one of the two predicted instances gives expected competitive ratio at least

$$1 + \frac{1-\beta}{3+\beta} = \frac{4}{3+\beta}.$$

The adversary selects the instance using the algorithm's distribution, not its realized random bits, so the lower bound holds against an oblivious adversary. Replacing unit-rate demand by trips of cost δ gives finite instances whose lower bound converges to the displayed value as $\delta \to 0$. □

Combining Theorems 10 and 11 gives

$$R_{2,\beta}^{\text{rand}} = \frac{4}{3+\beta} \qquad (\beta \in [0,1]). \tag{48}$$

At the worst boundary configuration in the upper-bound proof, $p = 1 - d = 1/2$. The mixing probability is then

$$\alpha = \frac{1}{3+\beta},$$

exactly the probability of buying before the short horizon in the balanced lower-bound instance. Thus the upper- and lower-bound minimax equations agree with equality.

D BAHNCARD: RANDOMIZED CONSISTENCY–ROBUSTNESS TRADEOFF

D.1 RANDOMIZED ALGORITHMS: AN IMPOSSIBILITY AT OPTIMAL CONSISTENCY

The randomized situation is qualitatively different. The exact consistency value $4/(3+\beta)$ forces an atom at an immediate card purchase, which is incompatible with any finite strict robustness guarantee.

Theorem 12 (Optimal randomized consistency forces infinite robustness)**.** *For every $\beta \in [0,1)$, every randomized Bahncard algorithm satisfying*

$$\text{cons}(\mathcal{A}) \le \frac{4}{3+\beta} \tag{49}$$

has

$$\text{rob}(\mathcal{A}) = \infty. \tag{50}$$

The claim holds against an oblivious adversary. For $\beta = 1$, the optimum consistency and robustness are both one.

Proof. Fix $\beta < 1$, put $q = 1 - \beta$, and use the two ski-rental predictions from the tight randomized consistency lower bound:

$$\theta := \frac{1}{2q}, \qquad L := \frac{3}{2q}.$$

Let

$$r := \Pr[\text{the algorithm buys strictly before } \theta].$$

If the short prediction is realized, the lower-bound calculation gives

$$\frac{\mathbb{E}[\text{ALG}(\theta)]}{\text{OPT}_\beta(\theta)} \ge 1 + rq. \tag{51}$$

If the long prediction is realized, it gives

$$\frac{\mathbb{E}[\text{ALG}(L)]}{\text{OPT}_\beta(L)} \ge 1 + (1-r)\frac{q}{2+\beta}. \tag{52}$$

Since $2 + \beta = 3 - q$, requiring both expressions to be at most $4/(3+\beta) = 1 + q/(3+\beta)$ forces

$$r = \frac{1}{3+\beta}. \tag{53}$$

Equality must also hold in the conditional cost bound used to derive (51). Conditional on buying before θ, a purchase at time τ costs at least

$$1 + \beta\theta + q\tau$$

on the short instance. Therefore equality in (51) forces $\tau = 0$ almost surely on the purchase event. In particular, with probability $1/(3+\beta)$ the algorithm buys a card at time zero.

Now fix the same two predictions but let the actual ski-rental horizon be $\varepsilon > 0$. This third instance is chosen before the random bits are drawn, so the adversary is oblivious. Its offline optimum is ε, whereas the algorithm pays at least one with probability $1/(3+\beta)$. Hence

$$\frac{\mathbb{E}[\mathrm{ALG}(\varepsilon)]}{\mathrm{OPT}_\beta(\varepsilon)} \geq \frac{1}{(3+\beta)\varepsilon} \longrightarrow \infty \qquad (\varepsilon \downarrow 0).$$

This proves (50). □

Thus a finite randomized consistency–robustness tradeoff requires an arbitrarily small relaxation of the optimal consistency value. If additive constants are allowed, however, the exact consistency bound can still be combined with the optimal classical leading coefficient.

Theorem 13 (Additive randomized robustness)**.** *There is a randomized algorithm that retains consistency* $4/(3+\beta)$ *and, for every actual instance and every two predictions, satisfies*

$$\mathbb{E}[\mathrm{ALG}_\beta(I; P_1, P_2)] \leq \frac{e}{e-1+\beta}\mathrm{OPT}_\beta(I) + 1. \tag{54}$$

The leading coefficient $e/(e-1+\beta)$ *is the optimal classical randomized Bahncard ratio.*

Proof. Follow the optimal randomized boundary-state algorithm while at least one prediction is viable. When both predictions fail, retain the current card state and switch to the classical randomized Bahncard algorithm of Karlin, Kenyon, and Randall Karlin et al. (2003). If one prediction is exact, the switch never occurs, so consistency is unchanged.

For an arbitrary actual instance, overlay the prediction-driven and offline card intervals and uncross them from left to right. Every completed prediction-driven interval can be matched to an offline card or to the fare saved on an offline-uncovered block. At the switch, at most one prediction-driven card interval crosses the cut; charge its unit purchase cost additively. The remaining instance is precisely a classical Bahncard instance with an inherited coverage state. Applying the KKR randomized interval-profile inequality to that residual instance and restoring the matched prefix yields

$$\mathbb{E}[\mathrm{ALG}_\beta(I; P_1, P_2)] \leq \frac{e}{e-1+\beta}\mathrm{OPT}_\beta(I) + 1.$$

The classical lower bound, obtained by giving two predictions that are falsified before the hard instance begins, shows that the coefficient of $\mathrm{OPT}_\beta(I)$ cannot be decreased. □

The additive unit in (54) cannot be converted to a strict multiplicative constant because $\mathrm{OPT}_\beta(I)$ may be arbitrarily small. Theorem 12 shows that this is not merely a weakness of the analysis: at the exact randomized consistency optimum, no finite strict robustness factor exists.

E Dynamic TCP acknowledgment: deferred deterministic consistency proofs

E.1 Boundary-state inequalities

Proof of Lemma 4. Choose a time $s < t$ such that acknowledging at s clears exactly the requests in $B \setminus B'$ and leaves precisely B' outstanding at t. Starting from a minimum-cost prefix for B, add an

acknowledgment at s. This costs one and can only reduce the latency accumulated after s. Therefore $P(B') \le P(B) + 1$.

Starting with the smaller backlog B' cannot be worse than starting with B, proving $F_i(B') \le F_i(B)$. In the other direction, start from B, send one acknowledgment immediately at t, and then follow an optimal continuation for B'. The extra acknowledgment also clears the requests of B', which can only help. Thus $F_i(B) \le F_i(B') + 1$.

Strict containment means that B contains a request, while reaching B' requires some pre-t acknowledgment that clears a request in $B \setminus B'$. Finally, a request outstanding in B must eventually be acknowledged. These observations prove (13). □

E.2 Full proof of the deterministic upper bound

Proof. Only the case in which the chosen state is not the optimal state for the realized prediction requires analysis.

First suppose that the algorithm chooses the more expensive prefix state B_2, but prediction 1 is realized. By (12),

$$C_1(B_2) - O_1 = P(B_2) - P(B_1) + F_1(B_2) - F_1(B_1) \le d.$$

The unit lower bounds in Lemma 4 give

$$P(B_1) = P(B_2) - d \ge 1 - d, \qquad F_1(B_1) \ge 1.$$

Consequently,

$$O_1 = P(B_1) + F_1(B_1) \ge 2 - d.$$

If $d \le d^*$, then

$$\frac{C_1(B_2)}{O_1} \le 1 + \frac{d}{2-d} = \frac{2}{2-d} \le \sqrt{2}.$$

Conversely, suppose the algorithm chooses B_1, but prediction 2 is realized. Again using Lemma 4,

$$C_2(B_1) - O_2 = P(B_1) - P(B_2) + F_2(B_1) - F_2(B_2) \le 1 - d.$$

Since every prefix realizing B_2 sends an acknowledgment, $O_2 \ge P(B_2) \ge 1$. If $d > d^*$, it follows that

$$\frac{C_2(B_1)}{O_2} \le 1 + \frac{1-d}{O_2} \le 2 - d < \sqrt{2}.$$

The threshold is obtained by balancing the two worst-case expressions:

$$\frac{2}{2-d} = 2 - d.$$

The solution in $[0, 1]$ is $2 - d = \sqrt{2}$, or $d = 2 - \sqrt{2}$. This proves the theorem. □

E.3 Matching lower bound

We now give a simple matching lower bound.

Theorem 14 (Lower bound)**.** *No deterministic online algorithm for dynamic TCP acknowledgment with two predictions, one of which is correct, has competitive ratio smaller than* $\sqrt{2}$.

Proof. Let

$$\tau := \sqrt{2} - 1.$$

Give the algorithm the following two predictions:

I_1 : one request released at time 0,
I_2 : one request released at time 0 and one request released at time τ.

The predictions are indistinguishable before τ. Their offline optimum costs are

$$O_1 = 1, \qquad O_2 = 1 + \tau = \sqrt{2}.$$

For I_1, acknowledge immediately at time 0. For I_2, one acknowledgment at time τ incurs unit acknowledgment cost and latency τ for the first request; this is cheaper than using two acknowledgments because $\tau < 1$.

Consider any deterministic algorithm. If it sends an acknowledgment at some time strictly before τ, realize I_2. The request released at time τ requires another acknowledgment, so the algorithm pays at least two. Its ratio is at least

$$\frac{2}{O_2} = \frac{2}{\sqrt{2}} = \sqrt{2}.$$

If it sends no acknowledgment before τ, realize I_1. The request at time 0 has already accumulated latency τ and still requires an acknowledgment. The algorithm therefore pays at least

$$1 + \tau = \sqrt{2} = \sqrt{2}\, O_1.$$

The realized sequence is one of the two predictions in both cases, proving the lower bound. □

Combining Theorems 5 and 14 gives the exact value

$$R_2^{\mathrm{TCP}} = \sqrt{2}. \tag{55}$$

F Dynamic TCP acknowledgment: deterministic consistency–robustness tradeoff

F.1 Robustness Bounds for Dynamic TCP Acknowledgment

We now remove the promise that one of the two predictions is correct. For a deterministic algorithm $\mathcal{A}$, define its consistency and robustness by

$$\mathrm{cons}(\mathcal{A}) := \sup_{P_1,P_2} \sup_{I \in \{P_1,P_2\}} \frac{\mathcal{A}(I; P_1, P_2)}{\mathrm{OPT}(I)},$$
$$\mathrm{rob}(\mathcal{A}) := \sup_{P_1,P_2} \sup_{I} \frac{\mathcal{A}(I; P_1, P_2)}{\mathrm{OPT}(I)}.$$

Thus consistency is measured only on instances for which one prediction is exact, whereas robustness is measured on arbitrary triples (I, P_1, P_2). The question is to minimize $\mathrm{rob}(\mathcal{A})$ subject only to $\mathrm{cons}(\mathcal{A}) \le \sqrt{2}$; in particular, we do not prescribe the actions of $\mathcal{A}$ before the predictions fail.

The first conclusion is a general lower bound that is independent of how the algorithm uses its predictions.

A robust recovery algorithm. Let $\mathcal{B}$ denote the $\sqrt{2}$-consistent boundary strategy from Section 5. At time s, call prediction P_i *viable* if its restriction through s agrees with the observed request sequence. Define

$$\sigma := \inf\{s \ge 0 : P_1 \text{ and } P_2 \text{ are both nonviable at time } s\}.$$

The algorithm follows $\mathcal{B}$ before σ. If $\sigma < \infty$, it irrevocably discards the predictions and runs a deterministic KKR savings-threshold policy from the state inherited at σ. In particular, outstanding requests and latency accumulated before σ are not erased.

For $z > 0$, the savings-threshold policy A_z sends its next acknowledgment at the first time at which an additional acknowledgment, placed optimally since the preceding acknowledgment, would have saved z units of latency. This is the deterministic family introduced by Karlin, Kenyon, and Randall Karlin et al. (2003); A_1 is the standard deterministic 2-competitive policy.

Lemma 15 (One-switch charging bound)**.** *For every $z > 0$, the algorithm that follows $\mathcal{B}$ until σ and then runs A_z from the inherited state satisfies*

$$\mathrm{ALG}(I; z) \le \max\left\{2 + z,\, 1 + \frac{1}{z}\right\} \mathrm{OPT}(I)$$

on every realized instance I.

Proof. We use the staircase representation of TCP acknowledgment from Karlin et al. (2003): the height at a time s is the number of outstanding requests, and latency is the area below this staircase. Overlay the online and offline staircases.

First consider acknowledgment blocks completed by $\mathcal{B}$ strictly before σ. Until σ, at least one prediction agrees with the observed prefix. The exchange argument underlying the boundary-state lemma matches every completed boundary block with an offline acknowledgment or with one unit of offline latency. Exchanging the matched blocks from left to right leaves at most one unmatched online acknowledgment: the last acknowledgment of the prediction-driven phase. All outstanding requests and their accrued latency remain in the residual instance.

Apply the KKR rectangle charging argument to this residual instance. Each post-switch acknowledgment closes a rectangle of saving z. A residual component can end in one of two ways. If the offline schedule acknowledges before the online threshold is reached, then the unmatched boundary acknowledgment, one post-switch acknowledgment, and at most z units of latency are charged to a unit offline acknowledgment, for a factor at most $2 + z$. Otherwise, the saving rectangle is completed first. The standard KKR basis-input inequality charges the acknowledgment and rectangle area to offline latency with factor at most $1 + 1/z$. The residual staircase is a nonnegative sum of these rectangles, so summing their charges and restoring the previously matched prefix blocks proves the lemma. □

Theorem 16 (Deterministic consistency and robustness)**.** *Let*

$$\varphi := \frac{1+\sqrt{5}}{2}.$$

The algorithm that follows $\mathcal{B}$ while a prediction remains viable and switches to $A_{1/\varphi}$ after both predictions fail is $\sqrt{2}$-consistent and φ^2-robust. Equivalently,

$$\mathrm{cons}(\mathcal{A}) \le \sqrt{2}, \qquad \mathrm{rob}(\mathcal{A}) \le \varphi^2 = \frac{3+\sqrt{5}}{2}.$$

Proof. If one prediction is exact, then $\sigma = \infty$, so the algorithm is identical to $\mathcal{B}$. Its consistency is therefore $\sqrt{2}$ by Theorem 5. On an arbitrary input, Lemma 15 gives robustness

$$\Gamma(z) := \max\left\{2 + z,\, 1 + \frac{1}{z}\right\}.$$

The first term is increasing and the second is decreasing. Balancing them gives $z^2 + z - 1 = 0$, and hence $z = 1/\varphi$. At this value,

$$\Gamma(1/\varphi) = 2 + \frac{1}{\varphi} = 1 + \varphi = \varphi^2.$$

□

The classical deterministic lower bound of 2 continues to apply: fix two predictions that are falsified before the hard request sequence begins. The algorithm then induces an ordinary deterministic online TCP acknowledgment algorithm. Consequently, if $R_{\mathrm{rob}}^{\mathrm{TCP,det}}$ denotes the smallest robustness attainable without weakening consistency $\sqrt{2}$, the bounds currently proved are

$$2 \le R_{\mathrm{rob}}^{\mathrm{TCP,det}} \le \varphi^2. \tag{56}$$

For comparison, switching to the classical policy A_1 gives the weaker upper bound 3. This factor is attained by the following simple example for the literal A_1 fallback. Take predictions $P_1 = \{0, T\}$ and $P_2 = \{0, 2T\}$, where $T > 1$. The boundary strategy acknowledges at time 0. Release an unexpected request at time $\varepsilon > 0$ and then stop. Policy A_1 incurs one unit of latency before its recovery acknowledgment, so its total cost is 3, whereas one offline acknowledgment at time ε costs $1 + \varepsilon$. Thus the ratio tends to 3 as $\varepsilon \downarrow 0$.

G DYNAMIC TCP ACKNOWLEDGMENT: RANDOMIZED CONSISTENCY BOUNDS

We now allow the online algorithm to randomize, while retaining the promise that one of the two complete predictions is the realized request sequence. Competitive ratios are in expectation against an oblivious adversary: the adversary may choose which of the two predictions is realized after seeing the algorithm and its probability distribution, but not after seeing its realized random bits. Randomization improves the tight ratio from $\sqrt{2}$ to

$$\rho_2^{\mathrm{rand}} := \frac{1+\sqrt{2}}{2} \approx 1.20710678. \tag{57}$$

G.1 THE RANDOMIZED BOUNDARY-STATE ALGORITHM

We retain the notation of Section 5. Let t be the first divergence time, let B_i be an optimal boundary state for prediction I_i, and recall that

$$C_i(B) = P(B) + F_i(B), \qquad C_i(B_i) = O_i = \mathrm{OPT}(I_i).$$

If $B_1 = B_2$, the algorithm follows an optimal prefix attaining this common state and is optimal under either realization. Otherwise, relabel the two predictions so that

$$B_1 \supsetneq B_2,$$

where B_1 has the larger backlog and B_2 corresponds to the more recent last acknowledgment. As before, define

$$d := P(B_2) - P(B_1) \in [0,1]. \tag{58}$$

The two possible errors have the following normalized upper bounds:

$$a(d) := \frac{d}{2-d}, \qquad \text{if the algorithm chooses } B_2 \text{ and } I_1 \text{ occurs,} \tag{59}$$

$$b(d) := 1-d, \qquad \text{if the algorithm chooses } B_1 \text{ and } I_2 \text{ occurs.} \tag{60}$$

Here $a(d)$ and $b(d)$ are the excesses over ratio one. Indeed, the proof of Theorem 5, using Lemma 4, gives

$$C_1(B_2) \le O_1 + d, \qquad O_1 \ge 2-d, \tag{61}$$

$$C_2(B_1) \le O_2 + 1 - d, \qquad O_2 \ge 1. \tag{62}$$

The randomized algorithm chooses the more recent state B_2 with probability

$$q(d) := \frac{b(d)}{a(d)+b(d)} = \frac{(1-d)(2-d)}{2-2d+d^2}, \tag{63}$$

and chooses B_1 with the complementary probability

$$1 - q(d) = \frac{d}{2-2d+d^2}. \tag{64}$$

The formulas include the endpoints: $q(0) = 1$ and $q(1) = 0$. Before t, the algorithm executes a minimum-cost prefix attaining the sampled state. At time t, the observed arrivals identify the realized prediction I_i, and the algorithm then executes an optimal continuation of cost $F_i(B)$ from the sampled state B.

The choice in (63) is the minimax mixture for the two boundary mistakes: it makes

$$q(d)a(d) = (1-q(d))b(d) = \frac{a(d)b(d)}{a(d)+b(d)}. \tag{65}$$

Theorem 17 (Randomized upper bound)**.** *For dynamic TCP acknowledgment with two complete predictions, one of which is correct, the randomized boundary-state algorithm is $\frac{1+\sqrt{2}}{2}$-competitive in expectation against an oblivious adversary.*

Proof. Fix the two predictions and the realized prediction. If the realized prediction is I_1, choosing B_1 gives the offline optimum, whereas (61) bounds the cost when B_2 is chosen. Consequently,

$$\frac{\mathbb{E}[\mathrm{ALG}(I_1)]}{O_1} \le (1-q(d)) + q(d)\left(1+\frac{d}{2-d}\right) = 1 + q(d)a(d). \tag{66}$$

If the realized prediction is I_2, choosing B_2 is optimal, while (62) bounds the cost when B_1 is chosen. Thus

$$\frac{\mathbb{E}[\mathrm{ALG}(I_2)]}{O_2} \le q(d) + (1-q(d))(1+1-d) = 1 + (1-q(d))b(d). \tag{67}$$

By (65), both expressions are at most

$$1 + h(d), \qquad h(d) := \frac{a(d)b(d)}{a(d)+b(d)} = \frac{d(1-d)}{2-2d+d^2}. \tag{68}$$

It remains to maximize this expression over $d \in [0,1]$, since the adversary chooses d, to obtain an upper bound on the competitive ratio of the algorithm. Put $x = 1-d$. Then

$$h(d) = \frac{x(1-x)}{1+x^2}, \qquad x \in [0,1]. \tag{69}$$

Differentiation gives

$$\frac{d}{dx}\left(\frac{x(1-x)}{1+x^2}\right) = \frac{1-2x-x^2}{(1+x^2)^2}.$$

The unique interior maximizer is therefore

$$x = \sqrt{2}-1, \qquad \text{equivalently} \qquad d = 2-\sqrt{2}.$$

At this point,

$$\max_{0\le d\le 1} h(d) = \frac{\sqrt{2}-1}{2}. \tag{70}$$

Combining (66), (67), and (70) proves

$$\frac{\mathbb{E}[\mathrm{ALG}(I_i)]}{\mathrm{OPT}(I_i)} \le 1 + \frac{\sqrt{2}-1}{2} = \frac{1+\sqrt{2}}{2} \qquad (i \in \{1,2\}).$$

□

G.2 Matching randomized lower bound

The upper bound is tight even for a pair of predictions containing at most two requests. The lower bound below is stated directly for randomized algorithms against an oblivious adversary; equivalently, it may be phrased through the minimax principle.

Theorem 18 (Randomized lower bound)**.** *No randomized online algorithm for dynamic TCP acknowledgment with two complete predictions, one of which is correct, has expected competitive ratio smaller than $\frac{1+\sqrt{2}}{2}$ against an oblivious adversary.*

Proof. Let

$$\tau := \sqrt{2}-1, \tag{71}$$

and give the algorithm the two predictions

I_1 : one request released at time 0,

I_2 : one request released at time 0 and one request released at time τ.

The request histories are identical strictly before τ, and

$$\mathrm{OPT}(I_1) = 1, \qquad \mathrm{OPT}(I_2) = 1+\tau = \sqrt{2}. \tag{72}$$

For I_2, the latter cost is attained by sending one acknowledgment at time τ; sending separate acknowledgments costs two and is more expensive because $\tau < 1$.

Consider any randomized online algorithm and let

$$q := \Pr[\text{the algorithm sends an acknowledgment strictly before time } \tau].$$

This probability is determined by the common pre-τ history and the algorithm's distribution; it does not depend on which prediction is realized.

Suppose first that I_1 is realized. On the event that no acknowledgment is sent before τ, the request at time zero accumulates latency at least τ and still requires an acknowledgment. On the complementary event, the algorithm incurs at least one acknowledgment cost. Hence

$$\frac{\mathbb{E}[\mathrm{ALG}(I_1)]}{\mathrm{OPT}(I_1)} \geq 1 + (1-q)\tau. \tag{73}$$

Now suppose that I_2 is realized. If an acknowledgment is sent strictly before τ, the request released at τ requires a further acknowledgment, so the total cost is at least two. If none is sent before τ, the request released at zero has accumulated latency at least τ, and at least one acknowledgment is still necessary; the cost is at least $1 + \tau$. Dividing by (72) yields

$$\frac{\mathbb{E}[\mathrm{ALG}(I_2)]}{\mathrm{OPT}(I_2)} \geq 1 + q\frac{1-\tau}{1+\tau}. \tag{74}$$

The oblivious adversary chooses whichever of I_1 and I_2 gives the larger of (73) and (74). For every $q \in [0, 1]$, this maximum is at least its minimum over q. The minimum occurs when the two excess terms are equal:

$$(1-q)\tau = q\frac{1-\tau}{1+\tau},$$

or

$$q = \frac{\tau(1+\tau)}{1+\tau^2}. \tag{75}$$

The resulting excess over one is

$$\frac{\tau(1-\tau)}{1+\tau^2} = \frac{\sqrt{2}-1}{2}, \tag{76}$$

where the equality uses $\tau = \sqrt{2} - 1$. Thus, for every randomized algorithm, at least one of the two predicted instances has expected ratio at least

$$1 + \frac{\sqrt{2}-1}{2} = \frac{1+\sqrt{2}}{2}.$$

The instance is selected using only the algorithm's distribution, before its random bits are realized, so this is an oblivious-adversary lower bound. □

Combining Theorems 17 and 18 gives the exact value

$$R_2^{\mathrm{TCP,rand}} = \frac{1+\sqrt{2}}{2} \approx 1.20710678. \tag{77}$$

The lower-bound instance realizes the worst boundary gap in the upper-bound analysis: its separation time $\tau = \sqrt{2} - 1$ corresponds to $d = 1 - \tau = 2 - \sqrt{2}$. Thus the balancing equations in the upper and lower bounds agree with equality. For comparison, the exact deterministic ratio with two correct-or-one-correct predictions is $\sqrt{2}$, whereas the optimal randomized ratio without predictions is $e/(e-1)$ Karlin et al. (2003).

H DYNAMIC TCP ACKNOWLEDGMENT: RANDOMIZED CONSISTENCY–ROBUSTNESS TRADEOFF

H.1 RANDOMIZED ROBUSTNESS FOR DYNAMIC TCP ACKNOWLEDGMENT

We now drop the promise that one of the two predictions is correct. The adversary is oblivious: it chooses the actual request sequence and the two predictions before seeing the algorithm's random bits. For a randomized algorithm $\mathcal{A}$, define

$$\mathrm{cons}(\mathcal{A}) := \sup_{P_1,P_2} \sup_{I\in\{P_1,P_2\}} \frac{\mathbb{E}[\mathcal{A}(I;P_1,P_2)]}{\mathrm{OPT}_{\mathrm{TCP}}(I)},$$
$$\mathrm{rob}(\mathcal{A}) := \sup_{P_1,P_2} \sup_{I} \frac{\mathbb{E}[\mathcal{A}(I;P_1,P_2)]}{\mathrm{OPT}_{\mathrm{TCP}}(I)}.$$

The expectation is over the internal randomness of $\mathcal{A}$. We retain the normalization in which one acknowledgment costs one. In particular, every nonempty instance satisfies

$$\mathrm{OPT}_{\mathrm{TCP}}(I) \geq 1. \tag{78}$$

The robust algorithm. Let $\mathcal{B}_{\mathrm{rand}}$ be the randomized boundary-state algorithm from Section G. At time s, call a prediction *viable* if its restriction through s agrees with the observed request sequence, and let

$$\sigma := \inf\{s \geq 0 : P_1 \text{ and } P_2 \text{ are both nonviable at time } s\}.$$

Until σ, the algorithm follows $\mathcal{B}_{\mathrm{rand}}$. If $\sigma < \infty$, it discards both predictions and switches, from the state inherited at σ, to the randomized algorithm of Karlin et al. (2003). Outstanding requests and latency accumulated before σ are retained.

For completeness, we recall the relevant form of the KKR algorithm. For $z \in [0,1]$, let A_z be the savings-threshold policy that sends an acknowledgment when the latency saving attainable by an additional acknowledgment reaches z. Independently of the random choice made by the boundary algorithm, sample one threshold Z with density

$$p(z) := \frac{e^z}{e-1}, \qquad 0 \leq z \leq 1, \tag{79}$$

and run A_Z after the switch. Its distribution function is

$$P(z) := \Pr[Z \leq z] = \frac{e^z - 1}{e-1}, \qquad p(z) - P(z) = \frac{1}{e-1}. \tag{80}$$

The only new issue relative to the classical KKR analysis is the last acknowledgment of the prediction-driven phase. The following lemma isolates its cost. It is useful more generally whenever a TCP acknowledgment policy is changed once.

Lemma 19 (Randomized one-switch inequality)**.** *Fix an actual instance I and the realized random choice of the boundary state. There is a decomposition*

$$\mathrm{OPT}_{\mathrm{TCP}}(I) = O_{\mathrm{pre}} + O_{\mathrm{res}}$$

and a nonincreasing threshold profile n_z for the residual instance such that the algorithm which switches to A_z satisfies

$$\mathcal{A}_z(I;P_1,P_2) \leq O_{\mathrm{pre}} + 1 + O_{\mathrm{res}} + n_z - \int_z^1 n_w\, dw. \tag{81}$$

Moreover,

$$\int_0^1 n_z\, dz \leq O_{\mathrm{res}}. \tag{82}$$

Consequently, for an arbitrary density p on $[0,1]$ with distribution function P,

$$\mathbb{E}_Z[\mathcal{A}_Z(I;P_1,P_2)] \leq \mathrm{OPT}_{\mathrm{TCP}}(I) + 1 + \int_0^1 \big(p(z) - P(z)\big) n_z\, dz. \tag{83}$$

Proof. Use the staircase representation of dynamic TCP acknowledgment: the height at time s is the number of outstanding requests and latency is the area under this staircase. Overlay the prediction-driven schedule with a fixed offline optimum. Apply the usual uncrossing operation from left to right to the acknowledgment blocks completed strictly before σ. Because a viable prediction agrees with the entire observed prefix, each completed block can be replaced by the corresponding offline block without changing the residual backlog or increasing the assigned prefix cost. After all such exchanges, only the final acknowledgment of the prediction-driven phase can cross the cut at σ. Charge this acknowledgment separately, at cost one. This gives a prefix charge of at most $O_{\mathrm{pre}} + 1$ and leaves a residual TCP instance with an offline schedule of cost O_{res}.

We now apply the KKR rectangle argument to the residual staircase. Let n_z be the number of acknowledgments sent by A_z on this residual instance. If E_z denotes the area lying above the offline staircase and below the staircase of A_z, the KKR accounting gives

$$\mathrm{cost}(A_z) \le n_z + O_{\mathrm{res}} - n_{\mathrm{off}} + z n_{\mathrm{off}} - E_z, \tag{84}$$

where n_{off} is the number of residual offline acknowledgments. Their geometric lemma states that

$$E_z \ge \int_z^1 n_w \, dw - (1-z) n_{\mathrm{off}}. \tag{85}$$

Substituting (85) into (84) cancels all terms containing n_{off} and yields

$$\mathrm{cost}(A_z) \le O_{\mathrm{res}} + n_z - \int_z^1 n_w \, dw.$$

Adding the prefix charge proves (81). The KKR integral lower bound applied to the residual instance is precisely (82).

Finally, integrate (81) against $p(z)$. Tonelli's theorem and a change in the order of integration give

$$\begin{aligned}
&\int_0^1 p(z) \left(n_z - \int_z^1 n_w \, dw \right) dz \\
&\quad = \int_0^1 p(z) n_z \, dz - \int_0^1 \left(\int_0^w p(z) \, dz \right) n_w \, dw \\
&\quad = \int_0^1 \big(p(w) - P(w) \big) n_w \, dw,
\end{aligned}$$

which proves (83). □

Theorem 20 (Randomized consistency and robustness)**.** *Against an oblivious adversary, the algorithm above satisfies*

$$\mathrm{cons}(\mathcal{A}) \le \frac{1+\sqrt{2}}{2}, \tag{86}$$

$$\mathbb{E}[\mathcal{A}(I; P_1, P_2)] \le \frac{e}{e-1} \mathrm{OPT}_{\mathrm{TCP}}(I) + 1 \qquad \textit{for every } I, P_1, P_2, \tag{87}$$

$$\mathrm{rob}(\mathcal{A}) \le 1 + \frac{e}{e-1} = \frac{2e-1}{e-1} \approx 2.58198. \tag{88}$$

Thus the asymptotic robustness, when an additive constant is permitted, is $e/(e-1)$.

Proof. If one prediction is exact, it remains viable forever, so $\sigma = \infty$. The fallback is never invoked and the algorithm is exactly $\mathcal{B}_{\mathrm{rand}}$. The consistency bound therefore follows from Theorem 17.

For an arbitrary actual sequence, substitute the density (79) into (83). By (80) and (82),

$$\begin{aligned}
\mathbb{E}[\mathcal{A}(I; P_1, P_2)] &\le \mathrm{OPT}_{\mathrm{TCP}}(I) + 1 + \frac{1}{e-1} \int_0^1 n_z \, dz \\
&\le \mathrm{OPT}_{\mathrm{TCP}}(I) + 1 + \frac{1}{e-1} O_{\mathrm{res}} \\
&\le \frac{e}{e-1} \mathrm{OPT}_{\mathrm{TCP}}(I) + 1.
\end{aligned}$$

This proves (87). For a nonempty instance, (78) absorbs the additive one into one copy of $\mathrm{OPT}_{\mathrm{TCP}}(I)$ and gives (88). The empty instance costs zero to both schedules. □

The leading constant in (87) is best possible.

Theorem 21 (Robustness lower bound)**.** *Every randomized online algorithm for dynamic TCP acknowledgment, including algorithms having two predictions and satisfying any consistency guarantee, has asymptotic robustness at least*

$$\frac{e}{e-1}. \tag{89}$$

The same number is therefore a lower bound on strict multiplicative robustness.

Proof. Give the algorithm two fixed predictions that are falsified before the hard request sequence begins. Conditioned on this fixed advice, the algorithm is an ordinary randomized online algorithm for dynamic TCP acknowledgment. The classical oblivious-adversary lower bound of Karlin et al. (2003) therefore applies and gives $e/(e-1)$. Notice that this construction places no restriction on the algorithm before or after the predictions fail, and hence also applies to the best algorithm subject only to the consistency requirement $\mathrm{cons}(\mathcal{A}) \le (1+\sqrt{2})/2$. □

Combining the two theorems, the strict multiplicative robustness currently proved without weakening optimal consistency lies in the interval

$$\frac{e}{e-1} \;\le\; \inf_{\mathrm{cons}(\mathcal{A})\le(1+\sqrt{2})/2} \mathrm{rob}(\mathcal{A}) \;\le\; 1+\frac{e}{e-1}. \tag{90}$$

The additive form (87) closes the corresponding asymptotic gap: its leading constant $e/(e-1)$ matches the classical lower bound. Closing the remaining gap for the strict multiplicative definition would require charging some or all of the single unmatched boundary acknowledgment, rather than paying for it as the additive constant in Lemma 19.

I Implementation Details of the Algorithms

The prediction-dependent preprocessing uses standard shortest-path dynamic programs. Once the boundary state or mixture has been selected, all online steps consist of executing the chosen prefix, testing prediction viability, and running the appropriate threshold recovery rule.

Bahncard. For a finite demand measure $\rho = \sum_{r=1}^{m} w_r \delta_{t_r}$ with sorted times, an optimal full-subsidy schedule can be chosen to buy only at demand times. A shortest-path dynamic program has one state per next uncovered demand: either pay w_r and advance one position, or buy a card at t_r and advance to the first demand at or after $t_r + D$. This computes $\mathrm{OPT}_0(\rho)$ and a corresponding schedule in $O(m \log m)$ time after sorting, using binary search for interval endpoints. To implement a boundary-state algorithm, compute an optimum for each prediction, extract its latest pre-divergence expiration e_i, and compute the two constrained prefix costs. The deterministic algorithm applies (7); the randomized algorithm samples according to (33). The certified robust implementation additionally maintains the residual fare accumulated since the relevant pending purchase and triggers the card when it reaches z_β.

Dynamic TCP acknowledgment. An offline optimum partitions the requests, in release-time order, into consecutive batches and acknowledges each batch at the release time of its last request. The resulting shortest-path dynamic program has one state per request index and a transition for every possible final batch; a direct implementation takes $O(n^2)$ time. Constraining the outstanding suffix at the divergence time gives the prefix values $P(B)$ and continuation values $F_i(B)$. The deterministic and randomized boundary choices then use (14) and (63), respectively.

Failure detection and recovery. The algorithm compares each observed event with both complete predictions and maintains their viability flags. When both flags become false, it retains the current coverage or backlog state and starts the prescribed savings-threshold recovery algorithm. This is an online operation: no past latency, uncovered fare, valid card, or outstanding request is discarded at the switch.

J More Experimental Results and Setup Details

J.1 Details of Experimental Setup in Zhao et al. (2024) for the Bahncard Problem

In Zhao et al. (2024), synthetic Bahncard instances and noisy predictions are generated over a discrete daily horizon as follows:

- **Time Horizon & Monte Carlo Trials:** Each problem instance spans a horizon of $N = 2000$ days. For every parameter configuration and perturbation level, the empirical competitive ratio $\mathrm{ALG}(I)/\mathrm{OPT}(I)$ is averaged over 100 independent trials.
- **Bahncard Problem Parameters** $\mathrm{BP}(C, \beta, T)$**:** A Bahncard costs $C > 0$, remains valid for T consecutive days ($[t, t + T - 1]$), and reduces every regular ticket price by a discount factor $\beta \in (0, 1)$. We use card cost $C = 5000$ and validity period $T = 500$ days for $\beta = 0.6$.
- **Traveler Profiles:**
  1. **Commuter:** Generates exactly one travel request on every day $d \in \{1, \ldots, N\}$.
  2. **Occasional Traveler:** Inter-request times are drawn i.i.d. from an Exponential distribution with mean 2.0 days ($\mathrm{Exp}(1/2)$). If multiple requests fall on the same calendar day $\lfloor t \rfloor$, their prices are summed into a single combined daily demand.
- **Ticket Price Distributions:** Each travel request's regular price is sampled i.i.d. from one of three distributions (all having mean 50):
  1. **Uniform (`U`):** $\mathrm{Uniform}(0, 100)$ with mean 50.
  2. **Normal (`N`):** $\mathcal{N}(\mu = 50, \sigma^2 = 25)$ (standard deviation $\sigma = 5$), truncated below at 0.
  3. **Pareto (`P`):** Heavy-tailed Lomax (Pareto Type II) distribution with shape parameter $\alpha = 2$ and scale parameter 50 (mean 50, infinite variance).
- **Prediction Perturbation Model ($p \in \{0.0, 0.1, \ldots, 1.0\}$):** Given a ground-truth daily price sequence $I = (I(1), \ldots, I(N))$ and a perturbing probability $p \in [0, 1]$, a noisy prediction $\widehat{I}_p$ is constructed by applying two independent operations to each perturbed day d:
  1. *Deletion:* With probability p, the existing travel request on day d is removed ($\widehat{I}_p(d) \leftarrow 0$).
  2. *Addition:* Independently, with probability p, a random price X_d sampled from the same ticket price distribution is added to day d ($\widehat{I}_p(d) \leftarrow \widehat{I}_p(d) + X_d$).

  40% Exact Prefix Variant: In our experiments, for the first 40% of the horizon ($1 \le d \le 800$), no noise is applied; the deletion/addition perturbations above are applied only for $801 \le d \le 2000$.

J.2 Details of Experimental Setup in Im et al. (2023) for Dynamc TCP Acknowledgment

In Im et al. (2023), the Dynamic TCP Acknowledgment Problem (DTAP) is evaluated on synthetic packet arrival traces as follows:

- **Time Horizon, Cost Model, & Trials:** Time is discretized into $T = 1000$ steps, and results are averaged over 5 independent runs per setting. Sending a single TCP acknowledgment (ACK) at any time step costs 1 and clears all currently unacknowledged packets. Each unacknowledged packet incurs a latency cost of $1/d$ per time step that it waits. We use $d = 500$ (unit latency cost $1/500$ per step) in our experiments.
- **Packet Arrival Distributions:** At each discrete time step $t \in \{1, \ldots, T\}$, the number of arriving packets $a_t \in \mathbb{Z}_{\ge 0}$ is generated from one of three arrival processes:
  1. **Pareto:** $a_t = \mathrm{round}(X_t)$ where $X_t \sim \mathrm{Pareto}(\alpha = 2)$, modeling heavy-tailed packet bursts.
  2. **Poisson:** $a_t \sim \mathrm{Poisson}(\lambda = 1)$, modeling steady homogeneous traffic.
  3. **Iterated Poisson:** A 10-step iterated Poisson process starting from $\mu_0 = 1$, where $\mu_k \sim \mathrm{Poisson}(\mu_{k-1})$ for $k = 1, \ldots, 10$ and $a_t = \mu_{10}$. Because $\mu_k = 0$ is an absorbing state, this process generates extreme bursts separated by long silent intervals.

- **Prediction Perturbation Model** ($r \in \{0.0, 0.05, 0.10, \ldots, 1.0\}$)**:** Given the true per-step packet counts $I = (a_1, \ldots, a_T)$ and a perturbing probability $r \in [0, 1]$, a noisy prediction $\widehat{I}_r$ is generated by applying two independent operations at each perturbed time step t:

  1. *Drop:* With probability r, all true packet arrivals at step t are dropped ($\hat{a}_t \leftarrow 0$; otherwise $\hat{a}_t \leftarrow a_t$).

  2. *Insert:* Independently, with probability r, an independent draw Y_t from the same arrival distribution is added to step t ($\hat{a}_t \leftarrow \hat{a}_t + Y_t$).

  40% Exact Prefix Variant: In our experiments, for the first 40% of the time steps ($1 \leq t \leq 400$), the noisy prediction matches the true trace exactly ($\hat{a}_t = a_t$); the drop/insert operations are applied only for $401 \leq t \leq 1000$.

J.3 Results of Additional Experiments

We now report the findings of the additional experiments in Figures 3 to 5. We outline the main findings:

- For the 1-EXACT + 1-NOISY regime with the Occasional Traveler pattern for the Bahncard problem (Figure 3), our Deterministic and Randomized algorithms significantly outperform PFSUM as well as the classical baseline algorithms. The worst ratio for Deterministic or Randomized is around 1.009 whereas that for PFSUM is around 1.035 and for the classical algorithms are above 1.05.

- For the 2-NOISY experiment for the Bahncard problem (Figure 4), the PFSUm algorithm performs the best, primarily because the two predictions are noisy versions of the actual instance and averaging them yields a single prediction of high quality. Since our algorithms detect that the predictions are incorrect, they revert to robust strategies while PFSUM continues to use this high quality prediction.

- For the 2-Noisy experiment for TDAP (Figure 5), our Robust Deterministic strictly outperforms Dooly, Karlin, and Robust Randomized across all noise levels $r > 0$, and beats ALA at moderate-to-high noise across all three distributions.

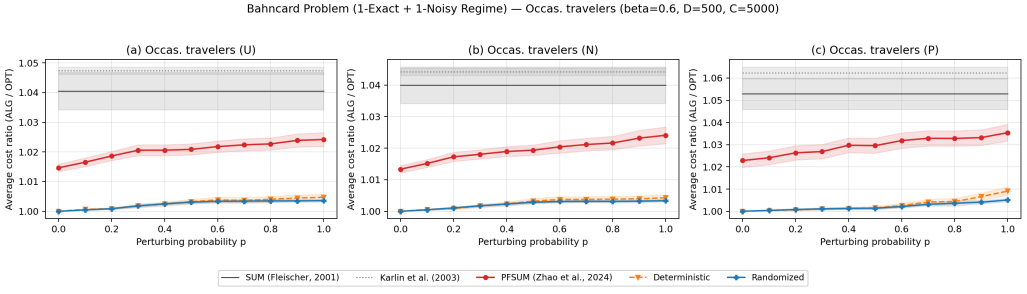


Figure 3: Bahncard [1-EXACT + 1-NOISY Case, $\beta = 0.6$]: Competitive ratio vs noise level p of the noisy prediction comparing **SUM** (Fleischer, 2001), **Karlin** (Karlin et al., 2003), **PFSUM** (Zhao et al., 2024), and our **Deterministic** and **Randomized** algorithms for the Occasional Traveler request pattern using Uniform, Normal, and Pareto distributions for generating trips. The line for **Deterministic** is partially hidden under that for **Randomized** since both algorithms essentially optimal solutions throughout the range.

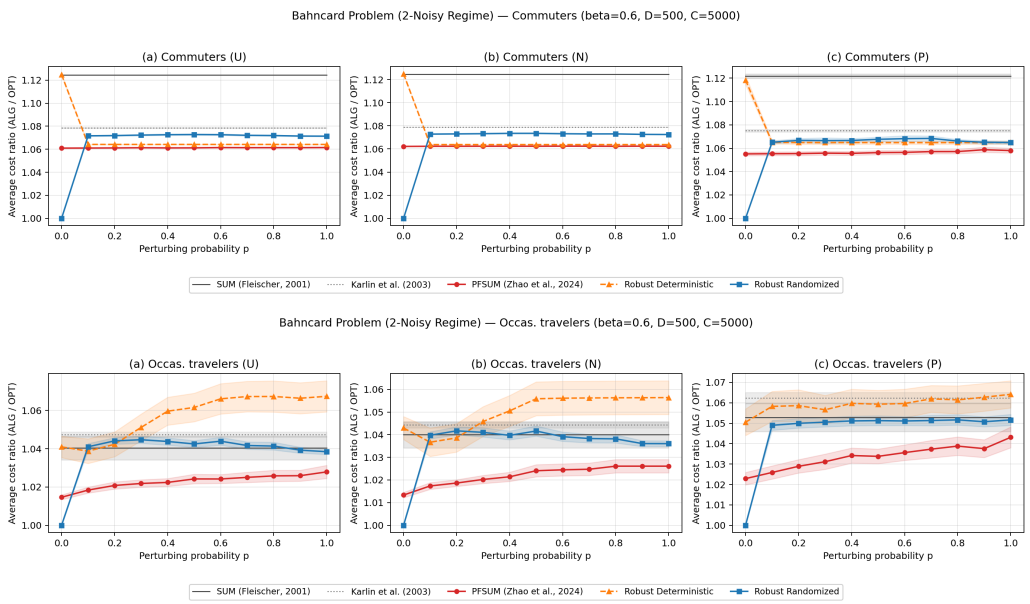


Figure 4: Bahncard [2-NOISY Case, $\beta = 0.6$]: Competitive ratio vs noise probability p when both predictions are are noisy for **Commuter** (top row) and **Occasional Traveler** (bottom row) request patterns. We compare **SUM** (Fleischer, 2001), **Karlin** (Karlin et al., 2003), **PFSUM** (Zhao et al., 2024), and our **Robust Deterministic** and **Robust Randomized** algorithms

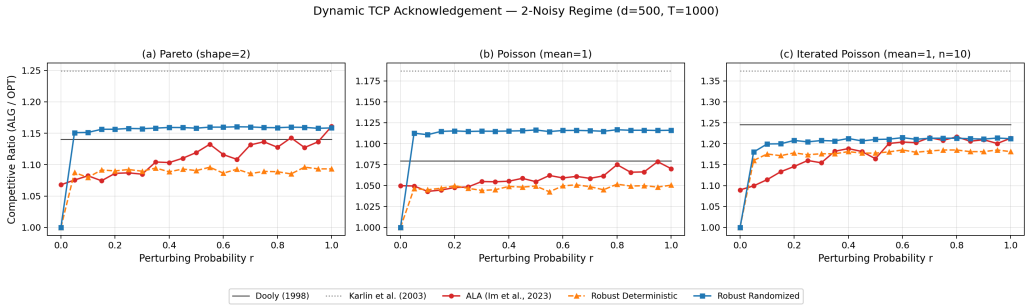


Figure 5: DTAP [2-NOISY Case]: Competitive ratio vs replacement rate $r \in [0, 1]$ comparing **Dooly** Dooly et al. (1998), **Karlin** Karlin et al. (2003), **ALA** Im et al. (2023), and our **Robust Deterministic** and **Robust Randomized** algorithms using Pareto, Poisson, and Iterated Poisson distributions for generating packets.