How Much Must a Private Mempool Hide? Exact Leakage Thresholds for Sandwich Attacks

Tingyi Lin[1†], **Jiazhuo Li**[2], **Ruoran Lai**[3]
[1]Tropic Research Institute [2]University of Michigan [3]Sun Yat-sen University

Abstract

Private and encrypted mempools hide pending transactions to stop sandwich attacks and other forms of maximal extractable value (MEV), but what they hide is rarely everything: a transaction's pair, direction, and a coarse range for its size can still leak. How much leakage makes sandwiching pay? We answer exactly for a fee-free constant-product automated market maker, the pricing rule behind Uniswap v2. Traders observe an interval containing the victim's size and bid in a first-price auction for the right to sandwich it, and the winning front-run must keep the victim's trade executable at every size in the interval. The answer turns on the smallest size consistent with the leak. It alone determines the feasible front-runs, the largest feasible front-run is optimal for pointwise, expected, and worst-case profit alike, and the guaranteed profit has a closed form. When execution is costly, a privacy layer that wants to rule out sandwiches profitable at every consistent size may therefore reveal anything about the size except a lower bound above an explicit threshold; the upper end of the range is irrelevant. With two or more symmetric traders, every pure-strategy perfect Bayesian equilibrium of the auction hands the entire expected net rent to the auctioneer. If the direction is hidden too, no non-contingent first leg front-runs both possible directions, while post-trade arbitrage can survive even perfect pre-trade hiding.

"A fool uttereth all his mind: but a wise man keepeth it in till afterwards."
– Proverbs 29:11 (King James Version)

1 Introduction

A hidden order has size $q > 0$, but outside agents observe only a coarse public signal. When does such leakage suffice to support profitable predatory behavior in equilibrium? We answer this question exactly in the trading game induced by a constant-product automated market maker (AMM), a smart contract that exchanges two tokens at prices set by its reserves. The motivating examples come from hidden-order execution environments, but only the induced information structure enters the theorems.

If an adversarial trader knows that a latent order intends to buy token Y using token X, then a front-run purchase of Y raises that order's execution price and creates the standard sandwich pattern: trade in the latent order's direction, let the latent order move the pool further, and then unwind in the opposite direction. Under full observability, the trader may know the hidden order exactly. Our interest is the intermediate regime in which the trader sees only coarse information. Sandwiching remains common on Ethereum [28, 25] and is not confined to simple Uniswap v2 paths: McLaughlin, Kruegel, and Vigna identify 63,257 sandwich attacks among apparent arbitrages, including one whose manipulation and unwind route through both Uniswap v2 and v3 [23], and Bai et al. detect 60,946 sandwich events, multi-token routes included, in 210,000 Ethereum blocks [6].

[†]Correspondence to: tingyi3@illinois.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

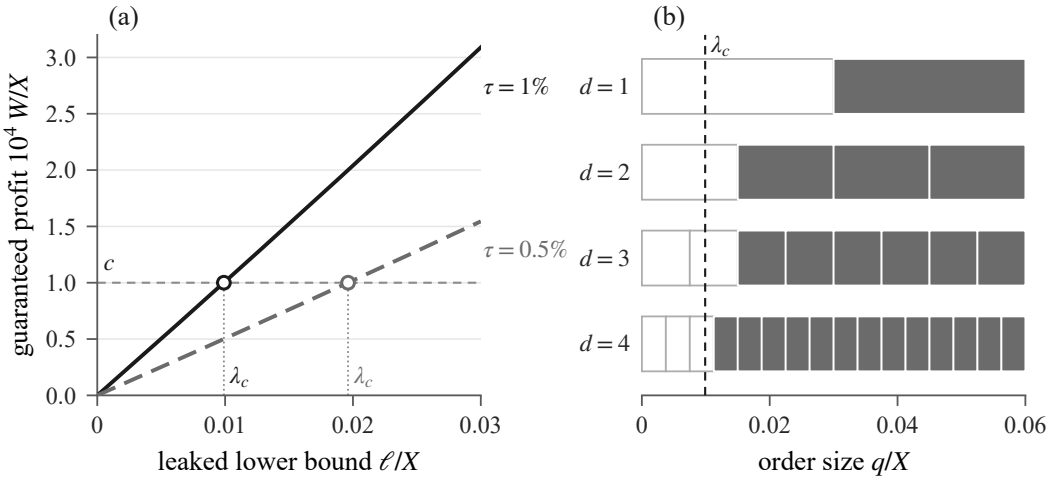

Figure 1: (a) Guaranteed gross profit W of the optimal robust sandwich against the leaked lower bound ℓ, for slippage tolerances $\tau = 1\%$ and $\tau = 0.5\%$, with execution cost $c = 10^{-4}X$ (dashed). A robust sandwich with positive net payoff at every consistent size exists exactly when $\ell > \lambda_c$ (Theorem 4.3); to first order $\lambda_c \approx c/\tau$. (b) Leakage of the first d bits of the size, for sizes up to $Q = 0.06X$ with $\tau = 1\%$ and $c = 10^{-4}X$. Shaded bins admit such a sandwich (Proposition 4.6); the unshaded range shrinks toward $\lambda_c \approx 0.0099X$ as d grows and always contains $(0, \lambda_c]$.

A round of our game has four steps. A victim submits a swap whose size is hidden, with a slippage guard that cancels the swap if its output falls too far below the honest output. A public signal then reveals an interval that contains the hidden size. Traders bid for the single right to surround the swap, with a front-run immediately before it and an unwind immediately after. The winner fixes its front-run size without learning the exact size, and the front-run must leave the swap executable at every size consistent with the signal.

Formally, a leakage map sends the hidden size q to a public signal s whose interval $I_s \ni q$ is either $[\ell_s, u_s]$ with lower endpoint $\ell_s > 0$ or $(0, u_s]$. The victim's direction and slippage tolerance $\tau \in (0, 1)$ are public. Before bidding, every trader and the auctioneer observe the pool reserves (X, Y), the trading pair, the direction, τ, and s, but not q, and they share a posterior μ_s on I_s derived from a common prior. Bids may use only *robustly admissible* front-runs, which keep the victim's trade valid for every $q \in I_s$.

Our first result characterizes the feasible front-runs and the profit they guarantee.

Theorem A (Admissibility, value, and threshold)**.** *Fix a leaked interval $[\ell_s, u_s]$ with $\ell_s > 0$.*

(i) The robustly admissible front-runs form an interval $[0, a_{\max}(\ell_s, \tau)]$ whose endpoint is an explicit function of X, τ, and ℓ_s alone.

(ii) The largest admissible front-run is the unique maximizer of pointwise, expected, and worst-case profit, and its worst-case gross profit is

$$\frac{\tau \ell_s (X + \ell_s)}{X + \tau \ell_s}.$$

(iii) With execution cost $c \geq 0$, a robustly admissible bundle with positive net payoff at every consistent size exists if and only if ℓ_s exceeds an explicit threshold λ_c that depends only on X, τ, and c.

Theorem A collects Theorems 4.1 to 4.3. Only the lower endpoint matters because a front-run hurts small victims most: the victim's output after the front-run, as a fraction of its honest output, rises with its size (Lemma 5.1). For a half-open interval $(0, u_s]$, parts (i) and (ii) hold with $\ell_s = 0$ for pointwise and expected profit, and with $c > 0$ no such bundle exists (Corollary 4.4 with u_s in place of Q). Figure 1 plots the threshold and the effect of leaking the leading bits of the size.

Our second result prices the execution right.

Theorem B (Execution auction)**.** *Let $m \geq 2$ symmetric risk-neutral traders bid for the slot, and let R_s be the posterior expected gross profit of the largest admissible front-run minus the execution cost c. If $R_s > 0$, every pure-strategy perfect Bayesian equilibrium awards the slot to that front-run at the payment R_s, so the auctioneer collects the entire expected net rent; if $R_s < 0$, no equilibrium places a sandwich.*

Theorem B is Theorem 4.5; the posterior enters only through R_s. With a positive execution cost, a signal that reveals no positive lower bound supports no guaranteed-profit sandwich, while post-trade arbitrage with gross profit up to $q^2/(X+q)$ can survive (Corollary 4.4 and Proposition 4.7).

Sandwich attacks on AMMs have been studied analytically under full observability, most directly in Heimbach and Wattenhofer's sandwich game [16]. Under partial observability the lower endpoint of the leaked interval is the sufficient statistic: it alone determines robust admissibility, the optimal robust attack, and the exact profitability threshold, a leakage-aware analogue of the minimum profitable victim input in public-order-flow analyses [31, 30].

The baseline hides the size while the pair, the direction, and the slippage tolerance are public, and the public direction is a substantive assumption. If the direction may be either way and the first leg must be chosen before it is revealed, no non-null first leg front-runs both directions (Proposition 6.4); a hidden slippage tolerance instead reduces to its public lower bound (Proposition 6.5). Hiding the direction thus blocks the non-contingent robust sandwich studied here, and when the direction leaks while the size is only interval-revealed, ℓ_s determines the exact threshold. The closed forms use the algebra of the fee-free constant-product invariant; Proposition 6.7 bounds the effect of swap fees in one direction, and Remark 6.10 discusses other pricing rules, for which we do not claim them.

2 Related Work

Analytical work on sandwiching and transaction reordering under full or nearly full observability provides the closest starting point. Daian et al.'s *Flash Boys 2.0* articulated the broader reordering perspective [13]. Zhou et al. analyze sandwich attacks under public order flow [31], and Heimbach and Wattenhofer study slippage-tolerance choice in a sandwich game [16]. Park shows that under transparent execution every liquidity-invariance pricing rule admits sandwich attacks [24], and Kulkarni, Diamandis, and Chitra analyze routing and reordering MEV in CFMMs, showing that the price of anarchy of routing is constant when the impact of a sandwich attack is localized [21]. In our notation, the fully observed regime corresponds to point leakage $I = [q, q]$. Our results show that under interval leakage the correct analogue of the minimum profitable victim input intuition from A2MM [30] is the leaked lower endpoint ℓ.

A second adjacent line studies market design and mechanism design for strategic trading environments. Chan, Wu, and Shi analyze AMM mechanism design [11]. Budish, Cramton, and Shim show how batch processing destroys speed-based rents in another market-design setting [10]. Wadhwa et al. study order-policy enforcement under rationality and prove impossibility results in a fully rational setting [29]. Kelkar et al. introduce order-fairness for Byzantine consensus [20], Li et al. revisit transaction-fairness definitions for blockchains [22], and Ferreira and Parkes design verifiable sequencing rules that limit what a block producer gains by ordering trades around a user [15]. PROF studies protected order flow in a profit-seeking PBS environment [4]. Transaction fee mechanism design studies how block space should be allocated and priced when block producers are strategic [27, 12], including producers who extract MEV themselves [5]; the auctioneer of our execution-rights auction sells one such slot, and Theorem 4.5 shows that it collects the entire expected net rent. The leakage map is the traders' information structure, and Bergemann, Brooks, and Morris show how information structures shape bidding and revenue in first-price auctions [8]. Relative to these papers, our contribution is an exact evaluative theorem: given a residual information structure, when is the remaining leakage already enough to sustain robust sandwiching in equilibrium?

Hidden-order and encrypted-mempool systems provide the closest systems motivation. Ferveo formalizes mempool privacy via threshold decryption in BFT networks [7], while Shutter gives a threshold-cryptography approach to private transactions [14]. BlindPerm combines encrypted mempools with permutation-based ordering for MEV mitigation [19]. Rondelet and Kilbourn argue that private mempools should be evaluated through an economic lens, with sandwich attacks as a canonical case [26], and Heimbach and Wattenhofer survey defenses against transaction reordering [17]. Angeris, Evans, and Chitra show that the usual CFMM implementations cannot keep traded

quantities hidden from an adversary who sees the remaining public information [3], which complements Proposition 4.7: hiding an order before execution does not remove the arbitrage its execution creates. Our analysis abstracts away implementation details and treats the pre-trade privacy layer solely through a leakage map.

3 Preliminaries and Model

3.1 Constant-Product AMM and Sandwich Bundles

We work with a constant-product AMM holding reserves $(X, Y) \in \mathbb{R}^2_{>0}$ of tokens X and Y. The invariant is $XY = k$. This is the pricing rule of Uniswap v2 [1] without its swap fee (Proposition 6.7 treats a fee withheld from the reserves) and the usual benchmark in analyses of sandwich attacks [31, 16]; it has the price impact and slippage that sandwiching exploits and still admits exact solutions. We analyze a single victim order that buys token Y using token X. The opposite direction is symmetric under token exchange. This isolates the residual-information regime in which the pair and direction leak through the transaction interface but the exact size does not; Section 6 discusses what changes when direction or slippage is also hidden.

For an input of $\delta > 0$ units of token X, the AMM returns

$$\Delta_Y(\delta; X, Y) = Y - \frac{XY}{X + \delta} = \frac{Y\delta}{X + \delta}. \tag{1}$$

After this trade the reserves become $(X + \delta, XY/(X + \delta))$. Similarly, if the current reserves are (X', Y') and a trader inputs $\eta > 0$ units of token Y, then the AMM returns

$$\Delta_X(\eta; X', Y') = X' - \frac{X'Y'}{Y' + \eta} = \frac{X'\eta}{Y' + \eta}. \tag{2}$$

The victim order has hidden size $q > 0$. If executed honestly from the initial state, the victim receives

$$h(q) = \Delta_Y(q; X, Y) = \frac{Yq}{X + q}. \tag{3}$$

The victim also posts a slippage tolerance $\tau \in (0, 1)$ and therefore requires at least $(1 - \tau)h(q)$ output.

If the trader front-runs with size $a \geq 0$, it acquires

$$y_F(a) = \frac{Ya}{X + a} \tag{4}$$

units of token Y, after which the AMM state is $(X + a, XY/(X + a))$. The victim then receives

$$v(a, q) = \Delta_Y\left(q; X + a, \frac{XY}{X + a}\right) = \frac{XYq}{(X + a)(X + a + q)}. \tag{5}$$

Definition 3.1 (Admissible front-run for type q)**.** A front-run size $a \geq 0$ is *admissible for victim type* q if the victim order still executes, i.e.,

$$v(a, q) \geq (1 - \tau)h(q). \tag{6}$$

When the trader later sells the front-run output $y_F(a)$ back into the AMM after the victim trade, the resulting output in token X is

$$x_B(a, q) = \Delta_X\left(y_F(a); X + a + q, \frac{XY}{X + a + q}\right) = \frac{a(X + a + q)^2}{(X + a)^2 + aq}. \tag{7}$$

The trader's gross sandwich profit in token X is therefore

$$\pi(a, q) = x_B(a, q) - a = \frac{aq(2X + a + q)}{(X + a)^2 + aq}. \tag{8}$$

The profit does not depend on Y. Scaling the initial Y-reserve scales the amount of Y bought in the front-run and the Y-side reserve at the unwind by the same factor, and profit is measured in token X, so the factor cancels.

3.2 Leakage Signals

The victim order is not public. Instead, the information structure reveals a signal s that leaks only interval information about q.

Definition 3.2 (Interval leakage map)**.** An *interval leakage map* is a function $L : (0, Q] \to \Sigma$ for some finite signal set Σ, together with an interval assignment $s \mapsto I_s \subseteq (0, Q]$ such that

$$q \in I_{L(q)} \qquad \text{for every } q \in (0, Q].$$

Each I_s is either a closed interval $[\ell_s, u_s]$ with $\ell_s > 0$ or a half-open interval $(0, u_s]$, which reveals no positive lower bound; in the second case we write $\ell_s = 0$. The signal is public before traders bid for execution rights. The victim size q itself is not.

Definition 3.3 (Robust admissibility under signal s)**.** Fix a signal s with interval I_s. A front-run size $a \geq 0$ is *robustly admissible* for s if it is admissible for every $q \in I_s$. The robustly admissible set is

$$\mathcal{A}(I_s) = \{a \geq 0 : a \text{ is admissible for every } q \in I_s\}.$$

3.3 Execution-Rights Auction

Nature draws a victim size $q \in (0, Q]$ from a common prior and reveals the public signal $s = L(q)$. Every trader and the auctioneer observe s together with the pool reserves (X, Y), the trading pair, the direction, and the slippage tolerance τ; nobody observes q. All of them update to a posterior μ_s supported on I_s. We assume that after the signal, the traders are symmetric and risk neutral.

There are $m \geq 2$ traders and one auctioneer, who sells a single execution slot: the indivisible right to place a front-run immediately before the victim and the unwind immediately after it. Each trader i simultaneously submits either a null action or a pair (a_i, b_i) where $a_i \in \mathcal{A}(I_s)$ is a robustly admissible front-run size and $b_i \geq 0$ is a payment to the auctioneer. The auctioneer selects the admissible bundle with the highest payment, breaking ties arbitrarily but deterministically, and the winner pays its own bid, so the auction is first-price (pay-as-bid). We assume a fixed execution cost $c \geq 0$ for a non-null sandwich attempt.

If trader i wins with bundle (a_i, b_i) and the hidden type is q, its utility is $\pi(a_i, q) - b_i - c$; the payment b_i goes to the auctioneer and is separate from the trading profit π. If it loses, its utility is 0. The auctioneer's utility is the payment of the chosen bundle, or 0 if all traders submit null.

Robust admissibility and risk neutrality act at different layers. The first defines the action set $\mathcal{A}(I_s)$, and the second ranks the actions in it by $\mathbb{E}_{q\sim\mu_s}[\pi(a, q)] - b - c$. A trader therefore does not evaluate profit by a max-min criterion: the model imposes an execution guarantee and uses posterior expected utility among the bundles that meet it. The guarantee asks a distribution-free question, namely how much leakage suffices for one bundle that remains a valid sandwich at every size consistent with the signal. Section 6.1 replaces it by a bound on the failure probability, which, for a posterior with full support on I_s, moves ℓ_s to a lower quantile of the posterior.

Remark 3.4 (Equilibrium convention). Throughout, *equilibrium* means *pure-strategy* perfect Bayesian equilibrium. The results characterize equilibrium outcomes of the post-signal game; mixed strategies would enter with discrete bids, asymmetric costs, or random tie-breaking, which the model excludes.

4 Main Results

Theorem 4.1 (Exact robust admissibility)**.** *Fix a signal interval $I = [\ell, u] \subseteq (0, Q]$. In the fee-free constant-product AMM model above, the robustly admissible set is the interval*

$$\mathcal{A}(I) = [0, a_{\max}(\ell, \tau)],$$

where

$$a_{\max}(\ell, \tau) = \frac{-(2X + \ell) + \sqrt{(2X + \ell)^2 + \frac{4\tau X(X+\ell)}{1-\tau}}}{2}. \tag{9}$$

Equivalently, a is robustly admissible if and only if

$$(X + a)(X + a + \ell) \leq \frac{X(X + \ell)}{1 - \tau}. \tag{10}$$

For a half-open interval $I = (0, u]$, the robustly admissible set is $\mathcal{A}(I) = [0, a_0]$, where $a_0 := a_{\max}(0, \tau) = X\big((1 - \tau)^{-1/2} - 1\big)$.

Theorem 4.2 (Optimal robust size and exact robust value)**.** *Fix a signal interval $I = [\ell, u] \subseteq (0, Q]$ and any posterior μ supported on I.*

(a) The gross sandwich profit $\pi(a, q)$ is strictly increasing in a and in q on $\mathbb{R}^2_{>0}$.

(b) Consequently, the unique optimizer of each of the following problems is

$$a^\star(I) = a_{\max}(\ell, \tau) : \tag{11}$$

$$\max_{a \in \mathcal{A}(I)} \pi(a, q) \quad \textit{for every fixed } q \in I,$$

$$\max_{a \in \mathcal{A}(I)} \mathbb{E}_{q\sim\mu}[\pi(a, q)],$$

and

$$\max_{a \in \mathcal{A}(I)} \min_{q \in I} \pi(a, q).$$

(c) The exact worst-case gross profit under signal I is

$$W(I) := \max_{a \in \mathcal{A}(I)} \min_{q \in I} \pi(a, q) = \pi(a^\star(I), \ell) = \frac{\tau\ell(X + \ell)}{X + \tau\ell}. \tag{12}$$

(d) The optimal posterior expected gross profit is

$$V_\mu(I) := \max_{a \in \mathcal{A}(I)} \mathbb{E}_{q\sim\mu}[\pi(a, q)] = \mathbb{E}_{q\sim\mu}[\pi(a^\star(I), q)].$$

For a half-open interval $I = (0, u]$ and any posterior μ on I, parts (a) and (d) and the pointwise and expected problems of part (b) hold with $\ell = 0$, so $a^\star(I) = a_0$.

Theorem 4.3 (Distribution-free leakage threshold)**.** *Fix a signal interval $I = [\ell, u] \subseteq (0, Q]$ and a fixed execution cost $c \geq 0$. There exists a robustly admissible sandwich bundle whose net payoff is strictly positive for* every *hidden type $q \in I$ if and only if*

$$W(I) > c. \tag{13}$$

Equivalently, this holds if and only if

$$\ell > \lambda_c := \frac{-(X - c) + \sqrt{(X - c)^2 + \frac{4cX}{\tau}}}{2}. \tag{14}$$

Corollary 4.4 (Perfect hiding eliminates universal robust sandwiching)**.** *Suppose a signal $s_\perp$ reveals no positive lower bound on the victim size, in the sense that $I_{s_\perp} = (0, Q]$. Then*

$$\inf_{\ell \downarrow 0} W([\ell, Q]) = 0,$$

and if $c > 0$, no universally profitable robust sandwich exists after signal $s_\perp$. If, moreover,

$$R_{s_\perp} := \sup_{a \in \mathcal{A}(I_{s_\perp})} \mathbb{E}_{q\sim\mu_{s_\perp}}[\pi(a, q)] - c < 0,$$

then every pure-strategy equilibrium continuation after $s_\perp$ has no on-path sandwich.

Theorem 4.3 and the first two claims of Corollary 4.4 concern the distribution-free guarantee W, which asks for positive net payoff at every size consistent with the signal. Whether the auction places a sandwich depends instead on the posterior continuation value R_s of Theorem 4.5. The two criteria part ways under perfect hiding. By Theorem 4.1, the front-run $a_0 > 0$ stays robustly admissible on $(0, Q]$. Under a posterior with mass one at Q, a trader who bids $(a_0, 0)$ while every other trader submits null wins alone and earns $\pi(a_0, Q) - c$, which is positive for small c. No profile in which every trader submits null is then an equilibrium, so every equilibrium places a sandwich on path. Perfect hiding thus removes the uniform guarantee, while attacks that pay in expectation can remain.

Theorem 4.5 (Pure-strategy execution-auction equilibrium)**.** *Fix a signal s with interval I_s and posterior μ_s, and let*

$$R_s := V_{\mu_s}(I_s) - c. \tag{15}$$

In the pure-strategy game with $m \geq 2$ symmetric traders:

(a) If $R_s < 0$, then every pure-strategy perfect Bayesian equilibrium has no on-path sandwich bundle after signal s. If $R_s = 0$, then a no-attack pure equilibrium exists, and every on-path sandwich equilibrium outcome yields zero auctioneer revenue and zero trader payoffs.

(b) If $R_s > 0$, then every pure-strategy perfect Bayesian equilibrium has an on-path winning bundle $(a^\star(I_s), b_s^\star)$ with

$$b_s^\star = R_s. \tag{16}$$

In particular, the winner's expected utility is 0, every losing trader gets 0, and the auctioneer captures the full positive continuation rent R_s.

Proposition 4.6 (Bit-prefix leakage)**.** *Assume the global support is $(0, Q]$ and the leakage signal reveals the first $d \geq 1$ bits of the normalized trade size, thereby inducing dyadic intervals*

$$I_j = \left[\frac{jQ}{2^d}, \frac{(j+1)Q}{2^d}\right] \qquad \text{for } j = 1, \ldots, 2^d - 1,$$

and $I_0 = (0, Q/2^d]$. Then for $1 \leq j \leq 2^d - 1$ a universally profitable robust sandwich exists on bin I_j if and only if

$$\frac{jQ}{2^d} > \lambda_c.$$

If $c > 0$, the lowest bin never supports universal robust sandwiching, and the zero-bit case recovers Corollary 4.4.

Proposition 4.7 (Boundary theorem: hiding order contents does not remove post-trade arbitrage)**.** *Suppose every order size leads to the perfectly hiding signal $s_\perp$ of Corollary 4.4, and $c > \pi(a_0, Q)$. Then no universally profitable robust sandwich exists after $s_\perp$, and no pure-strategy equilibrium has an on-path sandwich. Yet if an external market trades token Y at the pool's initial price X/Y, a trader who observes the reserves* after *a victim trade of size $q > 0$ earns a strictly positive gross arbitrage profit, up to $q^2/(X + q)$.*

5 Proof Overview

Two monotonicity properties drive the results; the algebra is deferred to Appendix A.

Lemma 5.1 (Relative-output monotonicity)**.** *For every fixed $a > 0$, the ratio*

$$r(a, q) := \frac{v(a, q)}{h(q)} = \frac{X(X + q)}{(X + a)(X + a + q)}$$

is strictly increasing in $q > 0$; for $a = 0$ it equals 1.

Lemma 5.2 (Profit monotonicity)**.** *The sandwich profit $\pi(a, q)$ from* (8) *is strictly increasing in $a > 0$ and in $q > 0$.*

Because the trader does not know the exact victim size, its front-run must be safe for every type in the leaked interval, and by Lemma 5.1 this constraint binds at the smallest one. Robust admissibility thus reduces to a condition at the lower endpoint ℓ, and solving the resulting quadratic gives Theorem 4.1. Since profit rises in the front-run (Lemma 5.2), the largest robust front-run is optimal for the pointwise, expected, and worst-case objectives of Theorem 4.2. The worst case is at $q = \ell$, where the boundary identity $\sigma(\sigma + \ell) = X(X + \ell)/(1 - \tau)$ for $\sigma = X + a_{\max}(\ell, \tau)$ turns the numerator and the denominator of the profit formula into multiples of $X/(1 - \tau)$:

$$\pi\big(a_{\max}(\ell, \tau), \ell\big) = \frac{\ell(\sigma - X)(\sigma + X + \ell)}{\sigma^2 + (\sigma - X)\ell} = \frac{\ell\,\tau X(X + \ell)/(1 - \tau)}{X(X + \tau\ell)/(1 - \tau)} = \frac{\tau\ell(X + \ell)}{X + \tau\ell}. \tag{17}$$

Solving $W(I) > c$ gives Theorem 4.3, and the limit $\ell \downarrow 0$ gives Corollary 4.4.

The equilibrium theorem is then one-dimensional. Because every trader agrees after signal s that the optimal admissible bundle uses $a^\star(I_s)$ and has continuation value $R_s = V_{\mu_s}(I_s) - c$, competition among at least two symmetric traders is Bertrand-like. If $R_s < 0$, no one attacks, and at $R_s = 0$ an attack earns zero surplus. If $R_s > 0$, a trader can win alone when nobody bids, any winning fee below R_s can be beaten by a rival, and a winner paying more than R_s gains by dropping out, which pins down the equilibrium payoffs (Appendix A.3). The bit-prefix and post-trade-arbitrage propositions apply the threshold and the constant-product price movement. Complete proofs are in Appendices A and B.

6 Extensions, Boundary Cases, and Conservative Variants

The exact threshold in Theorem 4.3 depends only on the input-token reserve X, the slippage tolerance τ, the lower endpoint ℓ of the signal interval, and the fixed execution cost c. It does not depend on the upper endpoint u, because the trader worries only about the smallest hidden trade compatible with the signal. Perfect hiding eliminates universal robust sandwiching for every $c > 0$ only when the global type space has no positive lower bound. If every relevant victim order is known a priori to satisfy $q \geq q_{\min} > 0$, complete hiding still reveals $\ell = q_{\min}$, and universal robust sandwiching survives whenever $q_{\min} > \lambda_c$.

6.1 Relaxing Robustness

The baseline model insists that a bundle remain valid for *every* hidden type in the leaked interval. This is the strongest notion of feasibility, and it is exactly why Theorem 4.3 is distribution free. If a trader is willing to tolerate rare failures and the posterior has full support on the leaked interval, the lower endpoint ℓ is replaced by a lower posterior quantile.

Definition 6.1 (ε-admissibility)**.** Fix a signal interval $I = [\ell, u]$, a posterior μ on I whose cumulative distribution function F_μ is continuous and strictly increasing on I, and a tolerance $\varepsilon \in [0, 1)$. A front-run size $a \geq 0$ is *ε-admissible* if

$$\mu\big(\{q \in I : v(a, q) \geq (1 - \tau)h(q)\}\big) \geq 1 - \varepsilon.$$

For $\varepsilon = 0$, this reduces to robust admissibility.

Proposition 6.2 (Chance-constrained admissibility)**.** *Let I, μ, and F_μ be as in Definition 6.1. For $\varepsilon \in [0, 1)$ let*

$$q_\varepsilon := \inf\{z \in I : F_\mu(z) \geq \varepsilon\}$$

be the lower ε-quantile. Then the ε-admissible set is

$$\mathcal{A}_\varepsilon(I, \mu) = [0, a_{\max}(q_\varepsilon, \tau)].$$

In particular, allowing a positive failure probability weakly enlarges the feasible set, and the robust case is recovered at $\varepsilon = 0$ with $q_0 = \ell$.

The proof is the same lower-tail monotonicity argument as in Theorem 4.1, with ℓ replaced by the lower posterior quantile; see Appendix B.

Remark 6.3 (Why the robust threshold is distribution free). Theorem 4.3 depends only on ℓ because it asks for positive net payoff for every hidden type. Once the attacker instead maximizes expected profit with occasional failures, or once failed bundles incur a separate revert loss, the relevant objective depends on the full posterior on q, not just its support minimum. Costs need the same care. If one insists on uniform profitability against every cost realization in a bounded set, the robust benchmark replaces c by a worst-case upper bound. A random cost independent of q and priced in expectation is replaced by its mean, and the threshold stays distribution free; a cost that depends on q and is priced in expectation brings in the posterior, and the threshold is no longer distribution free.

6.2 Hidden Metadata Beyond Size

The base model assumes that size is partially hidden while direction and slippage are public.

Proposition 6.4 (Direction ambiguity rules out non-contingent front-runs)**.** *Suppose the signal reveals only that a single swap of size $q \in I$ will occur, while the direction may be either $X \to Y$ or $Y \to X$, and a trader must commit to a non-contingent first leg before the direction is revealed. Call a first leg a* front-run against *a direction if, for every $q \in I$, it lowers the output that a victim trading in that direction receives, relative to honest execution. Then no non-null first leg is a front-run against both directions.*

The proof observes that any non-null first leg moves the pool price in only one direction and therefore cannot worsen both possible victim directions; see Appendix B.

Proposition 6.5 (Hidden slippage reduces to a lower slippage bound)**.** *Suppose that after signal s the victim size lies in $I_s = [\ell_s, u_s]$ and the victim's slippage tolerance lies in a public set $T_s \subset (0, 1)$ with*

Table 1: Dimensionless leakage threshold λ_c/X for representative cost ratios and slippage tolerances.

c/X	$\tau = 0.001$	$\tau = 0.005$	$\tau = 0.01$	$\tau = 0.02$
10^{-6}	0.00100	0.00020	0.00010	0.00005
10^{-5}	0.00990	0.00200	0.00100	0.00050
10^{-4}	0.09162	0.01962	0.00990	0.00498

lower endpoint $\underline{\tau}_s := \inf T_s$. A front-run size a is robustly admissible for every pair $(q, \tau) \in I_s \times T_s$ if and only if

$$(X + a)(X + a + \ell_s) \le \frac{X(X + \ell_s)}{1 - \underline{\tau}_s}.$$

Consequently the optimal robust size is $a_{\max}(\ell_s, \underline{\tau}_s)$, and the worst-case gross value is

$$\frac{\underline{\tau}_s\, \ell_s(X + \ell_s)}{X + \underline{\tau}_s \ell_s}.$$

The proof reduces robustness over T_s to the smallest feasible slippage tolerance and then applies Theorems 4.1 and 4.2; see Appendix B.

6.3 Dimensionless Form and Conservative Variants

The threshold (14) is scale free.

Proposition 6.6 (Scale-free threshold form)**.** *Let $\widehat{c} := c/X$ and $\widehat{\lambda} := \lambda_c/X$. Then*

$$\widehat{\lambda} = \frac{-(1 - \widehat{c}) + \sqrt{(1 - \widehat{c})^2 + \frac{4\widehat{c}}{\tau}}}{2}.$$

Moreover, as $\widehat{c} \downarrow 0$,

$$\widehat{\lambda} = \frac{\widehat{c}}{\tau} + O(\widehat{c}^2).$$

The derivation is a normalization of (14) followed by a first-order Taylor expansion; see Appendix B.

Proposition 6.7 (Positive swap fees only make universal sandwiching harder)**.** *Suppose every swap applies a fee rate $\rho \in [0, 1)$ that is withheld from the input and not added to the reserves, as in Uniswap v3 [2], so that only the effective input $\alpha\delta$ with $\alpha = 1 - \rho$ enters the pricing function and the reserves. Then:*

(i) the victim-admissibility condition becomes

$$(X + \alpha a)(X + \alpha a + \alpha q) \le \frac{X(X + \alpha q)}{1 - \tau};$$

(ii) for every fixed $a, q > 0$, the trader's gross sandwich profit is weakly smaller than in the fee-free model; and

(iii) therefore the fee-free threshold from Theorem 4.3 is conservative: if universal robust sandwiching is impossible at $\rho = 0$, then it is also impossible at fee rate ρ.

The proof replaces each raw input by its effective input and compares every leg with the fee-free benchmark; see Appendix B. Under Uniswap v2 accounting the fee stays in the pool, and the condition in (i) becomes $(X + \alpha a)(X + a + \alpha q) \le X(X + \alpha q)/(1 - \tau)$.

Example 6.8 (Reading the scale-free threshold)**.** Suppose the input-token reserve is $X = 10^6$, the victim slippage tolerance is $\tau = 0.02$, and the total bundle cost is $c = 100$ in the same units, so that $c/X = 10^{-4}$. Proposition 6.6 gives, to more digits than Table 1,

$$\frac{\lambda_c}{X} \approx 0.0049757, \qquad \text{hence} \qquad \lambda_c \approx 4{,}976.$$

Thus a leaked lower bound of only a few hundred units does not support a universally profitable robust sandwich, while a leaked lower bound around 5×10^3 crosses the threshold.

Remark 6.9 (Multiple victims and cross-venue interactions). The single-victim assumption is what makes the robust feasibility region one-dimensional. If several same-direction victim buys execute after the front-run on the same pool, feasibility involves every victim's own slippage guard, and each later guard also depends on the orders ahead of it, while the unwind profit depends on the cumulative downstream buy flow. Monotonicity then suggests that the smallest consistent flow governs the worst-case profit, but feasibility is no longer a condition on one number. Opposite-direction orders, unknown interleavings, or cross-venue routing add further state variables to both admissibility and unwind profit.

Remark 6.10 (What should persist beyond constant-product pools). Among our arguments, the most robust is the lower-endpoint logic: whenever a pre-trade hurts smaller same-direction victims more than larger ones, a leakage signal should again matter mainly through a worst-case lower tail. The exact cancellations behind (12), however, are specific to the constant-product invariant; stableswap-style or piecewise invariants are likely to preserve only the monotonicity backbone, not the same closed-form threshold. Uniswap v3 is covered locally, as a fee-free benchmark. Within a tick range with fixed active liquidity Λ, a v3 pool trades against virtual reserves (X_v, Y_v) with $X_v Y_v = \Lambda^2$ [2]. If, for every front-run size $a \in [0, a_{\max}(\ell_s, \tau)]$ and every victim size in I_s, the front-run, the victim trade, and the unwind stay within one such range, the fee-free results apply with (X, Y) replaced by the virtual reserves; larger front-runs stay inadmissible, because the victim's output falls as the front-run grows. In particular, u_s matters for whether the results apply. Every v3 pool charges a fee, and with a fee Proposition 6.7 gives only a one-sided comparison. A swap that crosses an initialized tick changes the active liquidity, the calculation becomes piecewise, and the closed-form threshold need not survive; that case lies outside our theorems.

What a private mempool must hide. Suppose execution is costly, $c > 0$. To rule out sandwiches that profit at every consistent size, a privacy layer may then reveal any upper bound on an order's size but no lower bound above λ_c (Theorem 4.3; for a revealed upper bound u alone, Corollary 4.4 with u in place of Q). To first order in c/X that threshold is c/τ (Proposition 6.6), so halving the victims' slippage tolerance roughly doubles the lower bound that can leak. Revealing the first d bits of the size exposes exactly the bins whose lower edge exceeds λ_c (Proposition 4.6), and a swap fee withheld from the reserves can only shrink the set of exposed lower bounds (Proposition 6.7). Hiding the direction removes the non-contingent sandwich for every cost: when the first leg must be chosen before the direction is revealed, no non-null first leg front-runs both directions (Proposition 6.4).

7 Open Problems

Several natural extensions remain open.

1. **Multiple victims and bundle interactions.** The present paper studies one hidden victim order. In richer rounds there may be many orders, and the trader may condition on joint interval information or attack only a subset.
2. **General CFMMs.** Constant-product pools admit clean cancellations; which of our monotonicity arguments survive for broader classes of CFMMs is open.
3. **Richer allocation mechanisms.** We reduced the post-signal market to symmetric traders competing for a single execution slot. Extending the analysis to multi-stage or asymmetric allocation mechanisms while retaining exact theorems remains open.
4. **Endogenous leakage.** Our leakage map is exogenous. Optimizing the information structure subject to implementability or efficiency constraints, or letting it be released strategically over time, turns the model into an information-design problem [18, 9].

Acknowledgments and Disclosure of Funding

This work received no third-party funding. The authors declare no competing interests. The views expressed in this paper are those of the authors and do not necessarily reflect the views of Tropic.

References

[1] Hayden Adams, Noah Zinsmeister, and Dan Robinson. Uniswap v2 Core. Whitepaper, 2020.

[2] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 Core. Whitepaper, 2021.

[3] Guillermo Angeris, Alex Evans, and Tarun Chitra. A Note on Privacy in Constant Function Market Makers. *arXiv preprint arXiv:2103.01193*, 2021.

[4] Kushal Babel, Nerla Jean-Louis, Yan Ji, Ujval Misra, Mahimna Kelkar, Kosala Yapa Mudiyanselage, Andrew Miller, and Ari Juels. PROF: Protected Order Flow in a Profit-Seeking World. In *2026 IEEE 11th European Symposium on Security and Privacy (EuroS&P)*, pages 398–418, 2026.

[5] Maryam Bahrani, Pranav Garimidi, and Tim Roughgarden. Transaction Fee Mechanism Design with Active Block Producers. In *Financial Cryptography and Data Security. FC 2024 International Workshops*, Lecture Notes in Computer Science, pages 85–90. Springer, 2025.

[6] Jinyu Bai, Dongze Li, Zhenxuan Jiang, and Gang Du. Design and Implementation of a Real-Time Detection System for Multi-Token Sandwich Attacks in Ethereum Based on Geth Client. *Discover Computing*, 28:180, 2025.

[7] Joseph Bebel and Dev Ojha. Ferveo: Threshold Decryption for Mempool Privacy in BFT Networks. *Cryptology ePrint Archive*, Paper 2022/898, 2022.

[8] Dirk Bergemann, Benjamin Brooks, and Stephen Morris. First-Price Auctions with General Information Structures: Implications for Bidding and Revenue. *Econometrica*, 85(1):107–143, 2017.

[9] Dirk Bergemann and Stephen Morris. Information Design: A Unified Perspective. *Journal of Economic Literature*, 57(1):44–95, 2019.

[10] Eric Budish, Peter Cramton, and John Shim. The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response. *Quarterly Journal of Economics*, 130(4):1547–1621, 2015.

[11] T.-H. Hubert Chan, Ke Wu, and Elaine Shi. Mechanism Design for Automated Market Makers. In *7th Conference on Advances in Financial Technologies (AFT 2025)*, LIPIcs 354, pages 7:1–7:22, 2025.

[12] Hao Chung and Elaine Shi. Foundations of Transaction Fee Mechanism Design. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3856–3899, 2023.

[13] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability. In *2020 IEEE Symposium on Security and Privacy*, pages 910–927, 2020.

[14] Stefan Dziembowski, Sebastian Faust, and Jannik Luhn. Shutter Network: Private Transactions from Threshold Cryptography. *Cryptology ePrint Archive*, Paper 2024/1981, 2024.

[15] Matheus Venturyne Xavier Ferreira and David C. Parkes. Credible Decentralized Exchange Design via Verifiable Sequencing Rules. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 723–736, 2023.

[16] Lioba Heimbach and Roger Wattenhofer. Eliminating Sandwich Attacks with the Help of Game Theory. In *Proceedings of the 17th ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, pages 153–167, 2022.

[17] Lioba Heimbach and Roger Wattenhofer. SoK: Preventing Transaction Reordering Manipulations in Decentralized Finance. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies (AFT)*, pages 47–60, 2022.

[18] Emir Kamenica and Matthew Gentzkow. Bayesian Persuasion. *American Economic Review*, 101(6):2590–2615, 2011.

[19] Alireza Kavousi, Duc V. Le, Philipp Jovanovic, and George Danezis. BlindPerm: Efficient MEV Mitigation with an Encrypted Mempool and Permutation. In *29th International Conference on Principles of Distributed Systems (OPODIS 2025)*, volume 361 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:21, 2026.

[20] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. Order-Fairness for Byzantine Consensus. In *Advances in Cryptology – CRYPTO 2020*, Lecture Notes in Computer Science, pages 451–480. Springer, 2020.

[21] Kshitij Kulkarni, Theo Diamandis, and Tarun Chitra. Towards a Theory of Maximal Extractable Value I: Constant Function Market Makers. *arXiv preprint arXiv:2207.11835*, 2022.

[22] Rujia Li, Xuanwei Hu, Qin Wang, Sisi Duan, and Qi Wang. Transaction Fairness in Blockchains, Revisited. *IEEE Transactions on Dependable and Secure Computing*, 23(1):752–765, 2026.

[23] Robert McLaughlin, Christopher Kruegel, and Giovanni Vigna. A Large Scale Study of the Ethereum Arbitrage Ecosystem. In *32nd USENIX Security Symposium*, pages 3295–3312, 2023.

[24] Andreas Park. The Conceptual Flaws of Decentralized Automated Market Making. *Management Science*, 69(11):6731–6751, 2023.

[25] Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying Blockchain Extractable Value: How Dark is the Forest? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 198–214, 2022.

[26] Antoine Rondelet and Quintus Kilbourn. Mempool Privacy: An Economic Perspective. *arXiv preprint arXiv:2307.10878*, 2023.

[27] Tim Roughgarden. Transaction Fee Mechanism Design. *Journal of the ACM*, 71(4):1–25, 2024.

[28] Christof Ferreira Torres, Ramiro Camino, and Radu State. Frontrunner Jones and the Raiders of the Dark Forest: An Empirical Study of Frontrunning on the Ethereum Blockchain. In *30th USENIX Security Symposium*, pages 1343–1359, 2021.

[29] Sarisht Wadhwa, Luca Zanolini, Aditya Asgaonkar, Francesco D'Amato, Chengrui Fang, Fan Zhang, and Kartik Nayak. Data Independent Order Policy Enforcement: Limitations and Solutions. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pages 378–392, 2024.

[30] Liyi Zhou, Kaihua Qin, and Arthur Gervais. A2MM: Mitigating Frontrunning, Transaction Reordering and Consensus Instability in Decentralized Exchanges. *arXiv preprint arXiv:2106.07371*, 2021.

[31] Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V. Le, and Arthur Gervais. High-Frequency Trading on Decentralized On-Chain Exchanges. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 428–445, 2021.

A Deferred Proofs for the Main Results

A.1 Structural lemmas

Proof of Lemma 5.1. For fixed a, write

$$r(a,q) = \frac{X}{X+a} \cdot \frac{X+q}{X+a+q}.$$

The first factor is constant in q, and at $a = 0$ both factors equal 1. For $a > 0$ we differentiate the second factor:

$$\frac{d}{dq}\left(\frac{X+q}{X+a+q}\right) = \frac{a}{(X+a+q)^2} > 0.$$

□

Proof of Lemma 5.2. Let

$$N(a,q) = aq(2X+a+q), \qquad D(a,q) = (X+a)^2 + aq,$$

so that $\pi = N/D$.

For the a-derivative,

$$\frac{\partial N}{\partial a} = q(2X+2a+q) \quad \text{and} \quad \frac{\partial D}{\partial a} = 2X+2a+q.$$

Hence

$$\frac{\partial \pi}{\partial a} = \frac{\left(\frac{\partial N}{\partial a}\right) D - N\left(\frac{\partial D}{\partial a}\right)}{D^2} = \frac{(2X+2a+q)(qD-N)}{D^2}.$$

Now

$$qD - N = q\big((X+a)^2 + aq\big) - aq(2X+a+q) = qX^2,$$

so

$$\frac{\partial \pi}{\partial a} = \frac{qX^2(2X+2a+q)}{D^2} > 0.$$

For the q-derivative,

$$\frac{\partial N}{\partial q} = a(2X+a+2q) \quad \text{and} \quad \frac{\partial D}{\partial q} = a.$$

Therefore

$$\frac{\partial \pi}{\partial q} = \frac{a(2X+a+2q)D - aN}{D^2} = \frac{a\big((2X+a+2q)(X+a)^2 + aq^2\big)}{D^2} > 0.$$

□

A.2 Main results

Proof of Theorem 4.1. By (6), a front-run size a is admissible for type q if and only if

$$\frac{v(a,q)}{h(q)} \geq 1-\tau.$$

Using (3) and (5), this becomes

$$\frac{X(X+q)}{(X+a)(X+a+q)} \geq 1-\tau.$$

Equivalently,

$$(X+a)(X+a+q) \leq \frac{X(X+q)}{1-\tau}.$$

By Lemma 5.1, $r(a,q)$ is nondecreasing in q, so the condition $r(a,q) \geq 1-\tau$ is hardest to satisfy at the smallest feasible victim size. Hence a is robustly admissible for the entire interval $I = [\ell, u]$ if and only if

$$(X+a)(X+a+\ell) \leq \frac{X(X+\ell)}{1-\tau},$$

which is (10). Expanding gives

$$a^2 + (2X + \ell)a - \frac{\tau X(X + \ell)}{1 - \tau} \le 0.$$

Since the quadratic is convex and vanishes at exactly one nonnegative root, the robustly admissible set is the interval $[0, a_{\max}(\ell, \tau)]$, where the root is exactly (9).

For $I = (0, u]$, each $a \in \mathcal{A}(I)$ is robustly admissible on $[\ell, u]$ for every $\ell \in (0, u]$, so $a \le a_{\max}(\ell, \tau)$, and letting $\ell \downarrow 0$ gives $a \le a_{\max}(0, \tau) = a_0$. For the converse we note that the left-hand side of the admissibility condition increases in a, so it suffices that a_0 is admissible for every $q > 0$. This follows from $(X + a_0)^2 = X^2/(1 - \tau)$ and $X + a_0 \le X/(1 - \tau)$:

$$(X + a_0)(X + a_0 + q) = \frac{X^2}{1 - \tau} + (X + a_0)q \le \frac{X(X + q)}{1 - \tau}.$$

□

Proof of Theorem 4.2. Part (a) is Lemma 5.2.

For part (b), by Theorem 4.1 the admissible set is the interval $[0, a_{\max}(\ell, \tau)]$. Since $\pi(a, q)$ is strictly increasing in a for every fixed q, the maximizer of $\pi(a, q)$ over the admissible set is $a_{\max}(\ell, \tau)$ for every fixed $q \in I$. The same monotonicity implies that the posterior expectation $\mathbb{E}_{q\sim\mu}[\pi(a, q)]$ is strictly increasing in a, because it is the expectation of a pointwise increasing function. The worst-case value $\min_{q\in I} \pi(a, q)$ is also strictly increasing in a. Therefore the unique optimizer in all three problems is $a^\star(I) = a_{\max}(\ell, \tau)$. For $I = (0, u]$ we use $\mathcal{A}(I) = [0, a_0]$ from Theorem 4.1, and the same argument gives the pointwise and expected optimizers; the worst-case problem is excluded there because $\inf_{q\in I} \pi(a, q) = 0$ is not attained for $a > 0$. Part (d) follows immediately from part (b).

For part (c), Lemma 5.2 also implies that for fixed $a^\star(I)$ the smallest profit over $q \in I$ occurs at $q = \ell$, so $W(I) = \pi(a^\star(I), \ell)$, which (17) evaluates with $\sigma = X + a^\star(I)$. Its first step substitutes $a^\star(I) = \sigma - X$ into (8), and its middle step uses the admissibility equality $\sigma(\sigma + \ell) = X(X + \ell)/(1 - \tau)$ twice:

$$(\sigma - X)(\sigma + X + \ell) = \sigma^2 + \sigma\ell - X(X + \ell) = \frac{\tau X(X + \ell)}{1 - \tau}, \qquad \sigma^2 + (\sigma - X)\ell = \sigma(\sigma + \ell) - X\ell = \frac{X(X + \tau\ell)}{1 - \tau}.$$

□

Proof of Theorem 4.3. By Theorem 4.2, the largest guaranteed gross profit obtainable by any robustly admissible bundle on interval I is exactly $W(I)$. Therefore a robustly admissible bundle has strictly positive net payoff for every $q \in I$ if and only if $W(I) > c$, proving (13).

It remains to solve this inequality explicitly. By (12),

$$\frac{\tau\ell(X + \ell)}{X + \tau\ell} > c$$

if and only if

$$\tau\ell(X + \ell) > c(X + \tau\ell),$$

that is,

$$\tau\ell^2 + \tau(X - c)\ell - cX > 0.$$

The left-hand side is a convex quadratic in ℓ, and we write its larger root as

$$\lambda_c = \frac{-\tau(X - c) + \sqrt{\tau^2(X - c)^2 + 4\tau cX}}{2\tau} = \frac{-(X - c) + \sqrt{(X - c)^2 + \frac{4cX}{\tau}}}{2}.$$

For $c > 0$ the product of its roots is $-cX/\tau < 0$, so the other root is negative; for $c = 0$ the roots are $-X$ and $\lambda_0 = 0$. In both cases a positive ℓ makes the quadratic positive exactly when $\ell > \lambda_c$. □

Proof of Corollary 4.4. By Theorem 4.2,

$$W([\ell, Q]) = \frac{\tau\ell(X + \ell)}{X + \tau\ell},$$

which tends to 0 as $\ell \downarrow 0$.

Let $c > 0$, and suppose, for contradiction, that a robustly admissible a has $\pi(a, q) - c > 0$ for every $q \in (0, Q]$. For each $\ell \in (0, Q]$ the front-run a is robustly admissible on $[\ell, Q] \subseteq (0, Q]$ with positive net payoff there, so $W([\ell, Q]) > c$ by Theorem 4.3. This fails once ℓ is small enough that $W([\ell, Q]) < c$.

For the last claim we run the proof of Theorem 4.5(a) with $I = I_{s_\perp}$. It uses only that $\mathbb{E}[\pi(a, q) \mid s_\perp] \le V$ for every robustly admissible a, which holds with $V := R_{s_\perp} + c$ by the definition of $R_{s_\perp}$. So $R_{s_\perp} < 0$ rules out an on-path sandwich. □

A.3 Execution auction equilibrium

Proof of Theorem 4.5. Fix a signal s, write $I = I_s$, and abbreviate $a^\star = a^\star(I)$ and $V = V_{\mu_s}(I)$, so $R_s = V - c$. Two facts carry the argument. Every robustly admissible a has $\mathbb{E}[\pi(a, q) \mid s] \le V$, with equality only at $a = a^\star$ (Theorem 4.2). A trader can always submit null and get 0, so in an equilibrium the winner's expected utility is at least 0.

For part (a), suppose first that $R_s < 0$. A winning bundle (a, b) gives its owner

$$\mathbb{E}[\pi(a, q) \mid s] - b - c \le R_s - b < 0,$$

less than null secures, so no equilibrium has an on-path sandwich. Now let $R_s = 0$. If every trader submits null, a trader who deviates and wins with (a, b) gets at most $R_s - b = -b \le 0$, so this profile is an equilibrium. In an equilibrium whose winner bids (a, b), the winner's utility u satisfies $0 \le u \le R_s - b = -b \le 0$, so $b = 0$ and $u = 0$; the auctioneer receives 0 and every losing trader gets 0.

For part (b), let $R_s > 0$. Some trader attacks on path: if every trader submitted null, a trader who deviated to $(a^\star, b)$ with $0 \le b < R_s$ would win alone and earn $R_s - b > 0$. Let (a, b) be the winning bundle and $u \ge 0$ its owner's utility. We claim $a = a^\star$. Otherwise $\mathbb{E}[\pi(a, q) \mid s] < V$, so $V - b - c > u \ge 0$. A losing trader, who gets 0, could then bid $(a^\star, b + \varepsilon)$ with $0 < \varepsilon < V - b - c$; its payment exceeds every other payment, so it wins and earns $V - b - \varepsilon - c > 0$. Hence $a = a^\star$, and $u = R_s - b \ge 0$ gives $b \le R_s$. If $b < R_s$, a losing trader who bids $(a^\star, b + \varepsilon)$ with $b + \varepsilon < R_s$ wins and earns $R_s - b - \varepsilon > 0$. So $b = R_s$. The winner then gets 0, as does every losing trader, and the auctioneer receives R_s. □

A.4 Bit-prefix leakage and post-trade arbitrage

Proof of Proposition 4.6. For $j \ge 1$ the bin I_j is a closed interval with lower endpoint $\ell_j = jQ/2^d > 0$, so Theorem 4.3 makes universal robust profitability on it equivalent to $\ell_j > \lambda_c$. For the lowest bin we apply Corollary 4.4 with $Q/2^d$ in place of Q: since $I_0 = (0, Q/2^d]$ reveals no positive lower bound, for $c > 0$ it supports no universally profitable robust sandwich. With $d = 0$ it is $I_{s_\perp}$. □

Proof of Proposition 4.7. The first claim is Corollary 4.4. For the second we use $\mathcal{A}((0, Q]) = [0, a_0]$ from Theorem 4.1. Every posterior is supported on $(0, Q]$, so Lemma 5.2 gives $\mathbb{E}[\pi(a, q) \mid s_\perp] \le \pi(a_0, Q) < c$. Hence $R_{s_\perp} < 0$, and Corollary 4.4 rules out an on-path sandwich.

After a victim trade of size q the reserves are $(X + q, XY/(X + q))$. A trader who buys $\eta > 0$ units of Y at the external price X/Y and sells them to the pool earns, by (2),

$$f(\eta) = \frac{(X + q)^2 \eta}{XY + (X + q)\eta} - \frac{X\eta}{Y}.$$

So $f(\eta) > 0$ exactly when $\eta < Yq(2X + q)/(X(X + q))$, and f is largest at $\eta = Yq/(X + q)$:

$$f\left(\frac{Yq}{X + q}\right) = \frac{q^2}{X + q}.$$

□

B Deferred Proofs for Extensions

Proof of Proposition 6.2. We fix $a \geq 0$. By Lemma 5.1, if a is admissible for type q, it is admissible for every larger type, and (6) is a non-strict inequality between continuous functions of q. So the failure set is $\{q \in I : q < q^*\}$ for some $q^* \in [\ell, \infty]$, and the failure probability is $F_\mu(\min\{q^*, u\})$. Since F_μ is continuous and strictly increasing on I, this is at most ε exactly when $q^* \leq q_\varepsilon$, that is, when a is admissible for every $q \in [q_\varepsilon, u]$. By Theorem 4.1 this happens exactly when $a \leq a_{\max}(q_\varepsilon, \tau)$. □

Proof of Proposition 6.4. A non-null first leg either buys Y with X or sells Y for X; the two cases are symmetric under token exchange, so we treat a buy of size $a > 0$. It leaves the reserves at $(X + a, XY/(X + a))$. A victim who then sells q units of Y receives more than its honest output $Xq/(Y + q)$, so the leg does not front-run the direction $Y \to X$:

$$\frac{(X+a)^2 q}{XY + (X+a)q} = \frac{Xq}{Y+q} + \frac{aq(2XY + Xq + Ya + aq)}{(Y+q)(XY + Xq + aq)}.$$

□

Proof of Proposition 6.5. For fixed a and q, the right-hand side $X(X+q)/(1-\tau)$ of the admissibility condition is continuous and increasing in τ. So a is admissible for every $\tau \in T_s$ exactly when it is admissible at $\underline{\tau}_s$; if $\underline{\tau}_s \notin T_s$, we let $\tau \downarrow \underline{\tau}_s$ within T_s, which preserves the non-strict inequality. Robustness over $I_s \times T_s$ is therefore robustness over I_s at $\underline{\tau}_s$. For $\underline{\tau}_s > 0$, Theorems 4.1 and 4.2 with $\tau = \underline{\tau}_s$ give the stated formulas. For $\underline{\tau}_s = 0$, the condition $(X + a)(X + a + \ell_s) \leq X(X + \ell_s)$ forces $a = 0$, and both formulas give 0. □

Proof of Proposition 6.6. Substitute $c = \widehat{c}X$ and $\lambda_c = \widehat{\lambda}X$ into (14) and divide through by X. The expansion follows from a first-order Taylor series of the square root around $\widehat{c} = 0$. □

Proof of Proposition 6.7. With the fee withheld, each swap acts on the reserves exactly as the fee-free swap of its effective input, so (1) and (2) apply with every input δ replaced by $\alpha\delta$.

For (i), the victim's honest output is $h(\alpha q)$ and its output after the front-run is $v(\alpha a, \alpha q)$, so the condition $v \geq (1 - \tau)h$ is the fee-free condition (6) at $(\alpha a, \alpha q)$, which is the display in (i).

For (ii), write $\pi_\rho(a, q)$ for the gross profit with the fee. The front-run buys $y = y_F(\alpha a)$, the victim moves the pool by the effective input αq, and the unwind returns $\Delta_X(\alpha y; \cdot) \leq \Delta_X(y; \cdot)$ because Δ_X increases in its input. So, with the last step by Lemma 5.2,

$$\pi_\rho(a, q) \leq x_B(\alpha a, \alpha q) - a = \pi(\alpha a, \alpha q) - \rho a \leq \pi(a, q).$$

For (iii), we take an a that is robustly admissible with the fee on $I = [\ell, u]$ and has $\pi_\rho(a, q) > c$ for every $q \in I$. By (i), αa is fee-free admissible at the type $\alpha\ell \leq \ell$, hence by Lemma 5.1 at every $q \geq \alpha\ell$, so $\alpha a \in \mathcal{A}(I)$. By (ii) and Lemma 5.2, $\pi(\alpha a, q) \geq \pi(\alpha a, \alpha q) \geq \pi_\rho(a, q) > c$ for every $q \in I$. So αa is a universally profitable robust sandwich without the fee. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction state the theorem-level contributions and delimit the model as a stylized theoretical analysis. The main claims are matched to formal definitions, theorem statements, corollaries, and proof sections in the body of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes].

   Justification: The paper discusses its modeling scope, boundary cases, and open problems in the introduction, model discussion, boundary/extension discussion, and concluding open-problems material. In particular, the analysis is presented as a theoretical characterization rather than an empirical or deployed-system evaluation.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes].

Justification: The paper states the assumptions through formal definitions and theorem hypotheses, and provides proofs for the main theorems, lemmas, propositions, and corollaries. Results explicitly marked as sketches or boundary observations are identified as such.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [N/A].

Justification: The paper does not report experiments. The main results are mathematical statements whose verification depends on the formal model, theorem statements, and proofs provided in the paper.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [N/A].

   Justification: The paper does not use datasets, trained models, or experimental code to support its claims. Algorithmic statements, where present, are specified and analyzed mathematically in the text.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [N/A].

   Justification: The paper contains no training, test set, hyperparameter, optimizer, or empirical evaluation setup. Its results are derived from the stated theoretical model.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A].

   Justification: The paper does not include experiments or statistical estimates. Consequently, there are no empirical error bars, confidence intervals, or significance tests to report.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [N/A].

   Justification: The paper does not run computational experiments. No experimental compute resources are required to reproduce the paper's main claims.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: The work is a theoretical analysis and does not involve human subjects, private data, deployed interventions, or release of potentially harmful models or datasets. The paper preserves anonymity in the submission version.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes].

Justification: The paper discusses positive implications for robust, privacy-preserving, or incentive-compatible decentralized mechanisms, while also identifying boundary cases and residual attack surfaces. The work is theoretical and does not propose a deployed system, so the impact discussion is correspondingly scoped to technical security and market-design implications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A].

Justification: The paper does not release data, trained models, scraped datasets, or other assets with high misuse risk. No release-specific safeguards are therefore applicable.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [N/A].

Justification: The paper does not use existing code, datasets, models, benchmarks, or other external assets as research inputs. Prior scholarly work is credited through citations in the related-work and references sections.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [N/A].

Justification: The paper does not introduce or release new datasets, code packages, models, or benchmarks. Its contribution consists of formal models, theorems, and proofs.

Guidelines:

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

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

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

Answer: [N/A].

Justification: The paper does not involve crowdsourcing, surveys, experiments with human participants, or human-subject data collection. There are therefore no participant instructions, screenshots, or compensation details to report.

Guidelines:

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

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

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

    Answer: [N/A].

    Justification: The paper does not involve human subjects or crowdsourced participants. IRB or equivalent human-subjects review is therefore not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes].

    Justification: AI tools were used to help check the manuscript's derivations and proofs, and to provide suggestions for improving and revising the derivations.

    Guidelines:

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