Breaking Sandwich MEV in ERC-20 DeFi via Context-Aware Directional Constraints

Prerna Arote
Distributed Systems and Networks Group,
IMDEA Networks Institute,
Madrid, Spain
Email: prerna.arote@networks.imdea.org

***Abstract*—Decentralized exchanges (DEXs) are a core component of decentralized finance (DeFi) but remain vulnerable to Maximal Extractable Value (MEV) attacks, particularly sandwich attacks that exploit transaction ordering across accounts and blocks. Existing defenses often rely on protocol modifications or identity-based assumptions, limiting their practicality and robustness.**

We present *Context-Aware Directional Constraint* (CADC), a lightweight application-layer defense implemented as an ERC-20–compatible token. CADC enforces directional consistency within pool-specific execution contexts and maintains a per-direction entry price to prevent both immediate and delayed profit extraction. By constraining the temporal and price conditions required for attack execution, CADC remains effective against multi-address and cross-block strategies. Evaluation on a testnet shows that CADC prevents same-block and delayed sandwich attacks while preserving normal trading behavior, with minimal overhead (2–3k gas per pool interaction, 3–8% of typical swaps). CADC provides a practical and deployable MEV mitigation without requiring changes to the underlying blockchain protocol.

***Index Terms*—Decentralized Finance, Maximal Extractable Value, Sandwich Attacks, ERC-20 Tokens, Decentralized Exchanges**

I. Introduction

Decentralized Finance (DeFi) has emerged as a major application of blockchain technology, enabling services such as trading, lending, and asset exchange through smart contracts [1]. Within this ecosystem, Decentralized Exchanges (DEXs) allow users to trade assets directly without intermediaries [2]. However, the transparency of these systems exposes transactions to adversarial strategies known as Maximal Extractable Value (MEV), where attackers exploit transaction ordering for profit [3]. A prominent example is the *sandwich attack*, in which an adversary places transactions before and after a victim's trade to manipulate prices and extract risk-free profit. Such attacks are widespread, particularly on Ethereum, posing ongoing risks to fairness and user welfare in DeFi [4] [5]. Existing mitigation approaches span multiple layers, including protocol-level solutions like proposer-builder separation and encrypted mempools, but these often require substantial changes to consensus mechanisms and face practical deployment challenges [6] [7] [8].

At the application layer, prior defenses attempt to prevent sandwich attacks by detecting transaction patterns or enforcing cooldowns based on sender identity [9]. However, such approaches rely on the assumption that adversaries operate from a single account. In practice, blockchain identities are ephemeral: attackers can easily create and rotate multiple addresses at negligible cost. Empirical evidence shows that many sandwich attacks already exploit multiple accounts or distribute attack steps across blocks as shown in Figure 1, rendering identity-based defenses ineffective [5], [10], [11].

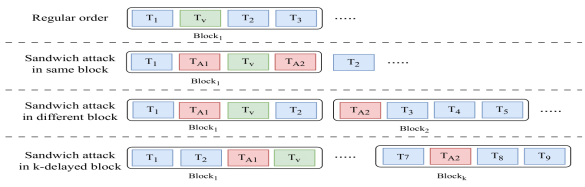

Fig. 1: Illustration of sandwich attack variants (across single or multiple blocks). The attacker's front-running transaction T_{A_1} precedes the victim transaction T_V, while the back-running transaction T_{A_2} follows it.

These limitations suggest that effective mitigation should avoid assumptions about attacker identity and instead constrain the structural conditions required for sandwich attacks. In particular, restricting how trades can be sequenced within a given execution context offers a robust approach against both single-address and coordinated multi-address adversaries.

Motivated by this insight, we propose ***Context-Aware Directional Constraint* (CADC)**, a lightweight application-layer mechanism that extends the ERC-20 token standard without requiring changes to the underlying blockchain protocol. CADC enforces **directional consistency** of trades within a pool-specific execution context over a configurable k-block window, and maintains a per-direction *entry price* p_{entry} to prevent delayed profit extraction through unfavorable reversals. By constraining both the temporal and price conditions necessary for profitable sandwich attacks, CADC remains effective against same-block, cross-block, and multi-address attacks while preserving normal trading behavior.

Our contributions:

- We design **CADC**, an ERC-20–compatible mechanism that enforces directional and price constraints per AMM pool over configurable k-block windows, preventing same-block, multi-address, and delayed sandwich attacks.

- We implement and evaluate CADC on Ethereum Sepolia, demonstrating complete attack prevention with low overhead (2–3k gas per pool interaction, 3–8% of typical swaps) and no cost for wallet-to-wallet transfers.
- We compare CADC with sender-based defenses, showing robustness against address rotation and coordinated multi-account attacks.

II. CADC Overview

We propose *Context-Aware Directional Constraint (CADC)*, an application-layer mitigation for sandwich attacks implemented as an ERC-20–compatible token. Unlike prior solutions that rely on protocol changes, batching, or solver-based architectures, CADC operates at the token layer and enforces constraints on transfers, enabling infrastructure-agnostic protection across AMMs without modifying underlying protocols.

Execution Context. CADC defines an execution context $C = (T, P)$, where T is the token and P is an AMM pool. The contract maintains minimal state per context: last trade block ℓ_C, active trade direction d_C, and an entry price p_{entry}. Trade direction is inferred from token transfers (e.g., $T \to P$ as sell, $P \to T$ as buy). This design supports multiple AMM types without altering pricing mechanisms or invariants (e.g., $x \cdot y = k$).

Directional Constraint. Given a configurable window k, CADC rejects transactions that reverse the trade direction within the same context before k blocks have elapsed. Same-direction trades are always permitted, ensuring continuous trading and preserving market liveness.

Price Constraint. To prevent delayed attacks, trades are grouped into directional epochs. For each epoch, CADC maintains an entry price p_{entry}, initialized at the first trade and updated only for the active direction:

$$p_{\text{entry}} = \max(p_{\text{market}})$$

A reversal is allowed only if:

$$b - \ell_C \geq k \quad \text{and} \quad p_{\text{market}} \geq p_{\text{entry}}$$

This prevents attackers from reversing direction at more favorable prices, eliminating delayed profit extraction.

Design Properties. CADC enforces constraints per execution context rather than sender identity, making it robust to address rotation and coordinated multi-account attacks. The mechanism is deterministic, requires no changes to consensus or AMM logic, and preserves liveness by allowing unrestricted same-direction trades. The protection window (e.g., $k = 12$ blocks) captures the vast majority of observed sandwich attacks while remaining configurable to balance security and usability.

Key Insight. By constraining the temporal and price conditions required for directional reversal, CADC prevents same-block, cross-block, and multi-address sandwich attacks while preserving normal trading behavior and AMM price discovery.

III. Implementation and Evaluation

Implementation and Setup. CADC is implemented as an ERC-20–compatible token and deployed on the Ethereum Sepolia testnet with an initial supply of 10^6 tokens and a configurable protection window of $k = 12$ blocks. We evaluate the system using a standard AMM pool with multiple accounts simulating attacker and victim behavior to capture both same-block and cross-block sandwich attacks. Implementation details are available at: https://github.com/prerna13/anti-sandwichERC20.

Attack Prevention. We simulate a sandwich attack consisting of (i) a frontrun, (ii) a victim trade, and (iii) a backrun from a different address. While the frontrun and victim transactions execute successfully, the backrun transaction is reverted due to violation of the directional constraint ($b - \ell_C < k$) as shown in Figure 2. This demonstrates that CADC prevents multi-address and cross-block sandwich attacks by blocking rapid directional reversals.

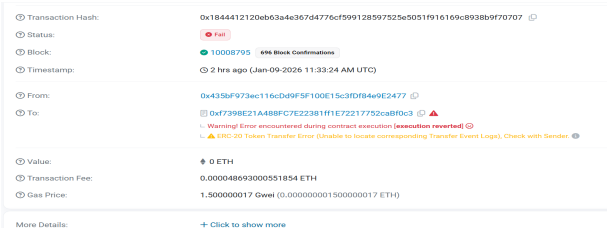

Fig. 2: Backrun transaction reverted under CADC.

Key Insight. Enforcement is scoped to the execution context (T, P) rather than sender identity, making address rotation ineffective while preserving normal same-direction trades. The comparison of CADC with Coo is given in Table I.

Overhead. CADC introduces a small overhead of 2–3k gas per pool-interacting transfer (3–8% of typical swaps), with no additional cost for wallet-to-wallet transfers.

TABLE I: The comparison of CADC with Sender-Based Cooldown token [9]

Property	CADC (Ours)	Coo
Enforcement Scope	Pool-specific (T, P)	Per-sender
Multi-address Sandwich	✓	✗
Cross-block Sandwich	✓	Limited
Liveness	Immediate	Delayed
ERC-20 Compliance	✓	✓

IV. Conclusion

CADC is a lightweight, ERC-20–compatible defense against sandwich attacks that enforces context-aware directional and price constraints over configurable k-block windows. It prevents same-block, cross-block, and multi-address attacks while preserving continuous trading and AMM price discovery, with minimal overhead (3–8% for pool interactions, none for wallet transfers). By enforcing constraints per execution context rather than sender identity, CADC achieves strong protection without protocol changes. Future work includes extending CADC to multi-pool and cross-chain settings.

REFERENCES

[1] G. Liao and D. Robinson, "The dominance of uniswap v3 liquidity," 2022.

[2] S. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. Knottenbelt, "Sok: Decentralized finance (defi)," in *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, 2022, pp. 30–46.

[3] E. Park, T. Yoon, H. Nam, D. Maram, and M. S. Kang, "On frontrunning risks in batch-order fair systems for blockchains," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, 2025, pp. 918–932.

[4] S. Jiang, J. Chen, J. Li, Z. Liu, and Y. Li, "Armm: Sandwich-attack resilient automated market maker," *IEEE Transactions on Services Computing*, 2025.

[5] K. Qin, L. Zhou, P. Gamito, P. Jovanovic, and A. Gervais, "An empirical study of defi liquidations: Incentives, risks, and instabilities," in *Proceedings of the 21st ACM internet measurement conference*, 2021, pp. 336–350.

[6] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *2020 IEEE symposium on security and privacy (SP)*. IEEE, 2020, pp. 910–927.

[7] A. Kavousi, D. V. Le, P. Jovanovic, and G. Danezis, "Blindperm: Efficient mev mitigation with an encrypted mempool and permutation," *Cryptology ePrint Archive*, 2023.

[8] A. Orda and O. Rottenstreich, "Enforcing fairness in blockchain transaction ordering," *Peer-to-peer Networking and Applications*, vol. 14, no. 6, pp. 3660–3673, 2021.

[9] B. Guidi, F. Massellucci, and A. Michienzi, "An anti-sandwich mechanism for evm's smart contracts," *Future Generation Computer Systems*, p. 108077, 2025.

[10] T. Chi, N. He, X. Hu, and H. Wang, "Remeasuring the arbitrage and sandwich attacks of maximal extractable value in ethereum," *arXiv preprint arXiv:2405.17944*, 2024.

[11] J. R. Jensen, V. von Wachter, and O. Ross, "Multi-block mev," *arXiv preprint arXiv:2303.04430*, 2023.