

Received 11 November 2022, accepted 27 November 2022, date of publication 1 December 2022,
date of current version 8 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225884

RESEARCH ARTICLE

Perpetual Contract NFT as Collateral for DeFi Composability

HYOUNGSUNG KIM^{ID}, HYUN-SIK KIM, AND YONG-SUK PARK^{ID}

Korea Electronics Technology Institute, Seoul 03924, Republic of Korea

Corresponding author: Yong-Suk Park (yspark@keti.re.kr)

This work was supported by the Culture, Sports and Tourism Research and Development Program through the Korea Creative Content Agency funded by the Ministry of Culture, Sports and Tourism, in 2022 under Project R2022020034 (Open Metaverse Asset Platform for Digital Copyrights Management) (100%).

ABSTRACT Decentralized Finance (DeFi) is an emerging financial service model based on blockchain technology. DeFi composability denotes the ability for different DeFi services to interact with one another resulting in new forms of financial services. The DeFi ecosystem is largely based on ERC-20 tokens that can represent the value of an asset. Collateralized assets in DeFi composability are locked and additional profit cannot be generated. In this paper, we propose a method to generate profit from locked assets by using ERC-721 Non-Fungible Tokens (NFTs) and perpetual contracts. NFT represents the rights to a certain asset. A perpetual contract is a futures contract that does not have an expiration date. We propose perpetual contract NFT, a new form of NFT that can be used as collateral, which exploits perpetual futures contracts in the cryptocurrency derivatives market. Collateral needs to be provided to back the value of a perpetual contract. If the perpetual contract is minted as NFT, the resulting NFT represents the rights to the perpetual contract and its collateral. Therefore, the perpetual contract NFT itself can be used as collateral for DeFi composability. A proof-of-concept smart contract and a web application for perpetual contract NFT are provided to demonstrate its functionality. To validate the profitability of the perpetual contract NFT using a real-world scenario, we experiment with the position NFT of Uniswap v3 decentralized exchange. The position NFT is a form of perpetual contract NFT. Specifically, we present validation with three types of pools: stablecoins, stablecoin/wrapped tokens pair, and wrapped tokens.

INDEX TERMS Automated market maker (AMM), blockchain, decentralized exchange (DEX), decentralized finance (DeFi), Ethereum, liquidity pool (LP), non-fungible token (NFT), uniswap.

I. INTRODUCTION

Since Satoshi Nakamoto proposed blockchain as a decentralized network and bitcoin as a cryptocurrency for peer-to-peer electronic cash systems [1], many studies have been conducted [2], [3], [4], [5], [6], [7], [8] to apply blockchain and cryptocurrency to various industries. Attempts to adopt cryptocurrencies as mediums for payment and exchange in the finance industry were initially unsuccessful due to their high price volatility. However, the advent of smart contracts and standardized token interfaces [9], [10] in the Ethereum blockchain [11] opened the era of Decentralized Finance (DeFi). DeFi is an emerging financial service model based on

blockchain smart contracts. It provides peer-to-peer financial services and eliminates the need for intermediaries, such as banks or brokerages, enabling faster transactions and increased service accessibility. Currently, most DeFi projects in the market are decentralized applications that run on top of the Ethereum blockchain. Each DeFi protocol can have its own native digital asset or token. The Ethereum standardized token interface ERC-20 [9] allows different smart-contract-enabled tokens to be interchangeable with other tokens, enabling the integration of different services and the creation of new products. Stablecoins, lending and borrowing, decentralized exchanges, derivatives, margin trading, and insurance are some of the services offered by the DeFi ecosystem. Stablecoins are ERC-20 tokens that can hold value equivalent to fiat currencies, such as the

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak^{ID}.

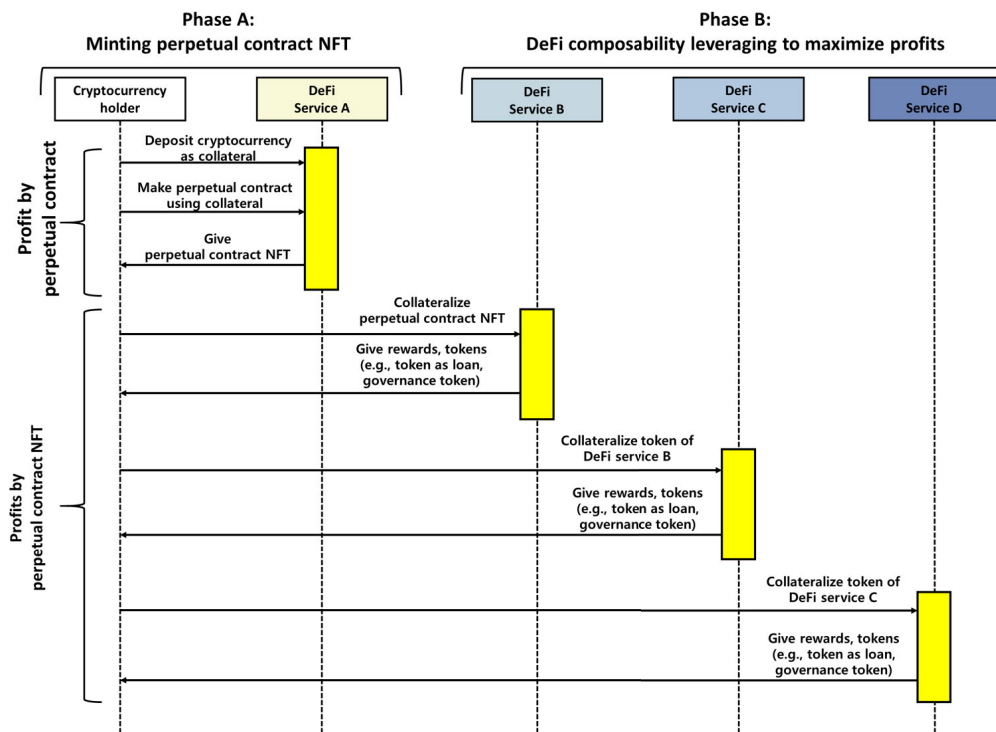


FIGURE 1. High-level architecture diagram of perpetual contract NFT. By exploiting DeFi composability, perpetual contract NFT holders can maximize their profits.

U.S. dollar, or other assets through pegging. This pegging provides price stability in stablecoins, allowing them to be served as mediums of exchange. Strictly speaking, the term “stablecoin” for the pegging “token” is a misnomer. However, in the cryptocurrency industry, this term is widely accepted and used instead of “stabletoken”.

Cryptocurrency lending is one of the most popular services in DeFi. In DeFi lending, cryptocurrency holders or lenders supply their tokens to a lending platform to earn interest. The supplied tokens are made available for other users to borrow by a smart contract. A user who wants to borrow funds must supply some tokens as collateral. Collateralization makes borrowers accountable for repaying the loan. The collateral factor, represented as the Loan-To-Value (LTV) ratio, determines how much can be borrowed based on the type of collateral. Let us assume that a user wants to borrow DAI, a stablecoin cryptocurrency, by providing Ether, the cryptocurrency of Ethereum, as collateral. If the LTV of Ether is 80%, this means that up to 80% of the value of the supplied Ether can be used to borrow DAI. If the value of the collateral decreases, and the loan is unable to meet the required collateral level, the collateral is liquidated by the lending platform to repay the borrowed funds. Notable DeFi lending services are provided by MakerDAO [12] and AAVE [13]. Another popular service in DeFi is Decentralized Exchange (DEX). A DEX is a smart-contract-based marketplace where users can trade cryptocurrencies without the need for an intermediary to facilitate transactions. Liquidity pool protocols are employed to set asset prices and facilitate

trades. Popular DEX services include Curve Finance [14] and Uniswap [15].

Most DeFi services rely on the use of ERC-20 tokens since they can represent the value associated with an asset. ERC-20 tokens of the same type hold the same value, enabling them to be traded and exchanged at equivalency. Ethereum has another type of standardized token interface known as ERC-721 [10]. ERC-721 tokens, also known as Non-Fungible Tokens (NFTs), represent the rights associated with the asset. For example, if an artist mints a piece of work as an NFT, the NFT denotes the work’s digital title deed. Characteristics of ERC-20 and ERC-721 tokens allow them to be widely used in various markets. Other token standards proposed after ERC-20 and ERC-721 are all based on the two token standards, expanding their functionality and usage scenarios. The ERC-1155 [16] is a multi-token standard that can handle the batch transfer of ERC-20 (fungible) or ERC-721 (non-fungible) tokens in a single transaction. By using the ERC-1155 standard, a contractor can mint a token that works either like an ERC-20 or an ERC-721 token.

There were attempts to exploit characteristics of ERC-721 for finance services [17], [18], [19], but the difficulty of NFT valuation restricts their use in DeFi. For comparison, let us assume the use of Wrapped Ether (WETH) in a DeFi lending service. WETH is an ERC-20 token pegged to Ether, meaning WETH and Ether have almost the same market value (i.e., a 1:1 swap ratio) [20]. The DeFi lending platform sets the LTV of collateralized WETH to handle

the fluctuation of WETH price. Users who possess WETH can borrow assets using WETH as collateral from the DeFi lending platform. When the price of WETH decreases steeply, the DeFi lending platform liquidates collateralized WETH to cover the loan. On the other hand, ERC-721, which is used for digital collection NFTs [22], [23], does not have clear criteria for valuation. If someone paid one WETH for an NFT representing the rights to a digital collection, there is no guarantee that it will retain the one WETH value in the future. Hence, it is difficult to use NFTs as primary collateral in DeFi lending platforms.

In this paper, we present a method that enables NFTs to be used as collateral in DeFi DEX services by exploiting perpetual contracts. A perpetual contract is a special form of a futures contract with no expiration date, allowing it to be held or traded for an indefinite amount of time. We propose to combine perpetual contract and NFT by minting perpetual contract as NFT. The conceptual architecture of the proposed method is shown in Fig. 1. The method consists of two phases: minting perpetual contract NFT and DeFi composability leveraging. In the minting phase, cryptocurrency holders deposit their assets to DeFi services. Using this collateral, the holder makes a perpetual contract and mints it as NFT. We name the minted NFT “perpetual contract NFT.” The perpetual contract NFT represents the rights to a perpetual contract, and its collateral is backed by the value of the perpetual contract NFT. In the leveraging phase, the holder can exploit perpetual contract NFT to maximize profit. Profit can be maximized by accumulating rewards from concatenated DeFi services. By locking the holders’ assets in the DeFi service, holders can obtain loans or receive governance tokens as rewards. Holders can lock the rewards received to other DeFi services to accumulate additional rewards. This concatenation of DeFi services is an example of DeFi composability. We present more details about DeFi composability in Section II.

The proposed perpetual contract NFT is not the first attempt to use NFTs as collateral in DeFi. Currently, the Nexo crypto asset digital platform is offering NFT-backed loan services [21], allowing two NFT collections to be used as collateral: Bored Ape Yacht Club [23] and CryptoPunks [24]. The LTV ratio provided is in the range of 10 to 20% of the momentary value of the NFT, and the borrowing rate is in the range of 12 to 20% [21]. In contrast, the proposed perpetual contract NFT mainly exploits the perpetual contract and its collateral ERC-20 tokens, providing many of the options in ERC-20 token-based services. Therefore, services based on perpetual contract NFTs can offer better LTV and borrow ratios compared to NFT-backed lending services like Nexo. Furthermore, since ERC-20 collateral has higher liquidity than NFT collections, the perpetual contract NFT can be combined seamlessly with DeFi composability to maximize profit. The primary contributions of our work are summarized as follows:

- We propose perpetual contract NFTs, a new form of NFT that represents the rights to a perpetual contract

and its collateral, that can be used as collateral for DeFi composability.

- We provide proof-of-concept implementation and demonstration of perpetual contract NFT.
- We provide use case scenarios of perpetual contract NFTs in DeFi services that enable DeFi composability.
- We validate the profitability of collateralizing perpetual contract NFTs using a real-world application.

Popular DeFi services with substantial Total Value Locked (TVL), such as MakerDAO, Lido Finance, and Curve Finance, exploit only ERC-20 tokens or use non-standard token interface service elements for DeFi composability. However, our work uses standard token interfaces and is not limited to ERC-20 tokens. NFT-backed DeFi services, such as Nexo, do not exploit the essence of the NFT. The essence of NFT is the representation of the right to an asset. Current NFT-backed DeFi services focus on the value or worth of the NFT, treating it as an ERC-20 token. On the other hand, the perpetual contract NFT proposed exploits the essence of NFT by using it to represent the right to collateral in DeFi services.

The remainder of this paper is organized as follows. Section II introduces perpetual contracts and DeFi composability in detail to help understand the concepts behind the perpetual contract NFT. The pseudo codes for the core smart contract functions are also presented. Section III presents a real-world use case of perpetual contract NFT using Uniswap v3 [25] to highlight trading strategies to maximize returns and validate profitability. Section IV summarizes our work and concludes the paper. The reference smart contract source code and demo web application for the perpetual contract NFT are provided in the Appendixes.

II. PERPETUAL CONTRACT NFT AND USE CASE SCENARIOS

In this section, we introduce perpetual contracts, which are essential in understanding the concepts behind perpetual contract NFTs. Afterward, we provide details of the perpetual contract NFT and outline a use case to demonstrate how perpetual contract NFTs can be used as collateral to maximize profit or hedge liquidation in futures trading.

A. PERPETUAL CONTRACTS

In a traditional derivatives market, traders agree to buy or sell some underlying asset, such as gold or oil, at a predetermined price and time in the future through a financial futures contract [26]. The transaction of the asset must take place at the set price at the contract expiration date. However, in the cryptocurrency derivatives market, most futures contracts do not have a predetermined expiration date. These types of contracts are called perpetual contracts [27]. Perpetual contracts allow traders to deposit cryptocurrencies like bitcoin, Ether, or stablecoins as collateral to derivatives exchanges. Using these collaterals, traders make financial futures contracts on select cryptocurrencies.

Traders can take a long or short position depending on their expectations over the value of the cryptocurrency. If a trader expects that the value of the cryptocurrency,

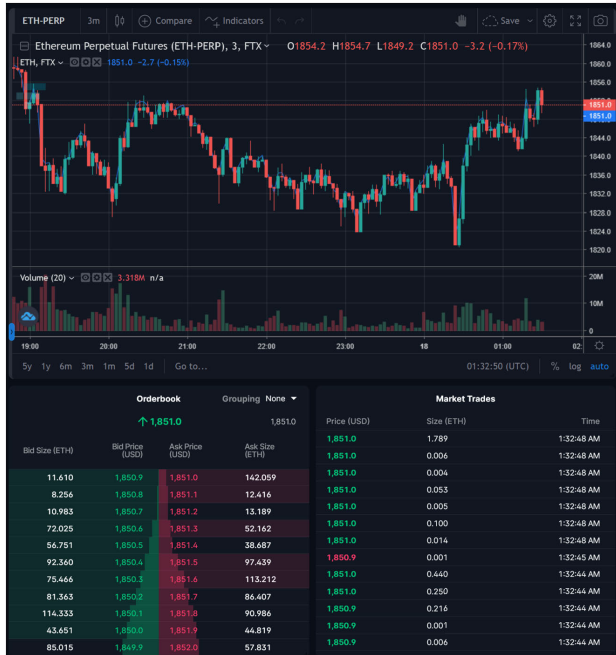


FIGURE 2. Perpetual contract trading in FTX exchange.

i.e., the underlying asset, will rise in the future, the trader goes long and buys perpetual contracts. A trader will go short and sell perpetual contracts if the value of the cryptocurrency is expected to fall in the future. Since there is no predetermined contract expiration date, traders can hold the position indefinitely, as long as they are able to pay for any funding rate associated fees. If the value of the cryptocurrency increases, the perpetual contract holder can claim profit at any time by closing the position. However, if the value of the cryptocurrency drops, and the margin balance drops below the minimum amount of collateral that one must hold to keep the trading position open, the collateral will be at risk of liquidation [28]. To avoid losing all collateral, traders must settle their perpetual contracts before liquidation.

Perpetual contracts of a cryptocurrencies exchange are transactions of futures and offer leverage opportunities. It allows a user to open a trading position that is bigger than the available trading capital. The leverage offered differs depending on the cryptocurrencies exchange. Fig. 2 shows perpetual contract trading in FTX, a popular cryptocurrency exchange. Users who want to trade perpetual contracts deposit their assets, such as Ether or stablecoins, to the FTX exchange. FTX sets a weight for each asset. For example, assume the price of each Ether is \$2000, and the weight of Ether is 0.95. Under this assumption, if a trader deposits one Ether as collateral, the trader can place a perpetual contract order worth \$1900. FTX presents with leveraging of up to 20 times. If a trader orders a perpetual contract with 20 times leveraging, the available capital will increase 20 times, and so will the profit. However, traders need to take precautions since leveraging magnifies not only potential profit but also potential loss. Performing high-leveraged trades can pose a

high risk of quick liquidation in a volatile market, resulting in total loss of capital.

B. PERPETUAL CONTRACT NFT

In a perpetual contract, the collateral backs the position with its value. The perpetual contract holder can settle the contract to claim their collateral with profit or loss whenever they want. Therefore, each perpetual contract denotes the right to claim the collateral. Since NFTs represent ownership rights to an asset, we can mint the perpetual contract as an NFT representing the rights to the collateral. For example, assume a trader makes a perpetual contract using 100 USD Coin (USDC), a stablecoin pegged to the U.S. dollar, as collateral for futures. A market that is capable of minting perpetual contracts as NFTs provides the corresponding perpetual contract NFT to the trader. After the perpetual contract NFT is issued, its holder can settle the contract to claim 100 USDC because holding the NFT represents the right to claim the collateral. Hence, a perpetual contract NFT can be regarded as possessing the value of the collateral. As a result, the perpetual contract NFT can be used as collateral in other DeFi services, such as lending, DEXs, or derivatives.

The smart contract for the perpetual contract NFT inherits the ERC-721 contract, providing seamless backward compatibility and extended functionality. **Algorithms 1 to 3** are pseudo codes of smart contract functions for perpetual contract NFT to collateralize, mint, and decrease collateral. **Algorithm 1** shows the procedure to set collateral. It takes a token id and asset amount as input arguments. The ERC-721 standard provides a unique identifier called token id to distinguish each token. Each token id has an associated collateral. If the input token id is already present in the blockchain network, its corresponding collateral is retrieved and updated.

Algorithm 1 Collateralize

Require: *tokenId*, *asset*

▷ *getCollateral(tokenId)* gives collateral of *tokenId*

- 1: **if** *tokenId* exists **then** ▷ Increase collateral
- 2: *collateral* ← *getCollateral(tokenId)*
- 3: (*collateral* + *asset*) ↦ *tokenId*
- 4: **else** ▷ Initialize collateral
- 5: *asset* ↦ *tokenId*
- 6: **end if**

The perpetual contract NFT minting procedure is outlined in **Algorithm 2**. The *_mint* function, which is inherited from the ERC-721 contract, is used to mint a new perpetual contract NFT. The *tokenId* is assigned to the newly created token. Once the token is minted, it is sent to the *address*, which is the address of the NFT owner. After minting, the algorithm uses the *asset* to initialize the collateral associated with the token. **Algorithm 3** presents the procedure to decrease or liquidate collateral. The algorithm verifies whether the given *address* has ownership of the NFT represented by the *tokenId*. If the *address* is the owner of the NFT, the collateral is decreased

by *asset* amount and returned to the *address*. As a proof-of-concept, we present the smart contract details for perpetual contract NFT as Solidity source code in the Appendixes section.

Algorithm 2 Mint Perpetual Contract NFT

Require: *asset*

```

1: tokenId  $\leftarrow$  getLastTokenId() + 1
2: _mint(address, tokenId)  $\triangleright$  Inherited from ERC-721
3: collateralize(tokenId, asset)
  
```

The smart contract for perpetual contract NFT inherits the ERC-721 contract, acquiring all of its attributes and properties. Therefore, it can be combined with other ERC-721 standard extensions, such as ERC-4907 [29]. ERC-4907 was approved by the Ethereum development team on June 2022, and it enables rentable NFTs by distinguishing between NFT ownership and usage rights. The address with NFT ownership is called the “owner”, and the address renting the NFT is called the “user”. The “user” is set to 0×00 when the NFT is not being rented. A rentable NFT delegates the right to use the NFT to the “user” for an agreed period of time.

To illustrate the interaction between the perpetual contract NFT smart contract and ERC-4907, let us assume a scenario where Alice has ownership of a perpetual contract NFT, and she wants to use the perpetual contract NFT as collateral for DeFi services to maximize returns. DeFi services can set the terms and conditions by adopting ERC-4907 and overriding the procedure described in **Algorithm 3**. Alice can delegate the right to liquidate assets backing the value of the perpetual contract to DeFi services for a specific period of time. In the next section, we present more details on how perpetual contract NFTs can be used to maximize the rate of return on capital.

Algorithm 3 Decrease or Liquidate Collateral

Require: *address*, *tokenId*, *asset*

```

1: if address is owner of tokenId then
2:   collateral  $\leftarrow$  getCollateral(tokenId)
3:   if asset < collateral then  $\triangleright$  Decrease collateral
4:     transfer(address, asset)
5:     (collateral - asset)  $\mapsto$  tokenId
6:   else  $\triangleright$  Liquidate collateral
7:     transfer(address, collateral)
8:     0  $\mapsto$  tokenId
9:   end if
10: else
11:   Give fail to decrease
12: end if
  
```

C. USE CASE

The valuation of ordinary NFTs, such as digital collections, game items, and lands in virtual worlds, depends on their previous selling price [21]. The assessment of the worth and value of an NFT may depend on its popularity or scarcity,

and there are no objective criteria for valuation. Hence, in general, these NFTs are not recommended to be used as collateral in DeFi services. However, the value of a perpetual contract NFT is backed by the collateral for the perpetual contract. Therefore, it is possible to exploit the valuation of perpetual contract NFT as collateral in various DeFi services. In addition, perpetual contract NFT follows standard interface ERC-721, enabling its use in peer-to-peer, peer-to-exchange, and exchange-to-exchange trading scenarios. Following a standard interface means that it can be integrated for DeFi composability. DeFi composability is the concatenation of DeFi services to maximize profit by giving liquidity to a locked asset. In the following subsections, we provide additional information on DeFi composability and illustrate practical examples of using perpetual contract NFTs in DeFi composability to maximize profit and avoid liquidation in the futures market.

1) DeFi COMPOSABILITY

Composability, which refers to the interoperability of components within a design system, is a key feature of DeFi. DeFi composability denotes the ability for DeFi application components and protocols to interact with one another resulting in a limitless combination of new forms of financial services. This is the reason why DeFi composability is also referred to as “DeFi Legos” or “Money Legos.” Specifically, DeFi composability provides a way to maximize profit by giving liquidity to a locked asset. Liquidity describes the degree to which an asset can be quickly converted into ready cash, maintaining its market price value.

Let us illustrate this with a practical example of DeFi services interaction. Beacon Chain [30] is a new consensus layer for Ethereum 2.0. A user willing to be a miner for Beacon Chain needs to stake 32 Ethers as a deposit. As an incentive for staking, miners can receive block generation rewards [31] from the Beacon Chain. However, they lose liquidity for the 32 Ethers deposited, since the Beacon Chain locks this deposit until Ethereum 2.0 is released. To deal with this loss of liquidity by staking, Lido Finance [32] released a DeFi staking delegation service. Under this service, users provide their Ether to Lido Finance, and Lido Finance stakes Ether to the Beacon Chain on behalf of the users. When users or delegators hand over their Ethers, Lido Finance provides the delegators an equivalent amount of staked Ether (stETH) derivative token which is pegged 1:1 with Ether. Since stETH and Ether have the same value, delegators can swap stETH for Ether and use it in other DeFi services. In addition, delegators can also get block generation rewards for staking to Beacon Chain from Lido Finance. Therefore, Lido Finance’s delegated staking provides liquidity for staked Ethers using stETH and block generation rewards in the Beacon Chain. The swap between stETH and Ether is provided by Curve Finance [14]. Curve Finance services a Liquidity Pool (LP) which helps the asset swap between stETH and Ether. Holders of stETH and Ether provide their assets to the pool and supply liquidity. As rewards for providing liquidity, providers receive LP tokens and swap

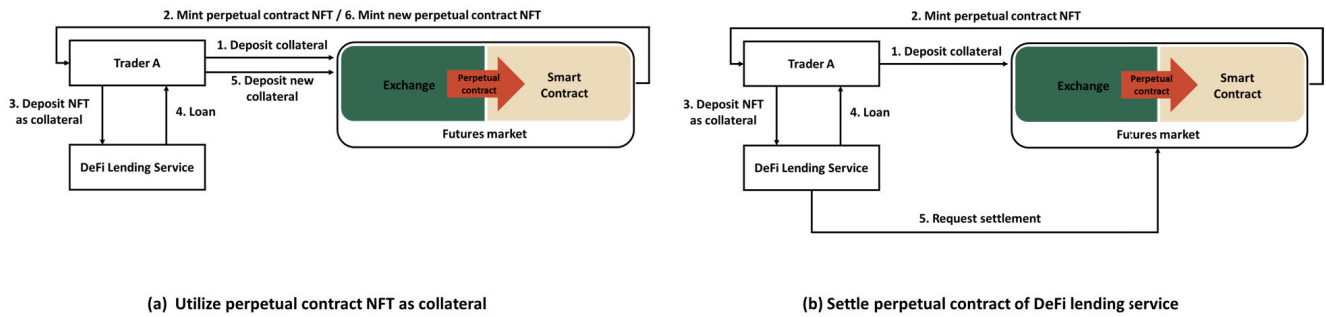


FIGURE 3. Use case of perpetual contract NFT. (a) denotes Trader A utilizing perpetual NFT as collateral to maximize profit or hedge liquidation. (b) denotes DeFi lending service requesting liquidation of the collateral by Trader A.

fees. In this example, concatenated DeFi services provided by Lido Finance and Curve Finance generated cumulative rewards, i.e., Beacon Chain rewards, swap fees, and LP tokens, for Ether holders. Furthermore, the LP tokens can be used in other DeFi services to generate additional profit. The generation of cumulative rewards through concatenated DeFi services is the essence of DeFi composability.

2) PERPETUAL CONTRACT NFT AS COLLATERAL FOR FUTURES

The value or worth of a perpetual contract NFT can be assessed objectively since it is backed by collateral. Therefore, perpetual contract NFTs can be collateralized for DeFi composability just like ERC-20 tokens. The position NFT, which was introduced in Uniswap v3 [25], can be considered a type of perpetual contract NFT. Uniswap is a DEX operating on the Ethereum blockchain. In Uniswap v3, liquidity providers can concentrate their assets and deploy liquidity to a specific price range of their choice. The liquidity range or position taken by the liquidity provider is represented as a position NFT. The position NFT is perpetual and has no predetermined expiration date of liquidity, allowing providers to modify their liquidity positions at any time. If liquidity providers exploit the position NFT as collateral for DeFi composability, additional profit can be generated without rebalancing liquidity. We discuss Uniswap v3 and its position NFT at greater length in Section III.

Traders can use perpetual contracts NFTs as collateral to maximize profit in the cryptocurrency futures market. Fig. 3 shows two cases where perpetual contract NFT is provided as collateral for loans. Fig. 3 (a) presents the borrowing process to maximize profit. Trader A opens a position in a perpetual contract for some underlying asset by providing collateral. The futures market generates a perpetual contract NFT, representing the rights of the perpetual contract and collateral. To maximize profit, Trader A delegates these rights to the DeFi lending service by providing perpetual contract NFT as collateral for a loan. Using this loan, Trader A makes a new perpetual contract. When Trader A wants to close all perpetual contracts to claim profits and collateral, Trader A settles the second perpetual contract NFT and repays for the loan to the DeFi lending service to take back the first perpetual contract NFT. Next, Trader A settles the first

perpetual contract NFT to take back the initial collateral and its profits. Fig. 3 (b) shows the process of loan recovery by the DeFi lending service. The delegation of a perpetual contract NFT by Trader A to the DeFi lending service denotes that the DeFi lending service has precedence over the futures market when it comes to the liquidation of the collateral. In the event of a decrease in the value of the underlying asset below a set margin threshold, the DeFi lending service can liquidate Trader A's collateral before the futures market to recover the loan given to Trader A.

In addition, by using a perpetual contract NFT as collateral, traders can avoid liquidation without the need to rebalance their position. In the derivatives market, futures prices do not move linearly. Even if a trader makes an accurate long-term price prediction, liquidation may occur due to short-term price movements. To avoid liquidation, traders reserve part of their assets to respond to price movements or to rebalance their contracts. However, perpetual contract NFTs enable traders to exploit their assets to the fullest for trading futures. For example, assume a trader opens a long position on a perpetual contract using all available assets as collateral. If short-term price fluctuations decrease the value of the underlying asset, the trader is at risk of losing all collateral due to liquidation. However, if a perpetual contract NFT is created, the trader can exploit it as collateral to borrow assets from other lending services to make a new position. With this new position, the trader can increase the profit margin to avoid liquidation without having to rebalance the position.

III. PROFITABILITY VALIDATION

In this section, we validate the profitability of perpetual contract NFT as collateral by using Uniswap v3. We introduce the fundamental protocols used in Uniswap v3, namely Automated Market Maker (AMM) [15] and concentrated liquidity. We then use the position NFTs associated with concentrated liquidity to prove the profitability of perpetual contract NFTs when used as collateral.

A. BACKGROUND

AMM is a popular liquidity pool algorithm proposed initially in Uniswap and forked by many other projects in the DeFi space, such as Curve Finance [14]. Concentrated liquidity is a mechanism aggregated to AMM in Uniswap v3 to maximize

capital efficiency. In this subsection, we introduce the key concepts behind AMM and concentrated liquidity.

1) AUTOMATED MARKET MAKER

In traditional centralized exchanges, the order book system is used for trading, where intermediaries or market makers match prices through “bid and ask” to help trade assets and provide liquidity in the market. AMM was introduced in decentralized exchanges to automate this process and play the role of market makers. AMM is a decentralized automated asset swapping algorithm that enables trading. AMM uses Liquidity Pools (LPs), which are pools or groups of assets (e.g., tokens) that are locked in smart contracts. Liquidity providers present assets as liquidity to the LP to receive swap fees as rewards. An LP in Uniswap consists of asset pairs, and their swap ratio is adjusted by AMM for automated algorithmic swaps. For example, if an LP is created for Uniswap governance token (UNI) and Wrapped Ether (WETH) pair, the AMM helps traders swap UNI with WETH and vice versa by adjusting their swap ratio. In Uniswap, the AMM uses the Constant Product Formula (CPF) to determine the pricing of the asset pair. CPF is expressed as:

$$x \cdot y = k$$

where x and y denote the respective reserves of two assets X and Y , and k is their product constant [15]. Fig. 4 (a) shows the example of AMM price curve of the WETH-UNI pair.

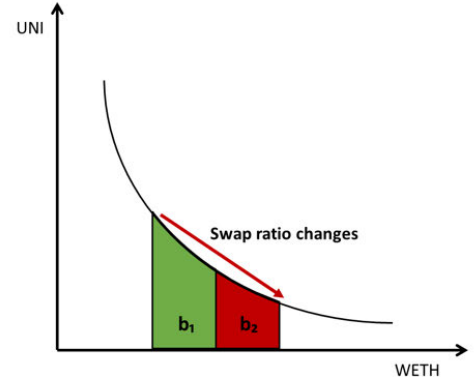
2) CONCENTRATED LIQUIDITY OF UNISWAP v3

In Uniswap v1 and v2, liquidity is allocated evenly across the entire price curve. The uniform distribution of liquidity allows asset trading across the entire price range. In Uniswap, swap fees earned depend on the number of liquidity shares allocated to a price interval. If liquidity is distributed evenly, the liquidity provider may miss out on swap fee rewards.

For example, assume all asset swaps occur at price band b_1 in Fig. 4 and that a liquidity provider has enough assets to own all the shares of one price band. If liquidity is distributed evenly to five price bands as shown in Fig. 4 (b), liquidity is also being allocated to price bands where swaps never occur. Hence, price bands other than band b_1 do not generate any swap fee rewards. On the other hand, if the liquidity provider can concentrate all assets to band b_1 and owns 100% of the liquidity shares as illustrated in Fig. 4 (c), the liquidity provider can collect all the swap fees generated from band b_1 resulting in maximum earnings.

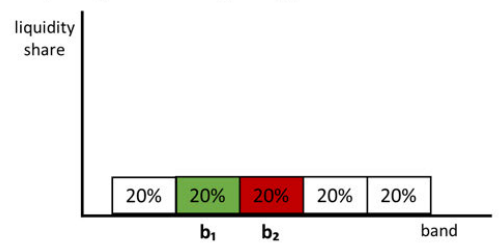
Uniswap v3 provides concentrated liquidity, which allows liquidity providers to select custom price bands of their choice and concentrate their assets only on select price bands to maximize swap fee earnings. Liquidity providers can concentrate capital in price bands where most swap activities occur to maximize returns. However, if the price band where swap activity occurs changes due to swap ratio changes, the liquidity provider stops earning fees. The swap ratio changes when swaps happen more frequently for one asset than the other. Fig. 4 (a) illustrates swap ratio changes in AMM

Automated market maker curve of WETH-UNI



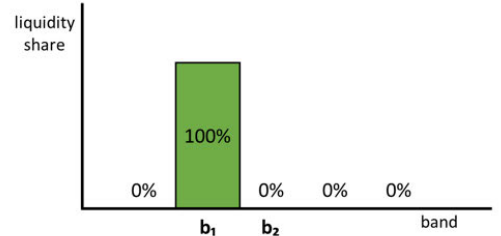
(a) Swap bands in WETH-UNI AMM curve

Liquidity shares of liquidity provider in AMM bands



(b) Liquidity distribution without concentration

Liquidity shares of liquidity provider in AMM bands



(c) Liquidity distribution with concentration

FIGURE 4. Uniswap AMM curve of WETH-UNI pair and liquidity distributions.

resulting in the shift of the active trading price band from b_1 to b_2 . If liquidity is uniformly distributed, as in Fig. 4 (b), liquidity providers can earn swap fee rewards regardless of swap ratio changes. However, if liquidity is concentrated only in b_1 , as in Fig. 4 (c), liquidity providers cannot receive any swap fees since no liquidity shares are allocated in b_2 .

With concentrated liquidity, Uniswap v3 presents LP with three fee tiers $f := \{0.05\%, 0.30\%, 1\%\}$. Additional fee tiers can be enabled by Uniswap governance [25]. We can estimate the swap fee rewards for liquidity provider j , who concentrates liquidity at price band i with fee tier x , as follows:

$$R_i^j := \frac{L_i^j}{\sum L_i} \cdot A_i^x \cdot f_x \quad (1)$$

where L_i^j denotes liquidity of the liquidity provider j in band i , A_i^x denotes accumulated trade volume of band i for 24 hours in

LP with fee tier f_x , and $f_x \in f \cdot L_i^j / \sum L_i$ denotes the liquidity shares of j in band i , and $A_i^x \cdot f_x$ means unclaimed fees in LP of fee tier f_x for 24 hours. In addition, for $n, m \in \mathbb{N}$, LP of n bands is $LP := \{L_1, L_2, \dots, L_n\}$ and liquidity band i consisting of m providers is $L_i := \{L_i^1, L_i^2, \dots, L_i^m\}$. We use the above definitions for profitability validation.

B. PROFITABILITY OF NFT AS COLLATERAL

The concentrated liquidity of Uniswap v3 allows liquidity providers to manually allocate liquidity positions to maximize profit. However, manual allocation raises concerns about rebalancing the position. When the swap ratio is changed, liquidity providers need to rebalance liquidity distribution to avoid losses. As a result, concerned liquidity providers opt for Uniswap v2 over Uniswap v3. They believe that without optimal position rebalancing, Uniswap v3 gives fewer returns than Uniswap v2 [35]. In this subsection, we propose a method that will motivate liquidity providers to migrate to Uniswap v3. By using Uniswap position NFT as collateral, liquidity position can be used as futures contracts in the derivatives market.

1) UNISWAP POSITION NFT AS COLLATERAL

Fig. 5 shows the position NFT of the Uniswap v3 of WETH-UNI with a 0.3% fee tier LP. When liquidity providers allocate assets to a specific band of LP as a liquidity position, liquidity providers get a position NFT representing their rights to the liquidity position in an AMM. We can consider the position NFT as a perpetual contract NFT because the position of Uniswap has similar characteristics to the perpetual contract used in the derivatives market. Namely, liquidity providers can hold their position indefinitely and close their position to claim assets whenever they want. Furthermore, reserved assets of liquidity providers back the value of the position, like collaterals backing the value of perpetual contracts. Therefore, position NFT holders get rights to the reserved assets and can claim the assets in that position. In addition, NFT is a standard interface, so it is possible to trade the position NFT in an NFT market. The buyer of a position NFT reserves the right to claim the assets bound to the position.

Position NFT holders can make use of DeFi composability to exploit position NFT as collateral to reduce concerns associated with position rebalancing. The value of the position NFT can be assessed by the assets in its position. Based on this valuation, the position NFT holder can borrow additional assets from other DeFi lending services with the position NFT as collateral. Using these assets, holders can make a new position in Uniswap v3 instead of rebalancing their position, or they can use the assets in other DeFi services to maximize returns.

2) PROFITABILITY VALIDATION

In Uniswap AMM, when trade volumes move from bands i to $i+1$, liquidity providers must decide whether to rebalance the position or not. Rebalancing raises concerns about losing profit. To mitigate the concerns, we propose the use of

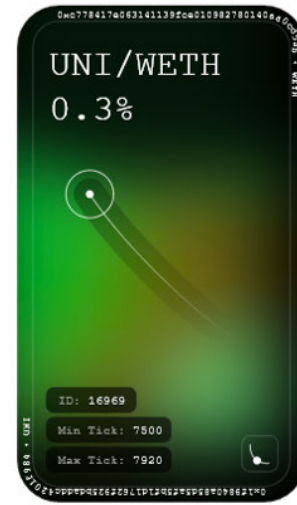


FIGURE 5. Liquidity position NFT of WETH-UNI pool in Uniswap v3.

position NFT as collateral for DeFi composability, e.g., a lending service, to borrow additional assets. To estimate the profitability from borrowed, assume that there is no additional liquidity except for the loan borrowed by provider j , and fee tier f_x is fixed until the provider removes all liquidity. Under these assumptions, we can estimate the borrowing profitability as follows:

$$\frac{L_i^j}{\sum L_i} \cdot A_i^x \cdot f_x \leq \frac{L_i^j}{\sum L_i} \cdot A_i^x \cdot f_x + \frac{L_{i+1}^j}{\sum L_{i+1}} \cdot A_{i+1}^x \cdot f_x - L_{i+1}^j \cdot I \quad (2)$$

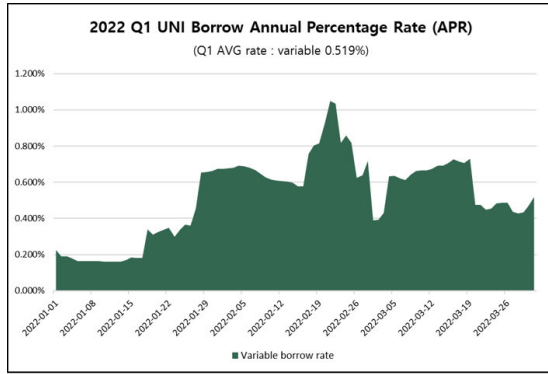
and

$$L_{i+1}^j = LTV \cdot L_i^j \quad (3)$$

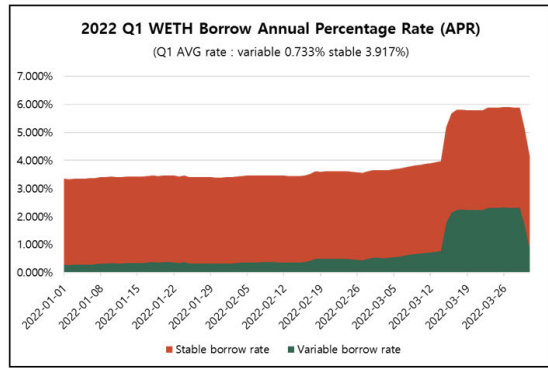
I denotes the daily loan interest rate of lending services, and LTV denotes loan-to-value. For example, when the price of a position NFT is \$100 and the LTV rate is 80%, the position NFT holder can borrow up to \$80 with position NFT as collateral. By rearranging (2), we obtain:

$$I \leq \frac{A_{i+1}^x}{\sum L_{i+1}} \cdot f_x \quad (4)$$

Inequation (4) shows that profits depend on A_{i+1}^x . Let us assume that a liquidity provider j borrows the same amount of UNI and WETH from AAVE [4] with a variable borrow rate for short-term from an LP with fee tier $f_x = 0.30\%$. Fig. 6 presents the borrow rate of two assets for 2022 Q1 in AAVE. AAVE provides loan services based on collateral. For UNI, only variable borrow rate is available for loans. For WETH, variable and stable rates are available for loans. The average of each asset's variable borrow Annual Percentage Rate (APR) for Q1 is 0.519% and 0.733%. Therefore, the variable borrow APR of Q1 for the total amount of assets is 0.626% when a liquidity provider borrows the same volume of each asset. Since Uniswap gives daily fees to liquidity



(a) Borrow APR of UNI



(b) Borrow APR of WETH

FIGURE 6. Borrow Annual Percentage Rate (APR) of UNI and WETH in AAVE for 2022 Q1. Data was extracted from a subgraph of AAVE: <https://github.com/aave/protocol-subgraphs>.

providers [35], we can estimate profitability as follows:

$$\frac{0.626}{365} \leq \frac{A_{i+1}^x}{\sum L_{i+1}} \cdot 0.30 \quad (5)$$

Therefore, from the futures contract perspective, if provider j expects trade volume in band $i + 1$ to be more than 0.57% of liquidity in band $i + 1$, provider j will use the position NFT as collateral for a loan to generate more profit. Since using the position NFT as collateral for a loan gives more returns than non-rebalancing, it motivates liquidity providers who are still staying in Uniswap v2 to migrate to Uniswap v3.

As of August 2022, the top five Uniswap v3 pools with the most Total Value Locked (TVL), which denote the total value of tokens locked in a pool as liquidity, consist of Wrapped Ether (WETH), Wrapped bitcoin (WBTC), Dai (DAI), and USD Coin (USDC) token pairs. The APR of each token is given in Table 1. WETH and WBTC are Ether and bitcoin wrapped in ERC-20 tokens, so the prices of these tokens are pegged to their respective coin value. DAI and USDC are stablecoins pegged to the U.S. dollar, so their value follows the U.S. dollar closely. It is interesting to see that stablecoins have higher borrow APRs than other tokens. The risk of depegging or bank run [36] yields higher borrow APR. Table 2 shows the profitable threshold values obtained using (4) for the five Uniswap

TABLE 1. UNI is a governance token of Uniswap. As of August 2022, the five pools with the most TVL consist of WETH, WBTC, DAI, and USDC pairs.

2022 Q1 Variable Borrow APR		
Token Name	APR	Pegged to
UNI	0.519%	-
WETH	0.733%	Ether
WBTC	0.293%	Bitcoin
DAI	3.473%	U.S. Dollar
USDC	3.208%	U.S. Dollar

pools with the most TVL. We consider the profitability of three types of pools with different token pairings: stablecoins, stablecoin/wrapped token, and wrapped tokens pair pools. The profitability of each pool depends on the fee tier. The DAI/USDC stablecoins pair pool shows a profitable threshold of 91.52% when the fee tier is 0.01%. This implies that it is not profitable to borrow additional assets until the trade volume for the liquidity of a band reaches 91.52%. However, with a 0.05% fee tier, profitability is achieved when trade volume for the liquidity band reaches 18.30%. The stablecoin/wrapped token pair pool (i.e., USDC/ETH) and wrapped tokens pair pool (i.e., WBTC/ETH) show profitability with a lower profitable threshold compared with the stablecoin pair pool. Furthermore, when fee tiers are the same, the profitable threshold of the wrapped tokens pair pool is lower than the stablecoin/wrapped token pair pool because of the low average APR.

C. DISCUSSION

In this section, we used the position NFT to provide an example of perpetual contract NFT and prove its profitability. Unlike the traditional financial market, most of the futures contracts in the cryptocurrency market do not have settlement deadlines. Such contracts with no expiration dates are called perpetual contracts. We provided a novel perspective for utilizing perpetual contracts through perpetual contract NFT. Furthermore, we validated the profitability of perpetual contract NFTs using Uniswap v3 position NFTs, which will motivate liquidity provision in Uniswap v3. In our profitability validation process, only short-term loans were considered. For long-term loans, additional considerations are necessary to maximize profit. First, it needs to be clear that many factors can influence the borrow Annual Percentage Rate (APR), such as macroeconomics and serial liquidation in DeFi. Second, impermanent loss may occur during long-term borrowing due to cryptocurrency price fluctuations. Third, in the profitability validation, we assumed the liquidity of a band to be constant because the additional liquidity of one provider has little impact on the overall liquidity of the band. However, as more providers add or reduce liquidity, the liquidity can vary and fluctuate. Thus, in the long term, a game theoretic analysis [37] is required for profitability validation.

In addition, we estimated the profitability for three types of pools, among which the profitable threshold of stablecoins pair pool was notable. The stablecoins pair pool gives less impermanent loss compared with other token pools because

TABLE 2. Five Uniswap pools with the most TVL as of August 2022.

Pool Name	Fee tier	Average APR	Profitable threshold
DAI/USDC	0.01%	3.34%	91.52%
DAI/USDC	0.05%	3.34%	18.30%
USDC/ETH	0.05%	1.97%	10.82%
USDC/ETH	0.30%	1.97%	1.80%
WBTC/ETH	0.30%	0.51%	0.47%

pegged prices and swap ratio of the paired stablecoins do not change steeply [14]. For example, the swap ratio of stablecoins pegged to the U.S. dollar is close to one-to-one. Therefore, liquidity providers of stablecoin pools need to consider target traders to exploit perpetual contract NFT. A pool with a fee tier of 0.05% is appropriate for liquidity providers who want to get trading fees from arbitrage traders. The value of stablecoins pegged to fiat currencies follows the value of fiat currency closely, but they are not exactly equivalent. Thus, there are arbitrage traders who exploit the price difference of stablecoins. For this pool, liquidity providers need rebalancing as the stablecoin prices change, and perpetual contract NFT can help to rebalance without removing liquidity. The 0.01% fee tier pool, which needs a 91.52% profit threshold, is appropriate for liquidity providers who do not want frequent rebalancing. Specifically, this pool helps traders who want to swap stablecoins for other stablecoins to be used in other DeFi services. However, this pool gives slippage to arbitrage traders because most of the liquidities are concentrated in a band with near one-to-one swap ratio. For this pool, a loan obtained by using a perpetual contract NFT can maximize profit by accumulating assets in a specific band, such as a band with a one-to-one swap ratio.

Decentralized exchange, such as Uniswap, is one of the many service categories offered in DeFi. Perpetual contract NFTs can be easily applied to other DeFi categories, such as lending and derivatives, making it a valuable resource for enabling DeFi composability. In addition, the adoption of blockchain by the emerging metaverse [38] platforms will provide new service scenarios for the proposed perpetual contract NFTs, enabling them to be used with the metaverse NFT assets to build and sustain the new virtual economy.

IV. CONCLUSION

In this work, we introduced a new perspective on utilizing perpetual contracts as NFT for DeFi composability. Specifically, we proposed minting a perpetual contract as NFT to exploit it in lending services as collateral. Using this lending service, perpetual contract NFT holders can maximize their profit or avoid liquidations in the futures market. A real-world use case was provided through concentrated liquidity and position NFTs used in Uniswap v3. The position NFT, which is a form of perpetual contract NFT, was used for DeFi composability to maximize the liquidity provider's profit without rebalancing by considering the liquidity position in the futures contract.

APPENDIXES

To demonstrate the feasibility of the proposed perpetual contract NFT, we provide a proof-of-concept

Perpetual contract NFT Demo

FIGURE 7. Perpetual contract NFT demonstration page.

smart contract implemented in Solidity. A simple web application is also provided to demonstrate the smart contract execution. All sources provided are released on GitHub: <https://github.com/HyoungsungKim/Perpetual-Contract-NFT-PoC>.

A. SMART CONTRACT

The Solidity interface of the perpetual contract NFT is shown in Source code 1. The interface for the perpetual contract NFT consists of one event and five function definitions. The `tokenId` parameter points to an NFT id. The `amount` denotes a given amount of asset, such as Ether, or token. When the collateral of an NFT is changed, the `UpdateCollateral` event logs the updated collateral to the Ethereum network. The `_collateralize` function maps collateral to an NFT. This function can combine with other token interfaces like ERC-721 as an extension for backward compatibility. The `increaseCollateral` function increases the amount of collateral backing the value of an NFT. The `decreaseCollateral` function reduces the amount of collateral and gives it back to the NFT owner. The `liquidation` function reduces all collateral of an NFT. The `getBalanceById` function shows the collateral balance of an NFT.

The implementation of the interface is shown in Source code 2. The perpetual contract NFT implementation inherits ERC-721 contracts for backward compatibility. Inheritance allows the derived contract to use ERC-721 functions, such as `_mint`. The mapping data type enables the storage of data as key-value pairs. The NFT id and collateral are used as the key and its corresponding value, respectively. Therefore, it is possible to reference the amount of collateral using the id of an NFT. The `emit` keyword is used to trigger the `increaseCollateral` event to log or store an NFT id and its collateral to the Ethereum network, allowing interaction with other contracts. The `mint` function uses the inherited `_mint` function to generate an ERC-721 token. After minting, the `_collateralize` function maps the

```
// SPDX-License-Identifier: CC0-1.0

interface IPerpetualContractNFT {
    /// Logged when the collateral of an NFT is changed
    /// @notice Emitted when the amount of collateral is updated
    event UpdateCollateral(uint256 indexed tokenId, uint256 collateral);

    /// @notice Map the amount of an asset to NFT
    /// Throws if NFT does not exist
    /// @param tokenId An Id of existing NFT
    /// @param amount The amount of an asset, such as Ether, ERC-20 tokens
    function _collateralize(uint256 tokenId, uint256 amount) external;

    /// @notice Increase the amount of an collateral that backing an NFT
    /// Call _collateralize function inside
    /// @param tokenId An Id of existing NFT
    function increaseCollateral(uint256 tokenId) external payable;

    /// @notice Reduce the part of the collateral backing value of NFT and send it to an NFT owner
    /// Throws a message sender does not own NFT
    /// Throws a message sender reduces more collateral than existing
    /// @param tokenId An Id of existing NFT
    function decreaseCollateral(uint256 tokenId) external payable;

    /// @notice Reduce all collateral that backing NFT, and send it to an NFT owner
    /// Throws a message sender does not own NFT
    /// @param tokenId An Id of existing NFT
    function liquidation(uint256 tokenId) external payable;

    /// @notice Gives balance of the collateral backing value of NFT
    /// Throws if NFT does not exist
    /// @param tokenId An Id of existing NFT
    /// @return The balance of the collateral backing value of NFT
    function getBalanceById(uint256 tokenId) external view returns (uint256);
}
```

Source code 1. Solidity interface of perpetual contract NFT.

given asset to a token id denoting an NFT and emits the event to write collateral changes to the Ethereum network.

The `increaseCollateral` function increases the amount of collateral. The `decreaseCollateral` function checks if the requester is the owner of the token. The `ownerOf` function, inherited from the ERC-721 contract, provides the address of the NFT owner. If the addresses match, the contract gives the collateral to the requester and emits the `UpdateCollateral` event to log the changes. The `liquidation` function gives back all collateral to the requester after verifying that the requester is the owner. The `getBalanceById` function shows the balance of the collateral corresponding to the token Id. One notable thing about `increaseCollateral` and `decreaseCollateral` functions is their execution requirement. Increasing collateral does not require matching the owner and the requester, but decreasing collateral requires matching. Increasing the amount of collateral is allowed even without NFT ownership verification.

B. SMART CONTRACT DEMONSTRATION

This section describes the web application for perpetual contract NFT demonstration. The demonstration source code and the Docker image for application execution are provided via GitHub. We assume that the user has a basic understanding of Web3 and is comfortable with using blockchain for web services. The smart contract demonstration requires the

installation of Metamask Ethereum wallet as a prerequisite. The contract is deployed in the Goerli Ethereum testing network. Goerli testnet ETH is required to execute the smart contract. Fig. 7 shows the demonstration page executed which contains eight functions. In addition to the six functions presented in Source code 2, a function to connect to Metamask and a function to transfer NFT have been included. To start the demonstration, the user's Metamask wallet needs to be connected to the web application using the "CONNECT TO METAMASK" button.

After connection, a perpetual contract NFT can be minted by collateralizing the user's Goerli ETH using the "MINT AND COLLATERALIZE" button. The button click executes the `mint` function of Source code 2 and creates a transaction. If the transaction is confirmed by the network, the contract transfers a perpetual contract NFT representing the right for 0.00001 Goerli ETH to the connected wallet. The contract references the collateral using an NFT id. Therefore, a token id needs to be provided as input to obtain the balance of the collateral. The initially disabled "LIQUIDATION" button is enabled after obtaining the balance. If there is enough balance backing the value of NFT, the user can execute the `liquidation` function to collect all collateral. The "GET BALANCE" button executes the `getBalanceById` function. This function uses the `view` keyword, which means that it will not change the state of the contract. Therefore, no transactions are made. In addition,

```
// SPDX-License-Identifier: CC0-1.0
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/utils/Counters.sol";
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "../IperpetualContract.sol";

contract PerpetualContractNFT is ERC721URIStorage, IPerpetualContractNFT {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIds;
    mapping (uint256 => uint256) public collateralOfNFT;

    constructor(string memory name_, string memory symbol_) ERC721(name_, symbol_) { }

    function mint(address _to) external payable {
        _tokenIds.increment();
        _mint(_to, _tokenIds.current());
        _collateralize(_tokenIds.current(), msg.value);
    }

    function _collateralize(uint256 tokenId, uint256 amount) public override {
        require(tokenId <= _tokenIds.current(), "This token does not exist");

        collateralOfNFT[tokenId] += amount;
        emit UpdateCollateral(tokenId, collateralOfNFT[tokenId]);
    }

    function increaseCollateral(uint256 tokenId) external payable override {
        _collateralize(tokenId, msg.value);
    }

    function decreaseCollateral(uint256 tokenId) external payable override {
        require(msg.sender == ownerOf(tokenId), "Not own this token");
        require(msg.value <= collateralOfNFT[tokenId], "Not enough collateral");

        payable(msg.sender).transfer(msg.value);
        collateralOfNFT[tokenId] -= msg.value;
        emit UpdateCollateral(tokenId, collateralOfNFT[tokenId]);
    }

    function liquidation(uint256 tokenId) external payable override {
        require(msg.sender == ownerOf(tokenId), "Not own this token");

        payable(msg.sender).transfer(collateralOfNFT[tokenId]);
        collateralOfNFT[tokenId] = 0;
        emit UpdateCollateral(tokenId, collateralOfNFT[tokenId]);
    }

    function getBalanceById(uint256 tokenId) public view override returns (uint256) {
        require(tokenId <= _tokenIds.current(), "This token does not exist");
        return collateralOfNFT[tokenId];
    }
}
```

Source code 2. Solidity code of perpetual contract NFT.

the user can increase or decrease the amount of collateral by referring to the token id. The “INCREASE COLLATERAL” button executes the `increaseCollateral` function and generates a transaction. After transaction confirmation, the contract adds 0.00001 Goerli ETH to the collateral. The “DECREASE COLLATERAL” button executes the `decreaseCollateral` function to create a transaction that results in collateral decrease by 0.00001 Goerli ETH. This button is initially disabled but gets activated when the balance becomes positive. The “TRANSFER” button executes the inherited transfer function from the ERC-721 contract. Transferring the NFT to another account denotes that the user loses the right to collect the collateral of the NFT.

REFERENCES

- [1] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Jun. 20, 2022. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [2] S. Ozcan and S. Unalan, “Blockchain as a general-purpose technology: Patentometric evidence of science, technologies, and actors,” *IEEE Trans. Eng. Manag.*, vol. 69, no. 3, pp. 792–809, Jun. 2022.
- [3] M. Du, Q. Chen, J. Chen, and X. Ma, “An optimized consortium blockchain for medical information sharing,” *IEEE Trans. Eng. Manag.*, vol. 68, no. 6, pp. 1677–1689, Dec. 2021.
- [4] M. A. Ferrag and L. Maglaras, “DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids,” *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1285–1297, Nov. 2020.
- [5] M. Kuperberg, “Blockchain-based identity management: A survey from the enterprise and ecosystem perspective,” *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1008–1027, Nov. 2020.

- [6] H. M. Kim, H. Turesson, M. Laskowski, and A. F. Bahreini, "Permissionless and permissioned, technology-focused and business needs-driven: Understanding the hybrid opportunity in blockchain through a case study of insolar," *IEEE Trans. Eng. Manag.*, vol. 69, no. 3, pp. 776–791, Jun. 2022.
- [7] S. Farshidi, S. Jansen, S. España, and J. Verkleij, "Decision support for blockchain platform selection: Three industry case studies," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1109–1128, Nov. 2020.
- [8] M. Du, Q. Chen, J. Xiao, H. Yang, and X. Ma, "Supply chain finance innovation using blockchain," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1045–1058, Nov. 2020.
- [9] F. Vogelsteller and V. Buterin. (Nov. 19, 2015). *EIP-20: Token Standard*. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-20>
- [10] W. Entri, D. Shirley, J. Evans, and N. Sachs. (Jan. 24, 2018). *EIP-721: Non-Fungible Token Standard*. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-721>
- [11] V. Buterin. *A Next-Generation Smart Contract and Decentralized Application Platform*. Accessed: Jun. 20, 2022. [Online]. Available: <https://ethereum.org/en/whitepaper/>
- [12] M. Dao. *The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System*. Accessed: Jun. 20, 2022. [Online]. Available: <https://makerdao.com/en/whitepaper/>
- [13] AAVE. *AAVE Protocol Whitepaper V1.0*. Accessed: Jun. 17, 2022. [Online]. Available: https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf
- [14] Curve.fi. *Curve Documentation Release 1.0.0*. Accessed: Jun. 23, 2022. [Online]. Available: https://curve.readthedocs.io/_/downloads/en/latest/pdf/
- [15] H. Adams. *Uniswap V2 Core*. Accessed: Jun. 23, 2022. [Online]. Available: <https://uniswap.org/whitepaper.pdf>
- [16] W. Radomski, A. Cooke, P. Castonguay, J. Therien, E. Binet, and R. Sandford, "EIP-1155: Multi token standard," Ethereum Improvement Proposals, Bern, Switzerland, Tech. Rep. 1155, Jun. 2018, Accessed: Oct. 26, 2022. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-1155>
- [17] D. Weijers and H. J. Turton, "Environmentally smart contracts for artists using non-fungible tokens," in *Proc. IEEE Int. Symp. Technol. Soc. (ISTAS)*, Oct. 2021, pp. 1–4.
- [18] M. Duguleana and F. Girbacia, "Augmented reality meets non-fungible tokens: Insights towards preserving property rights," in *Proc. IEEE Int. Symp. Mixed Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2021, pp. 359–361.
- [19] P. Boonpam, P. Bumrungsook, K. Sookhnaphibarn, and W. Choensawat, "Social data analysis on play-to-earn non-fungible tokens (NFT) games," in *Proc. IEEE 4th Global Conf. Life Sci. Technol. (LifeTech)*, Mar. 2022, pp. 263–264.
- [20] RADAR. *WETH | ERC20 Tradable Version of ETH*. Accessed: Jun. 20, 2022. [Online]. Available: <https://WETH.io/>
- [21] Nexo. *NFT Lending*. Accessed: Aug. 17, 2022. [Online]. Available: <https://nexo.io/nft-lending>
- [22] A. Zen. *CryptoKitties*. Accessed: Jun. 20, 2022. [Online]. Available: <https://www.cryptokitties.co/>
- [23] BAYC. *Bored Ape Yacht Club*. Accessed: Jun. 20, 2022. [Online]. Available: <https://boredapeyachtclub.com>
- [24] Larva Labs. *CryptoPunks*. Accessed: Aug. 20, 2022. [Online]. Available: <https://www.larvalabs.com/cryptopunks>
- [25] H. Adams. *Uniswap V3 Core*. Accessed: Jun. 23, 2022. [Online]. Available: <https://uniswap.org/whitepaper-v3.pdf>
- [26] J. C. Hull, *Options, Futures, and Other Derivatives*, 9th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2014.
- [27] FTX Crypto Derivatives Exchange. *What Are Futures?*. Accessed: Aug. 17, 2022. [Online]. Available: <https://help.ftx.com/hc/en-us/articles/360024780791-What-Are-Futures>
- [28] FTX Crypto Derivatives Exchange. *Liquidations*. Accessed: Aug. 17, 2022. [Online]. Available: <https://help.ftx.com/hc/en-us/articles/360027668712-Liquidations>
- [29] L. Anders and Shrug. *Rental NFT, an Extension of EIP-721*. Accessed: Oct. 26, 2022. [Online]. Available: <https://github.com/ethereum/EIPS/blob/master/EIPS/eip-4907.md>
- [30] V. Buterin, D. Hernandez, T. Kamphefner, K. Pham, Z. Qiao, D. Ryan, J. Sin, Y. Wang, and Y. X Zhang, "Combining GHOST and casper," 2020, *arXiv:2003.03052*.
- [31] V. Buterin, D. Reijersbergen, S. Leonardos, and G. Piliouras, "Incentives in ethereum's hybrid casper protocol," 2019, *arXiv:1903.04205*.
- [32] Lido. *Lido: Ethereum Liquid Staking*. Accessed: Jun. 20, 2022. [Online]. Available: <https://lido.fi/static/Lido:Ethereum-Liquid-Staking.pdf>
- [33] Immutable X. *Immutable X Whitepaper*. Accessed: Jun. 20, 2022. [Online]. Available: https://support.immutable.com/hc/en-us/article_attachments/4408240189583/Immutable_X_Whitepaper.pdf
- [34] The Sandbox Team. *The SANDBOX Play Create Own Government Earn*. Accessed: Jun. 20, 2022. [Online]. Available: https://installers.sandbox.game/The_Sandbox_Whitepaper_2020.pdf
- [35] A. Adams and G. Liao. *When Uniswap V3 Returns More Fees For Passive LPs*. Accessed: Jun. 23, 2022. [Online]. Available: <https://uniswap.org/SuperiorReturnsForLiquidityProviders.pdf>
- [36] M. Shen, "A \$60 billion crypto collapse reveals a new kind of bank run," Bloomberg Businessweek Finance, New York, NY, USA, Tech. Rep., May 2022. Accessed: Aug. 17, 2022. [Online]. Available: <https://www.bloomberg.com/news/articles/2022-05-19/luna-terra-collapse-reveal-crypto-price-volatility?leadSource=uverify%20wall>
- [37] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton, NJ, USA: Princeton Univ. Press, 1944.
- [38] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Commun. Surveys Tuts.*, early access, Sep. 7, 2022, doi: 10.1109/COMST.2022.3202047.



HYOUNGSUNG KIM received the B.S. degree in electronics and computer engineering from Chonnam National University, Gwangju, South Korea, in 2019, and the M.S. degree from the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju. Since 2021, he has been working as a Researcher with the Korea Electronics Technology Institute (KETI), Seoul, South Korea. His research interests include blockchain, decentralized finance, and statistical analysis.



HYUN-SIK KIM received the Ph.D. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2017. Since 2004, he has been with the Contents Convergence Research Center, Korea Electronics Technology Institute (KETI), Seoul, where he is currently a Principal Researcher. His research interests include blockchain, immersive contents, contents copyright protection, and image processing.



YONG-SUK PARK received the B.S. and M.S. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 1997 and 1998, respectively, and the Ph.D. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2018. He is currently a Principal Researcher at the Korea Electronics Technology Institute (KETI), Seoul. Before joining KETI, in 2003, he was with I&C Technology and Samsung S1, where he worked in projects relevant to Wi-Fi networks and security system integration. His current research interest includes blockchain-based content protection in metaverse ecosystems.

...