1a) What's new in Uniswap v4, compared with Uniswap v3?

As the upgrade to Uniswap v3, Uniswap v4 released these new properties below.

- A time-weighted average market maker (TWAMM)
- Dynamic fees based on volatility or other inputs
- Onchain limit orders
- Depositing out-of-range liquidity into lending protocols
- Customized on-chain oracles, such as geomean oracles
- Autocompounded LP fees back into the LP positions
- Internalized MEV profits are distributed back to LPs

1b) An easier question is as follows: What's new in Uniswap v3, compared with Uniswap v2?

As the upgrade to Uniswap v2, Uniswap v3 further introduces two major new features.
Firstly, Uniswap v3 has “concentrated liquidity”. It gives individual liquidity providers (LPs) granular control over what price ranges their capital is allocated to. Individual positions are aggregated together into a single pool, forming one combined curve for users to trade against.
Secondly, Uniswap v3 has “multiple fee tiers”. It allows LPs to be appropriately compensated for taking on varying degrees of risk.

The aforementioned features further make Uniswap v3 the most flexible and efficient AMM ever designed for four reasons. Firstly, LPs can provide liquidity with up to 4000x capital efficiency relative to Uniswap v2, earning higher returns on their capital. Secondly, capital efficiency paves the way for low-slippage trade execution that can surpass both centralized exchanges and stablecoin-focused AMMs. Thirdly, LPs can significantly increase their exposure to preferred assets and reduce their downside risk. Last but not least, LPs can sell one asset for another by adding liquidity to a price range entirely above or below the market price, approximating a fee-earning limit order that executes along a smooth curve.

For the cost, the gas cost of v3 swaps on Ethereum mainnet is slightly cheaper than v2 even with these groundbreaking design improvements. Furthermore, Uniswap’s V3 oracles are capable of providing time-weighted average prices (TWAPs) on demand for any period within the last ~9 days. This removes the need for integrators to checkpoint historical values.

Concentrated liquidity is worth mentioning because LP's can concentrate their capital within custom price ranges, providing greater amounts of liquidity at desired prices. In doing so, LPs construct individualized price curves that reflect their own preferences.

2a) As Uniswap's Singleton smart contract (See Uniswap) 在 ERC 4337 的定義當中出現, explain ERC 4337 in the light of Singleton smart contract.

ERC-4337 is an account abstraction proposal which completely avoids the need for consensus-layer protocol changes. **This means it holds all the pools in one singleton contract.** Instead of adding new protocol features and changing the bottom-layer transaction type, this proposal introduces a higher-layer pseudo-transaction object called a UserOperation. Users send UserOperation objects into a new separate mem-pool. **Bundlers package up a set of these objects into a single transaction by making a call to a special contract, and that transaction then gets included in a block.** With doing so, ERC-4337 will provide important gas savings because swaps will no longer need to transfer tokens between pools held in different contracts.

2b) Uniswap mentioned hook. Explain the hook here.

Hooks are contracts that run at various points of a pool action's lifecycle. Uniswap v4’s pools can make the same tradeoffs as v3, or they can add totally new functionality. For example, v4 will allow pools that natively support dynamic fees, add on-chain limit orders, or act as a time-weighted average market maker (TWAMM) to spread out large orders over time.

2c) Uniswap mentioned flash accounting. How is flash accounting enabled by "transient storage" in ERC 1153?

Uniswap v3 deployed a new contract for every pool, making creating pools and performing multi-pool swaps more expensive. In v4, Uniswap holds all the pools in one singleton contract, which will provide important gas savings because swaps will no longer need to transfer tokens between pools held in different contracts. Early estimates show that v4 reduces pool creation gas costs by 99%. Hooks introduce a world with endless options and singleton allows you to efficiently route across all of them.

This singleton architecture is complemented by a new "flash accounting" system. Instead of transferring assets in and out of pools at the end of every swap in v3, this system transfers only on net balances -- meaning a far more efficient system that provides additional gas savings in Uniswap v4.

Storage refunds accumulated due to inter frame communication are also limited to 20% of gas spent by a transaction due to EIP-3529 (introduced in the London hard fork). This greatly reduces the refunds for transiently-set storage slots in otherwise low-cost transactions. For example, in order to receive the full refund of one re-entrancy lock, the transaction must spend ~80k gas on other operations.

"Transient storage" can be introduced (similar to the existing qualifiers memory and storage, and Java's own transient keyword with a similar meaning). Since the addressing scheme of TSTORE and TLOAD is the same as for SSTORE and SLOAD, code generation routines that exist for storage variables, can be easily generalized to also support transient storage.

Thanks to the "transient storage" technique, EIP-1153 is being considered as part of the Ethereum Cancun hard fork and will bring even bigger gas improvements and cleaner contract designs across a wide variety of applications.

3) Why does VItalik emphasize AA (Account Abstraction) and intent-centric recently?

The reasons that VItalik emphasizes AA (Account Abstraction) may be its properties of Better Security, Beginner-friendliness, Easy Interaction With Web3, Automation and Customization, Gas Fee Management, and A user-friendlier UX without sacrificing control.
All the properties that are listed above could explain that AA has the potential to empower the wallets to enlarge the business ability.

As for Intent-centric, Intent-centric could be a turnaround opportunity for wallets.
Originally, users' interaction needs, such as bridge, swap, staking, withdrawal, etc., all directly interacted with the on-chain contracts in the form of instructions. Most DeFi protocols would independently design front-end interactive pages for users to use. The role played by Wallet was only Connect Wallet+Signature.
The intent process is essentially an off-chain pre-processing black box. The wallet can completely become a user intent pre-processing center, used to collect and pre-process the user's transaction needs, and then link all back-end DeFi protocols to interact, and the DeFi protocol also No longer relies on front-end traffic. Only in this way can the commercial potential of using this wallet as an entrance to purchase road money can emerge. Of course, to be a good Solver, the challenges faced by the wallet will also be arduous.